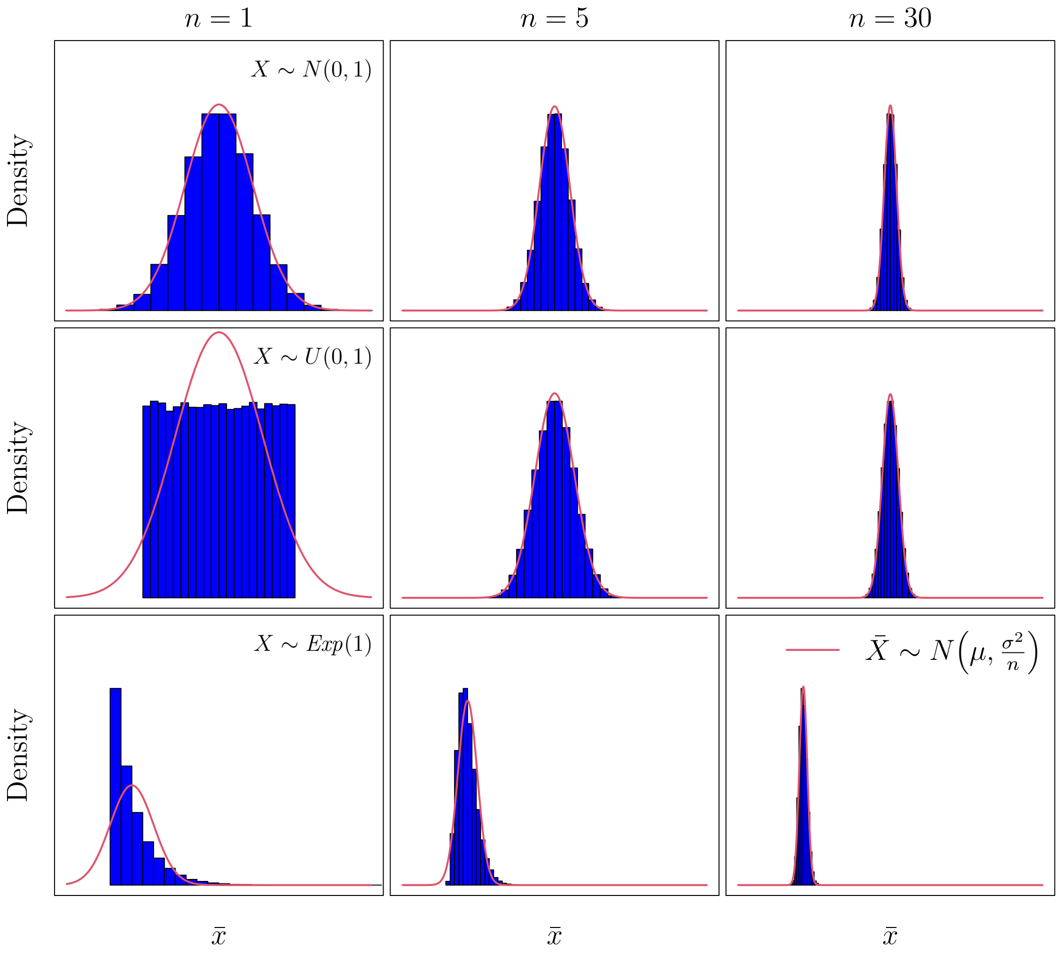
Introduction to Statistics at DTU

Per B. Brockhoff, Jan K. Møller, Elisabeth W. Andersen
Peder Bacher, Lasse E. Christiansen

2024-11-24 13:49



Contents

The plot on the front page is an illustration of the Central Limit Theorem (CLT). To put it shortly, it states that when sampling a population: as the sample size increases, then the mean of the sample converges to a normal distribution – no matter the distribution of the population. The thumb rule is that the normal distribution can be used for the sample mean when the sample size n is above 30 observations (n is the number observations in the sample). The plot is created by simulating 100000 sample means $\bar{X} = \sum_{i=1}^{n} X_i$ (where X_i is an observation from a distribution) and plotting their histogram with the CLT distribution on top (the red linie). The upper is for the normal, the mid is for the uniform and the lower is for the exponential distribution. We can thus see that as n increase, then the distribution of the simulated sample means $\bar{x}$ approaches the distribution stated by the CLT (it is the normal distribution $\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$, where μ is the mean and σ is the standard deviation of the population), see more in Section 3.1.4.

Chapter 1

Introduction, descriptive statistics, Python and data visualization

This is the first chapter in the eight-chapter DTU Introduction to Statistics book. It consists of eight chapters:

1. Introduction, descriptive statistics, Python and data visualization
2. Probability and simulation
3. Statistical analysis of one and two sample data
4. Statistics by simulation
5. Simple linear regression
6. Multiple linear regression
7. Analysis of categorical data
8. Analysis of variance (analysis of multi-group data)

In this first chapter the idea of statistics is introduced together with some of the basic summary statistics and data visualization methods. This book is available in in two versions: one using the open souce environment R and one using the open source environment Python. You are currently reading the Python version, and this is the software used throughout the book for working with statistics, probability and data analysis. An introduction to Python is included in this chapter.

1.1 What is Statistics - a primer

To catch your attention we will start out trying to give an impression of the importance of statistics in modern science and engineering.

In the well respected *New England Journal of medicine* a millennium editorial on the development of medical research in a thousand years was written:

EDITORIAL: Looking Back on the Millennium in Medicine, *N Engl J Med*, 342:42-49, January 6, 2000, NEJM200001063420108.

They came up with a list of 11 points summarizing the most important developments for the health of mankind in a millennium:

- Elucidation of human anatomy and physiology
- Discovery of cells and their substructures
- Elucidation of the chemistry of life
- Application of statistics to medicine
- Development of anaesthesia
- Discovery of the relation of microbes to disease
- Elucidation of inheritance and genetics
- Knowledge of the immune system
- Development of body imaging
- Discovery of antimicrobial agents
- Development of molecular pharmacotherapy

The reason for showing the list here is pretty obvious: one of the points is *Application of Statistics to Medicine*! Considering the other points on the list, and what the state of medical knowledge was around 1000 years ago, it is obviously a very impressive list of developments. The reasons for statistics to be on this list are several and we mention two very important historical landmarks here. Quoting the paper:

"One of the earliest clinical trials took place in 1747, when James Lind treated 12 scorbutic ship passengers with cider, an elixir of vitriol, vinegar, sea water, oranges and lemons, or an electuary recommended by the ship's surgeon. The success of the citrus-containing treatment eventually led the British Admiralty to mandate the provision of lime juice to all sailors, thereby eliminating scurvy from the navy." (See also James_Lind).

Still today, clinical trials, including the statistical analysis of the outcomes, are taking place in massive numbers. The medical industry needs to do this in order to find out if their new developed drugs are working and to provide documentation to have them accepted for the World markets. The medical industry is probably the sector recruiting the highest number of statisticians among all sectors. Another quote from the paper:

"The origin of modern epidemiology is often traced to 1854, when John Snow demonstrated the transmission of cholera from contaminated water by analyzing disease rates among citizens served by the Broad Street Pump in London's Golden Square. He arrested the further spread of the disease by removing the pump handle from the polluted well." (See also John_Snow_(physician)).

Still today, epidemiology, both human and veterinarian, maintains to be an extremely important field of research (and still using a lot of statistics). An important topic, for instance, is the spread of diseases in populations, e.g. virus spreads like Ebola and others.

Actually, today more numbers/data than ever are being collected and the amounts are still increasing exponentially. One example is Internet data, that internet companies like Google, Facebook, IBM and others are using extensively. A quote from New York Times, 5. August 2009, from the article titled "For Today's Graduate, Just One Word: Statistics" is:

"I keep saying that the sexy job in the next 10 years will be statisticians," said Hal Varian, chief economist at Google. 'and I'm not kidding.' "

The article ends with the following quote:

"The key is to let computers do what they are good at, which is trawling these massive data sets for something that is mathematically odd," said Daniel Gruhl, an I.B.M. researcher whose recent work includes mining medical data to improve treatment. "And that makes it easier for humans to do what they are good at - explain those anomalies."

1.2 Statistics at DTU Compute

At DTU Compute at the Technical University of Denmark statistics is used, taught and researched mainly within four research sections:

- Statistics and Data Analysis
- Dynamical Systems
- Image Analysis & Computer Graphics
- Cognitive Systems

Each of these sections have their own focus area within statistics, modelling and data analysis. On the master level it is an important option within DTU Compute studies to specialize in statistics of some kind on the joint master programme in Mathematical Modelling and Computation (MMC). And a *Statistician* is a well-known profession in industry, research and public sector institutions.

The high relevance of the topic of statistics and data analysis today is also illustrated by the extensive list of ongoing research projects involving many and diverse industrial partners within these four sections. Neither society nor industry can cope with all the available data without using highly specialized people in statistical techniques, nor can they cope and be internationally competitive without continuously further developing these methodologies in research projects. Statistics is and will continue to be a relevant, viable and dynamic field. And the amount of experts in the field continues to be small compared to the demand for experts, hence obtaining skills in statistics is for sure a wise career choice for an engineer. Still for any engineer not specialising in statistics, a basic level of statistics understanding and data handling ability is crucial for the ability to navigate in modern society and business, which will be heavily influenced by data of many kinds in the future.

1.3 Statistics - why, what, how?

Often in society and media, the word *statistics* is used simply as the name for a summary of some numbers, also called data, by means of a summary table and/or plot. We also embrace this basic notion of statistics, but will call such basic data summaries *descriptive statistics* or *explorative statistics*. The meaning of *statistics* goes beyond this and will rather mean *"how to learn from data in an insightful way and how to use data for clever decision making"*, in short we call this *inferential statistics*. This could be on the national/societal level, and could be related to any kind of topic, such as, e.g., health, economy or environment, where data is collected and used for learning and decision making. For example:

- Cancer registries
- Health registries in general
- Nutritional databases
- Climate data
- Macro economic data (Unemployment rates, GNP etc.)
- etc.

The latter is the type of data that historically gave name to the word *statistics*. It originates from the Latin ‘statisticum collegium’ (state advisor) and the Italian word ‘statista’ (statesman/politician). The word was brought to Denmark by the Gottfried Achenwall from Germany in 1749 and originally described the processing of data for the state, see also History_of_statistics.

Or it could be for industrial and business applications:

- Is machine A more effective than machine B?
- How many products are we selling on different markets?
- Predicting wind and solar power for optimizing energy systems
- Do we produce at the specified quality level?
- Experiments and surveys for innovative product development
- Drug development at all levels at e.g. Novo Nordisk A/S or other pharmaceutical companies
- Learning from "Big Data"
- etc.

In general, it can be said that we learn from data by analysing the data with statistical methods. Therefore *statistics* will in practice involve *mathematical modelling*, i.e. using some linear or non-linear function to model the particular phenomenon. Similarly, the use of *probability theory* as the concept to describe randomness is extremely important and at the heart of being able to “be clever” in our use of the data. Randomness expresses that the data just as well could have come up differently due to the inherent random nature of the data collection and the phenomenon we are investigating.

Probability theory is in its own right an important topic in engineering relevant applied mathematics. Probability based modelling is used for e.g. queuing systems (queuing for e.g. servers, websites, call centers etc.), for reliability modelling, and for risk analysis in general. Risk analysis encompasses a vast diversity of engineering fields: food safety risk (toxicological and/or allergenic), environmental risk, civil engineering risks, e.g. risk analysis of large building constructions, transport risk, etc. The present material focuses on the statistical issues, and treats probability theory at a minimum level, focusing solely on the purpose of being able to do proper *statistical inference* and leaving more elaborate probability theory and modelling to other texts.

There is a conceptual frame for doing *statistical inference*: in *Statistical inference* the observed data is a *sample*, that is (has been) taken from a *population*. Based on the sample, we try to generalize to (infer about) the population. Formal definitions of what the sample and the population is are given by:

Definition 1.1 Sample and population

- An *observational unit* is the single entity about which information is sought (e.g. a person)
- An *observational variable* is a property which can be measured on the observational unit (e.g. the height of a person)
- The *statistical population* consists of the value of the observational variable for all observational units (e.g. the heights of all people in Denmark)
- The *sample* is a subset of the statistical population, which has been chosen to represent the population (e.g. the heights of 20 persons in Denmark).

See also the illustration in Figure 1.1.

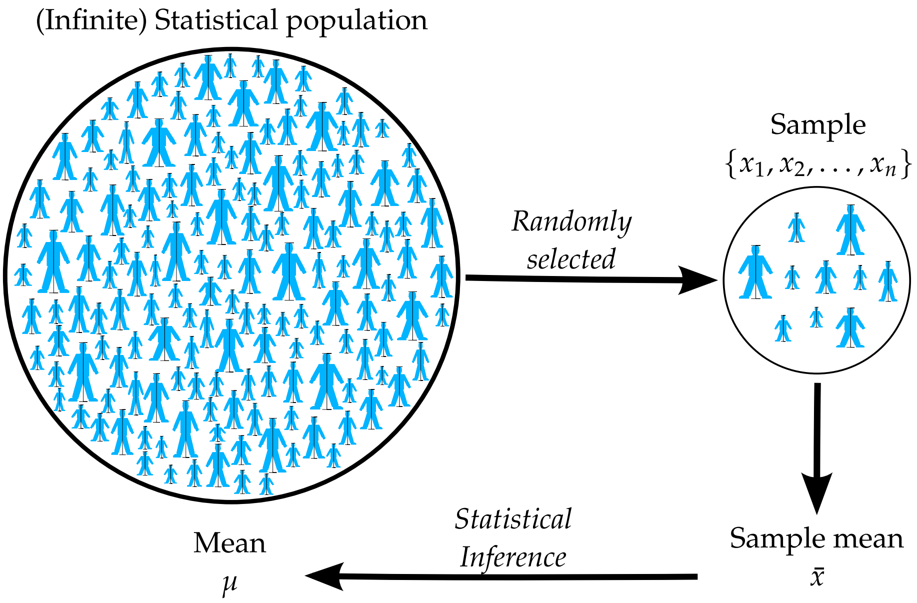


Figure 1.1: Illustration of statistical population and sample, and statistical inference. Note that the bar on each person indicates that the it is the height (the observational variable) and not the person (the observational unit), which are the elements in the statistical population and the sample. Notice, that in all analysis methods presented in this text the statistical population is assumed to be very large (or infinite) compared to the sample size.

This is all a bit abstract at this point. And likely adding to the potential confusion about this is the fact that the words *population* and *sample* will have a "less precise" meaning when used in everyday language. When they are used in a statistical context the meaning is very specific, as given by the definition above. Let us consider a simple example:

|||| Example 1.2

The following study is carried out (actual data collection): the height of 20 persons in Denmark is measured. This will give us 20 values $x_1, \ldots, x_{20}$ in cm. The *sample* is then simply these 20 values. The statistical *population* is the height values of all people in Denmark. The *observational unit* is a person.

The meaning of *sample* in statistics is clearly different from how a chemist or medical doctor would use the word, where a sample would be the actual substance in e.g. the petri dish. Within this book, when using the word sample, then it is always in the statistical meaning i.e. a set of values taken from a statistical population.

With regards to the meaning of *population* within statistics the difference to the everyday meaning is less obvious: but note that the *statistical population* in the example is defined to be the height values of people, not actually the people. Had we measured the weights instead the statistical population would be quite different. Also later we will realize that statistical populations in engineering contexts can refer to many other things than populations as in a group of organisms, hence stretching the use of the word beyond the everyday meaning. From this point: *population* will be used instead of *statistical population* in order to simplify the text.

The population in a given situation will be linked with the actual study and/or experiment carried out - the data collection procedure sometimes also denoted the *data generating process*. For the sample to represent relevant information about the population it should be *representative* for that population. In the example, had we only measured male heights, the population we can say anything about would be the male height population only, not the entire height population.

A way to achieve a representative sample is that each observation (i.e. each value) selected from the population, is randomly and independently selected of each other, and then the sample is called a *random sample*.

1.4 Summary statistics

The descriptive part of studying data maintains to be an important part of statistics. This implies that it is recommended to study the given data, the sample, by means of *descriptive statistics* as a first step, even though the purpose of a full statistical analysis is to eventually perform some of the new inferential tools taught in this book, that will go beyond the pure descriptive part. The aims of the initial descriptive part are several, and when moving to more complex data settings later in the book, it will be even more clear how the initial descriptive part serves as a way to prepare for and guide yourself in the subsequent more formal inferential statistical analysis.

The initial part is also called an *explorative* analysis of the data. We use a number of summary statistics to summarize and describe a sample consisting of one or two variables:

- Measures of centrality:
  - Mean
  - Median
  - Quantiles
- Measures of "spread":
  - Variance
  - Standard deviation
  - Coefficient of variation
  - Inter Quartile Range (IQR)
- Measures of relation (between two variables):
  - Covariance
  - Correlation

One important point to notice is that these statistics can only be calculated for the sample and not for the population - we simply don't know all the values in the population! But we want to learn about the population from the sample. For example when we have a random sample from a population we say that the *sample mean* ($\bar{x}$) is an *estimate* of the *mean* of the population, often then denoted μ, as illustrated in Figure 1.1.

Remark 1.3

Notice, that we put 'sample' in front of the name of the statistic, when it is calculated for the sample, but we don't put 'population' in front when we refer to it for the population (e.g. we can think of the *mean* as the true mean).

HOWEVER we don't put *sample* in front of the name every time it should be there! This is to keep the text simpler and since traditionally this is not strictly done, for example the median is rarely called the sample median, even though it makes perfect sense to distinguish between the sample median and the median (i.e. the population median). Further, it should be clear from the context if the statistic refers to the sample or the population, when it is not clear then we distinguish in the text. Most of the way we do distinguish strictly for the *mean*, *standard deviation*, *variance*, *covariance* and *correlation*.

1.4.1 Measures of centrality

The sample mean is a key number that indicates the centre of gravity or centring of the sample. Given a sample of n observations $x_1, \ldots, x_n$, it is defined as follows:

Definition 1.4 Sample mean

The sample mean is the sum of observations divided by the number of observations

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{1-1}$$

Sometimes this is refereed to as the *average*.

The median is also a key number indicating the center of sample (note that to be strict we should call it 'sample median', see Remark 1.3 above). In some cases, for example in the case of extreme values or skewed distributions, the median can be preferable to the mean. The median is the observation in the middle of the sample (in sorted order). One may express the ordered observations as $x_{(1)}, \ldots, x_{(n)}$, where then $x_{(1)}$ is the smallest of all $x_1, \ldots, x_n$ (also called the minimum) and $x_{(n)}$ is the largest of all $x_1, \ldots, x_n$ (also called the maximum).

|||| Definition 1.5 Median

Order the n observations $x_1, \ldots, x_n$ from the smallest to largest: $x_{(1)}, \ldots, x_{(n)}$. The median is defined as:

- If n is odd the median is the observation in position $\frac{n+1}{2}$:

$$Q_2 = x_{(\frac{n+1}{2})}. \tag{1-2}$$

- If n is even the median is the average of the two observations in positions $\frac{n}{2}$ and $\frac{n+2}{2}$:

$$Q_2 = \frac{x_{(\frac{n}{2})} + x_{(\frac{n+2}{2})}}{2}. \tag{1-3}$$

The reason why it is denoted with Q_2 is explained below in Definition 1.8.

|||| Example 1.6 Student heights

A random sample of the heights (in cm) of 10 students in a statistics class was

168 161 167 179 184 166 198 187 191 179 .

The sample mean height is

$$\bar{x} = \frac{1}{10}(168 + 161 + 167 + 179 + 184 + 166 + 198 + 187 + 191 + 179) = 178.$$

To find the sample median we first order the observations from smallest to largest

$x_{(1)}$	$x_{(2)}$	$x_{(3)}$	$x_{(4)}$	$x_{(5)}$	$x_{(6)}$	$x_{(7)}$	$x_{(8)}$	$x_{(9)}$	$x_{(10)}$
161	166	167	168	179	179	184	187	191	198

Note that having duplicate observations (like e.g. two of 179) is not a problem - they all just have to appear in the ordered list. Since $n = 10$ is an even number the median becomes the average of the 5th and 6th observations

$$\frac{x_{(\frac{n}{2})} + x_{(\frac{n+2}{2})}}{2} = \frac{x_{(5)} + x_{(6)}}{2} = \frac{179 + 179}{2} = 179.$$

As an illustration, let's look at the results if the sample did not include the 198 cm height, hence for $n = 9$

$$\bar{x} = \frac{1}{9}(168 + 161 + 167 + 179 + 184 + 166 + 187 + 191 + 179) = 175.78.$$

then the median would have been

$$x_{(\frac{n+1}{2})} = x_{(5)} = 179.$$

This illustrates the robustness of the median compared to the sample mean: the sample mean changes a lot more by the inclusion/exclusion of a single "extreme" measurement. Similarly, it is clear that the median does not depend at all on the actual values of the most extreme ones.

The median is the point that divides the observations into two halves. It is of course possible to find other points that divide into other proportions, they are called quantiles or percentiles (note, that this is actually the *sample quantile* or *sample percentile*, see Remark 1.3).

|||| Definition 1.7 Quantiles and percentiles

The p *quantile* also called the $100p\%$ quantile or $100p$'th percentile, can be defined by the following procedure: [a]

1. Order the n observations from smallest to largest: $x_{(1)}, \ldots, x_{(n)}$
2. Compute pn
3. If pn is an integer: average the pn'th and $(pn+1)$'th ordered observations. Then the p quantile is

$$q_p = \left(x_{(np)} + x_{(np+1)}\right)/2 \tag{1-4}$$

4. If pn is a non-integer: take the "next one" in the ordered list. Then the p'th quantile is

$$q_p = x_{(\lceil np \rceil)}, \tag{1-5}$$

where $\lceil np \rceil$ is the *ceiling* of np, that is, the smallest integer larger than np

[a] There exist several other formal definitions. To obtain this definition of quantiles/percentiles in Python use `percentile(..., method='averaged_inverted_cdf')`. Using the default method is also a perfectly valid approach - just a different one.

Often calculated percentiles are the so-called *quartiles* (splitting the sample in quarters, i.e. 0%, 25%, 50%, 75% and 100%):

- q_0, $q_{0.25}$, $q_{0.50}$, $q_{0.75}$ and q_1

Note that the 0'th percentile is the minimum (smallest) observation and the 100'th percentile is the maximum (largest) observation. We have specific names for the three other quartiles:

Definition 1.8 Quartiles

$$\begin{aligned}
Q_1 &= q_{0.25} = \text{"lower quartile"} = \text{"0.25 quantile"} = \text{"25'th percentile"}\\
Q_2 &= q_{0.50} = \text{"median"} = \text{"0.50 quantile"} = \text{"50'th percentile"}\\
Q_3 &= q_{0.75} = \text{"upper quartile"} = \text{"0.75 quartile"} = \text{"75'th percentile"}
\end{aligned}$$

Example 1.9 Student heights

Using the $n = 10$ sample from Example 1.6 and the ordered data table from there, let us find the lower and upper quartiles (i.e. Q_1 and Q_3), as we already found $Q_2 = 179$.

First, the Q_1: with $p = 0.25$, we get that $np = 2.5$ and we find that

$$Q_1 = x_{(\lceil 2.5 \rceil)} = x_{(3)} = 167,$$

and since $n \cdot 0.75 = 7.5$, the upper quartile becomes

$$Q_3 = x_{(\lceil 7.5 \rceil)} = x_{(8)} = 187.$$

We could also find the 0'th percentile

$$q_0 = \min(x_1, \ldots, x_n) = x_{(1)} = 161,$$

and the 100'th percentile

$$q_1 = \max(x_1, \ldots, x_n) = x_{(10)} = 198.$$

Finally, 10'th percentile (i.e. 0.10 quantile) is

$$q_{0.10} = \frac{x_{(1)} + x_{(2)}}{2} = \frac{161 + 166}{2} = 163.5,$$

since $np = 1$ for $p = 0.10$.

1.4.2 Measures of variability

A crucial aspect to understand when dealing with statistics is the concept of variability - the obvious fact that not everyone in a population, nor in a sample, will be exactly the same. If that was the case they would all equal the mean of the population or sample. But different phenomena will have different degrees of variation: An adult (non dwarf) height population will maybe spread from around 150 cm up to around 210 cm with very few exceptions. A kitchen scale measurement error population might span from -5 g to $+5$ g. We need a way to quantify the degree of variability in a population and in a sample. The most commonly used measure of sample variability is the sample variance or its square root, called the sample standard deviation:

Definition 1.10 Sample variance

The *sample variance* of a sample $x_1, \ldots, x_n$ is the sum of squared differences from the sample mean divided by $n-1$

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2. \tag{1-6}$$

Definition 1.11 Sample standard deviation

The *sample standard deviation* is the square root of the sample variance

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{1-7}$$

The sample standard deviation and the sample variance are key numbers of absolute variation. If it is of interest to compare variation between different samples, it might be a good idea to use a relative measure - most obvious is the coefficient of variation:

|||| Definition 1.12 Coefficient of variation

The *coefficient of variation* is the sample standard deviation seen relative to the sample mean

$$V = \frac{s}{\bar{x}}. \tag{1-8}$$

We interpret the standard deviation as the *average absolute deviation from the mean* or simply: the *average level of differences*, and this is by far the most used measure of spread. Two (relevant) questions are often asked at this point (it is perfectly fine if you didn't wonder about them by now and you might skip the answers and return to them later):

|||| Remark 1.13

Question: Why not actually compute directly what the interpretation is stating, which would be: $\frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$?

Answer: This is indeed an alternative, called the *mean absolute deviation*, that one could use. The reason for most often measuring "mean deviation" NOT by the *Mean Absolute Deviation* statistic, but rather by the sample standard deviation s, is the so-called theoretical statistical properties of the sample variance s^2. This is a bit early in the material for going into details about this, but in short: inferential statistics is heavily based on probability considerations, and it turns out that it is theoretically much easier to put probabilities related to the sample variance s^2 on explicit mathematical formulas than probabilities related to most other alternative measures of variability. Further, in many cases this choice is in fact also the optimal choice in many ways.

|||| Remark 1.14

Question: Why divide by $n-1$ and not n in the formulas of s and s^2? (which *also* appears to fit better with the stated interpretation)

Answer: The sample variance s^2 will most often be used as an estimate of the (true but unknown) population variance σ^2, which is the average of $(x_i - \mu)^2$ in the population. In doing that, one should ideally compare each observation x_i with the population mean, usually called μ. However, we do not know μ and instead we use $\bar{x}$ in the computation of s^2. In doing so, the squared differences $(x_i - \bar{x})^2$ that we compute in this way will tend to be slightly smaller than those we ideally should have used: $(x_i - \mu)^2$ (as the observations themselves were used to find $\bar{x}$ so they will be closer to $\bar{x}$ than to μ). It turns out, that the correct way to correct for this is by dividing by $n-1$ instead of n.

Spread in the sample can also be described and quantified by quartiles:

|||| Definition 1.15 Range

The *range* of the sample is

$$\text{Range} = \text{Maximum} - \text{Minimum} = Q_4 - Q_0 = x_{(n)} - x_{(1)}. \qquad (1\text{-}9)$$

The Inter Quartile Range (IQR) is the middle 50% range of data defined as

$$IQR = q_{0.75} - q_{0.25} = Q_3 - Q_1. \qquad (1\text{-}10)$$

|||| Example 1.16 Student heights

Consider again the $n = 10$ data from Example 1.6. To find the variance let us compute the $n = 10$ differences to the mean, that is $(x_i - 178)$

-10 -17 -11 1 6 -12 20 9 13 1 .

So, if we square these and add them up we get

$$\sum_{i=1}^{10}(x_i - \bar{x})^2 = 10^2 + 17^2 + 11^2 + 1^2 + 6^2 + 12^2 + 20^2 + 9^2 + 13^2 + 1^2 = 1342.$$

Therefore the sample variance is

$$s^2 = \frac{1}{9}1342 = 149.1,$$

and the sample standard deviation is

$$s = 12.21.$$

We can interpret this as: people are on average around 12 cm away from the mean height of 178 cm. The Range and Inter Quartile Range (IQR) are easily found from the ordered data table in Example 1.6 and the earlier found quartiles in Example 1.9

$$\text{Range} = \text{maximum} - \text{minimum} = 198 - 161 = 37,$$

$$IQR = Q_3 - Q_1 = 187 - 167 = 20.$$

Hence 50% of all people (in the sample) lie within 20 cm.

Note, that the standard deviation in the example has the physical unit cm, whereas the variance has cm^2. This illustrates the fact that the standard deviation has a more direct interpretation than the variance in general.

1.4.3 Measures of relation: correlation and covariance

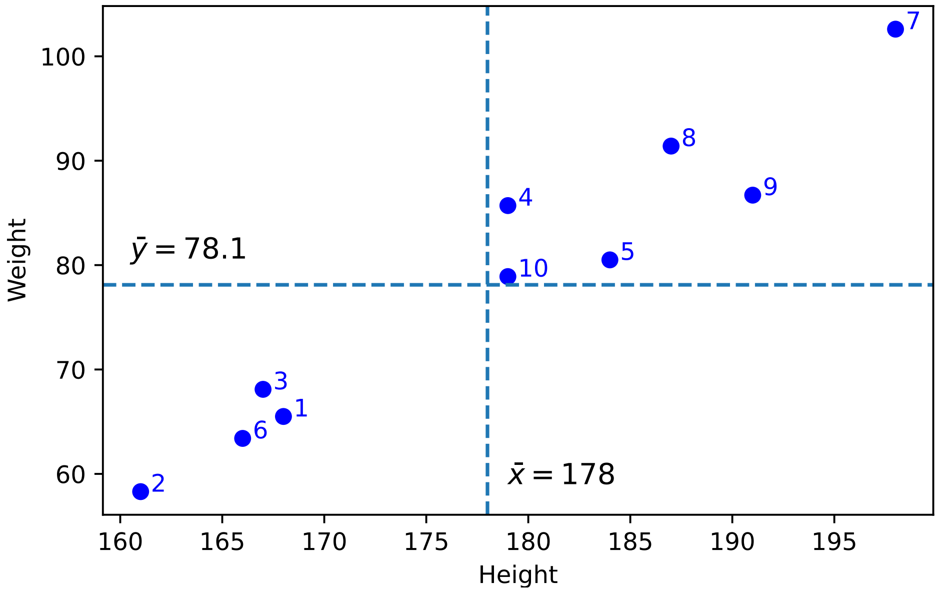
When two observational variables are available for each observational unit, it may be of interest to quantify the relation between the two, that is to quantify how the two variables *co-vary* with each other, their *sample covariance* and/or *sample correlation*.

Example 1.17 Student heights and weights

In addition to the previously given student heights we also have their weights (in kg) available

Heights (x_i)	168	161	167	179	184	166	198	187	191	179
Weights (y_i)	65.5	58.3	68.1	85.7	80.5	63.4	102.6	91.4	86.7	78.9 .

The relation between weights and heights can be illustrated by the so-called scatter-plot, cf. Section 1.6.4, where e.g. weights are plotted versus heights:



Each point in the plot corresponds to one student - here illustrated by using the observation number as plot symbol. The (expected) relation is pretty clear now - different wordings could be used for what we see:

- Weights and heights are related to each other
- Higher students tend to weigh more than smaller students
- There is an increasing pattern from left to right in the "point cloud”
- If the point cloud is seen as an (approximate) ellipse, then the ellipse clearly is horizontally upwards ”tilted”.
- Weights and heights are (positively) *correlated* to each other

The sample covariance and sample correlation coefficients are a summary statistics that can be calculated for two (related) sets of observations. They quantify the (linear) strength of the relation between the two. They are calculated by combining the two sets of observations (and the means and standard deviations from the two) in the following ways:

Definition 1.18 Sample covariance

The sample covariance is

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}). \tag{1-11}$$

Definition 1.19 Sample correlation

The sample correlation coefficient is

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right) = \frac{s_{xy}}{s_x \cdot s_y}, \tag{1-12}$$

where s_x and s_y is the sample standard deviation for x and y respectively.

When $x_i - \bar{x}$ and $y_i - \bar{y}$ have the same sign, then the point (x_i, y_i) give a positive contribution to the sample correlation coefficient and when they have opposite signs the point give a negative contribution to the sample correlation coefficient, as illustrated here:

Example 1.20 Student heights and weights

The sample means are found to be

$$\bar{x} = 178 \text{ and } \bar{y} = 78.1.$$

Using these we can show how each student deviate from the average height and weight (these deviations are exactly used for the sample correlation and covariance computations)

Student	1	2	3	4	5	6	7	8	9	10
Height (x_i)	168	161	167	179	184	166	198	187	191	179
Weight (y_i)	65.5	58.3	68.1	85.7	80.5	63.4	102.6	91.4	86.7	78.9
$(x_i - \bar{x})$	-10	-17	-11	1	6	-12	20	9	13	1
$(y_i - \bar{y})$	-12.6	-19.8	-10	7.6	2.4	-14.7	24.5	13.3	8.6	0.8
$(x_i - \bar{x})(y_i - \bar{y})$	126.1	336.8	110.1	7.6	14.3	176.5	489.8	119.6	111.7	0.8

Student 1 is below average on both height and weight $(-10 \text{ and } -12.6)$. Student 10 is above average on both height and weight $(+1 \text{ and } +0.8)$.s

The sample covariance is then given by the sum of the 10 numbers in the last row of the table

$$\begin{aligned} s_{xy} &= \frac{1}{9}(126.1 + 336.8 + 110.1 + 7.6 + 14.3 + 176.5 + 489.8 + 119.6 + 111.7 + 0.8) \\ &= \frac{1}{9} \cdot 1493.3 \\ &= 165.9 \end{aligned}$$

And the sample correlation is then found from this number and the standard deviations

$$s_x = 12.21 \text{ and } s_y = 14.07.$$

(the details of the s_y computation is not shown). Thus we get the sample correlation as

$$r = \frac{165.9}{12.21 \cdot 14.07} = 0.97.$$

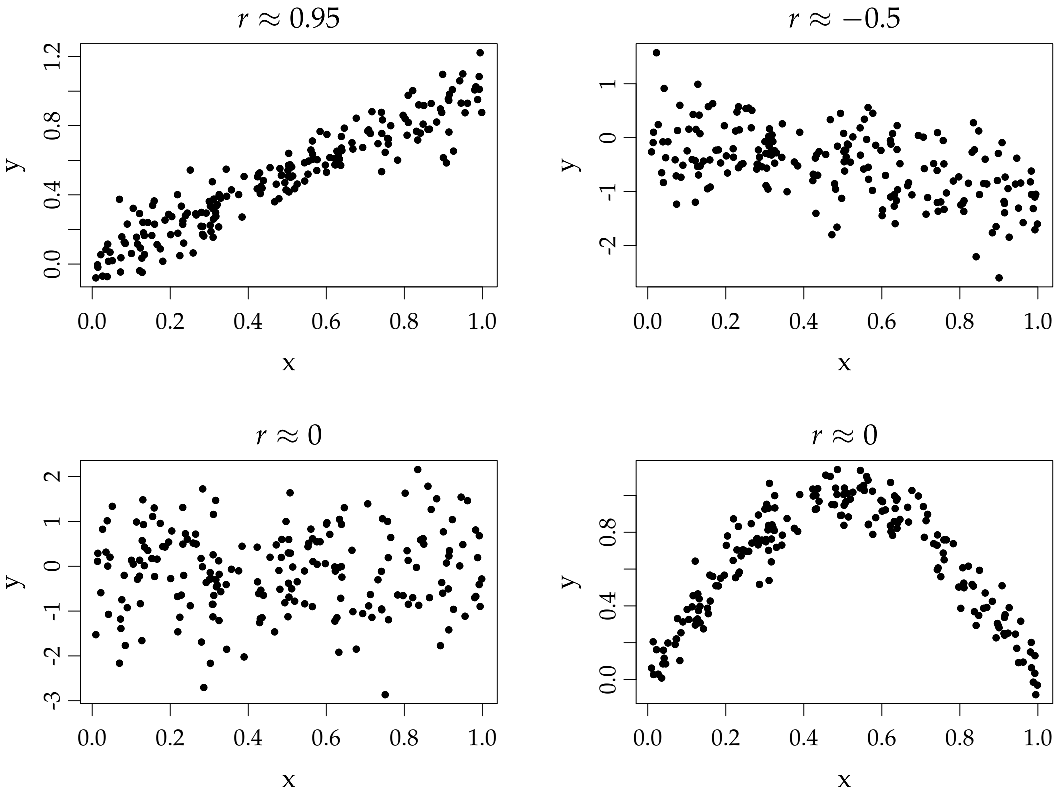
Note how all 10 contributions to the sample covariance are positive in the example case - in line with the fact that all observations are found in the first and third quadrants of the scatter plot (where the quadrants are defined by the sample means of x and y). Observations in second and fourth quadrant would contribute with negative numbers to the sum, hence such observations would be from students with below average on one feature while above average on the other. Then it is clear that: had all students been like that, then the covariance and the correlation would have been negative, in line with a negative (downwards) trend in the relation.

We can state (without proofs) a number of properties of the sample correlation r:

|||| Remark 1.21 Properties of the sample correlation, r

- r is always between -1 and 1: $-1 \leq r \leq 1$
- r measures the degree of linear relation between x and y
- $r = \pm 1$ if and only if all points in the scatterplot are exactly on a line
- $r > 0$ if and only if the general trend in the scatterplot is positive
- $r < 0$ if and only if the general trend in the scatterplot is negative

The sample correlation coefficient measures the degree of linear relation between x and y, which imply that we might fail to detect non-linear relationships, illustrated in the following plot of four different point clouds and their sample correlations:



The sample correlation in both the bottom plots are close to zero, but as we see from the plot this number itself doesn't imply that there no relation between y and x - which clearly is the case in the bottom right and highly non-linear case.

Sample covariances and correlation are closely related to the topic of linear regression, treated in Chapter 5 and 6 , where we will treat in more detail how we can find the line that could be added to such scatter-plots to describe the relation between x and y in a different (but related) way, as well as the statistical analysis used for this.

1.5 Introduction to Python

Python is an open source software that you can download to your own laptop for free. Please follow the instructions provided by the Python support team.

Visual Studio Code (VS Code) is an integrated development environment (IDE) that can be used to write, edit and run Python code (and many other types of code). VS Code will be used in the course and we will primarily be running Jupyter Notebooks. Please follow the instructions provided by the Python support team in order to install VS Code on your own laptop.

In this course (and throughout the book) we will work with several Python Libraries (also referred to as packages or modules). Libraries are collections of functions, methods, and types that extend the capabilities of Python. Libraries need to be installed before you can "import" them to use in your code. This is done with "conda install" (or "pip install"). Once installed (which you typically only do once and for all in the terminal) the libraries must be imported (which you typically do in the beginning of each of your scripts or notebooks).

In this book we will use the following abbreviations for some commonly used libraries:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as stats
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels.stats.power as smp
import statsmodels.stats.proportion as smprop
```

1.5.1 Executing code

Once you have opened a Jupyer Notebook in VS Code, you can execute a "code cell" by hitting Shift+Enter or by using the "Run" button. For instance:

```
# Add two numbers in the console
2+3

5
```

You can also assign a value to a variable, for instance:

```
# Assign the value 3 to y
y = 3
print(y)

3
```

The execution of code cells in a Jupyter Notebook is independent of their order in the notebook interface. This means that the state of the notebook (variables, functions, imports, etc.) is determined by the order in which cells are executed, not by their position in the notebook. If you execute cells out of order, you might get unexpected results because the notebook's state depends on the order of execution. It's a good practice to execute cells sequentially from top to bottom to ensure the notebook state is consistent. Occasionally, it can be helpful to use the "Kernel" -> "Restart & Run All" option to ensure that all cells are executed in order from a clean state.

1.5.2 Vectors and Numpy Arrays

We will often want to work with a datatype that behaves like a vector - for this we use Numpy Arrays:

```
import numpy as np
x = np.array([1, 4, 6, 2])
print(x)

[1 4 6 2]


print(type(x))
```

```
<class 'numpy.ndarray'>
```

You can use "arange()", if you need a sequence, e.g. 1 to 10:

```
import numpy as np
x = np.arange(10)
print(x)

[0 1 2 3 4 5 6 7 8 9]
```

Python has many different data types and Numpy Arrays is just one of them. In Python it is also very common to work with lists. Beware that Python lists can contain elements of different data types (e.g., both numbers and text) and generally do not support vectorized operations.

1.5.3 Descriptive statistics

All the summary statistics measures presented in Section 1.4 can be found as functions in the Numpy library:

- `np.mean(x)` - mean value of the vector `x`
- `np.var(x, ddof=1)` - sample variance (notice `ddof = 1`)
- `np.std(x, ddof=1)` - sample standard deviation
- `np.median(x)` - median
- `np.percentile(x,p, method=’averaged_inverted_cdf’)` - finds the pth percentiletile. p can consist of several different values, e.g. `np.percentile(x,[25,75], method=’averaged_inverted_cdf’)`
- `np.cov(x, y, ddof=1)` - the covariance of the vectors `x` and `y`
- `np.corrcoef(x, y)` - the correlation

Please again note that the words *quantiles* and *percentiles* are used interchangeably - they are essentially synonyms meaning exactly the same, even though the formal distinction has been clarified earlier.

Example 1.22 Summary statistics

Consider again the $n = 10$ data from Example 1.6. We can compute the sample mean and sample median as follows:

```
# Sample Mean and Median
x = np.array([168, 161, 167, 179, 184, 166, 198, 187, 191, 179])
np.mean(x)

np.float64(178.0)


np.median(x)

np.float64(179.0)
```

The sample variance and sample standard deviation are found as follows:

```
# Sample variance and standard deviation
np.var(x, ddof=1)

np.float64(149.11111111111111)


np.sqrt(np.var(x, ddof=1))

np.float64(12.211106056009468)


np.std(x, ddof=1)

np.float64(12.211106056009468)
```

The sample quartiles can be found by using the `quantile` function as follows:

```
# Sample percentiles
np.percentile(x, [0,25,50,75,100], method=’averaged_inverted_cdf’)

array([161.000, 167.000, 179.000, 187.000, 198.000])
```

The option “`method=’averaged_inverted_cdf’`” makes sure that the percentiles found by the function is found using the definition given in Definition 1.7. By default, the `percentile` function would use another definition (not detailed here). Generally, we consider this default choice just as valid as the one explicitly given here, it is merely a different one.

You can also generate percentiles using `range(start, stop, step)` (recall that Python is "left inclusive, right exclusive"). For instance:

```
# Sample percentiles
np.percentile(x, list(range(0,110,10)), method=’averaged_inverted_cdf’)


array([161.000, 163.500, 166.500, 167.500, 173.500, 179.000, 181.500,
       185.500, 189.000, 194.500, 198.000])
```

1.5.4 Use of Python in the course and at the exam

You should bring your laptop with Python installed with you to the teaching activity and to the exam. We will need access to the so-called probability distributions to do statistical computations, and the values of these distributions are not otherwise part of the written material: These probability distributions are part of many different software, also Excel, but it is part of the syllabus to be able to work with these within Python.

Apart from access to these probability distributions, Python is used in three ways in our course

1. As a pedagogical learning tool: The random variable simulation tools inbuilt in Python enables the use of software as a way to illustrate and learn the principles of statistical reasoning that are the main purposes of this course.

2. As a pocket calculator substitute - that is making Python calculate ”manually” - by simple routines - plus, minus, square root etc. whatever needs to be calculated, that you have identified by applying the right formulas from the proper definitions and methods in the written material.

3. As a ”probability calculus and statistical analysis machine” where e.g. with some data fed into it, it will, by inbuilt functions and procedures do all relevant computations for you and present the final results in some overview tables and plots.

We will see and present all three types of applications of Python during the course. For the first type, the aim is not to learn how to use the given Python-code itself but rather to learn from the insights that the code together with the results of applying it is providing. It will be stated clearly whenever a Python-example is of this type. Types 2 and 3 are specific tools that should be learned as a part of the course and represent tools that are explicitly relevant in your future engineering activity. It is clear that at some point one would love to just do the last kind of applications. However, it must be stressed that even though the program is able to calculate things for the user, understanding the details of the calculations must NOT be forgotten - understanding the methods and knowing the formulas is an important part of the syllabus, and will be checked at the exam.

|||| Remark 1.23 BRING and USE pen and paper PRIOR to using software

For many of the exercises that you are asked to do it will not be possible to just directly identify what Python-command(s) should be used to find the results. The exercises are often to be seen as what could be termed “problem mathematics” exercises. So, it is recommended to also bring and use pen and paper to work with the exercises to be able to subsequently know how to finally finish them by some Python-calculations. (If you adjusted yourself to some digital version of ”pen-and-paper”, then this is fine of course.)

|||| Remark 1.24 Python is not a substitute for your brain activity in this course!

The Python software should be seen as the most fantastic and easy computational companion that we can have for doing statistical computations that we could have done ”manually”, if we wanted to spend the time doing it. All definitions, formulas, methods, theorems etc. in the written material should be known by the student, as should also certain Python-routines and functions.

A good question to ask yourself each time that you apply en inbuilt Python-function is: ”Would I know how to make this computation ”manually”?”. There are few exceptions to this requirement in the course, but only a few. And for these the question would be: ”Do I really understand what Python is computing for me now?”

1.6 Plotting, graphics - data visualisation

A really important part of working with data analysis is the visualization of the raw data, as well as the results of the statistical analysis – the combination of the two leads to reliable results. Let us focus on the first part now, which can be seen as being part of the explorative descriptive analysis also mentioned in Section 1.4. Depending on the data at hand different types of plots and graphics could be relevant. One can distinguish between *quantitative* vs. *categorical* data. We will touch on the following type of basic plots:

- Quantitative data:
  - Frequency plots and histograms
  - box plots
  - cumulative distribution
  - Scatter plot (xy plot)
- Categorical data:
  - Bar charts
  - Pie charts

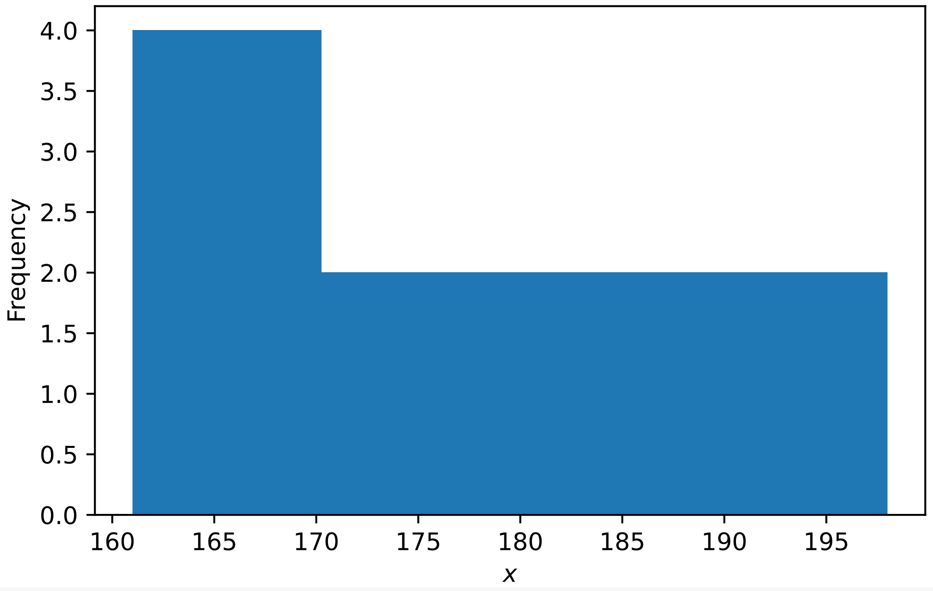
1.6.1 Frequency distributions and the histogram

The frequency distribution is the count of occurrences of values in the sample for different classes using some classification, for example in intervals or by some other property. It is nicely depicted by the histogram, which is a bar plot of the occurrences in each classes.

|||| Example 1.25 Histogram in Python

Consider again the $n = 10$ sample from Example 1.6.

```python
# A histogram of the heights
plt.hist(x)
plt.show()
```



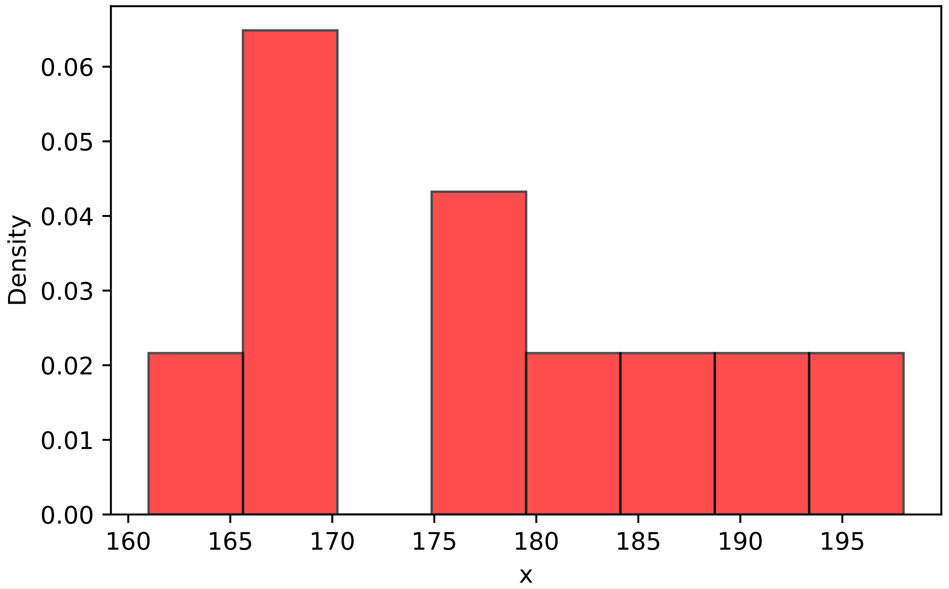
The default histogram uses equidistant interval widths (the same width for all intervals) and depicts the raw frequencies/counts in each interval. One may change the scale into showing what we will learn to be *densities* by dividing the raw counts by n and the interval width, i.e.

$$\frac{\text{"Interval count"}}{n \cdot (\text{"Interval width"})}.$$

By plotting the densities a density histogram also called the empirical density the area of all the bars add up to 1:

|||| Example 1.26 Empirical density in Python

```
# A density histogram or empirical density of the heights
plt.hist(x, bins=8, edgecolor=’black’, color=’red’, alpha=0.7,
density=True)
plt.show()
```



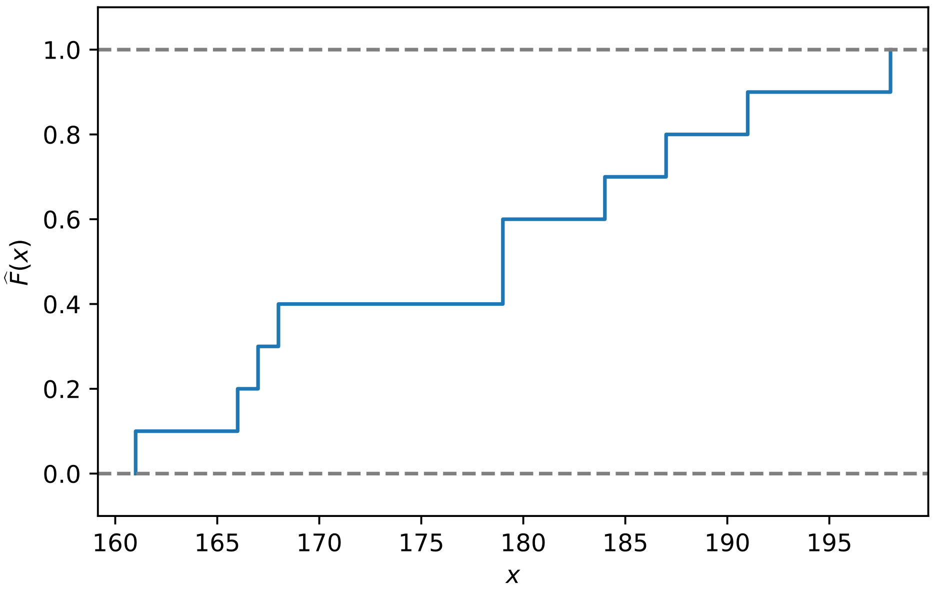
The function `hist` makes some choice of the number of classess based on the number of observations - it may be changed by the user option `nclass` as illustrated here.

1.6.2 Cumulative distributions

The cumulative distribution can be visualized simply as the cumulated relative frequencies either across classes, as also used in the histogram, or individual data points, which is then called the *empirical cumulative distribution function:*

|||| Example 1.27 Cumulative distribution plot in Python

```
# Empirical cumulative distribution plot
plt.ecdf(x)
plt.show()
```



The empirical cumulative distribution function F_n is a step function with jumps i/n at observation values, where i is the number of identical(tied) observations at that value.

For observations $(x_1, x_2, \ldots, x_n)$, $F_n(x)$ is the fraction of observations less or equal to x, that mathematically can be expressed as

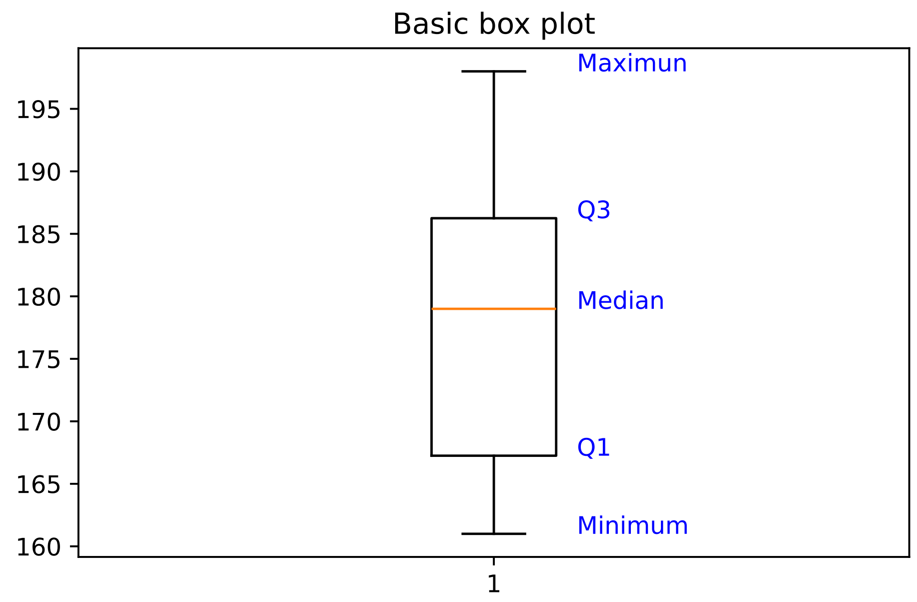
$$F_n(x) = \sum_{j \text{ where } x_j \leq x} \frac{1}{n}. \tag{1-13}$$

1.6.3 The box plot and the modified box plot

The so-called box plot in its basic form depicts the five quartiles (min, Q_1, median, Q_3, max) with a box from Q_1 to Q_3 emphasizing the Inter Quartile Range (IQR):

|||| Example 1.28 Box plot in Python

```
plt.boxplot(x)
plt.show()
```

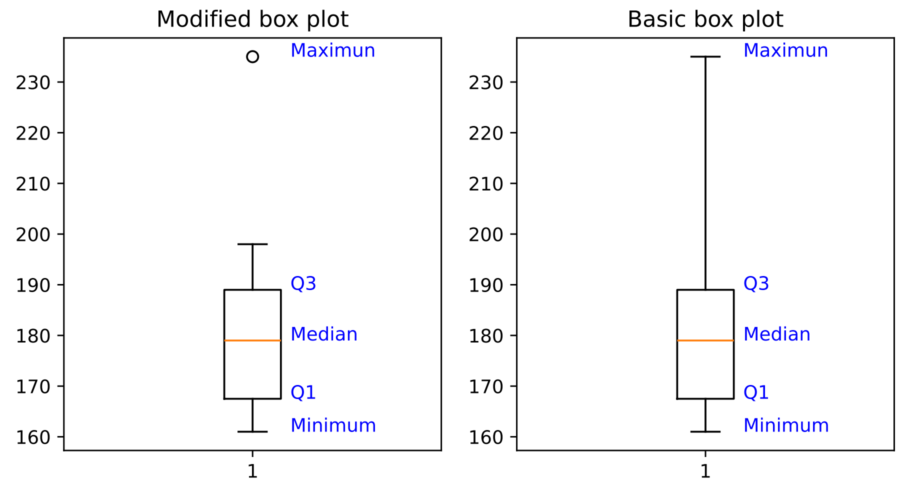


In the modified box plot the *whiskers* only extend to the min. and max. observation if they are not too far away from the box: defined to be $1.5 \times IQR$. Observations further away are considered as *extreme observations* (also called *fliers*) and will be plotted individually - hence the whiskers extend from the smallest to the largest observation within a distance of $1.5 \times IQR$ of the box (defined as either $1.5 \times IQR$ larger than Q_3 or $1.5 \times IQR$ smaller than Q_1).

|||| Example 1.29 Box plot in Python

If we add an extreme observation, 235 cm, to the heights sample and make the *modified box plot* - the default - and the basic *box plot*, then we have:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
# with default whiskers
ax1.boxplot(np.append(x, [235]))
# with whiskers covering 100% of the data:
ax2.boxplot(np.append(x, [235]), whis=(0,100))
plt.show()
```



Note that since there was no extreme observations among the original 10 observations, the two ”different” plots would be the same if we didn’t add the extreme 235 cm observation.

The box plot hence is an alternative to the histogram in visualising the distribution of the sample. It is a convenient way of comparing distributions in different groups, if such data is at hand.

Example 1.30 Box plot in Python

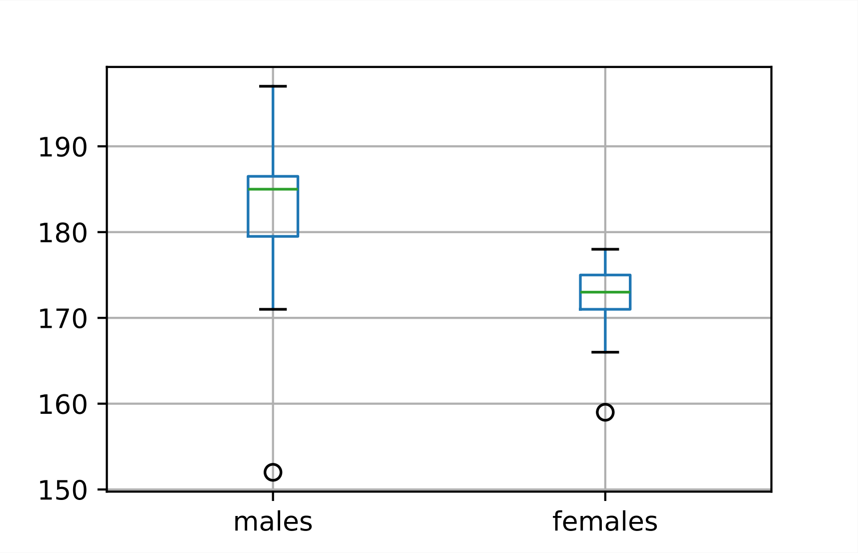
This example shows some ways of working with Python to illustrate data.

In another statistics course the following heights of 17 female and 23 male students were found:

Males	152	171	173	173	178	179	180	180	182	182	182	185
	185	185	185	185	186	187	190	190	192	192	197	
Females	159	166	168	168	171	171	172	172	173	174	175	175
	175	175	175	177	178							

When working with datasets in Python it is often useful to use the Pandas. Here we show an example of making a boxplot with the Pandas library (data is stored as a 'Pandas DataFrame'):

```
# Box plot with two groups
data = pd.DataFrame({
    'males':  [152, 171, 173, 173, 178, 179, 180, 180, 182, 182, 182,
               185, 185, 185, 185, 185, 186, 187, 190, 190, 192, 192,
               197],
    'females':[159, 166, 168, 168, 171, 171, 172, 172, 173, 174, 175,
               175, 175, 175, 175, 177, 178, np.nan, np.nan, np.nan,
               np.nan, np.nan, np.nan]
})
data.boxplot()
```



At this point, it should be noted that in real work with data using Python, one would generally not import data by explicit listings as we did above. This only works for very small data sets. Usually the data is imported from somewhere else, e.g. from a spread sheet exported in a `.csv` (*comma separated values*) format as shown here:

Example 1.31 Read and explore data in Python

The gender grouped student heights data used in Example 1.30 is available as a `.csv`-file via `http://www2.compute.dtu.dk/courses/introstat/data/studentheights.csv`. The structure of the data file, as it would appear in a spread sheet program (e.g. LibreOffice Calc or Excel) is two columns and 40+1 rows including a header row:

```
1 Height Gender
2    152 male
3    171 male
4    173 male
.      . .
.      . .
24   197 male
25   159 female
26   166 female
27   168 female
.      . .
.      . .
39   175 female
40   177 female
41   178 female
```

The data can now be imported into Python using the Pandas function `read_csv`:

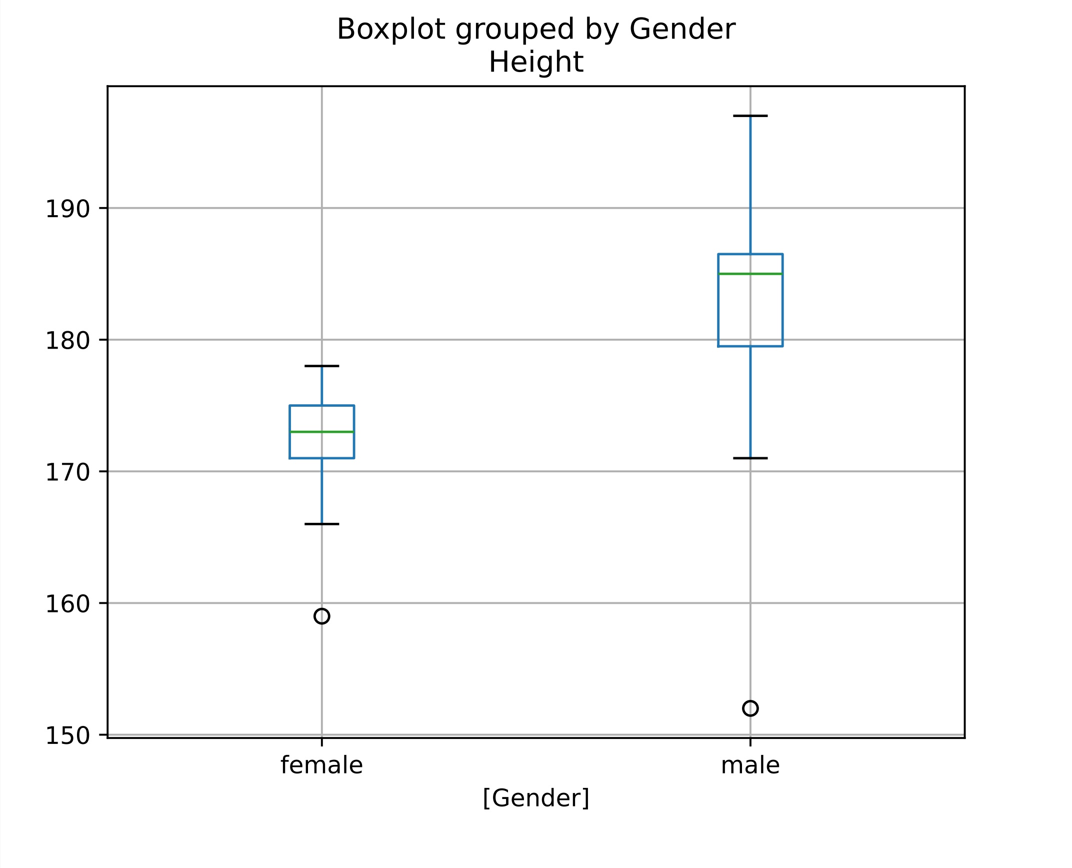
```
# Read the data (note that per default sep="," but here semicolon)
studentheights = pd.read_csv(’studentheights.csv’, sep=’;’)
```

The resulting object `studentheights` is now a Pandas `DataFrame`, which is very useful for working with tabular data in Python. There are some ways of getting a quick look at what kind of data is really in a `DataFrame`:

```
# Have a look at the first 6 rows of the data
studentheights.head()

   Height Gender
0     152   male
1     171   male
2     173   male
3     173   male
4     178   male


# Get an overview
studentheights.info(verbose=True)

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Height  40 non-null     int64
 1   Gender  40 non-null     object
dtypes: int64(1), object(1)
memory usage: 768.0+ bytes


# Get a summary of each column/variable in the data
studentheights.describe(include=’all’)

           Height Gender
count    40.00000     40
unique        NaN      2
top           NaN   male
freq          NaN     23
mean    177.87500    NaN
std       9.09265    NaN
min     152.00000    NaN
25%     172.75000    NaN
50%     177.50000    NaN
75%     185.00000    NaN
max     197.00000    NaN
```

The `describe` functions outputs different statistics for numeric and non-numeric columns (see documentation for `Pandas.DataFrame.describe`). A data structure like this is commonly encountered (and often the only needed) for statistical analysis. The gender grouped box plot can now be generated by:

```
# Box plot for each gender
studentheights.boxplot(by='Gender')
```



1.6.4 The Scatter plot

The scatter plot can be used for two quantitative variables. It is simply one variable plotted versus the other using some plotting symbol.

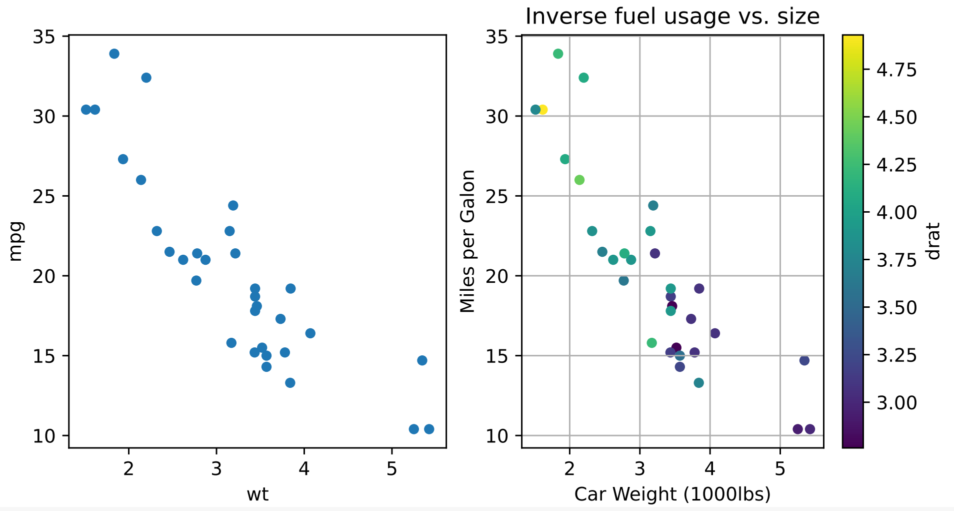
|||| Example 1.32 Explore `mtcars` data from R

Now we will use a data set which is available in the Statsmodels Library, and which is originally from the programming language R. Here we will use the `mtcars` data set, but there are many other data sets available, that may be useful for practising and testing. If you read the documentation you will find the following information about the `mtcars` dataset:

"The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models)."

Let us plot the gasoline use, (mpg=miles pr. gallon), versus the weight (wt):

```
# get mtcars data as a DataFrame
mtcars = sm.datasets.get_rdataset(’mtcars’).data
type(mtcars)

<class 'pandas.core.frame.DataFrame'>


# To make 2 plots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
# First the default version
mtcars.plot.scatter(’wt’, ’mpg’, ax=ax1)
# Then a nicer version
mtcars.plot.scatter(’wt’, ’mpg’, c=’drat’, colormap=’viridis’,
                    title=’Inverse fuel usage vs. size’,
                    xlabel = ’Car Weight (1000lbs)’,
                    ylabel=’Miles per Galon’,
                    grid=True, ax=ax2)
```



In the second plot we have added a third variable (the column ’`drat`’, rear axle ratio) using a colorbar. Notice also how the plotting commands are a little bit different as we are now using both Matplotlib and Pandas. There are many other ways to plot in Python and we recommend that you practice using the internet and relevant documentation to get help.

1.6.5 Bar plots and Pie charts

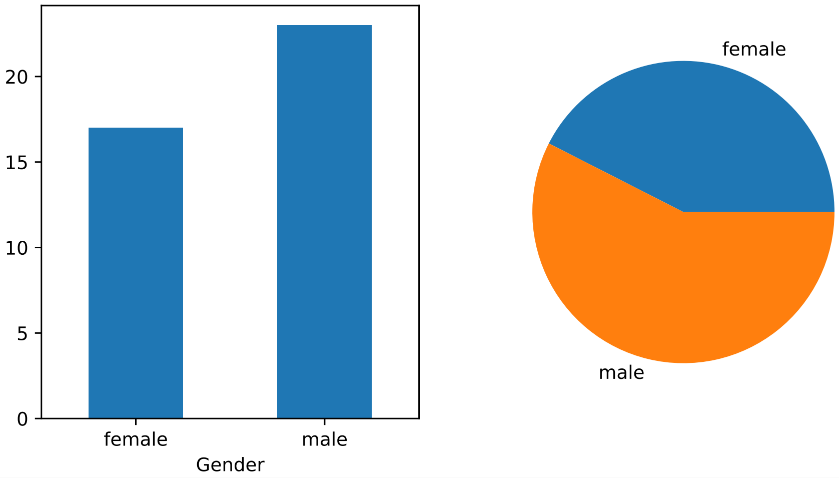
All the plots described so far were for quantitative variables. For categorical variables the natural basic plot would be a bar plot or pie chart visualizing the relative frequencies in each category.

|||| Example 1.33 Bar plots and Pie charts

For the gender grouped student heights data used in Example 1.30 we can produce a table with counts of each gender and plot the gender distribution by:

```
# Table
studentheights.groupby('Gender').size()

Gender
female    17
male      23
dtype: int64
```

```
# Barplot and Pie chart
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
studentheights.groupby('Gender').size().plot(kind='bar', ax=ax1, rot=0)
studentheights.groupby('Gender').size().plot(kind='pie', ax=ax2)
```



Notice that the table produced by 'studentheights.groupby('Gender').size()' return a Pandas Series, which is slightly different from a Pandas DataFrame (the main difference here is that a Series is a one-dimensional labeled array and a DataFrame is a two-dimensional labeled data structure).

1.6.6 More plots in Python

Python has many libraries that can produce statistical plots. In this course we will mostly use Matplotlib, but you can also explore the library Seaborn: https://seaborn.pydata.org/.

Chapter 2

Probability and simulation

In this chapter elements from probability theory are introduced. These are needed to form the basic mathematical description of randomness. For example for calculating the probabilities of outcomes in various types of experimental or observational study setups. Small illustrative examples, such as e.g. dice rolls and lottery draws, and natural phenomena such as the waiting time between radioactive decays are used as throughout. But the scope of probability theory and it's use in society, science and business, not least engineering endeavour, goes way beyond these small examples. The theory is introduced together with illustrative code examples, which the reader is encouraged to try and interact with in parallel to reading the text. Many of these are of the learning type, cf. the discussion of the way Python is used in the course in Section 1.5.

2.1 Random variable

The basic building blocks to describe random outcomes of an experiment are introduced in this section. The definition of an *experiment* is quite broad. It can be an experiment, which is carried out under controlled conditions e.g. in a laboratory or flipping a coin, as well as an experiment in conditions which are not controlled, where for example a process is observed e.g. observations of the GNP or measurements taken with a space telescope. Hence, an experiment can be thought of as any setting in which the outcome cannot be fully known. This for example also includes measurement noise, which are random "errors" related to the system used to observe with, maybe originating from noise in electrical circuits or small turbulence around the sensor. Measurements will always contain some noise.

First the *sample space* is defined:

|||| Definition 2.1

The *sample space* S is the set of all possible outcomes of an experiment.

|||| Example 2.2

Consider an experiment in which a person will throw two paper balls with the purpose of hitting a wastebasket. All the possible outcomes forms the sample space of this experiment as

$$S = \{(\text{miss,miss}), (\text{hit,miss}), (\text{miss,hit}), (\text{hit,hit})\}. \tag{2-1}$$

Now a *random variable* can be defined:

|||| Definition 2.3

A *random variable* is a function which assigns a numerical value to each outcome in the sample space. In this book random variables are denoted with capital letters, e.g.

$$X, Y, \ldots . \tag{2-2}$$

|||| Example 2.4

Continuing the paper ball example above, a random variable can be defined as the number of hits, thus

$$X\big((\text{miss,miss})\big) = 0, \tag{2-3}$$
$$X\big((\text{hit,miss})\big) = 1, \tag{2-4}$$
$$X\big((\text{miss,hit})\big) = 1, \tag{2-5}$$
$$X\big((\text{hit,hit})\big) = 2. \tag{2-6}$$

In this case the random variable is a function which maps the sample space S to positive integers, i.e. $X : S \rightarrow \mathbb{N}_0$.

Remark 2.5

The random variable represents a value of the outcome *before* the experiment is carried out. Usually the experiment is carried out n times and there are random variables for each of them

$$\{X_i : 1, 2, \ldots, n\}. \tag{2-7}$$

After the experiment has been carried out n times a set of values of the random variable is available as

$$\{x_i : 1, 2, \ldots, n\}. \tag{2-8}$$

Each value is called a *realization* or *observation* of the random variable and is denoted with a small letter sub-scripted with an index i, as introduced in Chapter 1.

Finally, in order to quantify probability, a random variable is associated with a *probability distribution*. The distribution can either be discrete or continuous depending on the nature of the outcomes:

- Discrete outcomes can for example be: the outcome of a dice roll, the number of children per family, or the number of failures of a machine per year. Hence some countable phenomena which can be represented by an integer.
- Continuous outcomes can for example by: the weight of the yearly harvest, the time spend on homework each week, or the electricity generation per hour. Hence a phenomena which can be represented by a continuous value.

Furthermore, the outcome can either be unlimited or limited. This is most obvious in the discrete case, e.g. a dice roll is limited to the values between 1 and 6. However it is also often the case for continuous random variables, for example many are non-negative (weights, distances, etc.) and proportions are limited to a range between 0 and 1.

Conceptually there is no difference between the discrete and the continuous case, however it is easier to distinguish since the formulas, which in the discrete case are with sums, in the continuous case are with integrals. In the remaining of this chapter, first the discrete case is presented and then the continuous.

2.2 Discrete random variables

In this section discrete distributions and their properties are introduced. A discrete random variable has discrete outcomes and follows a discrete distribution.

To exemplify, consider the outcome of one roll of a fair six-sided dice as the random variable X^{fair}. It has six possible outcomes, each with equal probability. This is specified with the *probability density function*.

Definition 2.6 The *pdf* of a discrete random variable

For a discrete random variable X the *probability density function* (*pdf*) is

$$f(x) = P(X = x). \tag{2-9}$$

It assigns a probability to every possible outcome value x.
A discrete *pdf* fulfils two properties: there are no negative probabilities for any outcome value

$$f(x) \geq 0 \text{ for all } x, \tag{2-10}$$

and the probabilities for all outcome values sum to one

$$\sum_{\text{all } x} f(x) = 1. \tag{2-11}$$

Example 2.7

For the fair dice the *pdf* is

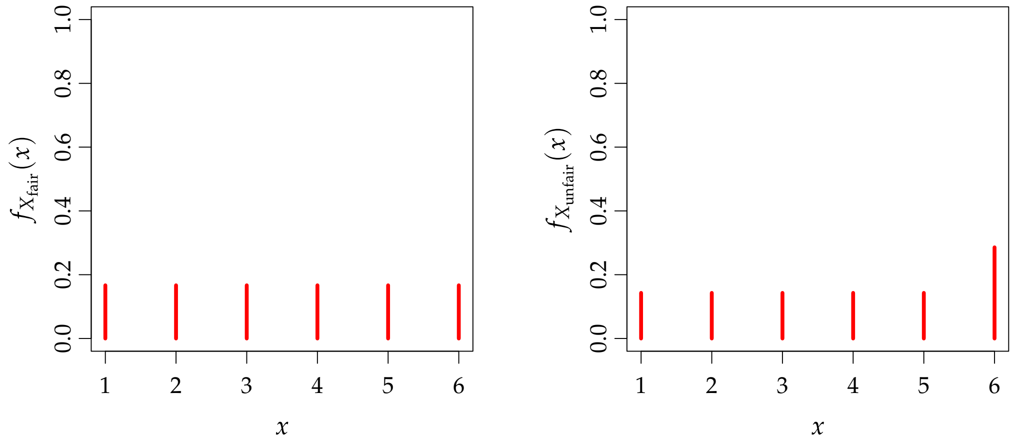
x	1	2	3	4	5	6
$f_{X^{\text{fair}}}(x)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

If the dice is not fair, maybe it has been modified to increase the probability of rolling a six, the *pdf* could for example be

x	1	2	3	4	5	6
$f_{X^{\text{unfair}}}(x)$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{2}{7}$

where X^{unfair} is a random variable representing the value of a roll with the unfair dice.

The pdfs are plotted: the left plot shows the pdf of a fair dice and the right plot the pdf of an unfair dice:



|||| Remark 2.8

Note that the *pdf*s has subscript with the symbol of the random variable to which they belong. This is done when there is a need to distinguish between *pdf*s e.g. for several random variables. For example if two random variables X and Y are used in same context, then: $f_X(x)$ is the *pdf* for X and $f_Y(x)$ for Y, similarly the sample standard deviation s_X is for X and s_Y is for Y, and so forth.

The *cumulated distribution function (cdf)*, or simply the *distribution function*, is often used.

|||| Definition 2.9 The *cdf* of a discrete random variable

The *cumulated distribution function* (*cdf*) for the discrete case is the probability of realizing an outcome below or equal to the value x

$$F(x) = P(X \leq x) = \sum_{j \text{ where } x_j \leq x} f(x_j) = \sum_{j \text{ where } x_j \leq x} P(X = x_j). \qquad \text{(2-12)}$$

The probability that the outcome of X is in a range is

$$P(a < X \leq b) = F(b) - F(a). \qquad \text{(2-13)}$$

For the fair dice the probability of an outcome below or equal to 4 can be calculated

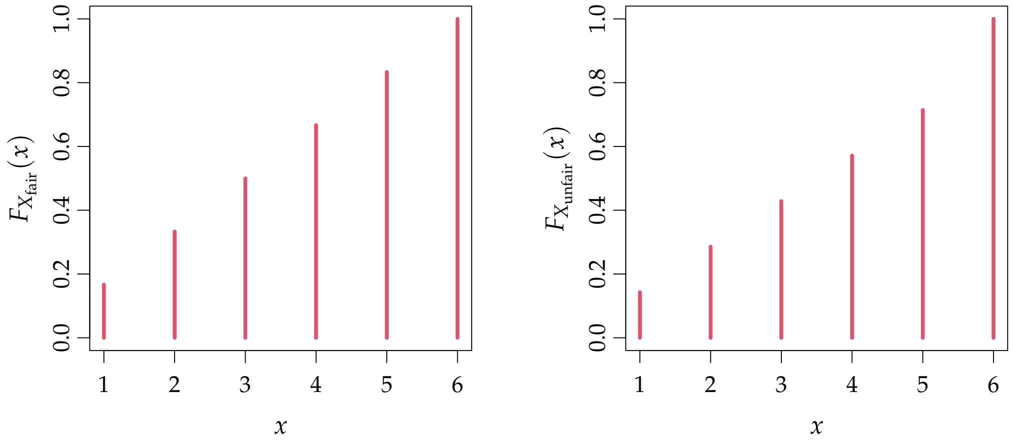
$$F_{X^{\text{fair}}}(4) = \sum_{j=1}^{4} f_{X^{\text{fair}}}(x_j) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{2}{3}. \tag{2-14}$$

Example 2.10

For the fair dice the *cdf* is

x	1	2	3	4	5	6
$F_{X^{\text{fair}}}(x)$	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1

The cdf for a fair dice is plotted in the left plot and the cdf for an unfair dice is plotted in the right plot:



2.2.1 Introduction to simulation

One nice thing about having computers available is that we try things in virtual reality - this we can here use here to play around while learning how probability and statistics work. With the *pdf* defined an experiment can easily be *simulated*, i.e. instead of carrying out the experiment in reality it is carried out using a model on the computer. When the simulation includes generating random numbers it is called a *stochastic simulation*. Such simulation tools are readily available within Python, and it can be used for as well learning purposes as a way to do large scale complex probabilistic and statistical computations. For now it will be used in the first way.

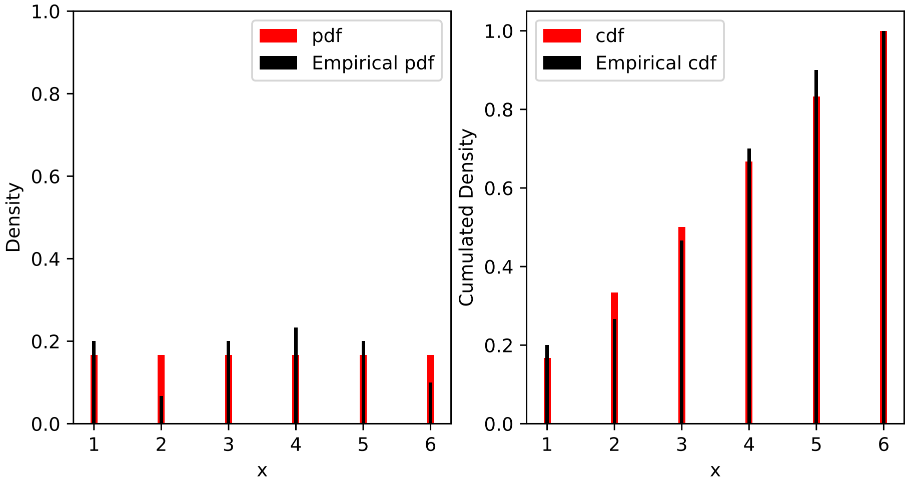
Example 2.11 Simulation of rolling a dice

Let’s simulate the experiment of rolling a dice using the following

```
# Make a random draw from (1,2,3,4,5,6) with equal probability for each
outcome
np.random.choice(range(1, 7), size=1)
```

The simulation becomes more interesting when the experiment is repeated many times, then we have a sample and can calculate the *empirical density function* (or *empirical pdf or density histogram*, see Section 1.6.1) as a discrete histogram and actually “see” the shape of the *pdf*

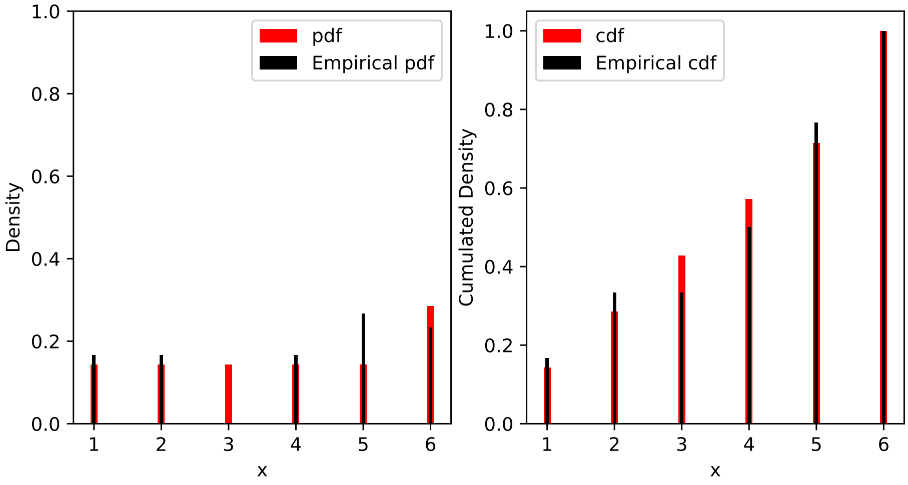
```
# Simulate a fair dice
# Number of simulated realizations
n = 30
# Draw independently from the set (1,2,3,4,5,6) with equal probability
xFair = np.random.choice(range(1, 7), size=n, replace=True)
# Count the number of each outcome using the bincount function
counts = np.bincount(xFair)
# Plot the pdf
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.bar(range(1,7), [1/6]*6, color=’red’)
# Plot the empirical pdf
ax1.bar(range(1,7), counts[1:7]/n)
# Plot the cdf
ax2.bar(range(1,7), np.cumsum([1/6]*6), color=’red’)
# Add the empirical cdf
ax2.bar(range(1,7), np.cumsum(counts[1:7]/n))
```



Try simulating with different number of rolls *n* and describe how this affects the accuracy of the empirical *pdf* compared to the *pdf*?

Now repeat this with the unfair dice

```
# Simulate an unfair dice
# Number of simulated realizations
n = 30
# Draw independently from the set (1,2,3,4,5,6) with higher
# probability for a six
probs = [1/7, 1/7, 1/7, 1/7, 1/7, 2/7]
xUnfair = np.random.choice(range(1, 7), size=n, replace=True, p=probs)
counts = np.bincount(xUnfair)
# Plot the pdf
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.bar(range(1,7), probs, color='red')
# Plot the empirical pdf
ax1.bar(range(1,7), counts[1:7]/n)
# Plot the cdf
ax2.bar(range(1,7), np.cumsum(probs), color='red')
# Add the empirical cdf
ax2.bar(range(1,7), np.cumsum(counts[1:7]/n))
```



Compare the fair and the unfair dice simulations:

How did the empirical *pdf* change?

By simply observing the empirical *pdf* can we be sure to distinguish between the fair and the unfair dice?

How does the number of rolls *n* affect how well we can distinguish the two dices?

One reason to simulate becomes quite clear here: it would take considerably more time to actually carry out these experiments. Furthermore, sometimes calculating the theoretical properties of random variables (e.g. products of several random variables etc.) are impossible and simulations can be a useful way to obtain such results.

Random number sequences generated with software algorithms have the properties of real random numbers, e.g. they are independent, but are in fact deterministic sequences depending on a *seed*, which sets an initial value of the sequence. Therefore they are named *pseudo random numbers*, since they behave like and are used as random numbers in simulations, but are in fact deterministic sequences.

Remark 2.12 Random numbers and seed in Python

In Python the initial values can be set with a single number called the *seed* as demonstrated with the following Python code. As default the seed is created from the time of start-up of a new instance of Python. A way to generate truly (i.e. non-pseudo) random numbers can be to sample some physical phenomena, for example atmospheric noise as done at www.random.org.

```python
# The random numbers generated depends on the seed

# Set the seed
np.random.seed(127)
# Generate a (pseudo) random sequence
print(np.random.rand(10))
```

```
[0.524 0.040 0.186 0.773 0.552 0.086 0.441 0.716 0.671 0.473]
```

```python
# Generate again and see that new numbers are generated
print(np.random.rand(10))
```

```
[0.906 0.105 0.175 0.089 0.650 0.071 0.460 0.907 0.094 0.633]
```

```python
# Set the seed and the same numbers as before just after the
# seed was set are generated
np.random.seed(127)
print(np.random.rand(10))
```

```
[0.524 0.040 0.186 0.773 0.552 0.086 0.441 0.716 0.671 0.473]
```

2.2.2 Mean and variance

In Chapter 1 the *sample mean* and the *sample variance* were introduced. They indicate respectively the centring and the spread of the observations in a sample. In this section the *mean* and *variance* are introduced. They are properties of the distribution of a random variable, they are called *population parameters*. The mean indicates where the distribution is centred. The variance indicates the spread of the distribution.

Mean and expected value

The *mean* (μ) of a random variable is the population parameter which most statistical analysis focus on. It is formally defined as a function $\mathrm{E}(X)$: the *expected value* of the random variable X.

|||| Definition 2.13 Mean value

The mean of a discrete random variable X is

$$\mu = \mathrm{E}(X) = \sum_{j=1}^{\infty} x_j f(x_j), \tag{2-15}$$

where x_j is the value and $f(x_j)$ is the probability that X takes the outcome value x_j.

The mean is simply the weighted average over all possible outcome values, weighted with the corresponding probability. As indicated in the definition there might be infinitely many possible outcome values, hence, even if the total sum of probabilities is one, then the probabilities must go sufficiently fast to zero for increasing values of X in order for the sum to be defined.

|||| Example 2.14

For the fair dice the mean is calculated by

$$\mu_{x^{\text{fair}}} = \mathrm{E}(X^{\text{fair}}) = 1\frac{1}{6} + 2\frac{1}{6} + 3\frac{1}{6} + 4\frac{1}{6} + 5\frac{1}{6} + 6\frac{1}{6} = 3.5,$$

for the unfair dice the mean is

$$\mu_{x^{\text{unfair}}} = \mathrm{E}(X^{\text{unfair}}) = 1\frac{1}{7} + 2\frac{1}{7} + 3\frac{1}{7} + 4\frac{1}{7} + 5\frac{1}{7} + 6\frac{2}{7} \approx 3.86.$$

The mean of a random variable express the limiting value of an average of many outcomes. If a fair dice is rolled a really high number of times the sample mean of these will be very close to 3.5. For the statistical reasoning related to the use of a sample mean as an estimate for μ, the same property ensures that envisioning many sample means (with the same n), a meta like thinking, then the mean of such many repeated sample means will be close to μ.

After an experiment has been carried out n times then the *sample mean* or *average* can be calculated as previously defined in Chapter 1

$$\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{i}^{n} x_i. \tag{2-16}$$

It is called a *statistic*, which means that it is calculated from a sample. Note the use of a hat in the notation over μ: this indicates that it is an *estimate* of the real underlying mean.

Our intuition tells us that the estimate ($\hat{\mu}$) will be close to true underlying expectation (μ) when n is large. This is indeed the case, to be more specific $\mathrm{E}\left[\frac{1}{n}\sum X_i\right] = \mu$ (when $\mathrm{E}[X_i] = \mu$), and we say that the average is a central estimator for the expectation. The exact quantification of these qualitative statements will be covered in Chapter 3.

Now play a little around with the mean and the sample mean with some simulations.

|||| Example 2.15 Simulate and estimate the mean

Carrying out the experiment more than one time an estimate of the mean, i.e. the sample mean, can be calculated. Simulate rolling the fair dice

```
# Number of realizations
n = 30
# Simulate rolls with a fair dice
xFair = np.random.choice(range(1, 7), size=n, replace=True)
# Calculate the sample mean
xFair.sum()/n

np.float64(3.3333333333333335)
```

```
# or
xFair.mean()

np.float64(3.3333333333333335)
```

Let us see what happens with the sample mean of the unfair dice by simulating the same number of rolls

```
# Simulate an unfair dice

# n realizations
probs = [1/7, 1/7, 1/7, 1/7, 1/7, 2/7]  # Higher probability for a six
xUnfair = np.random.choice(range(1, 7), size=n, replace=True, p=probs)
# Calculate the sample mean
xUnfair.mean()

np.float64(4.166666666666667)
```

Consider the mean of the unfair dice and compare it to the mean of the fair dice (see Example 2.14). Is this in accordance with your simulation results?

Let us again turn to how much we can "see" from the simulations and the impact of the number of realizations n on the estimation. In statistics the term *information* is used to refer to how much information is embedded in the data, and therefore how accurate different properties (parameters) can be estimated from the data.

Repeat the simulations several times with $n = 30$. By simply comparing the sample means from a single simulation can it then be determined if the two means really are different?

Repeat the simulations several times and increase n. What happens with to the 'accuracy' of the sample mean compared to the real mean? and thereby how well it can be inferred if the sample means are different?

Does the information embedded in the data increase or decrease when n is increased?

Variance and standard deviation

The second most used population parameter is the *variance* (or standard deviation). It is a measure describing the spread of the distribution, more specifically the spread away from the mean.

|||| Definition 2.16 Variance

The variance of a discrete random variable X is

$$\sigma^2 = \mathrm{V}(X) = \mathrm{E}[(X-\mu)^2] = \sum_{i=1}^{\infty}(x_i-\mu)^2 f(x_i), \qquad \text{(2-17)}$$

where x_i is the outcome value and $f(x_i)$ is the *pdf* of the ith outcome value. The *standard deviation* σ is the square root of the variance.

The variance is the expected value (i.e. average (weighted by probabilities)) of the squared distance between the outcome and the mean value.

|||| Remark 2.17

Notice that the variance cannot be negative.

The standard deviation is measured on the same scale (same units) as the random variable, which is not case for the variance. Therefore the standard deviation can much easier be interpreted, when communicating the spread of a distribution.

Consider how the expected value is calculated in Equation (2-15). One can think of the squared distance as a new random variable that has an expected value which is the variance of X.

|||| Example 2.18

The variance of rolls with the fair dice is

$$\begin{aligned}\sigma^2_{x^{\text{fair}}} &= \mathrm{E}[(X^{\text{fair}} - \mu_{X^{\text{fair}}})^2] \\ &= (1-3.5)^2\frac{1}{6} + (2-3.5)^2\frac{1}{6} + (3-3.5)^2\frac{1}{6} + (4-3.5)^2\frac{1}{6} + (5-3.5)^2\frac{1}{6} + (6-3.5)^2\frac{1}{6} \\ &= \frac{70}{24} \\ &\approx 2.92.\end{aligned}$$

It was seen in Chapter 1, that after an experiment has been carried out n times the *sample variance* can be calculated as defined previously by

$$s^2 = \hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2, \tag{2-18}$$

and hence thereby also *sample standard deviation* s.

Again our intuition tells us that the statistic (e.g. *sample variance*), should in some sense converge to the true variance - this is indeed the case and the we call the sample variance a central estimator for the true underlying variance. This convergence will be quantified for a special case in Chapter 3.

The sample variance is calculated by:

- Take the sample mean: $\bar{x}$
- Take the distance for each sample: $x_i - \bar{x}$
- Finally, take the average of the squared distances (using $n - 1$ in the denominator, see Chapter 1)

|||| Example 2.19 Simulate and estimate the variance

Return to the simulations. First calculate the sample variance from n rolls of a fair dice

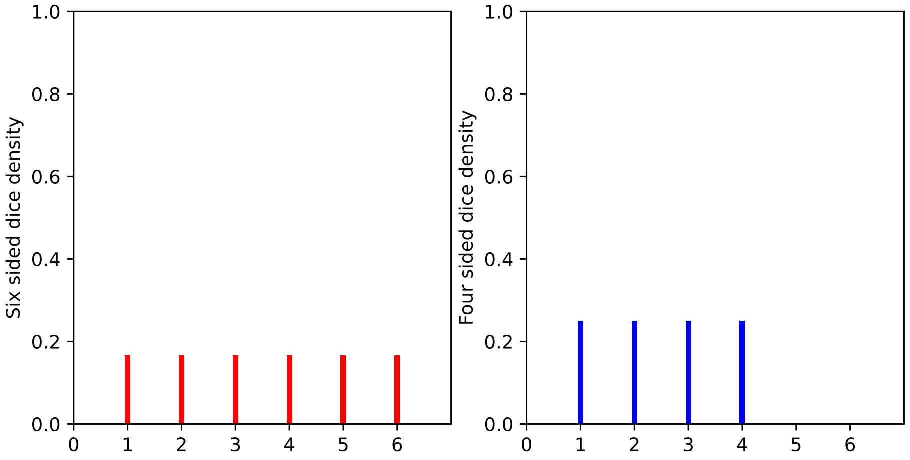
```
# Simulate a fair dice and calculate the sample variance

# Number of realizations
n = 30
# Simulate
xFair = np.random.choice(range(1,7), size=n, replace=True)
# Calculate the distance for each sample to the sample mean
distances = xFair - xFair.mean()
# Calculate the average of the squared distances
sum(distances**2)/(n-1)

np.float64(2.791954022988505)


# Or use the built in function
xFair.var(ddof=1)

np.float64(2.7919540229885054)
```

Let us then try to play with variance in the dice example. Let us now consider a four-sided dice. The *pdf* is

x	1	2	3	4
$F_{X\text{fairFour}}(x)$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Plot the *pdf* for both the six-sided dice and the four-sided dice

```
# Plot the pdf of the six-sided dice and the four-sided dice
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.bar(range(1, 7), [1/6] * 6, color='red')
ax2.bar(range(1, 5), [1/4] * 4, color='blue')
```



```
# Calculate the means and variances of the dices

# The means
muXSixsided = np.sum(np.array([1,2,3,4,5,6])*1/6)
muXFoursided = np.sum(np.array([1,2,3,4])*1/4)
# The variances
print(np.sum((np.array([1,2,3,4,5,6]) - muXSixsided)**2 * 1/6))

2.9166666666666667


print(np.sum((np.array([1,2,3,4]) - muXFoursided)**2 * 1/4))

1.25
```

Which dice outcome has the highest variance? is that as you had anticipated?

2.3 Discrete distributions

In this section the discrete distributions included in the material are presented. See the overview of all distributions in the collection of formulas Section A.2.1.

In the Python library Scipy, implementations of many different distributions are available. For each distribution at least the following is available

- The *pdf* is available by using `'.pmf()'`, e.g. for the binomial distribution `scipy.stats.binom.pmf()` (use `'.pmf()'` for discrete cases and `'.pdf()'` for continuous cases)
- The *cdf* is available by using `'.cdf()'`, e.g. `scipy.stats.binom.cdf()`
- The quantiles by using `'.ppf()'`, e.g. `scipy.stats.binom.ppf()`
- Random number generation by using `'.rvs()'`, e.g. `scipy.stats.binom.rvs()`

Examples of these functions are demonstrated below in this section for the discrete and later for the continuous distributions, see them demonstrated for the normal distribution in Example 2.45.

2.3.1 Binomial distribution

The binomial distribution is a very important discrete distribution and appears in many applications, it is presented in this section. In statistics it is typically used for proportions as explained in Chapter 7.

If an experiment has two possible outcomes (e.g. failure or success, no or yes, 0 or 1) and is repeated more than one time, then the number of successes may be binomial distributed. For example the number of heads obtained after a certain number of flips with a coin. Each repetition must be independent. In relation to random sampling this corresponds to successive draws with replacement (think of drawing notes from a hat, where after each draw the note is put back again, i.e. the drawn number is replaced again).

Definition 2.20 Binomial distribution

Let the random variable X be binomial distributed

$$X \sim B(n, p), \tag{2-19}$$

where n is number of independent draws and p is the probability of a success in each draw.
The binomial *pdf* describes probability of obtaining x successes

$$f(x; n, p) \;=\; P(X = x) \;=\; \binom{n}{x} p^x (1-p)^{n-x}, \tag{2-20}$$

where

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}, \tag{2-21}$$

is the number of distinct sets of x elements which can be chosen from a set of n elements. Remember that $n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$.

Theorem 2.21 Mean and variance

The mean of a binomial distributed random variable is

$$\mu = np, \tag{2-22}$$

and the variance is

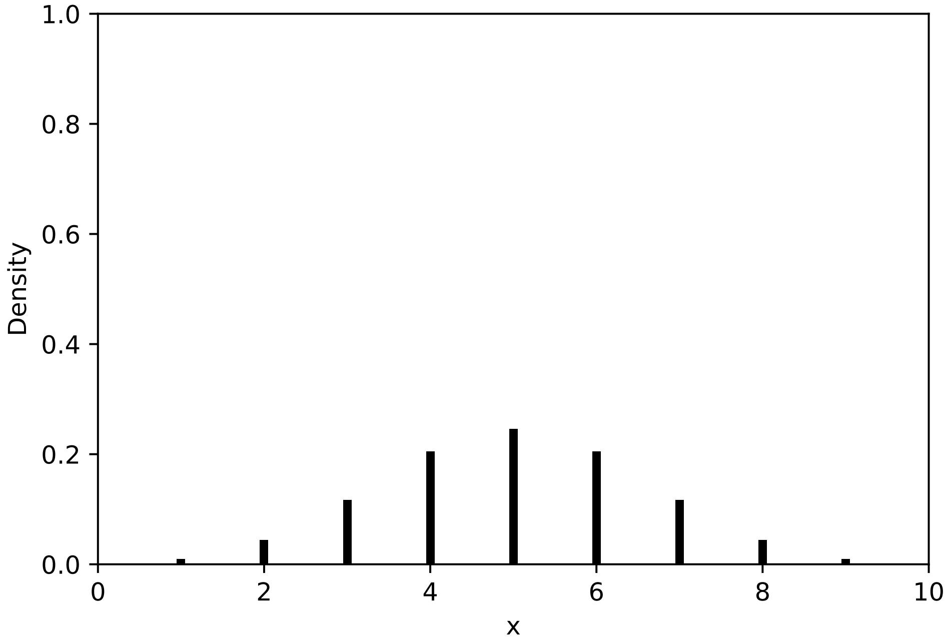
$$\sigma^2 = np(1-p). \tag{2-23}$$

Actually this can be proved by calculating the mean using Definition 2.13 and the variance using Definition 2.16.

Example 2.22 Simulation with a binomial distribution

The binomial distribution for 10 flips with a coin describe probabilities of getting x heads (or equivalently tails)

```
# Simulate a binomial distributed experiment

# Number of flips
nFlips = 10
# The possible outcomes are (0,1,...,nFlips)
xSeq = list(range(0,nFlips))
# Use the binom.pmf() function which returns the pdf
pdfSeq = stats.binom.pmf(xSeq, nFlips, 0.5)
# Plot the density
plt.bar(xSeq, pdfSeq, color=’black’,  width=0.1)
```



Example 2.23 Simulate 30 successive dice rolls

In the previous examples successive rolls of a dice was simulated. If a random variable which counts the number of sixes obtained X^{six} is defined, it follows a binomial distribution

```
# Simulate 30 successive dice rolls
Xfair = np.random.choice(range(1,7), size=30, replace=True)
# Count the number sixes obtained
sum(Xfair==6)
```

```
np.int64(5)
```

```
# This is equivalent to
stats.binom.rvs(n=30, p=1/6)
```

```
4
```

2.3.2 Hypergeometric distribution

The hypergeometric distribution describes number of successes from successive draws without replacement.

|||| Definition 2.24 Hypergeometric distribution

Let the random variable X be the number of successes in n draws without replacement. Then X follows the hypergeometric distribution

$$X \sim H(n, a, N), \tag{2-24}$$

where a is the number of successes in the N elements large population. The probability of obtaining x successes is described by the hypergeometric *pdf*

$$f(x; n, a, N) = P(X = x) = \frac{\binom{a}{x}\binom{N-a}{n-x}}{\binom{N}{n}}. \tag{2-25}$$

The notation

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}, \tag{2-26}$$

represents the number of distinct sets of b elements which can be chosen from a set of a elements.

|||| Theorem 2.25 Mean and variance

The mean of a hypergeometric distributed random variable is

$$\mu = n\frac{a}{N}, \tag{2-27}$$

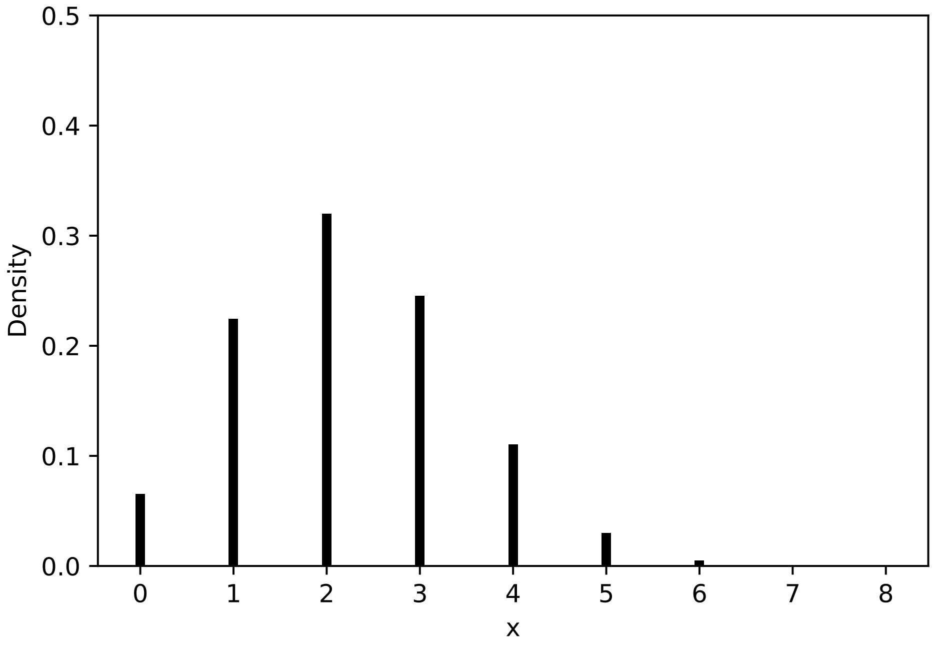
and the variance is

$$\sigma^2 = n\frac{a(N-a)}{N^2}\frac{N-n}{N-1}. \tag{2-28}$$

|||| Example 2.26 Lottery probabilities using the hypergeometric distribution

A lottery drawing is a good example where the hypergeometric distribution can be applied. The numbers from 1 to 90 are put in a bowl and randomly drawn without replacement (i.e. without putting back the number when it has been drawn). Say that you have the sheet with 8 numbers and want to calculate the probability of getting all 8 numbers in 25 draws.

```
# The probability of getting x numbers of the sheet in 25 drawings

# Number of successes in the population
a = 8
# Size of the population
N = 90
# Number of draws
n = 25
# Plot the pdf, note: parameters names are different in the
python-function (here using hypergeom)
plt.bar(np.arange(0,9), stats.hypergeom.pmf(np.arange(0,9), N, a, n),
color='black',  width=0.1)
```



2.3.3 Poisson distribution

The Poisson distribution describes the probability of a given number of events occurring in a fixed interval if these events occur with a known average rate and independently of the distance to the last event. Often it is events in a time interval, but can as well be counts in other intervals, e.g. of distance, area or volume. In statistics the Poisson distribution is usually applied for analyzing for example counts of: arrivals, traffic, failures and breakdowns.

Definition 2.27 Poisson distribution

Let the random variable X be Poisson distributed

$$X \sim Po(\lambda), \tag{2-29}$$

where λ is the rate (or intensity): the average number of events per interval. The Poisson *pdf* describes the probability of x events in an interval

$$f(x;\lambda) = \frac{\lambda^x}{x!}e^{-\lambda}. \tag{2-30}$$

Theorem 2.28 Mean and variance

A Poisson distributed random variable X has exactly the rate λ as the mean

$$\mu = \lambda, \tag{2-31}$$

and variance

$$\sigma^2 = \lambda. \tag{2-32}$$

Example 2.29

The Poisson distribution is typically used to describe phenomena such as:

- the number radioactive particle decays per time interval, i.e. the number of clicks per time interval of a Geiger counter
- calls to a call center per time interval (λ does vary over the day)
- number of mutations in a given stretch of DNA after a certain amount of radiation
- goals scored in a soccer match

One important feature is that the rate can be scaled, such that probabilities of occurrences in other interval lengths can be calculated. Usually the rate is denoted with the interval length, for example the hourly rate is denoted as λ^{hour}

and can be scaled to the minutely rate by

$$\lambda^{\text{minute}} = \frac{\lambda^{\text{hour}}}{60}, \tag{2-33}$$

such the probabilities of x events per minute can be calculated with the Poisson *pdf* with rate λ^{minute}.

Example 2.30 Rate scaling

You are enjoying a soccer match. Assuming that the scoring of goals per match in the league is Poisson distributed and on average 3.4 goals are scored per match. Calculate the probability that no goals will be scored while you leave the match for 10 minutes.

Let $\lambda^{\text{90minutes}} = 3.4$ be goals per match and scale this to the 10 minute rate by

$$\lambda^{\text{10minutes}} = \frac{\lambda^{\text{90minutes}}}{9} = \frac{3.4}{9}. \tag{2-34}$$

Let X be the number of goals in 10 minute intervals and use this to calculate the probability of no events a 10 minute interval by

$$P(X = 0) = f(0, \lambda^{\text{10minutes}}) \approx 0.685, \tag{2-35}$$

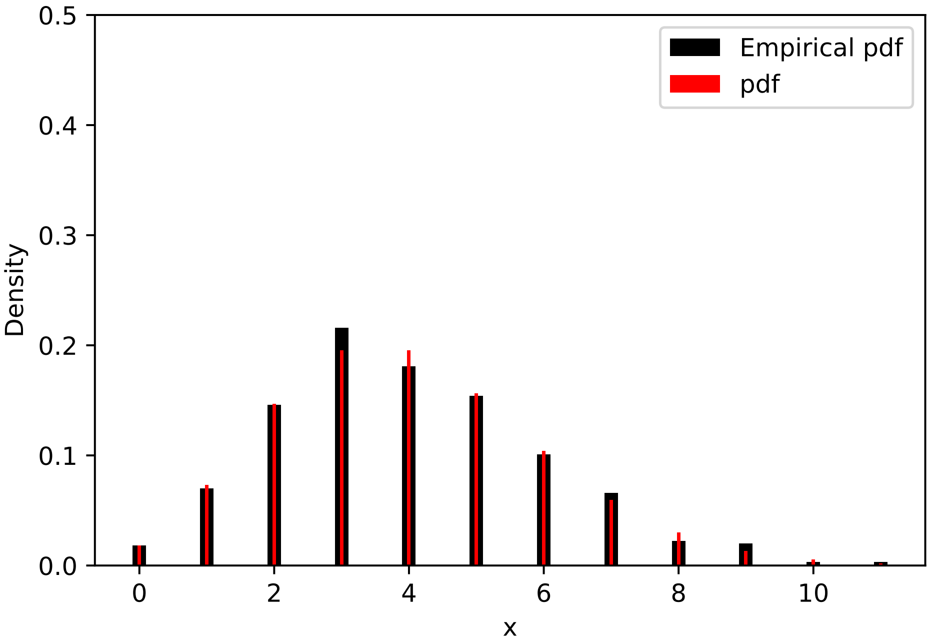
which was found with the following code

```
# Probability of no goals in 10 minutes

# The Poisson pdf (using poisson.pmf() function)
stats.poisson.pmf(0, 3.4/9)

np.float64(0.6853827910309876)
```

|||| Example 2.31 Poisson distributed random variable

Simulate a Poisson distributed random variable to see the Poisson distribution

```
# Simulate a Poisson random variable

# The mean rate of events per interval
lamb = 4
# Number of realizations
n = 1000
# Simulate
x = stats.poisson.rvs(lamb, size=n)
# Plot the empirical pdf
values, counts = np.unique(x, return_counts=True)
plt.bar(values, counts/n, color='black',  width=0.2, label='Empirical
pdf')
# Add the pdf to the plot
plt.bar(values, stats.poisson.pmf(values, lamb), color='red',
width=0.05, label='pdf')
plt.show()
```



2.4 Continuous random variables

If an outcome of an experiment takes a continuous value, for example: a distance, a temperature, a weight, etc., then it is represented by a continuous random variable.

|||| Definition 2.32 Density and probabilities

The *pdf* of a continuous random variable X is a non-negative function for all possible outcomes

$$f(x) \geq 0 \text{ for all } x, \tag{2-36}$$

and has an area below the function of one

$$\int_{-\infty}^{\infty} f(x)dx = 1. \tag{2-37}$$

It defines the probability of observing an outcome in the range from a to b by

$$P(a < X \leq b) = \int_{a}^{b} f(x)dx. \tag{2-38}$$

For the discrete case the probability of observing an outcome x is equal to the *pdf* of x, but this is not the case for a continuous random variable, where

$$P(X = x) = P(x < X \leq x) = \int_{x}^{x} f(u)du = 0, \tag{2-39}$$

i.e. the probability for a continuous random variable to be realized at a single number $P(X = x)$ is zero.

The plot in Figure 2.1 shows how the area below the *pdf* represents the probability of observing an outcome in a range. Note that the normal distribution is used here for the examples, it is introduced in Section 2.5.2.

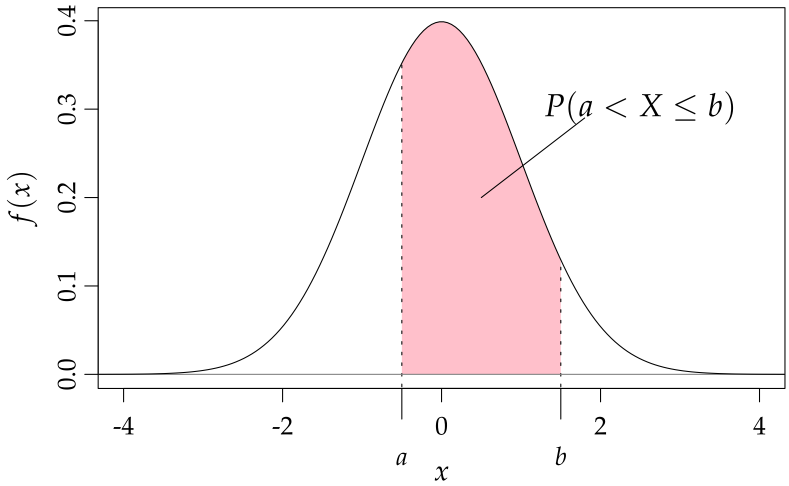


Figure 2.1: The probability of observing the outcome of X in the range between a and b is the area below the *pdf* spanning the range, as illustrated with the coloured area.

|||| Definition 2.33 Distribution

The *cdf* of a continuous variable is defined by

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du, \tag{2-40}$$

and has the properties (in both the discrete and continuous case): the *cdf* is non-decreasing and

$$\lim_{x\to-\infty} F(x) = 0 \text{ and } \lim_{x\to\infty} F(x) = 1. \tag{2-41}$$

The relation between the *cdf* and the *pdf* is

$$P(a < X \leq b) = F(b) - F(a) = \int_{a}^{b} f(x)dx, \tag{2-42}$$

as illustrated in Figures 2.1 and 2.2.

Also as the *cdf* is defined as the integral of the *pdf*, the *pdf* becomes the derivative of the *cdf*

$$f(x) = \frac{d}{dx}F(x) \tag{2-43}$$

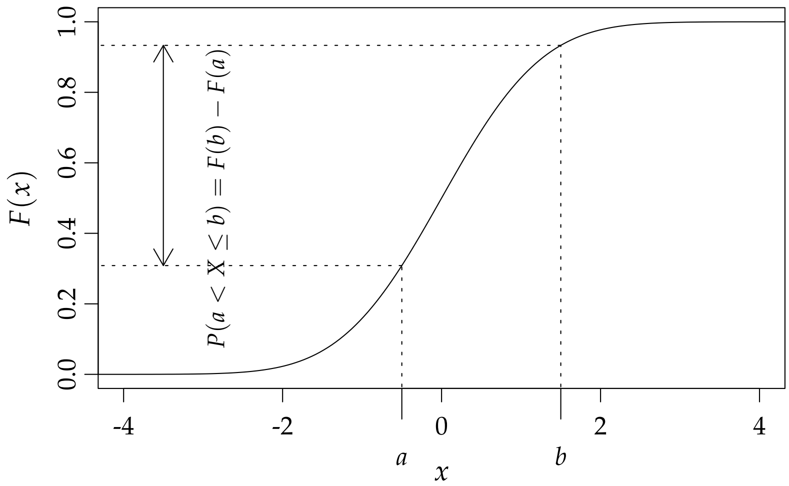


Figure 2.2: The probability of observing the outcome of X in the range between a and b is the distance between $F(a)$ and $F(b)$.

2.4.1 Mean and Variance

Definition 2.34 Mean and variance

For a continuous random variable the mean or expected value is

$$\mu = \mathrm{E}(X) = \int_{-\infty}^{\infty} x f(x) dx, \tag{2-44}$$

hence similar as for the discrete case the outcome is weighted with the *pdf*. The variance is

$$\sigma^2 = \mathrm{E}[(X-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx, \tag{2-45}$$

The differences between the discrete and the continuous case can be summed up in two points:

- In the continuous case integrals are used, in the discrete case sums are used.
- In the continuous case the probability of observing a single value is always zero. In the discrete case it can be positive or zero.

2.5 Continuous distributions

2.5.1 Uniform distribution

A random variable following the *uniform distribution* has equal density at any value within a defined range.

|||| Definition 2.35 Uniform distribution

Let X be a uniform distributed random variable

$$X \sim U(\alpha, \beta), \tag{2-46}$$

where α and β defines the range of possible outcomes. It has the *pdf*

$$f(x) = \begin{cases} \frac{1}{\beta-\alpha} & \text{for } x \in [\alpha, \beta] \\ 0 & \text{otherwise} \end{cases}. \tag{2-47}$$

The uniform *cdf* is

$$F(x) = \begin{cases} 0 & \text{for } x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \text{for } x \in [\alpha, \beta) \\ 1 & \text{for } x \geq \beta \end{cases}. \tag{2-48}$$

In Figure 2.3 the uniform *pdf* and *cdf* are plotted.

|||| Theorem 2.36 Mean and variance of the uniform distribution

The mean of a uniform distributed random variable X is

$$\mu = \frac{1}{2}(\alpha + \beta), \tag{2-49}$$

and the variance is

$$\sigma^2 = \frac{1}{12}(\beta - \alpha)^2. \tag{2-50}$$

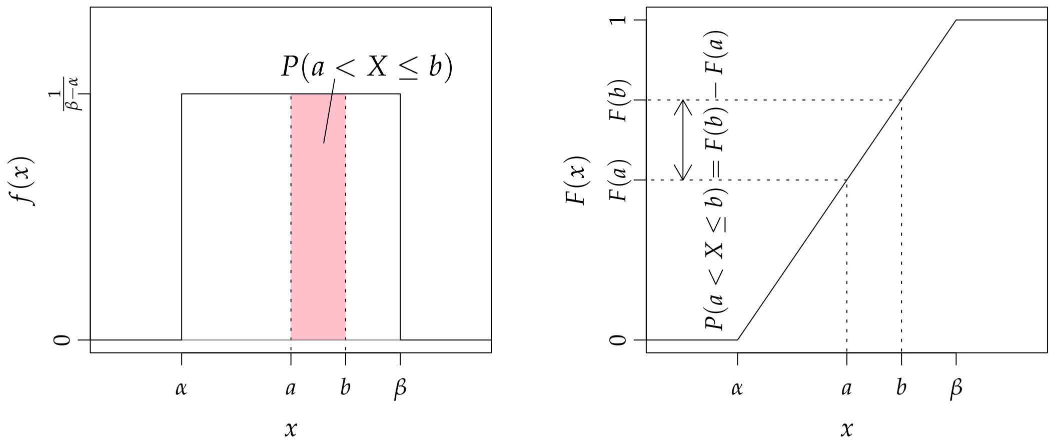


Figure 2.3: The uniform distribution *pdf* and cdf.

2.5.2 Normal distribution

The most famous continuous distribution is the normal distribution for many reasons. Often it is also called the Gaussian distribution. The normal distribution appears naturally for many phenomena and is therefore used in extremely many applications, which will be apparent in later chapters of the book.

|||| Definition 2.37 Normal distribution

Let X be a normal distributed random variable

$$X \sim N(\mu, \sigma^2), \tag{2-51}$$

where μ is the mean and σ^2 is the variance (remember that the standard deviation is σ). Note that the two parameters are actually the mean and variance of X.
It follows the normal *pdf*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2-52}$$

and the normal *cdf*

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(u-\mu)^2}{2\sigma^2}} du. \tag{2-53}$$

Theorem 2.38 Mean and variance

The mean of a Normal distributed random variable is

$$\mu, \tag{2-54}$$

and the variance is

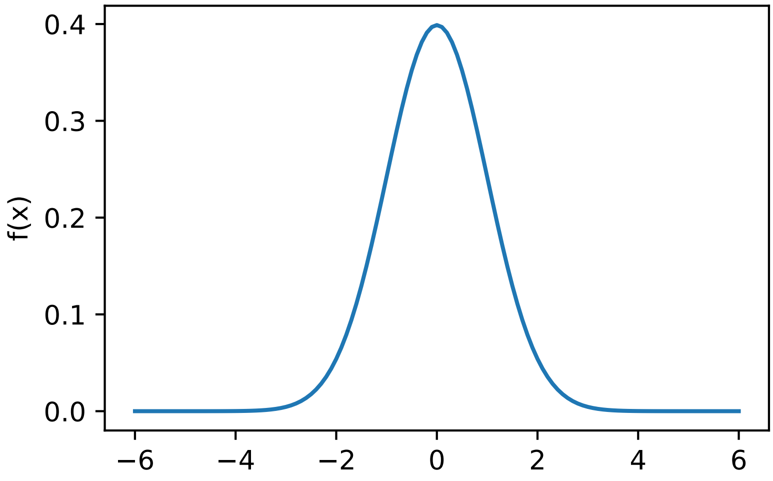
$$\sigma^2. \tag{2-55}$$

Hence simply the two parameters defining the distribution.

Example 2.39 The normal pdf

Example: Let us play with the normal *pdf*

```
# Play with the normal distribution

# The mean and standard deviation
muX = 0
sigmaX = 1
# A sequence of x values
xSeq = np.arange(-6, 6.1, 0.1)
##
pdfX = 1/(sigmaX*np.sqrt(2*np.pi)) * np.exp(-(xSeq-muX)**2/(2*sigmaX**2))
# Plot the pdf
plt.plot(xSeq, pdfX)
plt.ylabel('f(x)')
```



Try with different values of the mean and standard deviation. Describe how this change the position and spread of the *pdf*?

Theorem 2.40 Linear combinations of normal random variables

Let $X_1, \ldots, X_n$ be independent normal random variables, then any linear combination of $X_1, \ldots, X_n$ will follow a normal distribution, with mean and variance given in Theorem 2.56.

Use the mean and variance identities introduced in Section 2.7 to find the mean and variance of the linear combination as exemplified here:

Example 2.41

Consider two normal distributed random variables

$$X_1 \sim N(\mu_{X_1}, \sigma^2_{X_1}) \quad \text{and} \quad X_2 \sim N(\mu_{X_2}, \sigma^2_{X_2}). \tag{2-56}$$

The difference

$$Y = X_1 - X_2, \tag{2-57}$$

is normal distributed

$$Y \sim N(\mu_Y, \sigma^2_Y), \tag{2-58}$$

where the mean is

$$\mu_Y = \mu_{X_1} - \mu_{X_2}, \tag{2-59}$$

and

$$\sigma^2_Y = \sigma^2_{X_1} + \sigma^2_{X_2}, \tag{2-60}$$

where the mean and variance identities introduced in Section 2.7 have been used.

Standard normal distribution

Definition 2.42 Standard normal distribution

The standard normal distribution is the normal distribution with zero mean and unit variance

$$Z \sim N(0,1), \tag{2-61}$$

where Z is the standardized normal random variable.

Historically before the widespread use of computers the standardized random variables were used a lot, since it was not possible to easily evaluate the *pdf* and *cdf*, instead they were looked up in tables for the standardized distributions. This was smart since transformation into standardized distributions requires only a few simple operations.

Theorem 2.43 Transformation to the standardized normal random variable

A normal distributed random variable X can be transformed into a standardized normal random variable by

$$Z = \frac{X-\mu}{\sigma}. \tag{2-62}$$

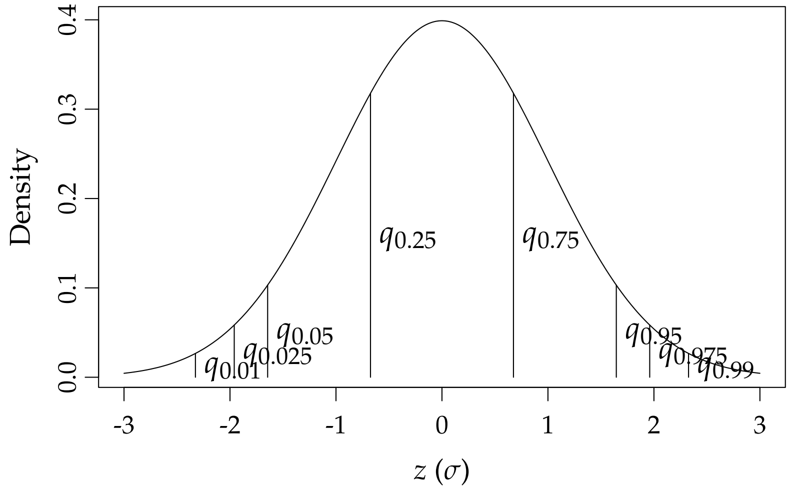
Example 2.44 Quantiles in the standard normal distribution

The most used quantiles (or percentiles) in the standard normal distribution are

Percentile	1%	2.5%	5%	25%	75%	95%	97.5%	99%
Quantile	0.01	0.025	0.05	0.25	0.75	0.95	0.975	0.99
Value	-2.33	-1.96	-1.64	-0.67	0.67	1.64	1.96	2.33

Note that the values can be considered as standard deviations (i.e. for Z the standardized normal then $\sigma_Z = 1$), which holds for any normal distribution.

The most used quantiles are marked on the plot



Note that the units on the x-axis is in standard deviations.

Normal *pdf* details

In order to get insight into how the normal distribution is formed consider the following steps. In Figure 2.4 the result of each step is plotted:

1. Take the distance to the mean: $x - \mu$
2. Square the distance: $(x - \mu)^2$
3. Make it negative and scale it: $\frac{-(x-\mu)^2}{(2\sigma^2)}$
4. Take the exponential: $e^{\frac{-(x-\mu)^2}{(2\sigma^2)}}$
5. Finally, scale it to have an area of one: $\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-\mu)^2}{(2\sigma^2)}}$

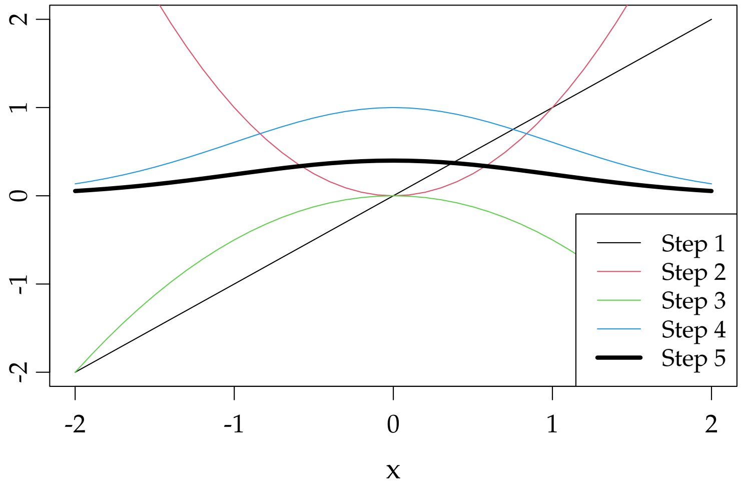


Figure 2.4: The steps involved in calculating the normal distribution *pdf*.

|||| Example 2.45 Python functions for the normal distribution

In Python functions to generate values from many distributions are implemented. For the normal distribution the following functions are available:

```
# Do it for a sequence of x values
xSeq = np.arange(-3, 4)
# The pdf
stats.norm.pdf(xSeq, 0, 1)

array([0.004, 0.054, 0.242, 0.399, 0.242, 0.054, 0.004])


# The cdf
stats.norm.cdf(xSeq, 0, 1)

array([0.001, 0.023, 0.159, 0.500, 0.841, 0.977, 0.999])


# The quantiles
stats.norm.ppf([0.01,0.025,0.05,0.5,0.95,0.975,0.99], 0, 1)

array([-2.326, -1.960, -1.645,  0.000,  1.645,  1.960,  2.326])


# Generate random normal distributed realizations
stats.norm.rvs(0, 1, size=10)

array([-1.043,  0.050, -0.592, -0.840,  0.460,  0.150,  0.021, -1.221,
       -0.638, -1.024])


# Calculate the probability that the outcome of X is between a and b
a = 0.2
b = 0.8
stats.norm.cdf(b, 0, 1) - stats.norm.cdf(a, 0, 1)

np.float64(0.20888489197750038)


# See more details in online documentation for scipy.stats.norm
```

Use the functions to make a plot of the normal *pdf* with marks of the 2.5%, 5%, 95%, 97.5% quantiles.

Make a plot of the normal *pdf* and a histogram (empirical *pdf*) of 100 simulated realizations.

2.5.3 Log-Normal distribution

If a random variable is log-normal distributed then its logarithm is normally distributed.

|||| Definition 2.46 Log-Normal distribution

A log-normal distributed random variable

$$X \sim LN(\alpha, \beta^2), \tag{2-63}$$

where α is the mean and β^2 is the variance of the normal distribution obtained when taking the natural logarithm to X.
The log-normal *pdf* is

$$f(x) = \frac{1}{x\sqrt{2\pi}\beta} e^{-\frac{(\ln x - \alpha)^2}{2\beta^2}}. \tag{2-64}$$

|||| Theorem 2.47 Mean and variance of log-normal distribution

Mean of the log-normal distribution

$$\mu = e^{\alpha+\beta^2/2}, \tag{2-65}$$

and variance

$$\sigma^2 = e^{2\alpha+\beta^2}(e^{\beta^2} - 1). \tag{2-66}$$

The log-normal distribution occurs in many fields, in particular: biology, finance and many technical applications.

2.5.4 Exponential distribution

The usual application of the exponential distribution is for describing the length (usually time) between events which, when counted, follows a Poisson distribution, see Section 2.3.3. Hence the length between events which occur continuously and independently at a constant average rate.

|||| Definition 2.48 Exponential distribution

Let X be an exponential distributed random variable

$$X \sim Exp(\lambda), \tag{2-67}$$

where λ is the average rate of events.

It follows the exponential *pdf*

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}. \tag{2-68}$$

|||| Theorem 2.49 Mean and variance of exponential distribution

Mean of an exponential distribution is

$$\mu = \frac{1}{\lambda}, \tag{2-69}$$

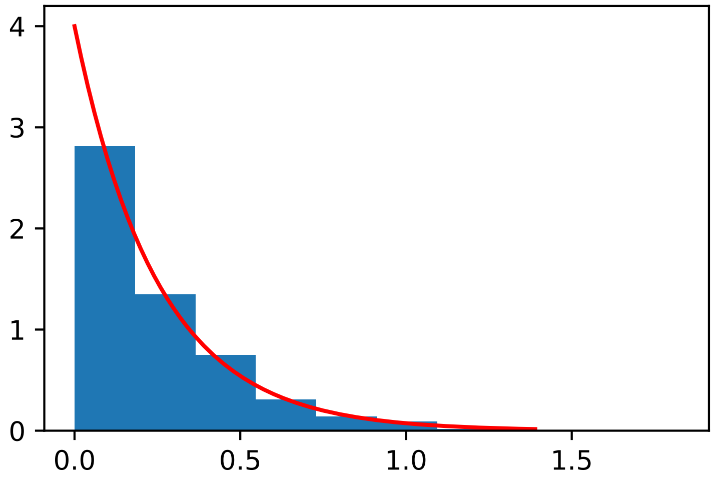
and the variance is

$$\sigma^2 = \frac{1}{\lambda^2}. \tag{2-70}$$

Example 2.50 Exponential distributed time intervals

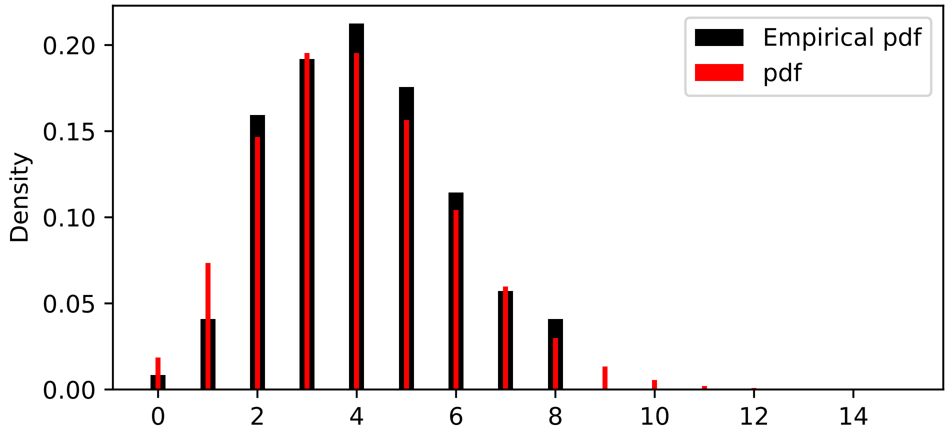
Simulate a so-called Poisson process, which has exponential distributed time interval between events

```
# Simulate exponential waiting times

# The rate parameter: events per time
lamb = 4
# Number of realizations
n = 1000
# Simulate
wait_times = stats.expon.rvs(loc=0, scale=1/lamb, size=n)
# The empirical pdf
plt.hist(wait_times, density=True)
# Add the pdf to the plot
x = np.arange(0,1.4,0.01)
plt.plot(x, stats.expon.pdf(x, loc=0, scale=1/lamb), color='red')
plt.show()
```



Furthermore check that by counting the events in fixed length intervals that they follow a Poisson distribution.

```
# Check the relation to the Poisson distribution
# by counting the events in each interval

# Sum up to get the running time
running_times = np.cumsum(wait_times)
# Use the hist function to count in intervals between the breaks,
# here 0,1,2,...
counts, bin_edges = np.histogram(running_times,
bins=np.arange(np.ceil(running_times.max())))
plt.bar(np.arange(len(np.bincount(counts))),
np.bincount(counts)/len(counts), color='black',  width=0.3,
label='Empirical pdf')

# Add the Poisson pdf to the plot
poisson_pmf = stats.poisson.pmf(np.arange(0, 16), lamb)
plt.bar(np.arange(0, 16), poisson_pmf, color='red', width=0.1,
label='pdf')
```



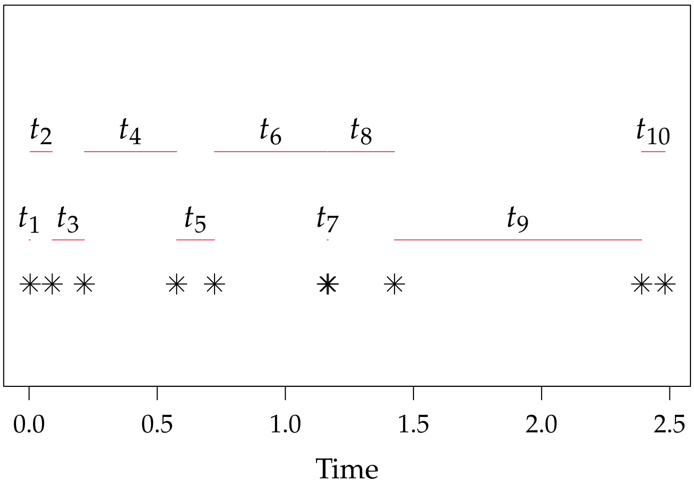


Figure 2.5: Exponential distributed time intervals between events forms a so-called Poisson process.

2.6 Simulation of random variables

The basic concept of simulation was introduced in Section 2.2.1 and we have already applied the in-built functions in Python for generating random numbers from any implemented distribution, see how in Section 2.3.1. In this section it is explained how realizations of a random variable can be generated from any probability distribution – it is the same technique for both discrete and continuous distributions.

Basically, a computer obviously cannot create a result/number, which is random. A computer can give an output as a function of an input. (Pseudo) random numbers from a computer are generated from a specially designed algorithm - called a random number generator, which once started can make the number x_{i+1} from the number x_i. The algorithm is designed in such a way that when looking at a sequence of these values, in practice one cannot tell the difference between them and a sequence of real random numbers. The algorithm needs a start input, called the "seed", as explained above Remark 2.12. Usually, you can manage just fine without having to worry about the seed issue since the program itself finds out how to handle it appropriately. Only if you want to be able to recreate exactly the same results you need to set seed value.

Actually, a basic random number generator typically generates (pseudo) random numbers between 0 and 1 in the sense that numbers in practice follow the uniform distribution on the interval 0 to 1, see Section 2.35. Actually, there is a simple way how to come from the uniform distribution to any kind of distribution:

Theorem 2.51

If $U \sim \text{Uniform}(0,1)$ and F is a distribution function for any probability distribution, then $F^{-1}(U)$ follow the distribution given by F

Recall, that the distribution function F in Python is given by the `'.cdf()'` versions of the distributions, while F^{-1} is given by the `'.ppf()'` versions.

Example 2.52 Random numbers in Python

We can generate 100 normally distributed $N(2, 3^2)$ numbers similarly the following two ways:

```
# Generate 100 normal distributed values
random_numbers = stats.norm.rvs(loc=2, scale=3, size=100)
# Similarly, generate 100 uniform distributed values from 0 to 1 and
# # put them through the inverse normal cdf
uniform_random_numbers = stats.uniform.rvs(loc=0, scale=1, size=100)
stats.norm.ppf(uniform_random_numbers, loc=2, scale=3)
```

Example 2.53 Simulating the exponential distribution

Consider the exponential distribution with $\lambda = 1/\beta = 1/2$, that is, with density function

$$f(x) = \lambda e^{-\lambda x},$$

for $x > 0$ and 0 otherwise. The distribution function is

$$F(x) = \int_0^x f(t)dt = 1 - e^{-0.5x}.$$

The inverse of this distribution function can be found by solving

$$u = 1 - e^{-0.5x} \Leftrightarrow x = -2\log(1-u).$$

So if random numbers $U \sim \text{Uniform}(0,1)$ then $-2\log(1-U)$ follows the exponential distribution with $\lambda = 1/2$ (and $\beta = 2$). We confirm this in the code given below:

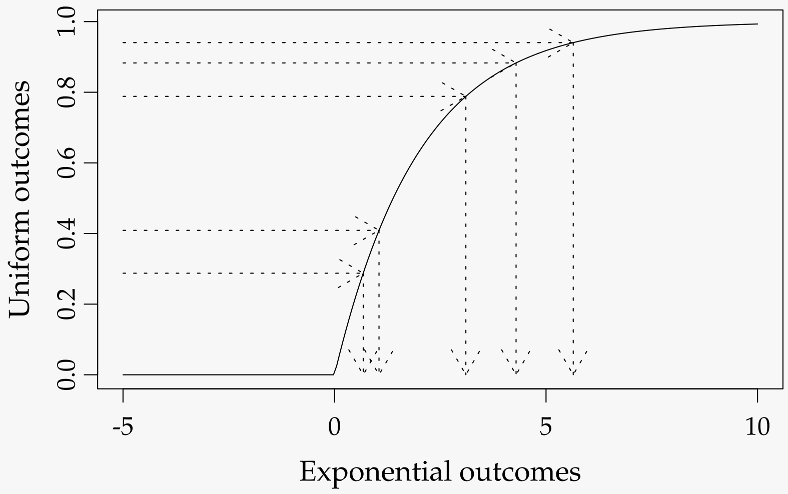
```
# Three equivalent ways of simulating the exponential distribution
# with lambda=1/2
re1 = -2*np.log(1-stats.uniform.rvs(loc=0, scale=1, size=10000))
re2 = stats.expon.ppf(stats.uniform.rvs(loc=0, scale=1, size=10000),
loc=0, scale=2)
re3 = stats.expon.rvs(loc=0, scale=2, size=10000)

# Check the means and variances of each
print(re1.mean(), re2.mean(), re2.mean())
```

```
2.0039553521283056 1.9948804494574823 1.9948804494574823
```

```
print(re1.var(), re2.var(), re2.var())
```

```
3.89520722053301 3.7967058951210935 3.7967058951210935
```

This can be illustrated by plotting the distribution function (cdf) for the exponential distribution with $\lambda = 1/2$ and 5 random outcomes



But since Python has already done all this for us, we do not really need this as long as we only use distributions that have already been implemented in Python.

2.7 Identities for the mean and variance

Rules for calculation of the mean and variance of linear combinations of independent random variables are introduced here. They are valid for both the discrete and continuous case.

Theorem 2.54 Mean and variance of linear functions

Let $Y = aX + b$ then

$$\mathrm{E}(Y) = \mathrm{E}(aX + b) = a\,\mathrm{E}(X) + b, \tag{2-71}$$

and

$$\mathrm{V}(Y) = \mathrm{V}(aX + b) = a^2\,\mathrm{V}(X). \tag{2-72}$$

Random variables are often scaled (i.e. aX) for example when shifting units:

Example 2.55

The mean of a bike shops sale is 100 bikes per month and varies with a standard deviation of 15. They earn 200 Euros per bike. What is the mean and standard deviation of their earnings per month?

Let X be the number of bikes sold per month. On average they sell $\mu_X = 100$ bikes per month and it varies with a variance of $\sigma_X^2 = 225$. The shops monthly earnings

$$Y = 200X,$$

has then a mean and standard deviation of

$$\begin{aligned}\mu_Y &= \mathrm{E}(Y) = \mathrm{E}(200X) = 200\,\mathrm{E}(X) = 200 \cdot 100 = 20000 \text{ Euro/month},\\ \sigma_Y &= \sqrt{\mathrm{V}(Y)} = \sqrt{\mathrm{V}(200X)} = \sqrt{200^2\,\mathrm{V}(X)} = \sqrt{40000 \cdot 225} = 3000 \text{ Euro/month}.\end{aligned}$$

Theorem 2.56 Mean and variance of linear combinations

The mean of a linear combination of independent random variables is

$$\mathrm{E}(a_1 X_1 + a_2 X_2 + \cdots + a_n X_n) = a_1\,\mathrm{E}(X_1) + a_2\,\mathrm{E}(X_2) + \cdots + a_n\,\mathrm{E}(X_n), \tag{2-73}$$

and the variance

$$\mathrm{V}(a_1 X_1 + a_2 X_2 + \cdots + a_n X_n) = a_1^2\,\mathrm{V}(X_1) + a_2^2\,\mathrm{V}(X_2) + \cdots + a_n^2\,\mathrm{V}(X_n). \tag{2-74}$$

Example 2.57

Lets take a dice example to emphasize an important point. Let X_i represent the outcome of a roll with a dice with mean μ_X and standard deviation σ_X.

Now, consider a scaling of a single roll with a dice, say five times

$$Y^{\text{scale}} = 5X_1,$$

then the mean will scale linearly

$$\mathrm{E}(Y^{\text{scale}}) = \mathrm{E}(5X_1) = 5\,\mathrm{E}(X_1) = 5\,\mu_X,$$

and the standard deviation also scales linearly

$$\sigma^2_{Y^{\text{scale}}} = \mathrm{V}(5X_1) = 5^2\,\mathrm{V}(X_1) = 5^2\,\sigma_X^2 \;\Leftrightarrow\; \sigma_{Y^{\text{scale}}} = 5\,\sigma_X.$$

Whereas *for a sum* of five rolls

$$Y^{\text{sum}} = X_1 + X_2 + X_3 + X_4 + X_5,$$

the mean will similarly scale linearly

$$\begin{aligned}
\mathrm{E}(Y^{\text{sum}}) &= \mathrm{E}(X_1 + X_2 + X_3 + X_4 + X_5)\\
&= \mathrm{E}(X_1) + \mathrm{E}(X_2) + \mathrm{E}(X_3) + \mathrm{E}(X_4) + \mathrm{E}(X_5)\\
&= 5\,\mu_X,
\end{aligned}$$

however the standard deviation will increase only with the square root

$$\begin{aligned}
\sigma^2_{Y^{\text{sum}}} &= \mathrm{V}(X_1 + X_2 + X_3 + X_4 + X_5)\\
&= \mathrm{V}(X_1) + \mathrm{V}(X_2) + \mathrm{V}(X_3) + \mathrm{V}(X_4) + \mathrm{V}(X_5)\\
&= 5\,\sigma_X^2 \Leftrightarrow\\
\sigma_{Y^{\text{sum}}} &= \sqrt{5}\,\sigma_X.
\end{aligned}$$

> This is simply because when applying the sum to many random outcomes, then the high and low outcomes will even out each other, such that the variance will be smaller for a sum than for a scaling.

2.8 Covariance and correlation

In this chapter we have discussed mean and variance (or standard deviation), and the relation to the sample mean and sample variance, see Section 2.2.2. In Chapter 1 Section 1.4.3 we discussed the sample covariance and sample correlation, these two measures also have theoretical justification, namely covariance and correlation, which we will discuss in this section. We start by the definition of covariance.

|||| Definition 2.58 Covariance

Let X and Y be two random variables, then the covariance between X and Y, is

$$\mathrm{Cov}(X, Y) = \mathrm{E}[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])] \,. \tag{2-75}$$

|||| Remark 2.59

It follows immediately from the definition that $\mathrm{Cov}(X, X) = \mathrm{V}(X)$ and $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$.

An important concept in statistics is independence (see Section 2.9 for a formal definition). We often assume that realizations (random variables) are independent. If two random variables are independent then their covariance will be zero, the reverse is however not necessarily true (see also the discussion on sample correlation in Section 1.4.3).

The following calculation rule apply to covariance between two random variables X and Y:

|||| Theorem 2.60 Covariance between linear combinations

Let X and Y be two random variables, then

$$\mathrm{Cov}(a_0 + a_1X + a_2Y, b_0 + b_1X + b_2Y) = a_1b_1\,\mathrm{V}(X) + a_2b_2\,\mathrm{V}(Y) + (a_1b_2 + a_2b_1)\,\mathrm{Cov}(X, Y). \tag{2-76}$$

|||| Proof

Let $Z_1 = a_0 + a_1 X + a_2 Y$ and $Z_2 = b_0 + b_1 X + b_2 Y$ then

$$\begin{aligned}
\operatorname{Cov}(Z_1, Z_2) &= \mathrm{E}[(a_1(X - \mathrm{E}[X]) + a_2(Y - \mathrm{E}[Y]))(b_1(X - \mathrm{E}[X]) + b_2(Y - \mathrm{E}[Y]))] \\
&= \mathrm{E}[a_1(X - \mathrm{E}[X])b_1(X - \mathrm{E}[X])] + \mathrm{E}[a_1(X - \mathrm{E}[X])b_2(Y - \mathrm{E}[Y])] + \\
&\quad \mathrm{E}[a_2(Y - \mathrm{E}[Y])b_1(X - \mathrm{E}[X])] + \mathrm{E}[a_2(Y - \mathrm{E}[Y])b_2(Y - \mathrm{E}[Y])] \\
&= a_1 b_1 \mathrm{V}(X) + a_2 b_2 \mathrm{V}(Y) + (a_1 b_2 + a_2 b_2) \operatorname{Cov}(X, Y). \qquad (2\text{-}77)
\end{aligned}$$

■

|||| Example 2.61

Let $X \sim N(3, 2^2)$ and $Y \sim N(2, 1)$ and the covariance between X and Y given by $\operatorname{Cov}(X, Y) = 1$. What is the variance of the random variable $Z = 2X - Y$?

$$\begin{aligned}
\mathrm{V}(Z) = \operatorname{Cov}[2X - Y, 2X - Y] &= 2^2 \mathrm{V}(X) + \mathrm{V}(Y) - 4 \operatorname{Cov}(X, Y) \\
&= 2^2 2^2 + 1 - 4 = 13.
\end{aligned}$$

We have already seen in Section 1.4.3 that the sample correlation measures the observed degree of linear dependence between two random variables – calculated from samples observed on the same observational unit e.g. height and weight of people. The theoretical counterpart is the correlation between two random variables – the true linear dependence between the two variables:

|||| Definition 2.62 Correlation

Let X and Y be two random variables with $\mathrm{V}(X) = \sigma_x^2$, $\mathrm{V}(Y) = \sigma_y^2$, and $\operatorname{Cov}(X, Y) = \sigma_{xy}$, then the correlation between X and Y is

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}. \qquad (2\text{-}78)$$

|||| Remark 2.63

The correlation is a number between -1 and 1.

|||| Example 2.64

Let $X \sim N(1, 2^2)$ and $\epsilon \sim N(0, 0.5^2)$ be independent random variables, find the correlation between X and $Z = X + \epsilon$.

The variance of Z is

$$\mathrm{V}(Z) = \mathrm{V}(X + \epsilon) = \mathrm{V}(X) + \mathrm{V}(\epsilon) = 4 + 0.25 = 4.25.$$

The covariance between X and Z is

$$\mathrm{Cov}(X, Z) = \mathrm{Cov}(X, X + \epsilon) = \mathrm{V}(X) = 4,$$

and hence

$$\rho_{xz} = \frac{4}{\sqrt{4.25 \cdot 4}} = 0.97.$$

2.9 Independence of random variables

In statistics the concept of independence is very important, and in order to give a formal definition of independence we will need the definition of two-dimensional random variables. The probability density function of a two-dimensional discrete random variable, called the joint probability density function, is,

|||| Definition 2.65 Joint *pdf* of two-dimensional discrete random variables

The *pdf* of a two-dimensional discrete random variable $[X, Y]$ is

$$f(x,y) = P(X = x, Y = y), \tag{2-79}$$

with the properties

$$f(x,y) \geq 0 \text{ for all } (x,y), \tag{2-80}$$

$$\sum_{\text{all } x} \sum_{\text{all } y} f(x,y) = 1. \tag{2-81}$$

|||| Remark 2.66

$P(X = x, Y = y)$ should be read: the probability of $X = x$ *and* $Y = y$.

|||| Example 2.67

Imagine two throws with an fair coin: the possible outcome of each throw is either head or tail, which will be given the values 0 and 1 respectively. The complete set of outcomes is (0,0), (0,1), (1,0), and (1,1) each with probability 1/4. And hence the *pdf* is

$$f(x,y) = \frac{1}{4}; \quad x = \{0,1\}, y = \{0,1\},$$

further we see that

$$\begin{aligned}\sum_{x=0}^{1}\sum_{y=0}^{1} f(x,y) &= \sum_{x=0}^{1}(f(x,0)+f(x,1)) = f(0,0)+f(0,1)+f(1,0)+f(1,1)\\ &= 1.\end{aligned}$$

The formal definition of independence for a two dimensional discrete random variable is:

Definition 2.68 Independence of discrete random variables

Two discrete random variables X and Y are said to be independent if and only if

$$P(X = x, Y = y) \;=\; P(X = x)P(Y = y). \tag{2-82}$$

Example 2.69

Example 2.67 is an example of two independent random variables, to see this write the probabilities

$$P(X = 0) = \sum_{y=0}^{1} f(0, y) = \frac{1}{2},$$
$$P(X = 1) = \sum_{y=0}^{1} f(1, y) = \frac{1}{2}.$$

similarly $P(Y = 0) = \frac{1}{2}$ and $P(Y = 1) = \frac{1}{2}$, now we see that $P(X = x)P(Y = y) = \frac{1}{4}$ for all possible x and y, and hence

$$P(X = x)P(Y = y) \;=\; P(X = x, Y = y) \;=\; \frac{1}{4}.$$

Example 2.70

Now imagine that for the second throw we don't see the outcome of Y, but only observe the sum of X and Y, denote it by

$$Z = X + Y.$$

Lets find out if X and Z are independent. In this case the for all outcomes $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 2)$ the joint *pdf* is

$$P(X = 0, Z = 0) \;=\; P(X = 0, Z = 1) \;=\; P(X = 1, Z = 1) \;=\; P(X = 1, Z = 2) \;=\; \frac{1}{4}.$$

The *pdf* for each variable is: for X

$$P(X = 0) \;=\; P(X = 1) \;=\; \frac{1}{2},$$

and for Z

$$P(Z=0) \;=\; P(Z=2) \;=\; \frac{1}{4} \text{ and } P(Z=1) \;=\; \frac{1}{2},$$

thus for example for the particular outcome $(0,0)$

$$P(X=0)P(Z=0) \;=\; \frac{1}{2}\cdot\frac{1}{4} \;=\; \frac{1}{8} \;\neq\; \frac{1}{4} \;=\; P(X=0,Z=0),$$

the *pdf*s are not equal and hence we see that X and Z are not independent.

|||| Remark 2.71

In the example above it is quite clear that X and Z cannot be independent. In real applications we do not know exactly how the outcomes are realized and therefore we will need to assume independence (or test it).

To be able to define independence of continuous random variables, we will need the *pdf* of a two-dimensional random variable:

|||| Definition 2.72 *Pdf* of two dimensional continous random variables

The *pdf* of a two-dimensional continous random variable $[X,Y]$ is a function $f(x,y)$ from $\mathbb{R}^2$ into $\mathbb{R}_+$ with the properties

$$f(x,y) \geq 0 \text{ for all } (x,y), \tag{2-83}$$

$$\int\int f(x,y)dxdy = 1. \tag{2-84}$$

Just as for one-dimensional random variables the probability interpretation is in form of integrals

$$P\big((X,Y)\in A\big) = \int_A f(x,y)dxdy, \tag{2-85}$$

where A is an area.

|||| Example 2.73 Bivariate normal distribution

The most important two-dimensional distribution is the bivariate normal distribution

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\ &= \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}^2}} e^{-\frac{\sigma_{22}(x_1-\mu_1)^2 + \sigma_{11}(x_2-\mu_2)^2 - 2\sigma_{12}(x_1-\mu_1)(x_2-\mu_2)}{2(\sigma_{11}\sigma_{22}-\sigma_{12}^2)}}, \end{aligned}$$

where $\boldsymbol{x} = (x_1, x_2)$, and $\boldsymbol{\mu} = [\mathrm{E}(X_1), \mathrm{E}(X_2)]$, and Σ is the so-called variance-covariance matrix with elements $(\Sigma)_{ij} = \sigma_{ij} = \mathrm{Cov}(X_i, X_j)$, note that $\sigma_{12} = \sigma_{21}$, $|\cdot|$ is the determinant, and Σ^{-1} is the inverse of Σ.

|||| Definition 2.74 Independence of continous random variables

Two continous random variables X and Y are said to be independent if

$$f(x, y) = f(x)f(y). \tag{2-86}$$

We list here some properties of independent random variables.

|||| Theorem 2.75 Properties of independent random variables

If X and Y are independent then

$$\mathrm{E}(XY) = \mathrm{E}(X)\,\mathrm{E}(Y), \tag{2-87}$$

and

$$\mathrm{Cov}(X, Y) = 0. \tag{2-88}$$

Let $X_1, \ldots, X_n$ be independent and identically distributed random variables then

$$\mathrm{Cov}(\overline{X}, X_i - \overline{X}) = 0. \tag{2-89}$$

|||| Proof

$$\begin{aligned} \mathrm{E}(XY) &= \int\int xyf(x,y)dxdy = \int\int xyf(x)f(y)dxdy \\ &= \int xf(x)dx \int yf(y)dy = \mathrm{E}(X)\,\mathrm{E}(Y) \end{aligned} \tag{2-90}$$

$$\begin{aligned} \mathrm{Cov}(X,Y) &= \mathrm{E}[(X-\mathrm{E}(X))(Y-\mathrm{E}(Y))] \\ &= \mathrm{E}[XY] - \mathrm{E}[\mathrm{E}(X)Y] - \mathrm{E}[X\,\mathrm{E}(Y)] + \mathrm{E}(X)\,\mathrm{E}(Y) \\ &= 0. \end{aligned} \tag{2-91}$$

$$\begin{aligned} \mathrm{Cov}(\overline{X}, X_i - \overline{X}) &= \mathrm{Cov}(\overline{X}, X_i) - \mathrm{Cov}(\overline{X}, \overline{X}) \\ &= \frac{1}{n}\sigma^2 - \frac{1}{n^2}\mathrm{Cov}\left(\sum X_i, \sum X_i\right) \\ &= \frac{1}{n}\sigma^2 - \frac{1}{n^2}n\sigma^2 = 0. \end{aligned} \tag{2-92}$$

■

|||| Remark 2.76

Note that $\mathrm{Cov}(X,Y) = 0$ does not imply that X and Y are independent. However, if X and Y follow a bivariate normal distribution, then if X and Y are uncorrelated then they are also independent.

2.10 Functions of normal random variables

This section will cover some important functions of a normal random variable. In general the question of how an arbitrary function of a random variable is distributed cannot be answered on closed form (i.e. directly and exactly calculated) – for answering such questions we must use simulation as a tool, as covered details in Chapter 4. We have already discussed simulation as a learning tool, which will also be used in this section.

The simplest function we can think of is a *linear combination of normal random variables*, which we from Theorem 2.40 know *will follow a normal distribution*. The mean and variance of this normal distribution can be calculated using the identities given in Theorem 2.56.

Remark 2.77

Note that combining Theorems 2.40 and 2.75, and Remark 2.76 imply that $\overline{X}$ and $X_i - \overline{X}$ are independent.

In addition to the result given above we will cover three additional distributions: χ^2-distribution, t-distribution and the F-distribution, which are all very important for the statistical inference covered in the following chapters.

2.10.1 The χ^2-distribution

The χ^2-distribution (chi-square) is defined by:

Definition 2.78

Let X be χ^2 distributed, then its *pdf* is

$$f(x) = \frac{1}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)} x^{\frac{\nu}{2}-1}e^{-\frac{x}{2}}; \quad x \geq 0, \tag{2-93}$$

where $\Gamma\left(\frac{\nu}{2}\right)$ is the Γ-function and ν is the degrees of freedom.

An alternative definition (here formulated as a theorem) of the χ^2-distribution is:

Theorem 2.79

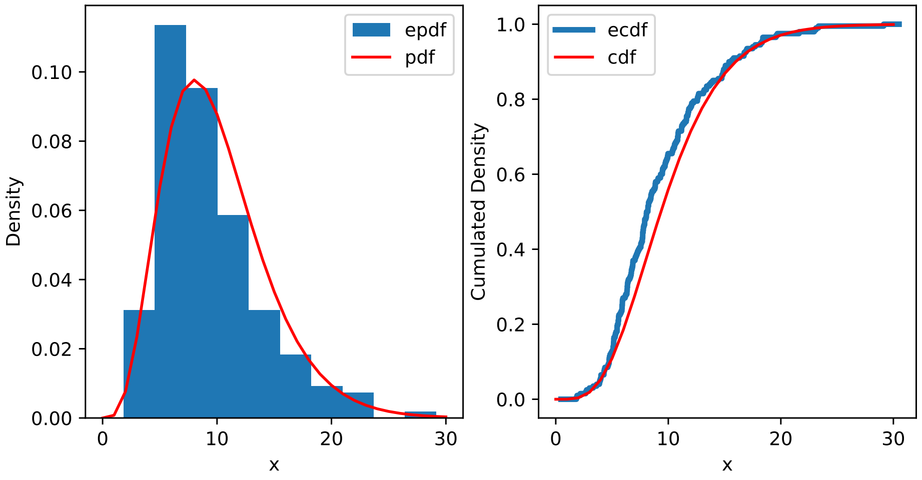
Let $Z_1, \ldots, Z_\nu$ be independent random variables following the standard normal distribution, then

$$\sum_{i=1}^{\nu} Z_i^2 \sim \chi^2(\nu). \tag{2-94}$$

We will omit the proof of the theorem as it requires more probabilty calculus than covered here. Rather a small example that illustrates how the theorem can be checked by simulation:

Example 2.80 simulation of χ^2-distribution

```
# Simulate 10 realizations from a standard normal distributed variable
n = 10
stats.norm.rvs(loc=0, scale=1, size=n)
# Now repeat this 200 times and calculate the sum of squares each time
# Note: the use of the function replicate: it repeats the
#       expression in the 2nd argument k times, see ?replicate
x = [np.sum(stats.norm.rvs(loc=0, scale=1, size=n)**2) for _ in
range(200)]
x = np.array(x)
# Plot the epdf of the sums and compare to the theoretical chisquare
pdf
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.hist(x, density=True, label='epdf')
ax1.plot(range(0,31), stats.chi2.pdf(range(0,31), df=n), color="red",
label='pdf')
# and the ecdf compared to the cdf
stats.ecdf(x).cdf.plot(ax2, label='ecdf')
ax2.plot(range(0,31), stats.chi2.cdf(range(0,31), df=n), color="red",
label='cdf')
```



In the left plot the empirical *pdf* is compared to the theoretical *pdf* and in the right plot the empirical *cdf* is compared to the theoretical *cdf*.

|||| Theorem 2.81

Given a sample of size n from the normal distributed random variables X_i with variance σ^2, then the sample variance S^2 (viewed as random variable) can be transformed into

$$\chi^2 = \frac{(n-1)S^2}{\sigma^2}, \tag{2-95}$$

which follows the χ^2-distribution with degrees of freedom $\nu = n - 1$.

|||| Proof

Start by rewriting the expression

$$\begin{aligned}\frac{(n-1)S^2}{\sigma^2} &= \sum_{i=1}^{n}\left(\frac{X_i - \overline{X}}{\sigma}\right)^2\\ &= \sum_{i=1}^{n}\left(\frac{X_i - \mu + \mu - \overline{X}}{\sigma}\right)^2\\ &= \sum_{i=1}^{n}\left(\frac{X_i - \mu}{\sigma}\right)^2 + \sum_{i=1}^{n}\left(\frac{\overline{X} - \mu}{\sigma}\right)^2 - 2\sum_{i=1}^{n}\frac{(\overline{X}-\mu)(X_i-\mu)}{\sigma^2}\\ &= \sum_{i=1}^{n}\left(\frac{X_i - \mu}{\sigma}\right)^2 + n\left(\frac{\overline{X} - \mu}{\sigma}\right)^2 - 2n\left(\frac{\overline{X} - \mu}{\sigma}\right)^2\\ &= \sum_{i=1}^{n}\left(\frac{X_i - \mu}{\sigma}\right)^2 - \left(\frac{\overline{X} - \mu}{\sigma/\sqrt{n}}\right)^2,\end{aligned} \tag{2-96}$$

we know that $\frac{X_i-\mu}{\sigma} \sim N(0,1)$ and $\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$, and hence the left hand side is a $\chi^2(n)$ distributed random variable minus a $\chi^2(1)$ distributed random variable (also $\overline{X}$ and S^2 are independent, see Theorems 2.75, and 2.40, and Remark 2.76). Hence the left hand side must be $\chi^2(n-1)$.

■

If someone claims that a sample comes from a specific normal distribution (i.e. $X_i \sim N(\mu, \sigma^2)$, then we can examine probabilities of specific outcomes of the sample variance. Such calculation will be termed hypothesis test in later chapters.

Example 2.82 Milk dose machines

A manufacture of machines for dosing milk claims that their machines can dose with a precision defined by the normal distribution with a standard deviation less than 2% of the dose volume in the operation range. A sample of $n = 20$ observations was taken to check if the precision was as claimed. The sample standard deviation was calculated to $s = 0.03$.

Hence the claim is that $\sigma \leq 0.02$, thus we want to answer the question: if $\sigma = 0.02$ (i.e. the upper limit of the claim), what is then the probability of getting the sampling deviation $s \geq 0.03$?

```python
# Chi-square milk dosing precision
# The sample size
n = 20
# The claimed deviation
sigma = 0.02
# The observed sample standard deviation
s = 0.03
# Calculate the chi-square statistic
chiSq = (n-1)*s**2 / sigma**2
# Use the cdf to calculate the probability of getting the observed
# sample standard deviation or higher
1 - stats.chi2.cdf(chiSq, df=n-1)
```

```
np.float64(0.0014022691601097703)
```

It seems very unlikely that the standard deviation is below 0.02 since the probability of obtaining the observed sample standard deviation under this condition is very small. The probability we just found will be termed a p-value in later chapters - the p-value a very fundamental in testing of hypothesis.

The probability calculated in the above example will be called the p-value in later chapters and it is a very fundamental concept in statistics.

Theorem 2.83 Mean and variance

Let $X \sim \chi^2(\nu)$ then the mean and variance of X is

$$\mathrm{E}(X) = \nu; \quad \mathrm{V}(X) = 2\nu. \tag{2-97}$$

We will omit the proof of this theorem, but it is easily checked by a symbolic calculation software (like e.g. Maple).

|||| Example 2.84

We want to calculate the expected value of the sample variance (S^2) based on n observations with $X_i \sim N(\mu, \sigma^2)$. We have already seen that $\frac{n-1}{\sigma^2} S^2 \sim \chi^2(n-1)$ and we can therefore write

$$\begin{aligned} \mathrm{E}(S^2) &= \frac{\sigma^2}{n-1}\frac{n-1}{\sigma^2}\,\mathrm{E}(S^2) \\ &= \frac{\sigma^2}{n-1}\,\mathrm{E}\left(\frac{n-1}{\sigma^2}S^2\right) \\ &= \frac{\sigma^2}{n-1}(n-1) = \sigma^2, \end{aligned}$$

and we say that S^2 is a central estimator for σ^2 (the term *estimator* is introduced in Section 3.1.3). We can also find the variance of the estimator

$$\begin{aligned} \mathrm{V}(S^2) &= \left(\frac{\sigma^2}{n-1}\right)^2 \mathrm{V}\left(\frac{n-1}{\sigma^2}S^2\right) \\ &= \frac{\sigma^4}{(n-1)^2}2(n-1) = 2\frac{\sigma^4}{n-1}. \end{aligned}$$

|||| Example 2.85 Pooled variance

Suppose now that we have two different samples (not yet realized) $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ with $X_i \sim N(\mu_1, \sigma^2)$ and $Y_i \sim N(\mu_2, \sigma^2)$ (both *i.i.d.*). Let S_1^2 be the sample variance based on the X's and S_2^2 be the sample variance based on the Y's. Now both S_1^2 and S_2^2 will be central estimators for σ^2, and so will any weighted average of the type

$$S^2 = aS_1^2 + (1-a)S_2^2; \quad a \in [0,1].$$

Now we would like to choose a such that the variance of S^2 is as small as possible, and hence we calculate the variance of S^2

$$\begin{aligned} \mathrm{V}(S^2) &= a^2 2\frac{\sigma^4}{n_1-1} + (1-a)^2 2\frac{\sigma^4}{n_2-1} \\ &= 2\sigma^4\left(a^2\frac{1}{n_1-1} + (1-a)^2\frac{1}{n_2-1}\right). \end{aligned}$$

In order to find the minimum we differentiate with respect to a

$$\begin{aligned}\frac{\partial \mathrm{V}(S^2)}{\partial a} &= 2\sigma^4\left(2a\frac{1}{n_1-1} - 2(1-a)\frac{1}{n_2-1}\right)\\ &= 4\sigma^4\left(a\left(\frac{1}{n_1-1}+\frac{1}{n_2-1}\right)-\frac{1}{n_2-1}\right)\\ &= 4\sigma^4\left(a\frac{n_1+n_2-2}{(n_1-1)(n_2-1)}-\frac{1}{n_2-1}\right),\end{aligned}$$

which is zero for

$$a = \frac{n_1-1}{n_1+n_2-2}.$$

In later chapters we will refer to this choice of a as the pooled variance (S_p^2), inserting in (2-98) gives

$$S_p^2 = \frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}.$$

Note that S_p^2 is a weighted (proportional to the number of observations) average of the sample variances. It can also be shown (you are invited to do this) that $\frac{n_1+n_2-2}{\sigma^2}S_p^2 \sim \chi^2(n_1+n_2-2)$. Further, note that the assumption of equal variance in the two samples is crucial in the calculations above.

2.10.2 The t-distribution

The *t-distribution* is the *sampling distribution* of the sample mean standardized with the sample variation. It is valid for all sample sizes, however for larger sample sizes ($n > 30$) the difference between the t-distribution and the normal distribution is very small. Hence for larger sample sizes the normal distribution is often applied.

|||| Definition 2.86

The t-distribution *pdf* is

$$f_T(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\,\Gamma(\frac{\nu}{2})}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{2-98}$$

where ν is the degrees of freedom and $\Gamma()$ is the Gamma function.

The relation between normal random variables and χ^2-distributed random variables are given in the following theo:rem

Theorem 2.87

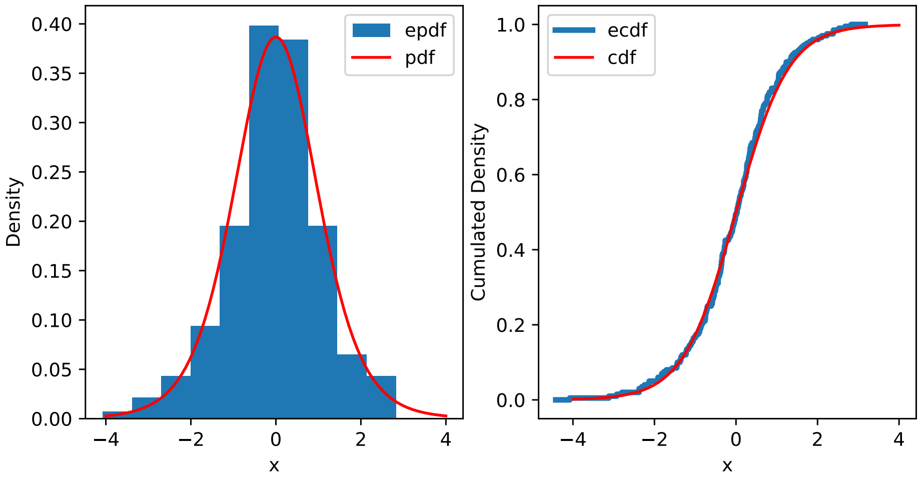
Let $Z \sim N(0,1)$ and $Y \sim \chi^2(\nu)$, then

$$X = \frac{Z}{\sqrt{Y/\nu}} \sim t(\nu). \tag{2-99}$$

We will not prove this theorem, but show by an example how this can be illustrated by simulation:

Example 2.88 Relation between normal and χ^2

```
# Set simulate parameters
nu = 8; k = 200
# Generate the simulated realizations
z = stats.norm.rvs(size=k)
y = stats.chi2.rvs(size=k, df=nu)
x = z/np.sqrt(y/nu)
# Plot
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.hist(x, density=True, label=’epdf’)
ax1.plot(range(-4,5), stats.t.pdf(range(-4,5), df=nu), color="red",
label=’pdf’)
stats.ecdf(x).cdf.plot(ax2, label=’ecdf’, linewidth=3)
ax2.plot(range(-4,5), stats.t.cdf(range(-4,5), df=nu), color="red",
label=’cdf’)
```



In the left plot the empirical *pdf* is compared to the theoretical *pdf* and in the right plot the empirical *cdf* is compared to the theoretical *cdf*.

The t-distribution arises when a sample is taken of a normal distributed random variable, then the sample mean standardized with the sample variance follows the t-distribution.

Theorem 2.89

Given a sample of normal distributed random variables $X_1, \dots, X_n$, then the random variable

$$T = \frac{\overline{X} - \mu}{S/\sqrt{n}} \sim t(n-1), \tag{2-100}$$

follows the t-distribution, where $\overline{X}$ is the sample mean, μ is the mean of X, n is the sample size and S is the sample standard deviation.

|||| Proof

Note that $\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$ and $\frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1)$ which inserted in Equation (2.87) gives

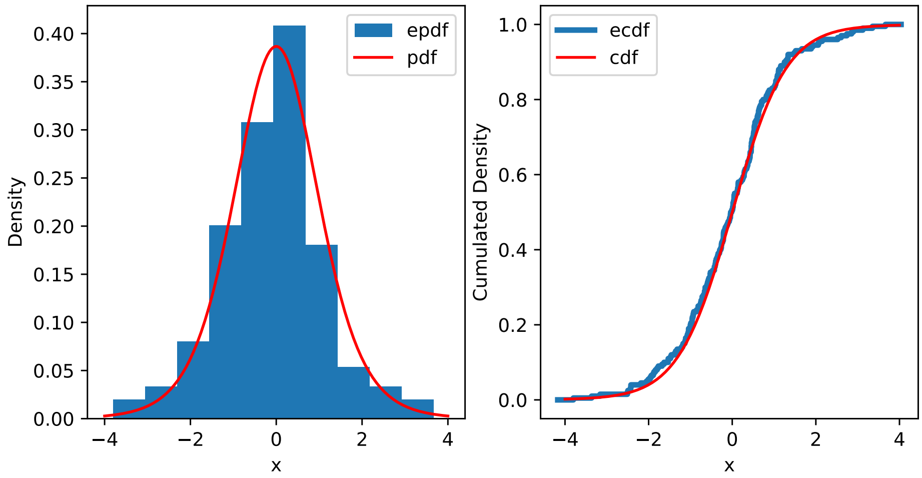
$$\begin{aligned} T &= \frac{\frac{\overline{X}-\mu}{\sigma/\sqrt{n}}}{\sqrt{\frac{(n-1)S^2}{\sigma^2(n-1)}}} = \frac{\frac{\overline{X}-\mu}{1/\sqrt{n}}}{\sqrt{S^2}} \\ &= \frac{\overline{X}-\mu}{S/\sqrt{n}} \sim t(n-1). \end{aligned} \tag{2-101}$$

■

We could also verify this by simulation:

|||| Example 2.90 Simulation of t-distribution

```
# Simulate
n = 8; k = 200; mu = 1.0; sigma = 2.0
# Repeat k times the simulation of a normal dist. sample:
# return the values in a (n x k) matrix
x = [stats.norm.rvs(loc=mu, scale=sigma, size=n) for _ in range(k)]
x = np.array(x)
xbar = np.array([np.mean(i) for i in x])
s    = np.array([np.std(i)  for i in x])
tobs = (xbar - mu)/(s/np.sqrt(n))
# Plot
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.hist(tobs, density=True, label=’epdf’)
ax1.plot(np.arange(-4,4.01,0.01), stats.t.pdf(np.arange(-4,4.01,0.01),
df=nu), color="red", label=’pdf’)
stats.ecdf(tobs).cdf.plot(ax2, label=’ecdf’, linewidth=3)
```



In the left plot the empirical *pdf* is compared to the theoretical *pdf* and in the right plot the empirical *cdf* is compared to the theoretical *cdf*.

Note that $\overline{X}$ and S are random variables, since they are the sample mean and standard deviation of a sample consisting of realizations of X, but the sample is not taken yet.

Very often samples with only few observations are available. In this case by assuming normality of the population (i.e. the X_i's are normal distributed) and for a some mean μ, the t-distribution can be used to calculate the probability of obtaining the sample mean in a given range.

|||| Example 2.91 Electric car driving distance

An electric car manufacture claims that their cars can drive on average 400 km on a full charge at a specified speed. From experience it is known that this full charge distance, denote it by X, is normal distributed. A test of $n = 10$ cars was carried out, which resulted in a sample mean of $\bar{x} = 382$ km and a sample deviation of $s = 14$.

Now we can use the t-distribution to calculate the probability of obtaining this value of the sample mean or lower, if their claim about the mean is actually true:

```python
# Calculate the probability of getting the sample mean under the
# conditions that the claim is actually the real mean

# A test of 10 cars was carried out
n = 10
# The claim is that the real mean is 400 km
muX = 400
# From the sample the sample mean was calculated to
xMean = 393
# And the sample deviation was
xSD = 14
# Use the cdf to calculate the probability of obtaining this
# sample mean or a lower value
stats.t.cdf((xMean-muX)/(xSD/np.sqrt(n)), df=n-1, loc=0, scale=1)
```

```
np.float64(0.0741523536832797)
```

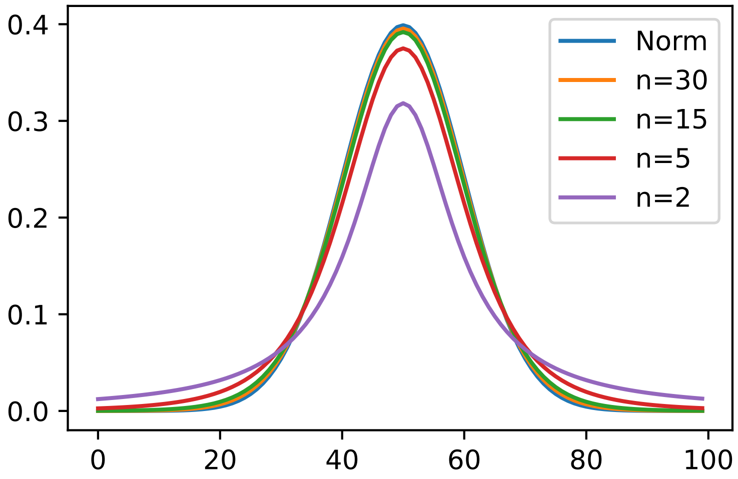
If we had the same sample mean and sample deviation, how do you think changing the number of observations will affect the calculated probability? Try it out.

The t-distribution converges to the normal distribution as the simple size increases. For small sample sizes it has a higher spread than the normal distribution. For larger sample sizes with $n > 30$ observations the difference between the normal and the t-distribution is very small.

|||| Example 2.92 *t*-distribution

Generate plots to see how the t-distribution is shaped compared to the normal distribution.

```
# Plot the t-distribution for different sample sizes

# First plot the standard normal distribution
x = np.arange(-5,5,0.1)
plt.plot(stats.norm.pdf(x, loc=0, scale=1), label=’Norm’)
# Add the t-distribution for 30 observations
plt.plot(stats.t.pdf(x, df=30-1, loc=0, scale=1), label=’n=30’)
# Add the t-distribution for 15, 5 and 2 observations
plt.plot(stats.t.pdf(x, df=15-1, loc=0, scale=1), label=’n=15’)
plt.plot(stats.t.pdf(x, df=5-1, loc=0, scale=1), label=’n=5’)
plt.plot(stats.t.pdf(x, df=2-1, loc=0, scale=1), label=’n=2’)
# Add a legend
plt.legend()
plt.show()
```



How does the number of observations affect the shape of the t-distribution *pdf* compared to the normal *pdf*?

|||| Theorem 2.93 Mean and variance

Let $X \sim t(\nu)$ then the mean and variance of X is

$$\mathrm{E}(X) = 0; \quad \nu > 1, \tag{2-102}$$

$$\mathrm{V}(X) = \frac{\nu}{\nu - 2}; \quad \nu > 2. \tag{2-103}$$

We will omit the proof of this theorem, but it is easily checked with a symbolic calculation software (like e.g. Maple).

|||| Remark 2.94

For $\nu \leq 1$ the expectation (and hence the variance) is not defined (the integral is not absolutely convergent), and for $\nu \in (1, 2]$ $(1 < \nu \leq 2)$ the variance is equal ∞. Note that this does not violate the general definition of probability density functions.

2.10.3 The F-distribution

The F-distribution is defined by:

|||| Definition 2.95

The F-distribution *pdf* is

$$f_F(x) = \frac{1}{B\left(\frac{\nu_1}{2}, \frac{\nu_2}{2}\right)} \left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}} x^{\frac{\nu_1}{2}-1} \left(1 + \frac{\nu_1}{\nu_2} x\right)^{-\frac{\nu_1+\nu_2}{2}}, \tag{2-104}$$

where ν_1 an ν_2 are the degrees of freedom and $B(\cdot, \cdot)$ is the Beta function.

The F-distribution appears as the ratio between two independent χ^2-distributed random variables:

Theorem 2.96

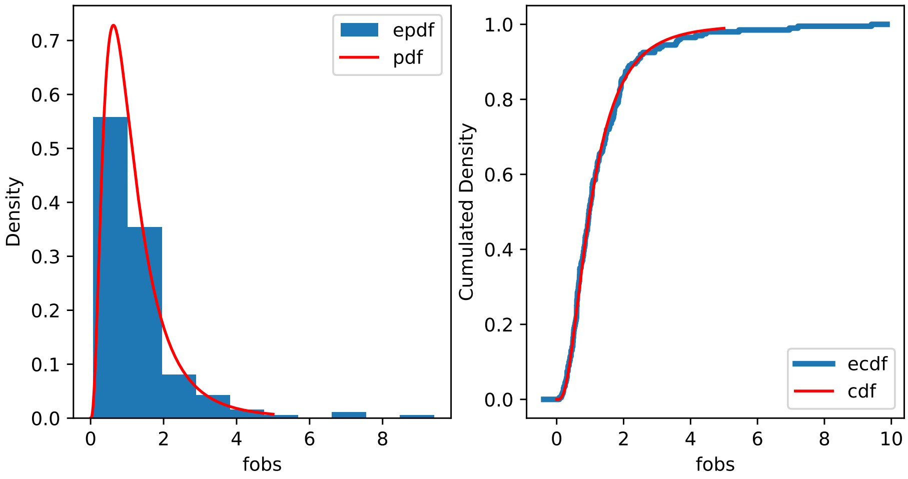
Let $U \sim \chi^2(\nu_1)$ and $V \sim \chi^2(\nu_2)$, be independent then

$$F = \frac{U/\nu_1}{V/\nu_2} \sim F(\nu_1, \nu_2). \tag{2-105}$$

Again we will omit the proof of the theorem and rather show how it can be visualized by simulation:

Example 2.97 *F*-distribution

```
# Simulate
nu1 = 8; nu2 = 10; k = 200
u = stats.chi2.rvs(size=k, df=nu1)
v = stats.chi2.rvs(size=k, df=nu2)
fobs = (u/nu1) / (v/nu2)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.hist(fobs, density=True, label='epdf')
ax1.plot(np.arange(0,5.01,0.01), stats.f.pdf(np.arange(0,5.01,0.01),
dfn=nu1, dfd=nu2), color="red", label='pdf')
stats.ecdf(fobs).cdf.plot(ax2, label='ecdf', linewidth=3)
ax2.plot(np.arange(0,5.01,0.01), stats.f.cdf(np.arange(0,5.01,0.01),
dfn=nu1, dfd=nu2), color="red", label='cdf')
```



In the left plot the empirical *pdf* is compared to the theoretical *pdf* and in the right plot the empirical *cdf* is compared to the theoretical *cdf*.

Theorem 2.98

Let $X_1, \ldots, X_{n_1}$ be independent and sampled from a normal distribution with mean μ_1 and variance σ_1^2, further let $Y_1, \ldots, Y_{n_2}$ be independent and sampled from a normal distribution with mean μ_2 and variance σ_2^2. Then the statistic

$$F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \sim F(n_1 - 1, n_2 - 1), \tag{2-106}$$

follows an F-distribution.

Proof

Note that $\frac{(n_1-1)S_1^2}{\sigma_1^2} \sim \chi^2(n_1 - 1)$ and $\frac{(n_2-1)S_2^2}{\sigma_2^2} \sim \chi^2(n_2 - 1)$ and hence

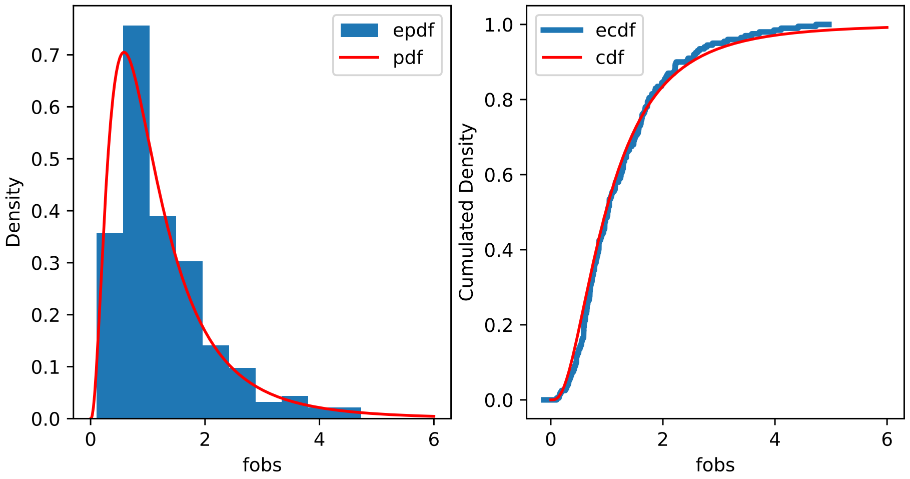
$$\frac{\frac{(n_1-1)S_1^2}{\sigma_1^2(n_1-1)}}{\frac{(n_2-1)S_2^2}{\sigma_2^2(n_2-1)}} = \frac{\frac{S_1^2}{\sigma_1^2}}{\frac{S_2^2}{\sigma_2^2}} \sim F(n_1 - 1, n_2 - 1). \tag{2-107}$$

■

We can also illustrate this sample version by simulation:

|||| Example 2.99 Relation between normal and F-distribution

```
# Simulate
n1 = 8; n2 = 10; k = 200
mu1 = 2; mu2 = -1
sigma1 = 2; sigma2 = 4
s1 =  np.array([np.std(stats.norm.rvs(size=n1, loc=mu1, scale=sigma1))
for _ in range(k)])
s2 =  np.array([np.std(stats.norm.rvs(size=n2, loc=mu2, scale=sigma2))
for _ in range(k)])
fobs = (s1**2 / sigma1**2) / (s2**2 / sigma2**2)
# Plot
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.hist(fobs, density=True, label=’epdf’)
ax1.plot(np.arange(0,6.01,0.01), stats.f.pdf(np.arange(0,6.01,0.01),
dfn=nu1-1, dfd=nu2-1), color="red", label=’pdf’)
stats.ecdf(fobs).cdf.plot(ax2, label=’ecdf’, linewidth=3)
ax2.plot(np.arange(0,6.01,0.01), stats.f.cdf(np.arange(0,6.01,0.01),
dfn=nu1-1, dfd=nu2-1), color="red", label=’cdf’)
```



In the left plot the empirical *pdf* is compared to the theoretical *pdf* and in the right plot the empirical *cdf* is compared to the theoretical *cdf*.

|||| Remark 2.100

Of particular importance in statistics is the case when $\sigma_1 = \sigma_2$, in this case

$$F = \frac{S_1^2}{S_2^2} \sim F(n_1 - 1, n_2 - 1). \tag{2-108}$$

|||| Theorem 2.101 Mean and variance

Let $F \sim F(\nu_1, \nu_2)$ then the mean and variance of F is

$$\mathrm{E}(F) = \frac{\nu_2}{\nu_2 - 2}; \quad \nu_2 > 2, \tag{2-109}$$

$$\mathrm{V}(F) = \frac{2\nu_2^2(\nu_1 + \nu_2 - 2)}{\nu_1(\nu_2 - 2)^2(\nu_2 - 4)}; \quad \nu_2 > 4. \tag{2-110}$$

Chapter 3

Statistics for one and two samples

3.1 Learning from one-sample quantitative data

Statistics is the art and science of learning from data, i.e. statistical inference. What we are usually interested in learning about is the population from which our sample was taken, as described in Section 1.3. More specifically, most of the time the aim is to learn about the mean of this population, as illustrated in Figure 1.1.

Example 3.1 Student heights

In examples in Chapter 1 we did descriptive statistics on the following random sample of the heights of 10 students in a statistics class (in cm):

$$168 \quad 161 \quad 167 \quad 179 \quad 184 \quad 166 \quad 198 \quad 187 \quad 191 \quad 179$$

and we computed the sample mean and standard deviation to be

$$\begin{aligned} \bar{x} &= 178, \\ s &= 12.21. \end{aligned}$$

The population distribution of heights will have some unknown mean μ and some unknown standard deviation σ. We use the sample values as point estimates for these population parameters

$$\begin{aligned} \hat{\mu} &= 178, \\ \hat{\sigma} &= 12.21. \end{aligned}$$

Since we only have a sample of 10 persons, we know that the point estimate of 178 cannot with 100% certainty be exactly the true value μ (if we collected a new sample with 10 different persons height and computed the sample mean we would definitely expect this to be different from 178). The way we will handle this uncertainty is by computing an interval called the *confidence interval* for μ. The confidence interval is a way to handle the uncertainty by the use of probability theory. The most commonly used confidence interval would in this case be

$$178 \pm 2.26 \cdot \frac{12.21}{\sqrt{10}},$$

which is

$$178 \pm 8.74.$$

The number 2.26 comes from a specific probability distribution called the t-distribution, presented in Section 2.86. The t-distributions are similar to the standard normal distribution presented in Section 2.5.2: they are symmetric and centred around 0.

The confidence interval interval

$$178 \pm 8.74 = [169.3,\ 186.7],$$

represents the plausible values of the unknown population mean μ in light of the data.

So in this section we will explain how to *estimate* the mean of a distribution and how to quantify the *precision*, or equivalently the *uncertainty*, of our estimate.

We will start by considering a population characterized by some distribution from which we take a sample $x_1, \ldots, x_n$ of size n. In the example above X_i would be the height of a randomly selected person and $x_1, \ldots, x_{10}$ our sample of student heights.

A crucial issue in the confidence interval is to use the correct probabilities, that is, we must use probability distributions that are properly representing the real life phenomena we are investigating. In the height example, the population distribution is the distribution of all heights in the entire population. So, this is what you would see if you sampled from a huge amount of heights, say $n = 1000000$, and then made a density histogram of these, see Example 1.25. Another way of saying the same is: the random variables X_i have a probability density function (*pdf* or $f(x)$) which describe exactly the distribution of all the values. Well, in our setting we have only a rather small sample, so in fact we may have to assume some specific *pdf* for X_i, since we don't know it and really can't see it well from the small sample. The most common type of assumption, or one could say *model*, for the population distribution is to assume it to be the normal distribution. This assumption makes the theoretical justification for the methods easier. In many cases real life phenomena actually indeed are nicely modelled by a normal distribution. In many other cases they are not. After taking you through the methodology based on a normal population distribution assumption, we will show and discuss what to do with the non-normal cases.

Hence, we will assume that the *random variable* X_i follows a *normal distribution* with mean μ and variance σ^2:

|||| Remark 3.2 How to write a statistical model

In all statistical analysis there must be an assumption of a *model*, which should be stated clearly in the presentation of the analysis. The model expressing that the sample was taken randomly from the population, which is normal distributed, can be written by

$$X_i \sim N(\mu, \sigma^2) \text{ and i.i.d., where } i = 1, \ldots, n. \tag{3-1}$$

Hence we n random variables representing the sample and they are *independent and identically distributed* (i.i.d).

Our goal is to learn about the mean of the population μ, in particular, we want to:

1. *Estimate* μ, that is calculate a best guess of μ based on the sample
2. Quantify the precision, or equivalently the uncertainty, of the estimate

Intuitively, the best guess of the population mean μ is the sample mean

$$\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Actually, there is a formal theoretical framework to support that this sort of obvious choice also is the theoretically best choice, when we have assumed that the underlying distribution is normal. The next sections will be concerned with answering the second question: quantifying how precisely $\bar{x}$ estimates μ, that is, how close we can expect the sample mean $\bar{x}$ to be to the true, but unknown, population mean μ. To answer this, we first, in Section 3.1.1, discuss the distribution of the sample mean, and then, in Section 3.1.2, discuss the *confidence interval* for μ, which is universally used to quantify precision or uncertainty.

3.1.1 Distribution of the sample mean

As indicated in Example 3.1 the challenge we have in using the sample mean $\bar{x}$ as an estimate of μ is the unpleasant fact that the next sample we take would give us a different result, so there is a clear element of randomness in our estimate. More formally, if we take a new sample from the population, let us call it $x_{2,1}, \ldots, x_{2,n}$, then the sample mean of this, $\bar{x}_2 = \frac{1}{n}\sum_{i=1}^{n} x_{2,i}$ will be different from the sample mean of the first sample we took. In fact, we can repeat this process as many times as we would like, and we would obtain:

1. Sample $x_{1,1}, \ldots, x_{1,n}$ and calculate the average $\bar{x}_1$
2. Sample $x_{2,1}, \ldots, x_{2,n}$ and calculate the average $\bar{x}_2$
3. Sample $x_{3,1}, \ldots, x_{3,n}$ and calculate the average $\bar{x}_3$
4. etc.

Since the sample means $\bar{x}_j$ will all be different, it is apparent that *the sample mean is also the realization of a random variable*. In fact it can be shown that if X is a random variable with a normal distribution with mean μ and variance σ^2, then the random sample mean $\bar{X}$ from a sample of size n is also a normally distributed random variable with mean μ and variance σ^2/n. This result is formally expressed in the following theorem:

|||| Theorem 3.3 The distribution of the mean of normal random variables

Assume that $X_1, \ldots, X_n$ are independent and identically normally distributed random variables, $X_i \sim N(\mu, \sigma^2), i = 1, \ldots, n$, then

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \sim N\left(\mu, \frac{\sigma^2}{n}\right). \tag{3-2}$$

Note how the formula in the theorem regarding the mean and variance of $\bar{X}$ is a consequence of the *mean and variance of linear combinations* Theorem 2.56

$$E(\bar{X}) = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \frac{1}{n}\sum_{i=1}^{n} \mu = \frac{1}{n}n\mu = \mu, \tag{3-3}$$

and

$$\mathrm{V}(\bar{\mathrm{X}}) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{V}(\mathrm{X}_i) = \frac{1}{n^2}\sum_{i=1}^{n}\sigma^2 = \frac{1}{n^2}n\sigma^2 = \frac{\sigma^2}{n}, \tag{3-4}$$

and using Theorem 2.40 it is clear that the mean of normal distributions also is a normal distribution.

> One important point to read from this theorem is that it tells us, at least theoretically, what the variance of the sample mean is, and hence also the standard deviation
>
> $$\sigma_{\bar{\mathrm{X}}} = \frac{\sigma}{\sqrt{n}}. \tag{3-5}$$
>
> Let us elaborate a little on the importance of this. Due to the basic rules for mean and variance calculations, i.e. Theorem 2.56, we know that the difference between $\bar{\mathrm{X}}$ and μ has the same standard deviation
>
> $$\sigma_{(\bar{\mathrm{X}}-\mu)} = \frac{\sigma}{\sqrt{n}}. \tag{3-6}$$
>
> This is the mean absolute difference between the sample estimate $\bar{\mathrm{X}}$ and the true μ, or in other words: this is the mean of the error we will make using the sample mean to estimate the population mean. This is exactly what we are interested in: to use a probability distribution to handle the possible error we make.

In our way of justifying and making explicit methods it is useful to consider the so-called *standardized sample mean*, where the $\bar{\mathrm{X}} - \mu$ is seen relative to its standard deviation, and using the standardization of normal distributions in Theorem 2.43, which states that the standardized sample mean has a standard normal distribution:

|||| Theorem 3.4 The distribution of the σ-standardized mean of normal random variables

Assume that $X_1, \ldots, X_n$ are independent and identically normally distributed random variables, $X_i \sim N\left(\mu, \sigma^2\right)$ where $i = 1, \ldots, n$, then

$$Z = \frac{\bar{\mathrm{X}} - \mu}{\sigma/\sqrt{n}} \sim N\left(0, 1^2\right). \tag{3-7}$$

That is, the standardized sample mean Z follows a standard normal distribution.

However, to somehow use the probabilities to say something clever about how close the estimate $\bar{x}$ is to μ, all these results have a flaw: the population standard deviation σ (true, but unknown) is part of the formula. And in most practical cases we don't know the true standard deviation σ. The natural thing to do is to use the sample standard deviation s as a substitute for (estimate of) σ. However, then the theory above breaks down: the sample mean standardized by the sample standard deviation instead of the true standard deviation no longer has a normal distribution! But luckily the distribution can be found (as a probability theoretical result) and we call such a distribution a t-distribution with $(n-1)$ *degrees of freedom* (for more details see Section 2.10.2):

|||| Theorem 3.5 The distribution of the S-standardized mean of normal random variables

Assume that $X_1, \ldots, X_n$ are independent and identically normally distributed random variables, where $X_i \sim N\left(\mu, \sigma^2\right)$ and $i = 1, \ldots, n$, then

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1), \tag{3-8}$$

where $t(n-1)$ is the t-distribution with $n-1$ degrees of freedom.

A t-distribution, as any other distribution, has a probability density function, presented in Definition 2.86. It is similar in shape to the standard normal distribution: it is symmetric and centred around 0, but it has thicker tails as illustrated in the figure of Example 2.92. Also, the t-distributions are directly available in Python, via the SciPy package as seen also for the other probability distributions, see the overview of distributions in A.2.1. So we can easily work with t-distributions in practice. As indicated, there is a different t-distribution for each n: the larger the n, the closer the t-distribution is to the standard normal distribution.

|||| Example 3.6 Normal and t probabilities and quantiles

In this example we compare some probabilities from the standard normal distribution with the corresponding ones from the t-distribution with various numbers of degrees of freedom.

Let us compare $P(T > 1.96)$ for some different values of n with $P(Z > 1.96)$:

```
# The P(T>1.96) probability  for n=10
print(1-stats.t.cdf(1.96,df=9))

0.04082220273020831

# The P(Z>1.96) probability
print(1-stats.norm.cdf(1.96))

0.024997895148220484

# The P(T>1.96) probability  for n-values, 10, 20, ... ,50
print(1-stats.t.cdf(1.96,df=np.linspace(10, 50, 5)-1))

[0.041 0.032 0.030 0.029 0.028]

# The P(T>1.96) probability  for n-values, 100, 200, ... ,500
print(1-stats.t.cdf(1.96,df=np.linspace(100, 500, 5)-1))

[0.026 0.026 0.025 0.025 0.025]
```

Note how the t-probabilities approach the standard normal probabilities as n increases. Similarly for the quantiles:

```
# The standard normal 97.5% quantile
print(stats.norm.ppf(0.975,loc=0,scale=1))

1.959963984540054


# The t-quantiles for n-values: 10, 20, ... ,50
# (rounded to 3 decimal points)
print(stats.t.ppf(0.975,df=np.linspace(10, 50, 5)-1))

[2.262 2.093 2.045 2.023 2.010]


# The t-quantiles for n-values: 100, 200, ... ,500
# (rounded to 3 decimal points)
print(stats.t.ppf(0.975,df=np.linspace(100, 500, 5)-1))

[1.984 1.972 1.968 1.966 1.965]
```

The sample version of the standard deviation of the sample mean $s/\sqrt{n}$ is called the *Standard Error of the Mean* (and is often abbreviated *SEM*):

|||| Definition 3.7 Standard Error of the mean

Given a sample $X_1, \ldots, X_n$, the *Standard Error of the Mean* is defined as

$$\sigma_{\bar{x}} = \frac{S}{\sqrt{n}}. \tag{3-9}$$

It can also be read as the *Sampling Error* of the mean, and can be called the standard deviation of the *sampling distribution* of the mean.

|||| **Remark 3.8**

Using the phrase *sampling distribution* as compared to just the *distribution* of the mean bears no mathematical/formal distinction: formally a probability distribution is a probability distribution and there exist only one definition of that. It is merely used to emphasize the role played by the distribution of the sample mean, namely to quantify how the sample mean changes from (potential) sample to sample, so more generally, the sample mean has a distribution (from sample to sample), so most textbooks and e.g. Wikipedia would call this distribution a sampling distribution.

3.1.2 Quantifying the precision of the sample mean - the confidence interval

As already discussed above, estimating the mean from a sample is usually not enough: we also want to know how close this estimate is to the true mean (i.e. the population mean). Using knowledge about probability distributions, we are able to quantify the uncertainty of our estimate even without knowing the true mean. Statistical practice is to quantify precision (or, equivalently, uncertainty) with a *confidence interval (CI)*.

In this section we will provide the explicit formula for and discuss confidence intervals for the population mean μ. The theoretical justification, and hence assumptions of the method, is a normal distribution of the population. However, it will be clear in a subsequent section that the applicability goes beyond this if the sample size n is large enough. The standard so-called one-sample confidence interval method is:

Method 3.9 The one sample confidence interval for μ

For a sample $x_1, \ldots, x_n$ the $100(1-\alpha)\%$ confidence interval is given by

$$\bar{x} \pm t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}, \tag{3-10}$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile from the t-distribution with $n-1$ degrees of freedom.[a]

Most commonly used is the 95%-confidence interval:

$$\bar{x} \pm t_{0.975} \cdot \frac{s}{\sqrt{n}}. \tag{3-11}$$

We will reserve the **Method** boxes for specific directly applicable statistical methods/formulas (as opposed to theorems and formulas used to explain, justify or prove various points).

Example 3.10 Student heights

We can now use Method 3.9 to find the 95% confidence interval for the population mean height from the height sample from Example 3.1. We need the 0.975-quantile from the t-distribution with $n-1=9$ degrees of freedom:

```
# The t-quantiles for n=10:
print(stats.t.ppf(0.975,df=9))

2.2621571628540993
```

And we can recognize the already stated result

$$178 \pm 2.26 \cdot \frac{12.21}{\sqrt{10}},$$

which is

$$178 \pm 8.74 = [169.3,\ 186.7].$$

Therefore with high confidence we conclude that the true mean height of the population of students to be between 169.3 and 186.7.

[a] Note how the dependence of n has been suppressed from the notation to leave room for using the quantile as index instead - since using two indices would appear less readable: $t_{n-1,1-\alpha/2}$

The confidence interval is widely used to summarize uncertainty, not only for the sample mean, but also for many other types of estimates, as we shall see in later sections of this chapter and in following chapters. It is quite common to use 95% confidence intervals, but other levels, e.g. 99% are also used (it is presented later in this chapter what the precise meaning of "other levels" is).

|||| Example 3.11 Student heights

Let us try to find the 99% confidence interval for μ for the height sample from Example 3.1. Now $\alpha = 0.01$ and we get that $1 - \alpha/2 = 0.995$, so we need the 0.995-quantile from the t-distribution with $n - 1 = 9$ degrees of freedom:

```
# The t-quantile for n=10
print(stats.t.ppf(0.995,df=9))

3.2498355415921254
```

And we can find the result as

$$178 \pm 3.25 \cdot \frac{12.21}{\sqrt{10}},$$

which is:

$$178 \pm 12.55 = [165.5,\ 190.5].$$

Or explicitly in Python:

```
# The 99% confidence interval for the mean
x = np.array([168, 161, 167, 179, 184, 166, 198, 187, 191, 179])
n = len(x)
print(x.mean() - stats.t.ppf(0.995,df=9) * x.std(ddof=1) / np.sqrt(n))


165.45078999139582


print(x.mean() + stats.t.ppf(0.995,df=9) * x.std(ddof=1) / np.sqrt(n))


190.54921000860418
```

Or using the function `stats.t.interval` from the SciPy package:

```
# The 99% confidence interval for the mean
stats.t.interval(0.99,df=n-1,loc=x.mean(),
scale=x.std(ddof=1)/np.sqrt(n))

(np.float64(165.45078999139582), np.float64(190.54921000860418))
```

Later we will introduce a function from the SciPy package that performs a "*t*-test", which can also be used to calculate confidence intervals.

In our motivation of the confidence interval we used the assumption that the population is normal distributed. Thankfully, as already pointed out above, the validity is not particularly sensitive to the normal distribution assumption. In later sections, we will discuss how to assess if the sample is sufficiently close to a normal distribution, and what we can do if the assumption is not satisfied.

3.1.3 The language of statistics and the process of learning from data

In this section we review what it means to make statistical inference using a confidence interval. We review the concepts, first presented in Section 1.3, of: a population, distribution, a parameter, an estimate, an estimator, and a statistic.

The basic idea in statistics is that there exists a *statistical* population (or just population) which we want to know about or learn about, but we only have a *sample* from that population. The idea is to use the sample to say something about the population. To generalize from the sample to the population, we characterize the population by a distribution (see Definition 1.1 and Figure 1.1).

For example, if we are interested in the weight of eggs lain by a particular species of hen, the population consists of the weights of all currently existing eggs as well as weights of eggs that formerly existed and will (potentially) exist in the future. We may characterize these weights by a normal distribution with mean μ and variance σ^2. If we let X denote the weight of a randomly chosen egg, then we may write $X \sim N(\mu, \sigma^2)$. We say that μ and σ^2 are the parameters of this distribution - we call them *population parameters*.

Naturally, we do not know the values of these *true* parameters, and it is impossible for us to ever know, since it would require that we weighed all possible eggs that have existed or could have existed. In fact the true parameters of the distribution $N(\mu, \sigma^2)$ are unknown and will forever remain unknown.

If we take a random sample of eggs from the population of egg weights, say we make 10 *observations*, then we have $x_1, \ldots, x_{10}$. We call this the *observed sample* or just *sample*. From the sample, we can calculate the *sample mean*, $\bar{x}$. We say that $\bar{x}$ is an *estimate* of the true *population mean* μ (or just *mean*, see Remark 1.3). In general we distinguish estimates of the parameters from the parameters themselves, by adding a hat (circumflex). For instance, when we use the sample mean as an estimate of the mean, we may write $\hat{\mu} = \bar{x}$ for the estimate and μ for the parameter, see the illustration of this process in Figure 1.1.

We denote parameters such as μ and σ^2 by Greek letters. Therefore parameter estimates are Greek letters with hats on them. Random variables such as X are denoted by capital Roman letters. The observed values of the random variables are denoted by lower case instead – we call them *realizations* of the random variables. For example, the sample $x_1, \ldots, x_{10}$ represents actually observed numbers (e.g. the weights of 10 eggs), so they are not random and therefore in lower case. If we consider a *hypothetical* sample it is yet unobserved and therefore random and denoted by, say, $X_1, \ldots, X_n$ and therefore in capital letters, see also Section 2.1.

To emphasize the difference, we say that $X_1, \ldots, X_n$ is a *random sample*, while we say that $x_1, \ldots, x_n$ is a *sample taken at random*; the observed sample is not random when it is observed, but it was produced as a result of n random experiments.

A *statistic* is a function of the data, and it can represent both a fixed value from an observed sample or a random variable from a random (yet unobserved) sample. For example sample average $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is a statistic computed from an observed sample, while $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ is also a statistic, but it is considered a function of a random (yet unobserved) sample. Therefore $\bar{X}$ is itself a random variable with a distribution. Similarly the sample variance S^2 is a random variable, while s^2 is its realized value and just a number.

An *estimator* (not to be confused with an *estimate*) is a *function* that produces an estimate. For example, μ is a parameter, $\hat{\mu}$ is the estimate and we use $\bar{X}$ as an *estimator* of μ. Here $\bar{X}$ is the function that produces the estimate of μ from a sample.

Learning from data is learning about parameters of distributions that describe populations. For this process to be meaningful, the sample should in a meaningful way be representative of the relevant population. One way to ensure that this is the case is to make sure that the sample is taken completely at random from the population, as formally defined here:

|||| Definition 3.12 Random sample

A random sample from an (infinite) population: A set of observations $X_1, ..., X_n$ constitutes a random sample of size n from the infinite population $f(x)$ if:

1. Each X_i is a random variable whose distribution is given by $f(x)$
2. The n random variables are independent

It is a bit difficult to fully comprehend what this definition really amounts to in practice, but in brief one can say that the observations should come from the same population distribution, and that they must each represent truly new information (the independence).

|||| Remark 3.13

Throughout previous sections and the rest of this chapter we assume infinite populations. Finite populations of course exists, but only when the sample constitutes a large proportion of the entire population, is it necessary to adjust the methods we discuss here. This occurs relatively infrequently in practice and we will not discuss such conditions.

3.1.4 When we cannot assume a normal distribution: the Central Limit Theorem

The Central Limit Theorem (CLT) states that the sample mean of independent identically distributed (i.i.d.) random variables converges to a normal distribution:

Theorem 3.14 Central Limit Theorem (CLT)

Let $\bar{X}$ be the sample mean of a random sample of size n taken from a population with mean μ and variance σ^2, then

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}, \tag{3-12}$$

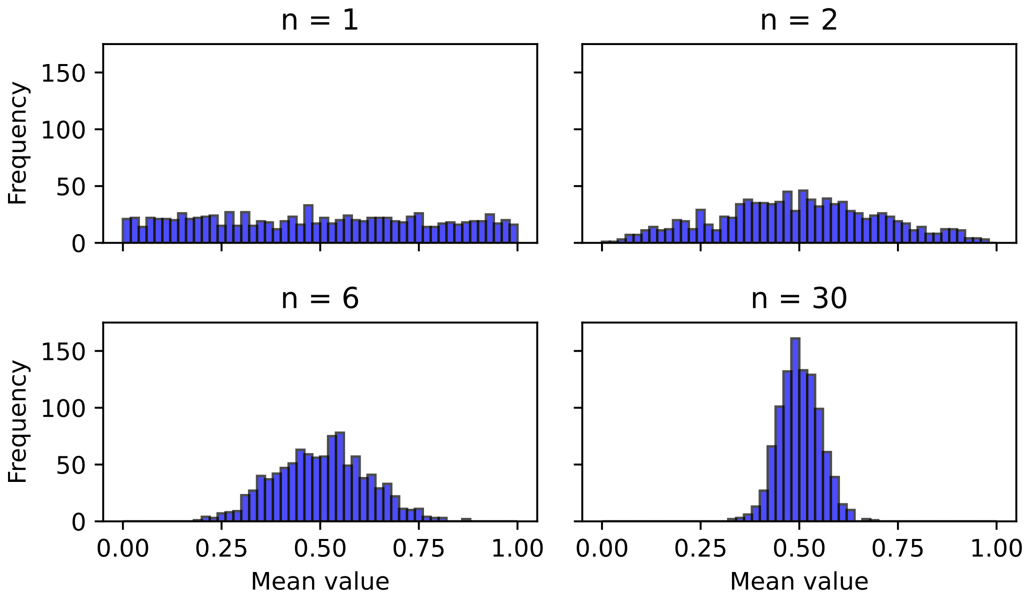
is a random variable which distribution function approaches that of the standard normal distribution, $N(0, 1^2)$, as $n \to \infty$. In other words, for large enough n, it holds approximately that

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1^2). \tag{3-13}$$

The powerful feature of the CLT is that, when the sample size n is large enough, the distribution of the sample mean $\bar{X}$ is (almost) independent of the distribution of the population X. This means that the underlying distribution of a sample can be disregarded when carrying out inference related to the mean. The variance of the sample mean can be estimated from the sample and it can be seen that as n increases the variance of the sample mean decreases, hence the "accuracy" with which we can infer increases.

|||| Example 3.15 Central Limit Theorem in practice

```
# Number of simulated samples
k = 1000
# Number of observations in each sample
n = 1
# Simulate k samples with n observations
Xbar1 = stats.uniform.rvs(0,1, size=(k,n))
# Increase the number of observations in each sample
n = 2
Xbar2 = pd.DataFrame(stats.uniform.rvs(0,1, size=(k,n))).mean(axis=1)
# Increase the number of observations in each sample
n = 6
Xbar6 = pd.DataFrame(stats.uniform.rvs(0,1, size=(k,n))).mean(axis=1)
# Increase the number of observations in each sample
n = 30
Xbar30 = pd.DataFrame(stats.uniform.rvs(0,1, size=(k,n))).mean(axis=1)
# Plot the histograms
fig, axs = plt.subplots(2,2)
axs[0,0].hist(Xbar1, bins=50, range=[0,1], edgecolor='black', color='blue',
alpha=0.7)
axs[0,1].hist(Xbar2, bins=50, range=[0,1], edgecolor='black', color='blue',
alpha=0.7)
axs[1,0].hist(Xbar6, bins=50, range=[0,1], edgecolor='black', color='blue',
alpha=0.7)
axs[1,1].hist(Xbar30, bins=50, range=[0,1], edgecolor='black', color='blue',
alpha=0.7)
plt.tight_layout()
plt.show()
```



Notice how the plot resembles the front page.

Due to the amazing result of the Central Limit Theorem 3.14 many expositions of classical statistics provides a version of the confidence interval based on the standard normal quantiles rather than the t-quantiles

$$\bar{x} \pm z_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}. \tag{3-14}$$

We present it here only as an interesting limit situation of the t-based interval in Method 3.9.

For large samples, the standard normal distribution and t-distribution are almost the same, so in practical situations, it doesn't matter whether the normal based or the t-based confidence interval (CI) is used. Since the t-based interval is also valid for small samples when a normal distribution is assumed, we recommend that the t-based interval in Method 3.9 is used in all situations. This recommendation also has the advantage that the SciPy-function `stats.t.interval`, which produces the t-based interval, can be used in all cases.

How large should the sample then be in a non-normal case to ensure the validity of the interval? No general answer can be given, but as a rule of thumb we recommend $n \geq 30$.

When we have a small sample for which we cannot or will not make a normality assumption, we have not yet presented a valid CI method. The classical solution is to use the so-called non-parametric methods. However, in the next chapter we will present the more widely applicable *simulation* or *re-sampling* based techniques.

3.1.5 Repeated sampling interpretation of confidence intervals

In this section we show that 95% of the 95% confidence intervals we make will cover the true value in the long run. Or, in general $100(1-\alpha)\%$ of the $100(1-\alpha)\%$ confidence intervals we make will cover the true value in the long run. For example, if we make 100 95% CI we cannot guarantee that exactly 95 of these will cover the true value, but if we repeatedly make 100 95% CIs then *on average* 95 of them will cover the true value.

|||| Example 3.16 Simulating many confidence intervals

To illustrate this with a simulation example, then we can generate 50 random $N(1, 1^2)$ distributed numbers and calculate the t-based CI given in Method 3.9, and then repeated this 1000 times to see how many times the true mean $\mu = 1$ is covered. The following code illustrates this:

```python
# Simulate 1000 samples each with 50 observations
x = pd.DataFrame(stats.norm.rvs(loc=1,scale=1,size=(1000,50)))
# Calculate a 95% CI from each sample
CIs = stats.t.interval(0.95,df=50-1,loc=x.mean(axis=1),
scale=x.std(ddof=1,axis=1)/np.sqrt(50))
# Count how often 1 is covered
print(np.sum((CIs[0] <= 1) & (CIs[1] >= 1)))
```

```
954
```

Hence in 954 of the 1000 repetitions (i.e. 95.4%) the CI covered the true value. If we repeat the whole simulation over, we would obtain 1000 different samples and therefore 1000 different CIs. Again we expect that approximately 95% of the CIs will cover the true value $\mu = 1$.

The result that we arrived at by simulation in the previous example can also be derived mathematically. Since

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1),$$

where t is the t-distribution with $n-1$ degrees of freedom, it holds that

$$1 - \alpha = P\left(-t_{1-\alpha/2} < \frac{\bar{X} - \mu}{S/\sqrt{n}} < t_{1-\alpha/2}\right),$$

which we can rewrite as

$$= P\left(\bar{X} - t_{1-\alpha/2}\frac{S}{\sqrt{n}} < \mu < \bar{X} + t_{1-\alpha/2}\frac{S}{\sqrt{n}}\right).$$

Thus, the probability that the interval with limits

$$\bar{X} \pm t_{1-\alpha/2}\frac{S}{\sqrt{n}}, \tag{3-15}$$

covers the true value μ is exactly $1 - \alpha$. One thing to note is that the only difference between the interval above and the interval in Method 3.9, is that the interval above is written with capital letters (simply indicating that it calculated with random variables rather than with observations).

This shows exactly that $100(1-\alpha)\%$ of the $100(1-\alpha)\%$ confidence interval we make will contain the true value in the long run.

3.1.6 Confidence interval for the variance

In previous sections we discussed how to calculate a confidence interval for the mean. In this section we discuss how to calculate a confidence interval for the variance or the standard deviation.

We will assume that the observations come from a normal distribution throughout this section, and we will not present any methods that are valid beyond this assumption. While the methods for the sample mean in the previous sections are not sensitive to (minor) deviations from the normal distribution, the methods discussed in this section for the sample variance rely much more heavily on the correctness of the normal distribution assumption.

|||| Example 3.17 Tablet production

In the production of tablets, an active matter is mixed with a powder and then the mixture is formed to tablets. It is important that the mixture is homogeneous, such that each tablet has the same strength.

We consider a mixture (of the active matter and powder) from where a large amount of tablets is to be produced.

We seek to produce the mixtures (and the final tablets) such that the mean content of the active matter is 1 mg/g with the smallest variance possible. A random sample is collected where the amount of active matter is measured. It is assumed that all the measurements follow a normal distribution.

The variance estimator, that is, the formula for the variance seen as a random variable, is

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2, \tag{3-16}$$

where n is the number of observations, X_i is observation number i where $i = 1, \ldots, n$, and $\bar{X}$ is the estimator of the mean of X.

The (sampling) distribution of the variance estimator is the χ^2-distribution distribution: let S^2 be the variance of a sample of size n from a normal distribution with variance σ^2, then

$$\chi^2 = \frac{(n-1)S^2}{\sigma^2}, \tag{3-17}$$

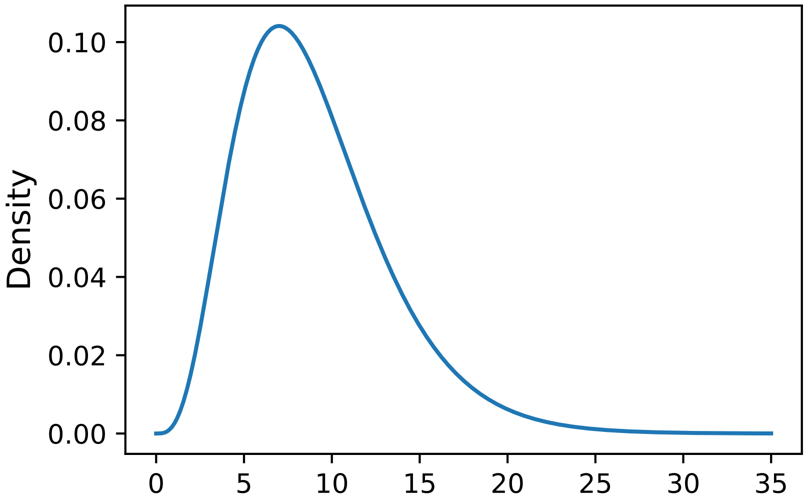
is a stochastic variable following the χ^2-distribution with $v = n - 1$ degrees of freedom.

The χ^2-distribution, as any other distribution, has a probability density function. It is a non-symmetric distribution on the positive axis. It is a distribution of squared normal random variables, for more details see Section 2.10.1. An example of a χ^2-distribution is given in the following:

Example 3.18 The χ^2-distribution

The density of the χ^2-distribution with 9 degrees of freedom is:

```python
# The chi-square-distribution with df=9 (the density)
x = np.linspace(0, 35, 1000)
plt.plot(x,stats.chi2.pdf(x,df=9))
plt.ylabel('Density',fontsize=12)
plt.tight_layout()
plt.show()
```



So, the χ^2-distributions are directly available in Python, via the SciPy package as seen for the other probability distributions presented in the distribution overview, see Appendix A.3.

Hence, we can easily work with χ^2-distributions in practice. As indicated there is a different χ^2-distribution for each n.

Method 3.19 Confidence interval for the variance/standard deviation

A $100(1-\alpha)\%$ confidence interval for the variance σ^2 is

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}},\ \frac{(n-1)s^2}{\chi^2_{\alpha/2}}\right], \tag{3-18}$$

where the quantiles come from a χ^2-distribution with $\nu = n-1$ degrees of freedom.

A $100(1-\alpha)\%$ confidence interval for the standard deviation σ is

$$\left[\sqrt{\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}},\ \sqrt{\frac{(n-1)s^2}{\chi^2_{\alpha/2}}}\right]. \tag{3-19}$$

Note: The confidence intervals for the variance and standard deviations are generally non-symmetric as opposed to the t-based interval for the mean μ.

Example 3.20 Tablet production

A random sample of $n = 20$ tablets is collected and from this the mean is estimated to $\bar{x} = 1.01$ and the variance to $s^2 = 0.07^2$. Let us find the 95%-confidence interval for the variance. To apply the method above we need the 0.025 and 0.975 quantiles of the χ^2-distribution with $\nu = 20 - 1 = 19$ degrees of freedom

$$\chi^2_{0.025} = 8.907,\ \chi^2_{0.975} = 32.85,$$

which we get from Python:

```
# Quantiles of the chi-square distribution:
print(stats.chi2.ppf([0.025,0.975],df=19))

[ 8.907 32.852]
```

Hence the confidence interval is

$$\left[\frac{19 \cdot 0.07^2}{32.85},\ \frac{19 \cdot 0.07^2}{8.907}\right] \approx [0.00283,\ 0.0105],$$

and for the standard deviation the confidence interval is

$$\left[\sqrt{0.002834},\ \sqrt{0.01045}\right] \approx [0.053,\ 0.102]\,.$$

3.1.7 Hypothesis testing, evidence, significance and the p-value

|||| Example 3.21 Sleeping medicine

In a study the aim is to compare two kinds of sleeping medicine A and B. 10 test persons tried both kinds of medicine and the following 10 DIFFERENCES between the two medicine types were measured (in hours):

Person	$x =$ Beffect - Aeffect
1	1.2
2	2.4
3	1.3
4	1.3
5	0.9
6	1.0
7	1.8
8	0.8
9	4.6
10	1.4

For Person 1, Medicine B provided 1.2 sleep hours more than Medicine A, etc.

Our aim is to use these data to investigate if the two treatments are different in their effect on length of sleep. We therefore let μ represent the mean difference in sleep length. In particular we will consider the so-called null hypothesis

$$H_0:\ \mu = 0,$$

which states that there is *no difference* in sleep length between the A and B Medicines.

If the observed sample turns out to be not very likely under this null hypothesis, we conclude that the null hypothesis is unlikely to be true.

First we compute the sample mean

$$\hat{\mu} = \bar{x}_1 = 1.67.$$

As of now, we don't know if this number is particularly small or large. If the true mean difference is zero, would it be unlikely to observe a mean difference this large?

Could it be due to just random variation? To answer this question we compute the probability of observing a sample mean that is 1.67 or further from 0 – in the case that the true mean difference is in fact zero. This probability is called a p-value. If the p-value is small (say less than 0.05), we conclude that the null hypothesis isn't true. If the p-value is not small (say larger than 0.05), we conclude that we haven't obtained sufficient evidence to falsify the null hypothesis.

After some computations that you will learn to perform later in this section, we obtain a p-value

$$p\text{-value} \approx 0.00117,$$

which indicates quite strong evidence against the null hypothesis. As a matter of fact, the probability of observing a mean difference as far from zero as 1.67 or further is only $\approx$ 0.001 (one out of thousand) and therefore very small.

We conclude that the null hypothesis is unlikely to be true as it is highly incompatible with the observed data. We say that *the observed mean* $\hat{\mu} = 1.67$ *is statistically significantly different from zero* (or simply *significant* implying that it is different from zero). Or that *there is a significant difference in treatment effects of B and A*, and we may conclude that Medicine B makes patients sleep significantly longer than Medicine A.

$p < 0.001$	Very strong evidence against H_0
$0.001 \leq p < 0.01$	Strong evidence against H_0
$0.01 \leq p < 0.05$	Some evidence against H_0
$0.05 \leq p < 0.1$	Weak evidence against H_0
$p \geq 0.1$	Little or no evidence against H_0

Table 3.1: A way to interpret the evidence for a given p-value.

The p-value

|||| Definition 3.22 The p-value

The p-value is the probability of obtaining a test statistic that is at least as extreme as the test statistic that was actually observed. This probability is calculated under the assumption that the null hypothesis is true.

Interpretations of a p-value:

1. The p-value measures evidence
2. The p-value measures extremeness/unusualness of the data under the null hypothesis ("under the null hypothesis" means "assuming the null hypothesis is true")

The p-value is used as a general measure of evidence against a null hypothesis: the smaller the p-value, the stronger the evidence against the null hypothesis H_0. A typical strength of evidence scale is given in Table 3.1.

As indicated, the definition and interpretations above are generic in the sense that they can be used for any kind of hypothesis testing in any kind of setup. In later sections and chapters of this material, we will indeed encounter many different such setups. For the specific setup in focus here, we can now give the key method:

Method 3.23 The one-sample t-test statistic and the p-value

For a (quantitative) one sample situation, the p-value is given by

$$p\text{-value} = 2 \cdot P(T > |t_{\text{obs}}|), \tag{3-20}$$

where T follows a t-distribution with $(n-1)$ degrees of freedom.
The observed value of the test statistics to be computed is

$$t_{\text{obs}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}, \tag{3-21}$$

where μ_0 is the value of μ under the null hypothesis

$$H_0: \ \mu = \mu_0. \tag{3-22}$$

The t-test and the p-value will in some cases be used to formalize actual decision making and the risks related to it:

Definition 3.24 The hypothesis test

We say that we carry out a hypothesis test when we decide against a null hypothesis or not, using the data.

A null hypothesis is *rejected* if the p-value, calculated after the data has been observed, is less than some α, that is if the p-value $< \alpha$, where α is some pre-specified (so-called) *significance level*. And if not, then the null hypothesis is said to be *accepted*.

Remark 3.25

Often chosen significance levels α are 0.05, 0.01 or 0.001 with the former being the globally chosen default value.

Remark 3.26

A note of caution in the use of the word *accepted* is in place: this should NOT be interpreted as having proved anything: *accepting* a null hypothesis in statistics simply means that we could not prove it wrong! And the reason for this could just potentially be that we did not collect sufficient amount of data, and *acceptance* hence proofs nothing at its own right.

Example 3.27 Sleeping medicine

Continuing from Example 3.21, we now illustrate how to compute the p-value using Method 3.23.

```
# Enter sleep difference observations
x = np.array([1.2, 2.4, 1.3, 1.3, 0.9, 1.0, 1.8, 0.8, 4.6, 1.4])
n = len(x)
# Compute the tobs - the observed test statistic
tobs = (x.mean() - 0)/(x.std(ddof=1) / np.sqrt(n))
print(tobs)

4.671645978656775
```

```
# Compute the p-value as a tail-probability in the t-distribution
pvalue = 2 * (1-stats.t.cdf(abs(tobs),df=n-1))
print(pvalue)

0.0011658764685527068
```

Naturally, a function in Python can do this for us (the results differ slightly due to numerical inaccuracies). This function can also be used to calculate confidence intervals:

```
stats.ttest_1samp(x,popmean=0).pvalue

np.float64(0.0011658764685528319)
```

```
stats.ttest_1samp(x,popmean=0).confidence_interval()

ConfidenceInterval(low=np.float64(0.8613337442036719), high=np.float64(2.47866625579632
```

The confidence interval and the p-value supplements each other, and often both the confidence interval and the p-value are reported. The confidence interval covers those values of the parameter that we accept given the data, while the p-value measures the extremeness of the data if the null hypothesis is true.

|||| Example 3.28 Sleeping medicine

In the sleep medicine example the 95% confidence interval is

$$[0.86,\ 2.48]\,,$$

so based on the data these are the values for the mean sleep difference of Medicine B versus Medicine A that we accept can be true. Only if the data is so extreme (i.e. rarely occurring) that we would only observe it 5% of the time the confidence interval does not cover the true mean difference in sleep.

The p-value for the null hypothesis $\mu = 0$ was ≈ 0.001 providing strong evidence against the correctness of the null hypothesis.

If the null hypothesis was true, we would only observe this large a difference in sleep medicine effect levels in around one out of a thousand times. Consequently we conclude that the null hypothesis is unlikely to be true and *reject* it.

Statistical significance

The word *significance* can mean *importance* or *the extent to which something matters* in our everyday language. In statistics, however, it has a very particular meaning: if we say that an effect is significant, it means that the p-value is so low that the null hypothesis stating *no effect* has been *rejected* at some *significance level* α.

|||| Definition 3.29 Significant effect

An *effect* is said to be *(statistically) significant* if the p-value is less than the significance level α. [a]

[a] Often, $\alpha = 0.05$ is adopted.

At this point an *effect* would amount to a μ-value different from μ_0. In other contexts we will see later, *effects* can be various features of interest to us.

|||| Example 3.30 Statistical significance

Consider the following two situations:

1. A researcher decides on a significance level of $\alpha = 0.05$ and obtains p-value $= 0.023$. She therefore concludes that the effect is *statistically significant*
2. Another researcher also adopts a significance level of $\alpha = 0.05$, but obtains p-value $= 0.067$. He concludes that the effect was not statistically significant

From a binary decision point of view the two researchers couldn't disagree more. However, from a scientific and more continuous evidence quantification point of view there is not a dramatic difference between the findings of the two researchers.

In daily statistical and/or scientific jargon the word ”statistically” will often be omitted, and when results then are communicated as *significant* further through media or other places, it gives the risk that the distinction between the two meanings gets lost. At first sight it may appear unimportant, but the big difference is the following: sometimes a statistically significant finding can be so small in real size that it is of no real importance. If data collection involves very big data sizes one may find statistically significant effects that for no practical situations matter much or anything at all.

The null hypothesis

The null hypothesis most often expresses the *status quo* or that “nothing is happening”. This is what we have to believe before we perform any experiments and observe any data. This is what we have to accept in the absence of any evidence that the situation is otherwise. For example the null hypothesis in the sleep medicine examples states that the difference in sleep medicine effect level is unchanged by the treatment: this is what we have to accept until we obtain evidence otherwise. In this particular example the observed data and the statistical theory provided such evidence and we could conclude a significant effect.

The null hypothesis has to be *falsifiable*. This means that it should be possible to collect evidence against it.

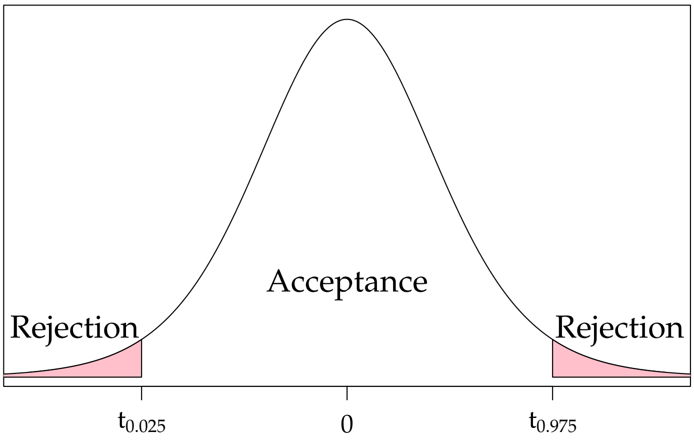


Figure 3.1: The 95% critical value. If t_{obs} falls in the pink area we would *reject*, otherwise we would *accept*

Confidence intervals, critical values and significance levels

A hypothesis test, that is, making the decision between *rejection* and *acceptance* of the null hypothesis, can also be carried out without actually finding the p-value. As an alternative one can use the so-called *critical values*, that is the values of the test-statistic which matches exactly the significance level, see Figure 3.1:

|||| Definition 3.31 The critical values

The $(1-\alpha)100\%$ critical values for the one-sample t-test are the $\alpha/2$- and $1-\alpha/2$-quantiles of the t-distribution with $n-1$ degrees of freedom

$$t_{\alpha/2} \text{ and } t_{1-\alpha/2}. \tag{3-23}$$

|||| Method 3.32 The one-sample hypothesis test by the critical value

A null hypothesis is *rejected* if the observed test-statistic is more extreme than the critical values

$$\text{If } |t_{\text{obs}}| > t_{1-\alpha/2} \text{ then } \textit{reject}, \tag{3-24}$$

otherwise *accept*.

The confidence interval covers the acceptable values of the parameter given the data:

|||| Theorem 3.33 Confidence interval for μ

We consider a $(1-\alpha)\cdot 100\%$ confidence interval for μ

$$\bar{x} \pm t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}. \tag{3-25}$$

The confidence interval corresponds to the acceptance region for H_0 when testing the hypothesis

$$H_0: \quad \mu = \mu_0. \tag{3-26}$$

|||| Remark 3.34

The proof of this theorem is almost straightforward: a μ_0 inside the confidence interval will fulfil that

$$|\bar{x} - \mu_0| < t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}, \tag{3-27}$$

which is equivalent to

$$\frac{|\bar{x} - \mu_0|}{\frac{s}{\sqrt{n}}} < t_{1-\alpha/2}, \tag{3-28}$$

and again to

$$|t_{\text{obs}}| < t_{1-\alpha/2}, \tag{3-29}$$

which then exactly states that μ_0 is accepted, since the t_{obs} is within the critical values.

The alternative hypothesis

Some times we may in addition to the null hypothesis, also explicitly state an *alternative hypothesis*. This completes the framework that allows us to control the rates at which we make correct and wrong conclusions in light of the alternative.

The alternative hypothesis is

$$H_1 : \mu \neq \mu_0. \tag{3-30}$$

This is sometimes called the two-sided (or non-directional) alternative hypothesis, because also one-sided (or directional) alternative hypothesis occur. However, the one-sided setup is not included in the book apart from a small discussion below.

|||| Example 3.35 Sleeping medicine – Alternative hypothesis

Continuing from Example 3.21 we can now set up the null hypothesis and the alternative hypothesis together

$$\begin{aligned} H_0 &: \ \mu = 0 \\ H_1 &: \ \mu \neq 0. \end{aligned}$$

> Which means that we have exactly the same setup just formalized by adding the alternative hypothesis. The conclusion is naturally exactly the same as in before.

A generic approach for tests of hypotheses is:

1. Formulate the hypotheses and choose the level of significance α (choose the "risk-level")

2. Calculate, using the data, the value of the test statistic

3. Calculate the p-value using the test statistic and the relevant sampling distribution, compare the p-value and the significance level α, and finally make a conclusion
   or
   Compare the value of the test statistic with the relevant critical value(s) and make a conclusion

Combining this generic hypothesis test approach with the specific method boxes of the previous section, we can now below give a method box for the one-sample t-test. This is hence a collection of what was presented in the previous section:

|||| Method 3.36 The level α one-sample t-test

1. Compute t_{obs} using Equation (3-21)

$$t_{\text{obs}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

2. Compute the evidence against the *null hypothesis*

$$H_0: \quad \mu = \mu_0, \tag{3-31}$$

vs. the *alternative hypothesis*

$$H_1: \quad \mu \neq \mu_0, \tag{3-32}$$

by the

$$p\text{-value} = 2 \cdot P(T > |t_{\text{obs}}|), \tag{3-33}$$

where the t-distribution with $n - 1$ degrees of freedom is used

3. If the p-value $< \alpha$, we $\underline{\text{reject } H_0}$, otherwise we $\underline{\text{accept } H_0}$,

or

The rejection/acceptance conclusion can equivalently be based on the critical value(s) $\pm t_{1-\alpha/2}$:
if $|t_{\text{obs}}| > t_{1-\alpha/2}$ we $\underline{\text{reject } H_0}$, otherwise we $\underline{\text{accept } H_0}$

The so-called one-sided (or directional) hypothesis setup, where the alternative hypothesis is either "less than" or "greater than", is opposed to the previous presented two-sided (or non-directional) setup, with a "different from" alternative hypothesis. In most situations the two-sided should be applied, since when setting up a null hypothesis with no knowledge about in which direction the outcome will be, then the notion of "extreme" is naturally in both directions. However, in some situations the one-sided setup makes sense to use. As for example in pharmacology where concentrations of drugs are studied and in some situations it is known that the concentration can only decrease from one time point of measurement to another (after the peak concentration). In such case a "less than" is the only meaningful alternative hypothesis – one can say that nature really has made the decision for us in that: either the concentration has not changed (the null hypothesis) or it has dropped (the alternative hypothesis). In other cases, e.g. more from the business and/or judicial perspective, one-sided hypothesis testing come up when for example a claim about the performance of some product is tested.

The one-sided "less than" hypothesis setup is: compute the evidence against the *null hypothesis* vs. the *one-sided alternative hypothesis*

$$H_0: \quad \mu \geq \mu_0 \tag{3-34}$$

$$H_1: \quad \mu < \mu_0, \tag{3-35}$$

by the

$$p\text{-value} = P(T < t_{\text{obs}}). \tag{3-36}$$

and equivalently for the "greater than" setup

$$H_0: \quad \mu \leq \mu_0 \tag{3-37}$$

$$H_1: \quad \mu > \mu_0, \tag{3-38}$$

by the

$$p\text{-value} = P(T > t_{\text{obs}}). \tag{3-39}$$

In both cases: if p-value $< \alpha$: We $\underline{\text{reject } H_0}$, otherwise we $\underline{\text{accept } H_0}$.

Note that there are no one-sided hypothesis testing involved in the exercises.

Errors in hypothesis testing

When testing statistical hypotheses, two kind of errors can occur:

$\underline{\text{Type I:}}$ Rejection of H_0 when H_0 is true

$\underline{\text{Type II:}}$ Non-rejection (acceptance) of H_0 when H_1 is true

|||| Example 3.37 Ambulance times

An ambulance company claims that on average it takes 20 minutes from a telephone call to their switchboard until an ambulance reaches the location.

We might have some measurements (in minutes): $21.1, 22.3, 19.6, 24.2, ...$

If our goal is to show that on average it takes longer than 20 minutes, the null- and the alternative hypotheses are

$$H_0: \quad \mu = 20,$$

$$H_1: \quad \mu \neq 20.$$

What kind of errors can occur?

Type I: Reject H_0 when H_0 is true, that is we mistakenly conclude that it takes longer (or shorter) than 20 minutes for the ambulance to be on location

Type II: Not reject H_0 when H_1 is true, that is we mistakenly conclude that it takes 20 minutes for the ambulance to be on location

|||| Example 3.38 Court of law analogy

A man is standing in a court of law accused of criminal activity.

The null- and the alternative hypotheses are

$$\begin{aligned} H_0 : &\quad \text{The man is not guilty,} \\ H_1 : &\quad \text{The man is guilty.} \end{aligned}$$

We consider a man not guilty until evidence beyond any doubt proves him guilty. This would correspond to an α of basically zero.

Clearly, we would prefer not to do any kinds of errors, however it is a fact of life that we cannot avoid to do so: if we would want to never do a Type I error, then we would never reject the null hypothesis, which means that we would e.g. never conclude that one medical treatment is better than another, and thus, that we would (more) often do a Type II error, since we would never detect when there was a significance effect.

For the same investment (sample size n), we will increase the risk of a Type II error by enforcing a lower risk of a Type I error. Only by increasing n we can lower both of them, but to get both of them very low can be extremely expensive and thus such decisions often involve economical considerations.

The statistical hypothesis testing framework is a way to formalize the handling of the risk of the errors we may make and in this way make decisions in an enlightened way knowing what the risks are. To that end we define the two possible risks as

$$\begin{aligned} P(\text{"Type I error"}) &= \alpha, \\ P(\text{"Type II error"}) &= \beta. \end{aligned} \tag{3-40}$$

This notation is globally in statistical literature. The name choice for the Type I error is in line with the use of α for the *significance level*, as:

|||| Theorem 3.39 Significance level and Type I error

The significance level α in hypothesis testing is the overall Type I risk

$$P(\text{"Type I error"}) = P(\text{"Rejection of } H_0 \text{ when } H_0 \text{ is true"}) = \alpha. \qquad (3\text{-}41)$$

So controlling the Type I risk is what is most commonly apparent in the use of statistics. Most published results are results that became significant, that is, the p-value was smaller than α, and hence the relevant risk to consider is the Type I risk.

Controlling/dealing with the Type II risk, that is: how to conclude on an experiment/study in which the null hypothesis was not rejected (i.e. no significant effect was found) is not so easy, and may lead to heavy discussions if the non-findings even get to the public. To which extent is a non-finding an evidence of the null hypothesis being true? Well, in the outset the following very important saying makes the point:

|||| Remark 3.40

Absence of evidence is NOT evidence of absence!

Or differently put:
Accepting a null hypothesis is NOT a statistical proof of the null hypothesis being true!

The main thing to consider here is that non-findings (non-significant results) may be due to large variances and small sample sizes, so sometimes a non-finding is indeed just that we know nothing. In other cases, if the sample sizes were high, a non-finding may actually, if not proving an effect equal to zero, which is not really possible, then at least indicate with some confidence that the possible effect is small or even very small. The confidence interval is a more clever method to use here, since the confidence interval will show the precision of what we know, whether it includes the zero effect or not.

In Section 3.3 we will use a joint consideration of both error types to formalize the planning of suitably sized studies/experiments.

3.1.8 Assumptions and how to check them

The t-tests that have been presented above are based on some assumptions about the sampling and the population. In Theorem 3.3 the formulations are that the random variables $X_1, \ldots, X_n$ are independent and identically normally distributed: $X_i \sim N(\mu, \sigma^2)$. In this statement there are two assumptions:

- Independent observations
- Normal distribution

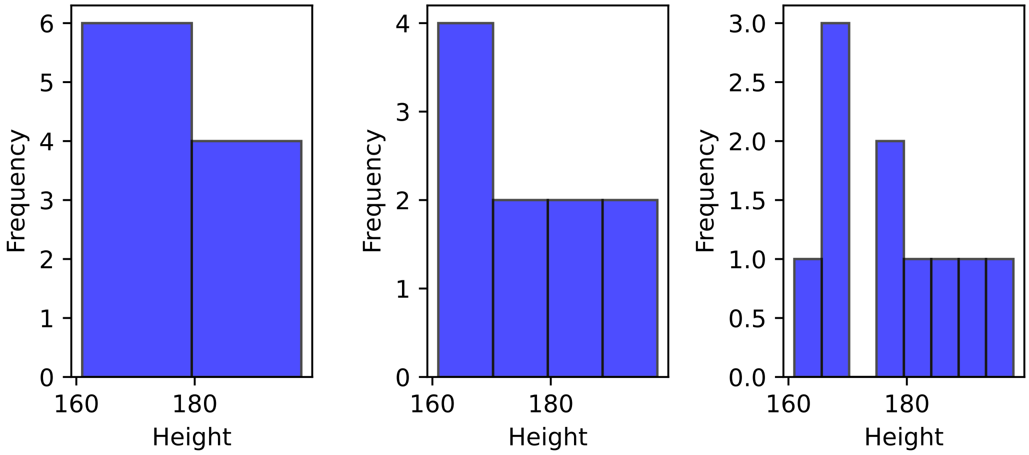
The assumption about independent observations can be difficult to check. It means that each observation must bring a unique new amount of information to the study. Independence will be violated if some measurements are not on randomly selected units and share some feature – returning to the student height example: we do not want to include twins or families in general. Having a sample of $n = 20$ heights, where 15 of them stem from a meeting with a large family group would not be 20 independent observations. The independence assumption is mainly checked by having information about the sampling procedure.

The assumption about normality can be checked graphically using the actual sample at hand.

|||| Example 3.41 Student heights

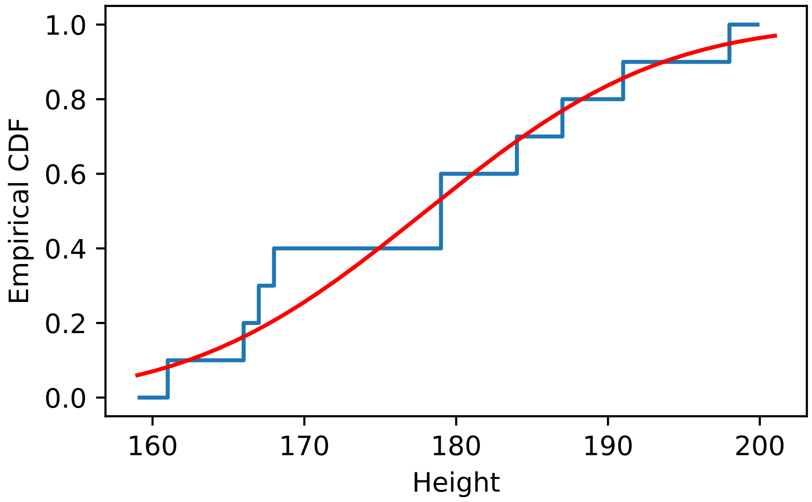
We will return to the height of the ten students from example 3.1. If we want to check whether the sample of heights could come from a normal distribution then we could plot a histogram and look for a symmetric bell-shape:

```
# The height sample
x = np.array([168,161,167,179,184,166,198,187,191,179])

# Using histograms
fig, (ax1, ax2, ax3) = plt.subplots(1, 3)
ax1.hist(x, bins=2, edgecolor='black', color='blue', alpha=0.7)
ax1.set(xlabel='Height', ylabel='Frequency')
ax2.hist(x, bins=4, edgecolor='black', color='blue', alpha=0.7)
ax2.set(xlabel='Height', ylabel='Frequency')
ax3.hist(x, bins=8, edgecolor='black', color='blue', alpha=0.7)
ax3.set(xlabel='Height', ylabel='Frequency')
plt.tight_layout()
plt.show()
```



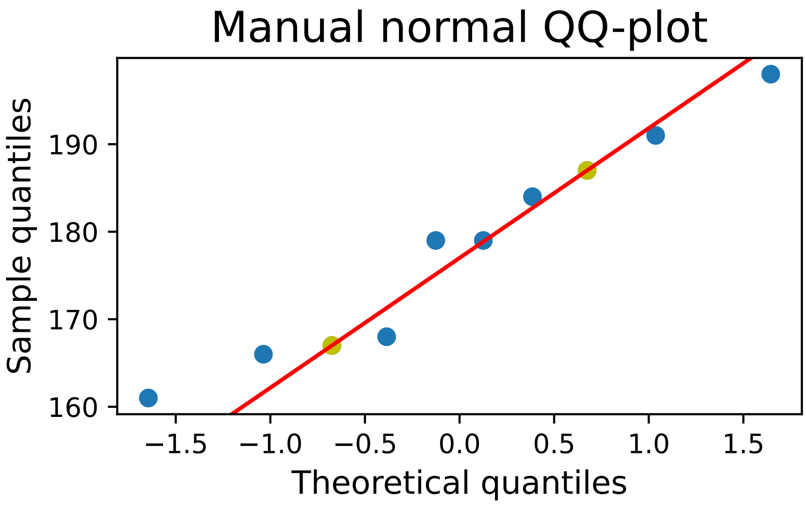
However, as we can see the histograms change shape depending on the number of breaks. Instead of using histograms, one can plot empirical cumulative distribution (see 1.6.2) and compare it with the best fitting normal distribution, in this case $N(\hat{\mu} = 178, \hat{\sigma}^2 = 12.21^2)$:

```
# Plot the empirical cdf
ecdf = stats.ecdf(x)
ax = plt.subplot()
ecdf.cdf.plot(ax)
ax.set(xlabel='Height', ylabel='Empirical CDF')
# Plot a normal cdf
y = np.linspace(159,201, 1000)
plt.plot(y,stats.norm.cdf(y,loc=x.mean(),
scale=x.std(ddof=1)),color="red")
plt.tight_layout()
plt.show()
```



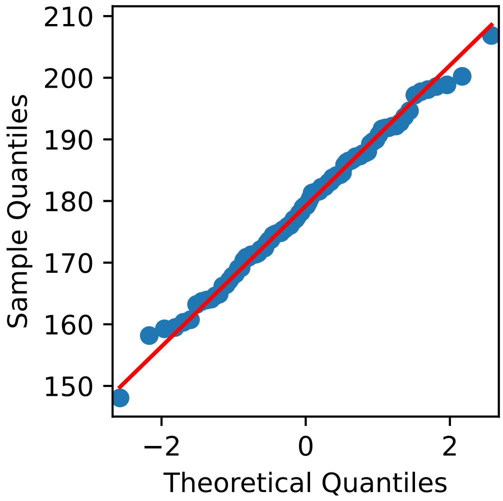
In the accumulated distribution plot it is easier to see how close the distributions are – compared to in the density histogram plot. However, we will go one step further and do the q-q plot: The observations (sorted from smallest to largest) are plotted against the expected quantiles – from the same normal distribution as above. If the observations are normally distributed then the observed are close to the expected and this plot is close to a straight line. In Python we can generate this plot by the following:

```
# A manual normal QQ-plot (normal Quantile-Quantile-plot)
# Calculate manual empirical CDF-values (p) for the observations in the
sample
n = len(x)
p = np.linspace(0.5/n,1-0.5/n,n)
# Plot the theoretical normal quantiles associated with p (x-axis)
against
# the observations. Note that the observations function as the sample
# quantiles. Thus, we compare the theoretical with the sample
quantiles.
plt.scatter(stats.norm.ppf(p),np.sort(x))
# Plot straight line thorugh (TQ1,SQ1) and (TQ3,SQ3).
# T: Theoretical - S: Sample
TQ1 = stats.norm.ppf(0.25)
TQ3 = stats.norm.ppf(0.75)
SQ1 = np.quantile(x,0.25,method='averaged_inverted_cdf')
SQ3 = np.quantile(x,0.75,method='averaged_inverted_cdf')
plt.plot((TQ1,TQ3),(SQ1,SQ3),'yo')
plt.axline((TQ1,SQ1),(TQ3,SQ3),color="red")
# Notice that this not generate the same plot as the standard functions
plt.xlabel('Theoretical quantiles',fontsize=12)
plt.ylabel('Sample quantiles',fontsize=12)
plt.title('Manual normal QQ-plot',fontsize=16)
plt.tight_layout()
plt.show()
```



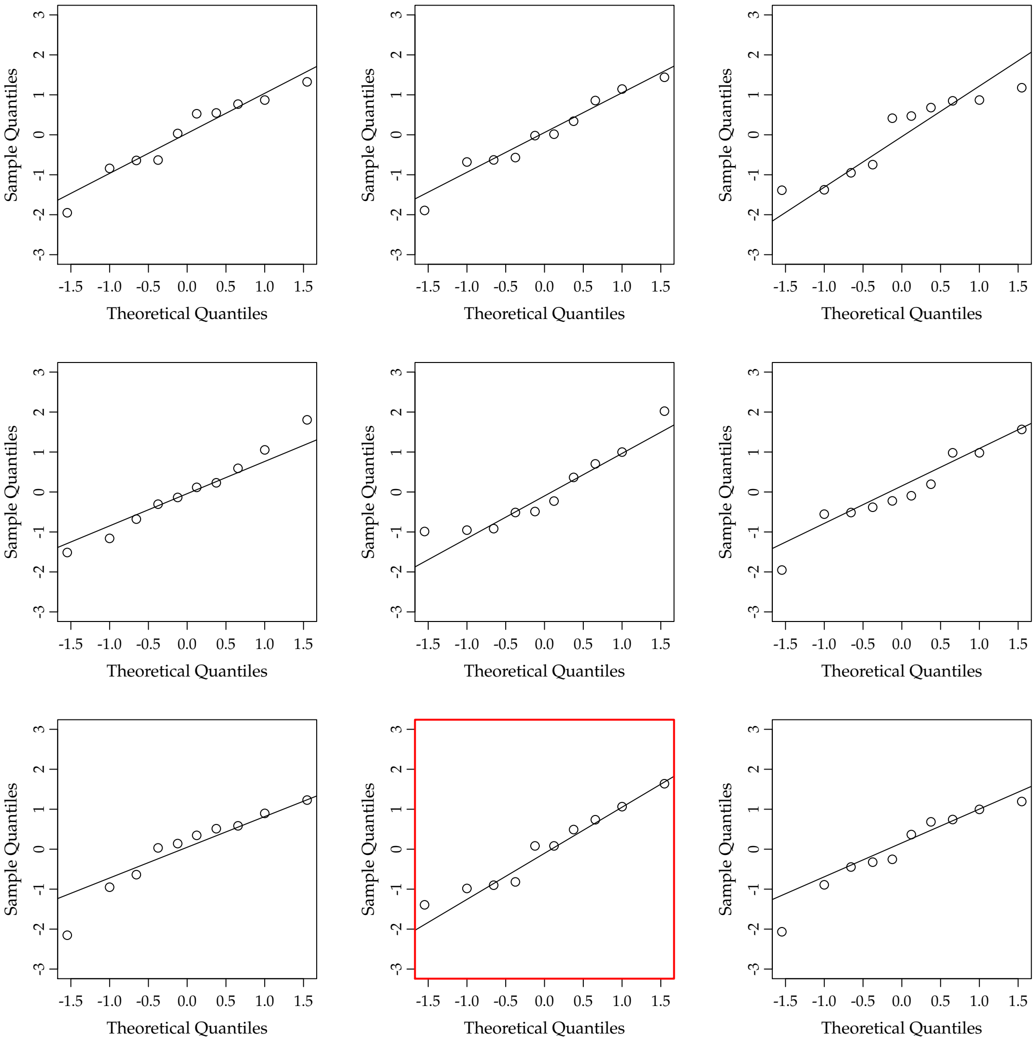
In the ideal normal case, the observations vs. the expected quantiles in the best possible normal distribution will be on a straight line, here plotted with the `line` argument of the `qqplot`-function from the `statsmodels` package:

```
# Simulate 100 observations
np.random.seed(31415)
simx = stats.norm.rvs(loc=x.mean(), scale=x.std(ddof=1), size=100)
# Do the normal QQ-plot and QQ-line with standard functions
sm.qqplot(simx,line="q",a=1/2)
plt.tight_layout()
plt.show()
```



Note that the inbuilt functions do exactly the same as the Python code generating the first q-q plot as described in Method 3.42.

In this example the points are close to a straight line and we can assume that the normal distribution holds. It can, however, be difficult to decide whether the plot is close enough to a straight line, so we write a function that generates one q-q plot of the observations and eight q-q plots with data simulated from a standard normal distribution. It is then possible to visually compare the plot based on the observed data to the simulated data and see whether the distribution of the observations is "worse" than they should be.



When we look at the nine plots then the original data are plotted in the frame with the red border. Comparing the observed data to the simulated data the straight line for the observed data is no worse than some of the simulated data, where the normality assumption is known to hold. So we conclude here that we apparently have no problem in assuming the normal distribution for these data.

|||| Method 3.42 The Normal q-q plot

The ordered observations $x_{(1)}, \ldots, x_{(n)}$, called the sample quantiles, are plotted versus a set of expected normal quantiles $z_{p_1}, \ldots, z_{p_n}$. If the points are not systematically deviating from a line, we accept the normal distribution assumption. The evaluation of this can be based on some simulations of a sample of the same size.

The usual definition of $p_1, \ldots, p_n$ to be used for finding the expected normal quantiles is

$$p_i = \frac{i - 0.5}{n}, \; i = 1, \ldots, n. \tag{3-42}$$

Hence, simply the equally distanced points between $0.5/n$ and $1 - 0.5/n$. This formula is suitable for samples with $n > 10$ and can be used in Python by specifying `qqplot(...,a=1/2)`. For samples with $n \leq 10$, the formula

$$p_i = \frac{i - 3/8}{n + 1/4}, \; i = 1, \ldots, n, \tag{3-43}$$

which can be used in Python by specifying `qqplot(...,a=3/8)`, is preferred.

|||| Example 3.43 Student heights

An example of how the expected normal quantile is calculated in Python can be seen if we take the second smallest height 166. There are 2 observations $\leq$ 166, so $166 = x_{(2)}$ can be said to be the observed $\frac{2-3/8}{10.25} = 0.1585$ quantile (where we use the formula for $n \leq 10$). The 0.1585 quantile in the normal distribution is `stats.norm.ppf(0.1585,loc=0,scale=1)` = -1.00 and the point $(-1.00, 166)$ can be seen on the q-q plot above.

3.1.9 Transformation towards normality

In the above we looked at methods to check for normality. When the data are not normally distributed it is often possible to choose a transformation of the sample, which improves the normality.

When the sample is positive with a long tail or a few large observations then the most common choice is to apply a logarithmic transformation, $\log(x)$. The log-transformation will make the large values smaller and also spread the observations on both positive and negative values. Even though the log-transformation is the most common there are also other possibilities such as $\sqrt{x}$ or $\frac{1}{x}$ for making large values smaller, or x^2 and x^3 for making large values larger.

When we have transformed the sample we can use all the statistical analyse we want. It is important to remember that we are now working on the transformed scale (e.g. the mean and its confidence interval is calculated for $\log(x)$) and perhaps it will be necessary to back-transform to the original scale.

Example 3.44 Radon in houses

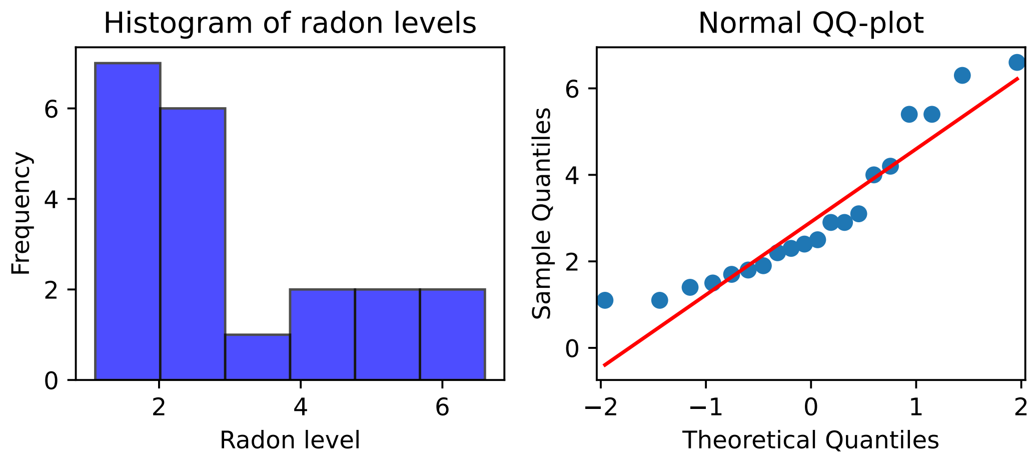
In an American study the radon level was measured in a number of houses. The Environmental Protection Agency's recommended action level is ≥ 4 pCi/L. Here we have the results for 20 of the houses (in pCi/L):

House	1	2	3	4	5	6	7	8	9	10
Radon level	2.4	4.2	1.8	2.5	5.4	2.2	4.0	1.1	1.5	5.4
House	11	12	13	14	15	16	17	18	19	20
Radon level	6.3	1.9	1.7	1.1	6.6	3.1	2.3	1.4	2.9	2.9

The sample mean, median and std. deviance is: $\bar{x} = 3.04$, $Q_2 = 2.45$ and $s_x = 1.72$.

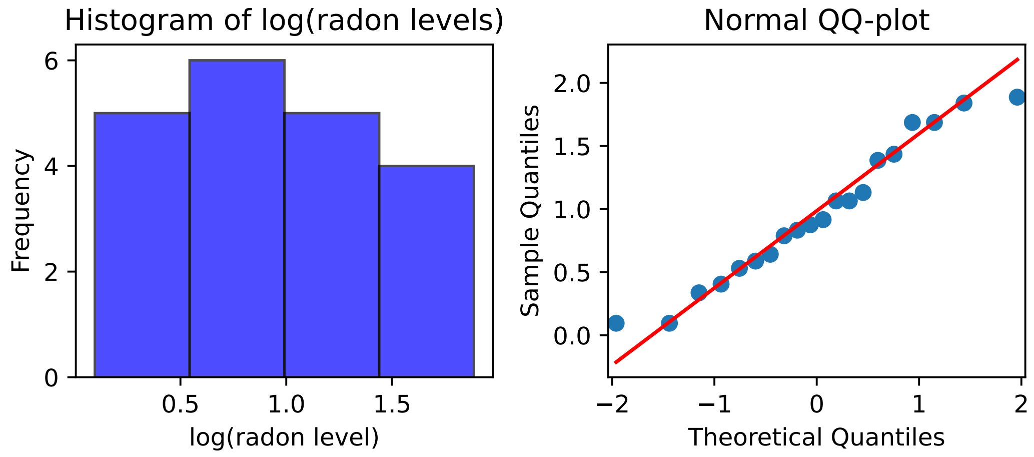
We would like to see whether these observed radon levels could be thought of as coming from a normal distribution. To do this we will plot the data:

```python
# Reading in the sample
radon = np.array([2.4, 4.2, 1.8, 2.5, 5.4, 2.2, 4.0, 1.1, 1.5, 5.4,
                  6.3, 1.9, 1.7, 1.1, 6.6, 3.1, 2.3, 1.4, 2.9, 2.9])

# A histrogram and normal QQ-plot
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.hist(radon, bins=6, edgecolor=’black’, color=’blue’, alpha=0.7)
ax1.set(title="Histogram of radon levels",xlabel="Radon
level",ylabel="Frequency")
sm.qqplot(radon,line="q",a=1/2,ax=ax2)
ax2.set(title="Normal QQ-plot")
plt.tight_layout()
plt.show()
```



From both plots we see that the data are positive and right skewed with a few large observations. Therefore a log-transformation is applied:

```
# Transform using the natural logarithm
logRadon = np.log(radon)

# A histrogram and normal QQ-plot
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.hist(logRadon, bins=4, edgecolor=’black’, color=’blue’, alpha=0.7)
ax1.set(title="Histogram of log(radon levels)",xlabel="log(radon
level)",ylabel="Frequency")
sm.qqplot(logRadon,line="q",a=1/2,ax=ax2)
ax2.set(title="Normal QQ-plot")
plt.tight_layout()
plt.show()
```



As we had expected the log-transformed data seem to be closer to a normal distribution.

We can now calculate the mean and 95% confidence interval for the log-transformed data. However, we are perhaps not interested in the mean of the log-radon levels, then we have to back-transform the estimated mean and confidence interval using $\exp(x)$. When we take the exponential of the estimated mean, then this is no longer a mean but a median on the original pCi/L scale. This gives a good interpretation, as medians are useful when the distributions are not symmetric.

```
# A confidence interval and t-test
n = len(logRadon)
test = stats.ttest_1samp(logRadon,popmean=0)
print(test.statistic,test.pvalue,test.df)

7.793651876947492 2.46529449526264e-07 19


CI = stats.ttest_1samp(logRadon,popmean=0).confidence_interval(0.95)
print(CI)

ConfidenceInterval(low=np.float64(0.7054264972507451), high=np.float64(1.22343071479501


# Alternatively, the CI can be obtained as
CI = stats.t.interval(0.95,df=n-1,loc=logRadon.mean(),
scale=logRadon.std(ddof=1)/np.sqrt(n))
print(CI)

(np.float64(0.7054264972507451), np.float64(1.2234307147950183))


# Back transform to original scale, now we get the median!
# This is a special case: In the lognormal distribution,
# the median coincides with the geometric mean value.
print(np.exp(logRadon.mean()))

2.623288297019726


# And the confidence interval on the original scale
print(np.exp(CI))

[2.025 3.399]
```

From the Python code we see that the mean log-radon level is 0.96 (95% CI: 0.71 to 1.22). On the original scale the estimated median radon level is 2.6 pCi/L (95% CI: 2.0 to 3.4).

|||| Theorem 3.45 Transformations and quantiles

In general, the data transformations discussed in this section will preserve the quantiles of the data. Or more precisely, if f is a data transformation function (an increasing function), then

$$\text{The } p\text{th quantile of } f(Y) = f(\text{The } p\text{th quantile of } Y). \tag{3-44}$$

The consequence of this theorem is that confidence limits on one scale transform easily to confidence limits on another scale even though the transforming function is non-linear.

3.2 Learning from two-sample quantitative data

In this section the setup, where we can learn about the difference between the means from two populations, is presented. This is very often a setup encountered in most fields of science and engineering: compare the quality of two products, compare the performance of two groups, compare a new drug to a placebo and so on. One could say, that it should be called a two-population setup, since it is really two populations (or groups) which are compared by taking a sample from each, however it is called a two-sample setup (probably it sounds better to say).

First, the two-sample setup is introduced with an example and then methods for confidence intervals and tests are presented.

|||| Example 3.46 Nutrition study

In a nutrition study the aim is to investigate if there is a difference in the energy usage for two different types of (moderately physically demanding) work. In the study, the energy usage of 9 nurses from hospital A and 9 (other) nurses from hospital B have been measured. The measurements are given in the following table in mega Joule (MJ):

Hospital A	Hospital B
7.53	9.21
7.48	11.51
8.08	12.79
8.09	11.85
10.15	9.97
8.40	8.79
10.88	9.69
6.13	9.68
7.90	9.19

Our aim is to assess the difference in energy usage between the two groups of nurses. If μ_A and μ_B are the mean energy expenditures for nurses from hospital A and B, then the estimates are just the sample means

$$\begin{aligned}\hat{\mu}_A &= \bar{x}_A = 8.293,\\ \hat{\mu}_B &= \bar{x}_B = 10.298.\end{aligned}$$

To assess the difference in means, $\delta = \mu_B - \mu_A$, we could consider the confidence interval for $\delta = \mu_B - \mu_A$. Clearly, the estimate for the difference is the difference of the sample means, $\hat{\delta} = \hat{\mu}_B - \hat{\mu}_A = 2.005$.

The 95% confidence interval is

$$2.005 \pm 1.412 = [0.59,\ 3.42],$$

which spans the mean differences in energy expenditure that we find acceptable based on the data. Thus we do not accept that the mean difference could be zero.

The interval width, given by 1.41, as we will learn below, comes from a simple computation using the two sample standard deviations, the two sample sizes and a t-quantile.

We can also compute a p-value to measure the evidence against the null hypothesis that the mean energy expenditures are the same. Thus we consider the following null hypothesis

$$H_0:\ \delta = 0.$$

Since the 95% confidence interval does not cover zero, we already know that the p-value for this significance test will be less than 0.05. In fact it turns out that the p-value for this significance test is 0.0083 indicating strong evidence against the null hypothesis that the mean energy expenditures are the same for the two nurse groups. We therefore have strong evidence that the mean energy expenditure of nurses from hospital B is higher than that of nurses from hospital A.

This section describes how to compute the confidence intervals and p-values in such two-sample setups.

3.2.1 Comparing two independent means - confidence Interval

We assume now that we have a sample $x_1, \ldots, x_n$ taken at random from one population with mean μ_1 and variance σ_1^2 and another sample $y_1, \ldots, y_n$ taken at random from another population with mean μ_2 and variance σ_2^2.

|||| Method 3.47 The two-sample confidence interval for $\mu_1 - \mu_2$

For two samples $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ the $100(1-\alpha)\%$ confidence interval for $\mu_1 - \mu_2$ is given by

$$\bar{x} - \bar{y} \pm t_{1-\alpha/2} \cdot \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}, \tag{3-45}$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile from the t-distribution with ν degrees of freedom given from Equation (3-50)

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}}. \tag{3-46}$$

Note how the t-quantile used for the confidence interval is exactly what we called the critical value above.

|||| Example 3.48 Nutrition study

Let us find the 95% confidence interval for $\mu_B - \mu_A$. Since the relevant t-quantile is, using $\nu = 15.99$,

$$t_{0.975} = 2.120,$$

the confidence interval becomes

$$10.298 - 8.293 \pm 2.120 \cdot \sqrt{\frac{2.0394}{9} + \frac{1.954}{9}},$$

which then gives the result as also seen above

$$[0.59,\ 3.42].$$

3.2.2 Comparing two independent means - hypothesis test

We describe the setup as having a random sample from each of two different populations, each described by a mean and a variance:

- Population 1: has mean μ_1, and variance σ_1^2

- Population 2: has mean μ_2, and variance σ_2^2

The interest lies in the comparisons of the means.

|||| Method 3.49 The (Welch) two-sample t-test statistic

When considering the null hypothesis about the difference between the means of two *independent* samples

$$\begin{aligned} \delta &= \mu_2 - \mu_1, \\ H_0: \ \delta &= \delta_0, \end{aligned} \tag{3-47}$$

the (Welch) two-sample t-test statistic is

$$t_{\text{obs}} = \frac{(\bar{x}_1 - \bar{x}_2) - \delta_0}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}. \tag{3-48}$$

|||| Theorem 3.50 The distribution of the (Welch) two-sample statistic

The (Welch) two-sample statistic seen as a random variable

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - \delta_0}{\sqrt{S_1^2/n_1 + S_2^2/n_2}}, \tag{3-49}$$

approximately, under the null hypothesis, follows a t-distribution with ν degrees of freedom, where

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}}, \tag{3-50}$$

if the two population distributions are normal or if the two sample sizes are large enough.

We can now, based on this, express the full hypothesis testing procedures for the two-sample setting:

|||| Method 3.51 The level α two-sample t-test

1. Compute the test statistic using Equation (3-48) and ν from Equation (3-50)

$$t_{\text{obs}} = \frac{(\bar{x}_1 - \bar{x}_2) - \delta_0}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} \text{ and } \nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}}$$

2. Compute the evidence against the *null hypothesis*[a]

$$H_0: \quad \mu_1 - \mu_2 = \delta_0,$$

vs. the *alternative hypothesis*

$$H_1: \quad \mu_1 - \mu_2 \neq \delta_0,$$

by the

$$p\text{-value} = 2 \cdot P(T > |t_{\text{obs}}|),$$

where the t-distribution with ν degrees of freedom is used

3. If p-value $< \alpha$: we reject H_0, otherwise we accept H_0,

or

The rejection/acceptance conclusion can equivalently be based on the critical value(s) $\pm t_{1-\alpha/2}$:
if $|t_{\text{obs}}| > t_{1-\alpha/2}$ we reject H_0, otherwise we accept H_0

[a] We are often interested in the test where $\delta_0 = 0$

An assumption that often is applied in statistical analyses of various kinds is that of the underlying variability being of the same size in different groups or at different conditions. The assumption is rarely crucial for actually carrying out some good statistics, but it may indeed make the theoretical justification for what is done more straightforward, and the actual computational procedures also may become more easily expressed. We will see in later chapters how this comes in play. Actually, the methods presented above do not make this assumption, which is nice. The fewer assumptions needed the better, obviously. Assumptions are problematic in the sense, that they may be questioned for particular applications of the methods.

However, below we will present a version of the two-sample t-test statistic, that actually is adapted to such an assumption, namely assuming that the two population variances are the same: $\sigma_1^2 = \sigma_2^2$. We present it here not because we really need it, we will use the above in all situations. But the version below will appear and be used many places and it also bears some nice relations to later multi-group analysis (Analysis of Variance (ANOVA)) that we will get to in Chapter 8.

If we believe in the equal variance assumption it is natural to compute a single joint – called the *pooled* – estimate of the variance based on the two individual variances:

|||| Method 3.52 The pooled two-sample estimate of variance

Under the assumption that $\sigma_1^2 = \sigma_2^2$ the *pooled* estimate of variance is the weighted average of the two sample variances

$$s_p^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}. \tag{3-51}$$

Note that when there is the same number of observations in the two groups, $n_1 = n_2$, the pooled variance estimate is simply the average of the two sample variances. Based on this the so-called pooled two-sample t-test statistic can be given:

|||| Method 3.53 The pooled two-sample *t*-test statistic

When considering the null hypothesis about the difference between the means of two *independent* samples

$$\begin{aligned} \delta &= \mu_1 - \mu_2, \\ H_0: \ \delta &= \delta_0. \end{aligned} \tag{3-52}$$

the pooled two-sample *t*-test statistic is

$$t_{\text{obs}} = \frac{(\bar{x}_1 - \bar{x}_2) - \delta_0}{\sqrt{s_p^2/n_1 + s_p^2/n_2}}. \tag{3-53}$$

And the following theorem would form the basis for hypothesis test procedures based on the pooled version:

|||| Theorem 3.54 The distribution of the pooled two-sample t-test statistic

The pooled two-sample statistic seen as a random variable:

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - \delta_0}{\sqrt{S_p^2/n_1 + S_p^2/n_2}}. \tag{3-54}$$

follows, under the null hypothesis and under the assumption that $\sigma_1^2 = \sigma_2^2$, a t-distribution with $n_1 + n_2 - 2$ degrees of freedom if the two population distributions are normal.

A little consideration will show why choosing the Welch-version as the approach to always use makes good sense: First of all if $s_1^2 = s_2^2$ the Welch and the Pooled test statistics are the same. Only when the two variances become really different the two test-statistics may differ in any important way, and if this is the case, we would not tend to favour the pooled version, since the assumption of equal variances appears questionable then.

Only for cases with a small sample sizes in at least one of the two groups the pooled approach may provide slightly higher power if you believe in the equal variance assumption. And for these cases the Welch approach is then a somewhat cautious approach.

|||| Example 3.55 Nutrition study

Let us consider the nurses example again, and test the null hypothesis expressing that the two groups have equal means

$$H_0: \ \delta = \mu_A - \mu_B = 0,$$

versus the alternative

$$H_0: \ \delta = \mu_A - \mu_B \neq 0,$$

using the most commonly used significance level, $\alpha = 0.05$. We follow the steps of Method 3.51: we should first compute the test-statistic t_{obs} and the degrees of freedom ν. These both come from the basic computations on the data:

```
# Load the two samples
xA = np.array([7.53, 7.48, 8.08, 8.09, 10.15, 8.4, 10.88, 6.13, 7.9])
xB = np.array([9.21, 11.51, 12.79, 11.85, 9.97, 8.79, 9.69, 9.68,
9.19])

# Summary statistics
print(xA.mean(),xB.mean())

8.293333333333335 10.297777777777776


print(xA.var(ddof=1),xB.var(ddof=1))

2.0394000000000005 1.9540444444444444


print(len(xA),len(xB))

9 9
```

So

$$t_{\text{obs}} = \frac{10.298 - 8.293}{\sqrt{2.0394/9 + 1.954/9}} = 3.01,$$

and

$$\nu = \frac{\left(\frac{2.0394}{9} + \frac{1.954}{9}\right)^2}{\frac{(2.0394/9)^2}{8} + \frac{(1.954/9)^2}{8}} = 15.99.$$

Or the same done in Python by ”manual” expression:

```
# Keep the summary statistics
ms = np.array([xA.mean(),xB.mean()])
vs = np.array([xA.var(ddof=1),xB.var(ddof=1)])
ns = np.array([len(xA),len(xB)])

# The observed statistic
t_obs = (ms[1]-ms[0])/np.sqrt(vs[0]/ns[0]+vs[1]/ns[1])

# The degrees of freedom
nu = ((vs[0]/ns[0]+vs[1]/ns[1])**2)/((vs[0]/ns[0])**2/(ns[0]-1)
+(vs[1]/ns[1])**2/(ns[1]-1))

# Print the result
print(t_obs)

3.009133495521211


print(nu)

15.992693827602634
```

Next step is then to find the p-value

$$p\text{-value} = 2 \cdot P(T > |t_{\text{obs}}|) = 2P(T > 3.01) = 2 \cdot 0.00415 = 0.0083,$$

where we use Python to find the probability $P(T > 3.01)$ based on a t-distribution with $\nu = 15.99$ degrees of freedom:

```
# The probability of observing a value greater that t_obs
print(1 - stats.t.cdf(t_obs,df=nu))

0.004161369978658014
```

To complete the hypothesis test, we compare the p-value with the given α-level, in this case $\alpha = 0.05$, and conclude:

> Since the p-value is less than α we *reject* the null hypothesis, and we have sufficient evidence for concluding: the two nurse groups have on average different energy usage work levels. We have shown this *effect* to be *statistically significant*.

In spite of a pre-defined α-level (whoever gave us that), it is always valuable to consider at what other α-levels the hypothesis would be rejected/accepted. Or in different words, interpret the size of the p-value using Table 3.1 and we thus sharpen the statement a little:

> Since the p-value in this case is between 0.001 and 0.01 conclude: there is a *strong evidence* against equality of the two population energy usage means and it is found that *the mean is significantly higher* on Hospital B compared to Hospital A.

The last part, that the mean is higher on Hospital B, can be concluded because it is rejected that they are equal and $\bar{x}_B > \bar{x}_A$ and we can thus add this to the conclusion.

Finally, the t-test computations are actually directly provided by the `ttest_ind`-function from the SciPy package using the two data input vectors

```
# Use the automatic function for a t-test
test = stats.ttest_ind(xB,xA,equal_var=False)
tobs = test.statistic
pvalue = test.pvalue
df = test.df
print(tobs,pvalue,df)

3.009133495521211 0.00832273995731614 15.992693827602634


stats.ttest_ind(xB,xA,equal_var=False).confidence_interval(0.95)

ConfidenceInterval(low=np.float64(0.5922803841924627), high=np.float64(3.41660850469642
```

Note, how the default choices of the Python-function compare to our exposition:

- Default test version: the pooled test (assuming equal variances)
- Default α-level: 0.05
- Default ”direction version”: the two-sided (or non-directional) alternative hypothesis (see Section 3.1.7 about other alternative hypotheses)

Actually, the final rejection/acceptance conclusion based on the default (or chosen) α-level is not given by Python.

In the `ttest_ind` results the α-level is used for the given confidence interval for the mean difference of the two populations, to be interpreted as: we accept that the true difference in mean energy levels between the two nurse groups is somewhere between 0.6 and 3.4.

Remark 3.56

Often "degrees of freedom" are integer values, but in fact t-distributions with non-integer valued degrees of freedom are also well defined. The $\nu = 15.99$ t-distribution (think of the density function) is a distribution in between the $\nu = 15$ and the $\nu = 16$ t-distributions. Clearly it will indeed be very close to the $\nu = 16$ one.

We did not in the example above use Step 4. of Method 3.51, which can be called the critical value approach. In fact this approach is directly linked to the confidence interval in the sense that one could make a rapid conclusion regarding rejection or not by looking at the confidence interval and checking whether the hypothesized value is in the interval or not. This would correspond to using the critical value approach.

Example 3.57 Nutrition study

In the nutrition example above, we can see that 0 is not in the confidence interval so we would reject the null hypothesis. Let us formally use Step 4 of Method 3.51 to see how this is exactly the same: the idea is that one can even before the experiment is carried out find the critical value(s), in this case:

$$\text{The 5\% critical values} = \pm t_{0.975} = \pm 2.120,$$

where the quantile is found from the t-distribution with $\nu = 15.99$ degrees of freedom:

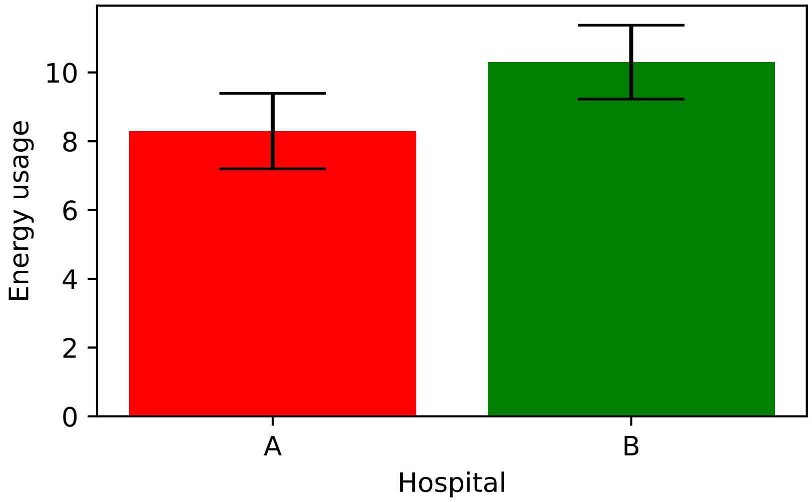
```
# The critical value for the test
print(stats.t.ppf(0.975,df=nu))

2.119984011855833
```

Now we conclude that since the observed t-statistic $t_{\text{obs}} = 3.01$ is beyond the critical values (either larger than 2.120 or smaller than -2.120) the null hypothesis is rejected, and further since it was higher, that $\mu_A - \mu_B > 0$ hence $\mu_B > \mu_A$.

Example 3.58 Overlapping confidence intervals?

A commonly encountered way to visualize the results of a two-sample comparison is to use a bar plot of the means together with some measure of uncertainty, either simply the standard errors of the means or the 95% confidence intervals within each group:

```
# The confidence intervals
CIA = stats.ttest_1samp(xA,popmean=0).confidence_interval(0.95)
CIB = stats.ttest_1samp(xB,popmean=0).confidence_interval(0.95)

# Barplots with error bars
fig, ax = plt.subplots(1, 1)
ax.bar(x=[0,1],height=[xA.mean(),xB.mean()],
yerr=[(CIA[1]-CIA[0])/2,(CIB[1]-CIB[0])/2],capsize=20,color=("r","g"))
ax.set(xlabel="Hospital",ylabel="Energy usage")
ax.set_xticks([0,1],("A","B"))
plt.tight_layout()
plt.show()
```



Here care must taken in the interpretation of this plot: it is natural, if your main aim is a comparison of the two means, to immediately visually check whether the shown error bars, in this case the confidence intervals, overlap or not, to make a conclusion about group difference. Here they actually just overlap - could be checked by looking at the actual CIs:

```
# The confidence intervals
print(CIA)

ConfidenceInterval(low=np.float64(7.195617231957511), high=np.float64(9.391049434709158

print(CIB)

ConfidenceInterval(low=np.float64(9.223278703268573), high=np.float64(11.37227685228698
```

And the conclusion would (incorrectly) be that the groups are not statistically different. However, remind that we found above that the p-value $= 0.008323$, so we concluded that there was strong evidence of a mean difference between the two nurse groups.

The problem of the ”overlapping CI interpretation” illustrated in the example comes technically from the fact that standard deviations are not additive but variances are

$$\begin{aligned} \sigma_{(\bar{X}_A-\bar{X}_B)} &\neq \sigma_{\bar{X}_A}+\sigma_{\bar{X}_B}, \\ \mathrm{V}(\bar{X}_A-\bar{X}_B) &= \mathrm{V}(\bar{X}_A)+\mathrm{V}(\bar{X}_B). \end{aligned} \tag{3-55}$$

The latter is what the confidence interval for the *mean difference* $\mu_A - \mu_B$ is using and what should be used for the proper statistical comparison of the means. The former is what you implicitly use in the ”overlapping CI interpretation approach”.

The proper standard deviation (sampling error) of the *sample mean difference* due to Pythagoras, is smaller than the sum of the two standard errors: assume that the two standard errors are 3 and 4. The sum is 7, but the square-root of the squares is $\sqrt{3^2+4^2}=5$. Or more generally

$$\sigma_{(\bar{X}_A-\bar{X}_B)} < \sigma_{\bar{X}_A}+\sigma_{\bar{X}_B}. \tag{3-56}$$

So we can say the following:

Remark 3.59

When interpreting two (and multi-) independent samples mean bar plots with added confidence intervals:

When two CIs do NOT overlap: The two groups are significantly different

When two CIs DO overlap: We do not know from this what the conclusion is (but then we can use the presented two-sample test method)

One can consider other types of plots for visualizing (multi)group differences. We will return to this in Chapter 8 on the multi-group data analysis, the so-called Analysis of Variance (ANOVA).

3.2.3 The paired design and analysis

Example 3.60 Sleeping medicine

In a study the aim is to compare two kinds of sleeping medicine A and B. 10 test persons tried both kinds of medicine and the following results are obtained, given in prolonged sleep length (in hours) for each medicine type:

Person	A	B	$D = B - A$
1	+0.7	+1.9	+1.2
2	-1.6	+0.8	+2.4
3	-0.2	+1.1	+1.3
4	-1.2	+0.1	+1.3
5	-1.0	-0.1	+0.9
6	+3.4	+4.4	+1.0
7	+3.7	+5.5	+1.8
8	+0.8	+1.6	+0.8
9	0.0	+4.6	+4.6
10	+2.0	+3.4	+1.4

Note that this is the same experiment as already treated in Example 3.21. We now in addition see the original measurements for each sleeping medicine rather than just individual differences given earlier. And we saw that we could obtain the relevant analysis (p-value and confidence interval) by a simple call to the `ttest_1samp` function using the 10 differences:

```
# Read the samples
x1 = np.array([0.7, -1.6, -0.2, -1.2, -1.0, 3.4, 3.7, 0.8, 0.0, 2.0])
x2 = np.array([1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 5.5, 1.6, 4.6, 3.4])

# Take the differences
dif = x2 - x1

# t-test on the differences
test = stats.ttest_1samp(dif,popmean=0)
print(test.statistic,test.pvalue,test.df)

4.671645978656774 0.0011658764685528319 9


stats.ttest_1samp(dif,popmean=0).confidence_interval(0.95)

ConfidenceInterval(low=np.float64(0.8613337442036719), high=np.float64(2.47866625579632
```

The example shows that this section actually could be avoided, as the right way to handle this so-called paired situation is to apply the one-sample theory and methods from Section 3.1 on the differences

$$d_i = x_i - y_i \quad \text{for } i = 1, 2, ..., n. \tag{3-57}$$

Then we can do all relevant statistics based on the mean $\bar{d}$ and the variance s_d^2 for these differences.

The reason for having an entire section devoted to *the paired t-test* is that it is an important topic for experimental work and statistical analysis. The paired design for experiments represents an important generic principle for doing experiments as opposed to the un-paired/independent samples design, and these important basic experimental principles will be important also for multi-group experiments and data, that we will encounter later in the material.

|||| Example 3.61 Sleeping medicine

And similarly in Python, they have prepared way to do the paired analysis directly on the two-sample data:

```
# Give both samples, but make paired t-test
test = stats.ttest_rel(x2,x1)
print(test.statistic,test.pvalue,test.df)

4.671645978656774 0.0011658764685528319 9


stats.ttest_rel(x2,x1).confidence_interval(0.95)

ConfidenceInterval(low=np.float64(0.8613337442036719), high=np.float64(2.47866625579632
```

Paired vs. completely randomized experiments

An experiment like the one exemplified here where two treatments are investigated can essentially be performed in two different ways:

Completely Randomized (independent samples) 20 patients are used and completely at random allocated to one of the two treatments (but usually making sure to have 10 patients in each group). So: different people in the different groups.

Paired (dependent samples) 10 patients are used, and each of them tests both of the treatments. Usually this will involve some time in between treatments to make sure that it becomes meaningful, and also one would typically make sure that some patients do A before B and others B before A. (and doing this allocation at random). So: the same people in the different groups.

Generally, one would expect that whatever the experiment is about and which observational units are involved (people, patients, animals) the outcome will be affected by the properties of each individual – the unit. In the example, some people will react positively to both treatments because they generally are more prone to react to sleeping medicines. Others will not respond as much to sleeping medicine. And these differences, the person-to-person variability, will give a high variance for the Welch independent samples t-test used for the independent samples case. So generally, one would often prefer to carry out a paired experiment, where the generic individual variability will not blur the signal – one can say that in a paired experiment, each individual serves as his/her own control – the effect of the two treatments are estimated for each individual. We illustrate this by analysing the example data wrongly, as if they were the results of a completely randomized experiment on 20 patients:

Example 3.62 Sleeping medicine - WRONG analysis

What happens when applying the wrong analysis:

```
# WRONG analysis
test = stats.ttest_ind(x2,x1,equal_var=False)
print(test.statistic,test.pvalue,test.df)

1.9334408348617207 0.06915652250932773 17.900065494971773
```

Note how the p-value here is around 0.07 as opposed to the 0.001 from the proper paired analysis. Also the confidence interval is much wider. Had we done the experiment with 20 patients and gotten the results here, then we would not be able to detect the difference between the two medicines. What happened is that the individual variabilities seen in each of the two groups now, incorrectly so, is being used for the statistical analysis and these are much larger than the variability of the differences:

```
# The sample variances of each sample and of the differences
print(x1.var(ddof=1))

3.4515555555555557

print(x2.var(ddof=1))

4.009

print((x2-x1).var(ddof=1))

1.2778888888888886
```

3.2.4 Validation of assumptions with normality investigations

For normality investigations in two-sample settings we use the tools given for one-sample data, presented in Section 3.1.8. For the paired setting, the investigation would be carried out for the differences. For the independent case the investigation is carried out within each of the two groups.

3.3 Planning a study: wanted precision and power

Experiments and observational studies are always better when they are carefully planned. Good planning covers many features of the study. The observations must be sampled appropriately from the population, reliable measurements must be made and the study must be "big enough" to be able to detect an effect of interest. And if the study becomes too big, effects of little practical interest may become statistically significant, and (some of) the money invested in the study will be wasted. Sample size is important for economic reasons: an oversized study uses more resources than necessary, this could be both financial but also ethical if subjecting objects to potentially harmful treatments, an undersized study can be wasted if it is not able to produce reliable results.

Sample size is very important to consider before a study is carried out.

3.3.1 Sample Size for wanted precision

One way of calculating the required sample size is to work back from the wanted precision. From (3-10) we see that the confidence interval is symmetric around $\bar{x}$ and the half width of the confidence interval (also called the margin of error (ME)) is given as

$$ME = t_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}. \tag{3-58}$$

Here $t_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile from the t-distribution with $n-1$ degrees of freedom. This quantile depends on both α and the sample size n, which is what we want to find.

The sample size now affects both n and $t_{1-\alpha/2}$, but if we have a large sample (e.g. $n \geq 30$) then we can use the normal approximation and replace $t_{1-\alpha/2}$ by the quantile from the normal distribution $z_{1-\alpha/2}$.

In the expression for ME in Equation (3-58) we also need σ, the standard deviation. An estimate of the standard deviation would usually only be available after the sample has been taken. Instead we use a guess for σ possibly based on a pilot study or from the literature, or we could use a scenario based choice (i.e. set σ to some value which we think is reasonable).

For a given choice of ME it is now possible to isolate n in Equation (3-58) (with the normal quantile inserted instead of the t-quantile):

|||| Method 3.63 The one-sample CI sample size formula

When σ is known or guessed at some value, we can calculate the sample size n needed to achieve a given margin of error, ME, with probability $1-\alpha$ as

$$n = \left(\frac{z_{1-\alpha/2} \cdot \sigma}{ME}\right)^2. \tag{3-59}$$

|||| Example 3.64 Student heights

In Example 3.1 we inferred using a sample of heights of 10 students and found the sample mean height to be $\bar{x} = 178$ and standard deviation $s = 12.21$. We can now calculate how many students we should include in a new study, if we want a margin of error of 3 cm with confidence 95%. Using the standard deviation from the pilot study with 10 students as our guess we can plug into Method 3.63

$$n = \left(\frac{1.96 \cdot 12.21}{3}\right)^2 = 63.64.$$

These calculations show that we should include 64 students, the nearest integer to 63.64.

The formula and approach here has the weakness that it only gives an "expected" behaviour of a coming experiment - at first reading this may seem good enough, but if you think about it, it means that approximately half of the times the actual width will be smaller and the other half, it will be larger than expected. If the uncertainty variability is not too large it might not be a big problem, but nothing in the approach helps us to know whether it is good enough – we cannot guarantee a minimum accuracy with a certain probability. A more advanced approach, that will help us control more precisely that a future experiment/study will meet our needs, is presented now.

3.3.2 Sample size and statistical power

Another way of calculating the necessary sample size is to use the power of the study. The *statistical power of a study is the probability of correctly rejecting H_0 if H_0 is false*. The relations between Type I error, Type II error and the power are seen in the table below.

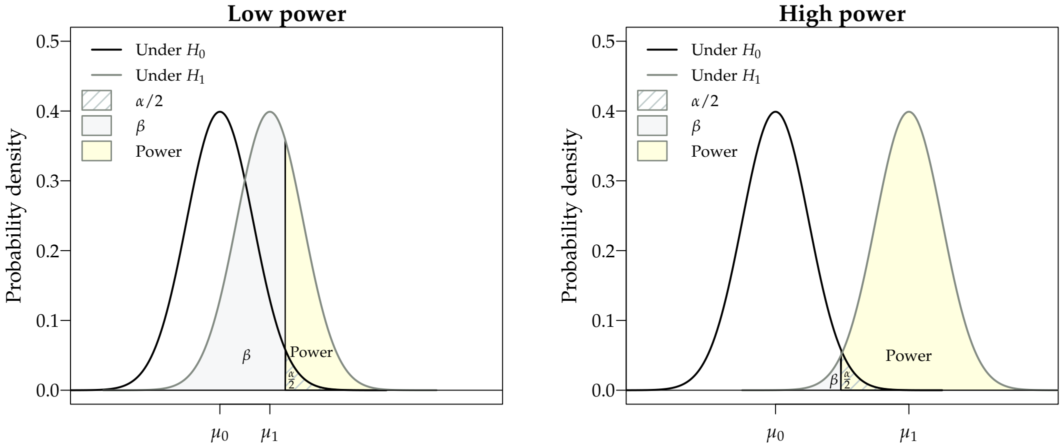


Figure 3.2: The mean μ_0 is the mean under H_0 and μ_1 the mean under H_1. When μ_1 increases (i.e. moving away from μ_0) so does the power (the yellow area on the graph).

	Reject H_0	Fail to reject H_0
H_0 is true	Type I error (α)	Correct acceptance of H_0
H_0 is false	Correct rejection of H_0 (Power)	Type II error (β)

The power has to do with the Type II error β, the probability of wrongly accepting H_0, when H_0 actually is false. We would like to have high power (low β), but it is clear that this will be impossible for all possible situations: it will depend on the scenario for the potential mean – small potential effects will be difficult to detect (low power), whereas large potential effects will be easier to detect (higher power), as illustrated in Figure 3.2. In the left plot we have the mean under H_0 (μ_0) close to the mean under the alternative hypothesis (μ_1) making it difficult to distinguish between the two and the power becomes low. In the right plot μ_0 and μ_1 are further apart and the statistical power is much higher.

The power approach to calculating the sample size first of all involves specifying the null hypothesis H_0. Then the following four elements must be specified/chosen:

- The significance level α of the test (in Python: `alpha`)
- A difference in the mean that you would want to detect, delta
- The standard deviation σ (`sd` in the code)
- The wanted power $(1 - \beta)$ (in Python: `power`)

When these values have been decided, it is possible to calculate the necessary sample size, n. In the one-sided, one-sample t-test there is an approximate closed form for n and this is also the case in some other simple situations.

Python offers easy to use functions for this not based on the approximate normal distribution assumption, but using the more proper t-distributions. In more complicated settings even it is possible to do some simulations to find the required sample size.

|||| Method 3.65 The one-sample sample size formula

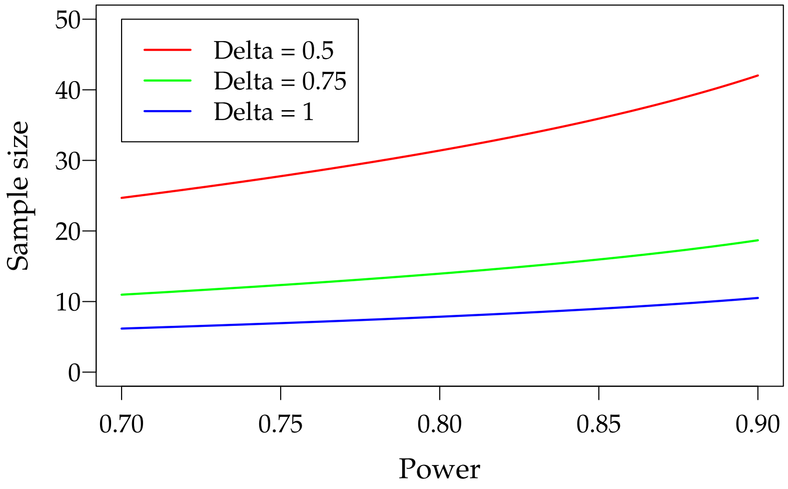
For the one-sample t-test for given α, β and σ

$$n = \left(\sigma \frac{z_{1-\beta} + z_{1-\alpha/2}}{(\mu_0 - \mu_1)}\right)^2,$$

where $\mu_0 - \mu_1$ is the difference in means that we would want to detect and $z_{1-\beta}$, $z_{1-\alpha/2}$ are quantiles of the standard normal distribution.

|||| Example 3.66 Sample size as function of power

The following figure shows how the sample size increases with increasing power using the formula in 3.65. Here we have chosen $\sigma = 1$ and $\alpha = 0.05$. Delta is $\mu_0 - \mu_1$.



Example 3.67 Student heights

If we return to the example with student heights 3.1, we might want to collect data for a new study to test the hypothesis about the mean height

$$H_0 : \mu = 180$$

Against the alternative

$$H_1 : \mu \neq 180$$

This is the first step in the power approach. The following four elements then are:

- Set the significance level α equal to 5%
- Specify that we want to be able to detect a difference of 4 cm
- We will use the standard deviation 12.21 from the study with 10 subjects as our guess for σ
- We want a power of 80%

Using the formula in 3.65 we get

$$n = \left(12.21 \cdot \frac{0.84 + 1.96}{4}\right)^2 = 73.05.$$

So we would need to include 74 students.
We could also use a Python-function for power and sample size based on the t-distributions:

```
# The sample size for power=0.80
delta = 4
sd = 12.21
alpha = 0.05
power = 0.8
smp.TTestPower().solve_power(effect_size=delta/sd, alpha=alpha,
power=power)

75.07715049712685
```

From the calculations in Python avoiding the normal approximation the required sample size is 76 students, very close to the number calculated by hand using the approximation above.

In fact the Python-function is really nice in the way that it could also be used to find the power for a given sample size, e.g. $n = 50$ (given all the other aspects):

```
delta = 4
sd = 12.21
nobs = 50
alpha = 0.05
smp.TTestPower().solve_power(effect_size=delta/sd, nobs=nobs,
alpha=alpha)

np.float64(0.6220915188555853)
```

This would only give the power 0.62 usually considered too low for a relevant effect size.

And finally the Python-function can tell us what effect size and delta that could be detected by, say, $n = 50$, and a power of 0.80:

```
nobs = 50
alpha = 0.05
power = 0.80
sd = 12.21
effect = smp.TTestPower().solve_power(nobs=nobs, alpha=alpha,
power=power)
delta = effect*sd
print(delta)

4.935074496518317
```

So with $n = 50$ only a delta as big as 4.9 would be detectable with probability 0.80.

To summarize: if we know/define 4 out the 5 values: significance level, power $(1 - \beta)$, n, *delta*, and σ, we can find the 5'th. In the Python-function, the arguments are called `alpha`, `power`, `nobs`, and `effect_size`, where `effect_size` is delta/σ.

In the practical planning of a study, often a number of scenario-based values of *delta* and σ are used to find a reasonable size of the study.

3.3.3 Power/Sample size in two-sample setup

For power and sample size one can generalize the tools presented for the one-sample setup in the previous section. We illustrate it here by an example of how to work with the inbuilt Python-function:

|||| Example 3.68 Two-sample power and sample size computations in Python

We consider the two-sample hypothesis test

$$H_0 : \mu_1 = \mu_2,$$
$$H_1 : \mu_1 \neq \mu_2$$

```
# Finding the power of detecting a group difference of 2
# with sigma=1 for n=10
delta = 2
sd = 1
nobs = 10
alpha = 0.05
smp.TTestIndPower().solve_power(effect_size=delta/sd, nobs1=nobs,
alpha=alpha, ratio=1.0)

np.float64(0.9881789691948746)
```

```
# Finding the sample size for detecting a group difference of 2
# with sigma=1 and power=0.9
delta = 2
sd = 1
alpha = 0.05
power = 0.90
smp.TTestIndPower().solve_power(effect_size=delta/sd, alpha=alpha,
power=power, ratio=1.0)

6.386755384175011
```

```
# Finding the detectable effect size (delta)
# with sigma=1, n=10 and power=0.9
nobs = 10
alpha = 0.05
power = 0.90
sd = 1
effect = smp.TTestIndPower().solve_power(nobs1=nobs, alpha=alpha,
power=power, ratio=1.0)
delta = effect*sd
print(delta)

1.5336931237722076
```

Note that we now use the function `TTestIndPower`, which uses the arguments `nobs1` and `ratio` to specify the number of observations in the two samples.

Chapter 4

Simulation Based Statistics

4.1 Probability and Simulation

4.1.1 Introduction

One of the really big gains for statistics and modelling of random phenomena, provided by computer technology during the last decades, is the ability to simulate random systems on the computer, as we have already seen much in use in Chapter 2. This provides possibilities to obtain results that otherwise from a mathematical analytical point of view would be impossible to calculate. And, even in cases where the highly educated mathematician/physicist might be able to find solutions, simulation is a general and simple calculation tool allowing solving complex problems without a need for deep theoretical insight.

An important reason for including this subject in an introductory statistics course, apart from using it as a pedagogical tool to aide the understanding of random phenomena, is the fact that the methods we are usually introducing in basic statistics are characterized by relying on one of two conditions:

1. The original data population density is assumed to be a normal distribution
2. Or: The sample size n is large enough to make this assumption irrelevant for what we do

And in real settings it may be challenging to know for sure whether any of these two are really satisfied, so to what extend can we trust the statistical conclusions that we make using our basic tools, as e.g. the one- and two-sample statistical methods presented in Chapter 3. And how should we do the basic statistical analysis if we even become convinced that none of these two conditions are fulfilled? Statistical data analysis based on simulation tools is a valuable tool to complete the tool box of introductory statistics. It can be used to do statistical computing for other features than just means, and for other population distributions than the normal. It can also be used to investigate whether some of our assumptions appear reasonable. We already saw an example of this in relation to the qq-plots in Chapter 3.1.9.

In fact, it will become clear that the simulation tools presented here will make us rapidly able to perform statistical analysis that goes way beyond what historically has been introduced in basic statistics classes or textbooks. Unfortunately, the complexity of real life engineering applications and data analysis challenges can easily go beyond the settings that we have time to cover within an introductory exposition. With the general simulation tool in our tool box, we have a multi-tool that can be used for (and adapted to) basically almost any level of complexity that we will meet in our future engineering activity.

The classical statistical practice would be to try to ensure that the data we're analyzing behaves like a normal distribution: symmetric and bell-shaped histogram. In Chapter 3 we also learned that we can make a normal q-q plot to verify this assumption in practice, and possibly transform the data to get them closer to being normal. The problem with small samples is that it even with these diagnostic tools can be difficult to know whether the underlying distribution really is ″normal″ or not.

And in some cases the assumption of normality after all simply may be obviuosly wrong. For example, when the response scale we work with is far from being quantitative and continuous - it could be a scale like ″small″, ″medium″ and ″large″ - coded as 1, 2 and 3. We need tools that can do statistical analysis for us WITHOUT the assumption that the normal distribution is the right model for the data we observe and work with.

Traditionally, the missing link would be covered by the so-called non-parametric tests. In short this is a collection of methods that make use of data at a more coarse level, typically by focusing on the rank of the observations instead of the actual values of the observations. So in a paired t-test setup, for example, one would just count how many times the observations in one sample is bigger than in the other – instead of calculating the differences. In that way you can make statistical tests without using the assumption of an underlying normal distribution. There are a large number of such non-parametric tests for different setups. Historically, before the computer age, it was the only way to really handle such situations in practice. These tests are all characterized by the fact that they are given by relatively simple computational formulas which in earlier times easily could be handled. For small sample statistics with questionable distributional settings, these tools maintain to offer a robust set of basic statistical procedures.

The simulation based methods that we now present instead have a couple of crucial advantages to the traditional non-parametric methods:

- Confidence intervals are much easier to achieve
- They are much easier to apply in more complex situations
- They scale better to modern time big data analysis

4.1.2 Simulation as a general computational tool

Basically, the strength of the simulation tool is that one can compute arbitrary functions of random variables and their outcomes. In other words one can find probabilities of complicated outcomes. As such, simulation is really not a statistical tool, but rather a probability calculus tool. However, since statistics essentially is about analysing and learning from real data in the light of certain probabilities, the simulation tool indeed becomes of statistical importance, which we will exemplify very specifically below. Before starting with exemplifying the power of simulation as a general computational tool, we refer to the introduction to simulation in Chapter 2 – in particular read first Section 2.6, Example **??** and thereafter Section 2.6.

|||| Example 4.1 Rectangular plates

A company produces rectangular plates. The length of plates (in meters), X is assumed to follow a normal distribution $N(2, 0.01^2)$ and the width of the plates (in meters), Y are assumed to follow a normal distribution $N(3, 0.02^2)$. We're hence dealing with plates of size 2×3 meters, but with errors in both length and width. Assume that these errors are completely independent. We are interested in the area of the plates which of course is given by $A = XY$. This is a non-linear function of X and Y, and actually it means that we, with the theoretical tools we presented so far in the material, cannot figure out what the mean area really is, and not at all what the standard deviation would be in the areas from plate to plate, and we would definitely not know how to calculate the probabilities of various possible outcomes. For example, how often such plates have an area that differ by more than 0.1 m^2 from the targeted 6 m^2? One statement summarizing all our lack of knowledge at this point: we do not know the probability distribution of the random variable A and we do not know how to find it! With simulation, it is straightforward: one can find all relevant information about A by just simulating the X and Y a high number of times, and from this compute A just as many times, and then observe what happens to the values of A. The first step is then given by:

```
# Number of simulations
k = 10000

# Simulate X and Y, then A
X = stats.norm.rvs(loc=2,scale=0.01,size=k)
Y = stats.norm.rvs(loc=3,scale=0.02,size=k)
A = X * Y
```

The Python object `A` now contains 10.000 observations of A. The expected value and the standard deviation for A are simply found by calculating the average and standard deviation for the simulated A-values:

```
# The mean and std. deviation of the simulated values
print(A.mean())

6.000707518857636


print(A.std(ddof=1))

0.050187379229233574
```

and the desired probability, $P(|A - 6| > 0.1) = 1 - P(5.9 \leq A \leq 6.1)$ is found by counting how often the incident actually occurs among the k outcomes of A:

```
1*(abs(A-6) > 0.1).mean()

np.float64(0.0454)
```

The code `abs(A-6) > 0.1` creates an array with values `TRUE` or `FALSE` depending on whether the absolute value of $A - 6$ is greater than 0.1 or not. When you multiply by 1 the `TRUE` is automatically translated into 1 and `FALSE` automatically translated to 0. To find the probability, we sum these binary values and divide by number of simulations k. This is equivalent to finding the mean of the binary values, and therefore we use the mean method.

Note, that if you do this yourself without using the same seed value you will not get exactly the same result. It is clear that this simulation uncertainty is something we must deal with in practice. The size of this will depend on the situation and on the number of simulations k. We can always get a first idea of it in a specific situation simply by repeating the calculation a few times and note how it varies. Indeed, one could then formalize such an investigation and repeat the simulation many times, to get an evaluation of the simulation uncertainty. We will not pursue this further here. When the target of the computation is in fact a probability, as in the latter example here, you can alternatively use standard binomial statistics, which is covered in Chapter 2 and Chapter 7. For example, with $k = 100000$ the uncertainty for a calculated proportion of around 0.044 is given by: $\sqrt{\frac{0.044(1-0.044)}{100000}} = 0.00065$. Or for example, with $k = 10000000$ the uncertainty is 0.000065. The result using such a k was 0.0455 and because we're a bit unlucky with the rounding position we can in practice say that the exact result rounded to 3 decimal places are either 0.045 or 0.046. In this way, a calculation which is actually based on simulation is turned into an exact one in the sense that rounded to 2 decimal places, the result is simply 0.05.

4.1.3 Propagation of error

Within chemistry and physics one may speak of measurement errors and how measurement errors propagate/accumulate if we have more measurements and/or use these measurements in subsequent formulas/calculations. First of all: The basic way to "measure an error", that is, to quantify a measurement error is by means of a standard deviation. As we know, the standard deviation expresses the average deviation from the mean. It is clear it may happen that a measuring instrument also on average measures wrongly (off the target). This is called "bias", but in the basic setting here, we assume that the instrument has no bias.

Hence, reformulated, an error propagation problem is a question about how the standard deviation of some function of the measurements depends on the standard deviations for the individual measurement: let $X_1, \ldots, X_n$ be n measurements with standard deviations (average measurement errors) $\sigma_1, \ldots, \sigma_n$. As usual in this material, we assume that these measurement errors are independent of each other. There are extensions of the formulas that can handle dependencies, but we omit those here. We must then in a general formulation be able to find

$$\sigma^2_{f(X_1,\ldots,X_n)} = \mathrm{V}(f(X_1,\ldots,X_n)). \tag{4-1}$$

|||| Remark 4.2

[For the thoughtful reader: Measurement errors, errors and variances] Although we motivate this entire treatment by the *measurement error* terminology, often used in chemistry and physics, actually everything is valid for *any kind of* errors, be it "time-to-time" production errors, or "substance-to-substance" or "tube-to-tube" errors. What the relevant kind of errors/variabilities are depends on the situation and may very well be mixed together in applications. But, the point is that as long as we have a relevant error variance, we can work with the concepts and tools here. It does not have to have a "pure measurement error" interpretation.

Actually, we have already in this course seen the linear error propagation rule, in Theorem in 2.56, which then can be restated here as

$$\text{If } f(X_1,\ldots,X_n)=\sum_{i=1}^{n} a_i X_i, \quad \text{then} \quad \sigma^2_{f(X_1,\ldots,X_n)}=\sum_{i=1}^{n} a_i^2\sigma_i^2.$$

There is a more general non-linear extension of this, albeit theoretically only an approximate result, which involves the partial derivative of the function f with respect to the n variables:

|||| Method 4.3 The non-linear approximative error propagation rule

If $X_1,\ldots,X_n$ are independent random variables with variances $\sigma_1^2,\ldots,\sigma_n^2$ and f is a (potentially non-linear) function of n variables, then the variance of the f-transformed variables can be approximated linearly by

$$\sigma^2_{f(X_1,\ldots,X_n)}=\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2\sigma_i^2, \tag{4-2}$$

where $\frac{\partial f}{\partial x_i}$ is the partial derivative of f with respect to the i'th variable

In practice one would have to insert the actual measurement values $x_1,\ldots,x_n$ of $X_1,\ldots,X_n$ in the partial derivatives to apply the formula in practice, see the example below. This is a pretty powerful tool for the general finding of (approximate) uncertainties for complicated functions of many measurements or for that matter: complex combinations of various statistical quantities. When the formula is used for the latter, it is also in some contexts called the "delta rule" (which is mathematically speaking a so-called first-order (linear) Taylor approximations to the non-linear function f). We bring it forward here, because as an alternative to this approximate formula one could use simulation in the following way:

|||| Method 4.4 Non-linear error propagation by simulation

Assume we have actual measurements $x_1, \ldots, x_n$ with known/assumed error variances $\sigma_1^2, \ldots, \sigma_n^2$:

1. Simulate k outcomes of all n measurements from assumed error distributions, e.g. $N(x_i, \sigma_i^2)$: $X_i^{(j)}, j = 1 \ldots, k$.
2. Calculate the standard deviation directly as the observed standard deviation of the k simulated values of f:

$$s_{f(X_1,\ldots,X_n)}^{\text{sim}} = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}(f_j - \bar{f})^2}, \tag{4-3}$$

where

$$f_j = f(X_1^{(j)}, \ldots, X_n^{(j)}). \tag{4-4}$$

|||| Example 4.5

Let us continue the example with $A = XY$ and X and Y defined as in the example above. First of all note, that we already above used the simulation based error propagation method, when we found the standard deviation to be 0.04957 based on the simulation. To exemplify the approximate error propagation rule, we must find the derivatives of the function $f(x, y) = xy$ with respect to both x and y

$$\frac{\partial f}{\partial x} = y \quad \frac{\partial f}{\partial y} = x.$$

Assume, that we now have two specific measurements of X and Y, for example $x = 2.00$ m and $y = 3.00$ m the error propagation law would provide the following approximate calculation of the "uncertainty error variance of the area result" 2.00 m $\cdot$ 3.00 m $= 6.00$ m^2, namely

$$\sigma_A^2 = y^2 \cdot 0.01^2 + x^2 \cdot 0.02^2 = 3.00^2 \cdot 0.01^2 + 2.00^2 \cdot 0.02^2 = 0.0025.$$

So, with the error propagation law we are managing a part of the challenge without simulating. Actually, we are pretty close to be able to find the correct theortical variance of $A = XY$ using tools provided in this course. By the definition and the following fundamental relationship

$$\mathrm{V}(X) = \mathrm{E}(X - \mathrm{E}(X))^2 = \mathrm{E}(X^2) - \mathrm{E}(X)^2. \tag{4-5}$$

So, one can actually deduce the variance of A theoretically, it is only necessary to know in addition that for independent random variables: $\mathrm{E}(XY) = \mathrm{E}(X)\,\mathrm{E}(Y)$ (which by the way then also tells us that $\mathrm{E}(A) = \mathrm{E}(X)\,\mathrm{E}(Y) = 6$)

$$\begin{aligned}
\mathrm{V}(XY) &= \mathrm{E}\left[(XY)^2\right] - \mathrm{E}(XY)^2\\
&= \mathrm{E}(X^2)\,\mathrm{E}(Y^2) - \mathrm{E}(X)^2\,\mathrm{E}(Y)^2\\
&= \left[\mathrm{V}(X) + \mathrm{E}(X)^2\right]\left[\mathrm{V}(Y) + \mathrm{E}(Y)^2\right] - \mathrm{E}(X)^2\,\mathrm{E}(Y)^2\\
&= \mathrm{V}(X)\,\mathrm{V}(Y) + \mathrm{V}(X)\,\mathrm{E}(Y)^2 + \mathrm{V}(Y)\,\mathrm{E}(X)^2\\
&= 0.01^2 \cdot 0.02^2 + 0.01^2 \cdot 3^2 + 0.02^2 \cdot 2^2\\
&= 0.00000004 + 0.0009 + 0.0016\\
&= 0.00250004.
\end{aligned}$$

Note, how the approximate error propagation rule actually corresponds to the two latter terms in the correct variance, while the first term – the product of the two variances is ignored. Fortunately, this term is the smallest of the three in this case. It does not always have to be like that. If you want to learn how to make a theoretical derivation of the density function for $A = XY$ then take a course in probability calculation.

Note, how we in the example actually found the "average error", that is, the error standard deviation by three different approaches:

1. The simulation based approach
2. The analytical, but approximate, error propagation method
3. A theoretical derivation

The simulation approach has a number of crucial advantages:

1. It offers a simple way to compute many other quantities than just the standard deviation (the theoretical derivations of such other quantities could be much more complicated than what was shown for the variance here)
2. It offers a simple way to use any other distribution than the normal – if we believe such better reflect reality
3. It does not rely on any linear approximations of the true non-linear relations

4.2 The parametric bootstrap

4.2.1 Introduction

Generally, a confidence interval for an unknown parameter μ is a way to express uncertainty using the sampling distribution of $\hat{\mu} = \bar{x}$. Hence, we use a distribution that expresses how our calculated value would vary from sample to sample. And the sampling distribution is a theoretical consequence of the original population distribution. As indicated, we have so far no method to do this if we only have a small sample size ($n < 30$), and the data cannot be assumed to follow a normal distribution. In principle there are two approaches for solving this problem:

1. Find/identify/assume a different and more suitable distribution for the population ("the system")
2. Do not assume any distribution whatsoever

The simulation method called bootstrapping, which in practice is to simulate many samples, exists in two versions that can handle either of these two challenges:

1. Parametric bootstrap: simulate multiple samples from the assumed distribution.
2. Non-parametric bootstrap: simulate multiple samples directly from the data.

Actually, the parametric bootstrap handles in addition the situation where data could perhaps be normally distributed, but where the calculation of interest is quite different than the average, for example, the coefficient of variation (standard deviation divided by average) or the median. This would be an example of a non-linear function of data – thus not having a normal distribution nor a t-distribution as a sampling distribution. So, the parametric bootstrap is basically just an example of the use of simulation as a general calculation tool, as introduced above. Both methods are hence very general and can be used in virtually all contexts.

In this material we have met a few of such alternative continuous distributions, e.g. the log-normal, uniform and exponential distributions. But if we think about it, we have not (yet) been taught how to do any statistics (confidence intervals and/or hypothesis testing) within an assumption of any of these. The parametric bootstrap is a way to do this without relying on theoretical derivations of everything. As for the theoretical variance deduction above, there are indeed methods for doing such general theoretical derivations, which would make us able to do statistics based on any kind of assumed distribution. The most welknown, and in many ways also optimal, overall approach for this is called *maximum likelihood* theory. The general theory and approach of maximum likelihood is not covered in this course, however it is good to know that, in fact, all the methods we present are indeed also maximum likelihood methods assuming normal distributions for the population(s).

4.2.2 One-sample confidence interval for μ

|||| Example 4.6 Confidence interval for the exponential rate or mean

Assume that we observed the following 10 call waiting times (in seconds) in a call center

$$32.6,\ 1.6,\ 42.1,\ 29.2,\ 53.4,\ 79.3,\ 2.3,\ 4.7,\ 13.6,\ 2.0.$$

If we model the waiting times using the exponential distribution, we can estimate the mean as

$$\hat{\mu} = \bar{x} = 26.08,$$

and hence the rate parameter $\lambda = 1/\beta$ in the exponential distribution as (cf. 2.48)

$$\hat{\lambda} = 1/26.08 = 0.03834356.$$

However, what if we want a 95% confidence interval for either $\mu = \beta$ or λ? We have not been tought the methods, that is, given any formulas for finding this. The following few lines of Python-code, a version of the simulation based error propagation approach from above, will do the job for us:

```
# Read the data
x = np.array([32.6, 1.6, 42.1, 29.2, 53.4, 79.3, 2.3 , 4.7, 13.6, 2.0])
n = len(x)
rate = 1/x.mean()

# Set the number of simulations
k = 100000

# 1. Simulate k samples each with n=10 observations from  an
#    exponential distribution with the estimated rate
simsamples = stats.expon.rvs(scale=1/rate, size=(k,n))

# 2. Compute the mean in each of the k samples
simsamples = pd.DataFrame(simsamples)
simmeans = simsamples.mean(axis=1)

# 3. Find the two relevant quantiles of the k generated means
print(np.quantile(simmeans,[0.025,0.975],
method='averaged_inverted_cdf'))
```

```
[12.575 44.563]
```

Explanation: we use `stats.expon.rvs` to generate 100.000 bootstrap samples each with 10 observations from an exponential distribution with the estimated mean value, and the results are collected in a 10×100.000 matrix. Then in a single call the 100.000 averages are calculated and subsequently the relevant quantiles found.

So the 95%-confidence interval for the mean μ is (in seconds)

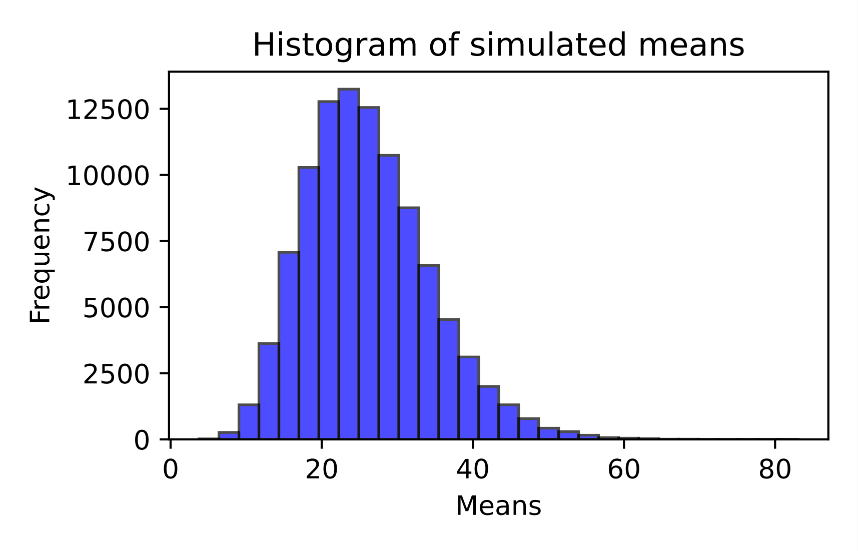
$$[12.6,\ 44.6].$$

And for the rate $\lambda = 1/\mu$ it can be found by a direct transformation (remember that the quantiles are 'invariant' to monotonic transformations, c.f. Chapter 3)

$$[1/44.6,\ 1/12.6] \Leftrightarrow [0.022,\ 0.0794].$$

The simulated sampling distribution of means that we use for our statistical analysis can be seen with the histogram:

```
# Histogram of the simulated means
plt.hist(simmeans, bins=30, edgecolor='black', color='blue', alpha=0.7)
plt.xlabel('Means')
plt.ylabel('Frequency')
plt.title('Histogram of simulated means')
plt.tight_layout()
plt.show()
```



We see clearly that the sampling distribution in this case is not a normal nor a t-distribution: it has a clear right skewed shape. So $n = 10$ is not quite large enough for this exponential distribution to make the Central Limit Theorem take over.

The general method which we have used in the example above is given below as Method 4.7.

4.2.3 One-sample confidence interval for any feature assuming any distribution

We saw in the example above that we could easily find a confidence interval for the rate $\lambda = 1/\mu$ assuming an exponential distribution. This was so, since the rate was a simple (monotonic) transformation of the mean, and the quantiles of simulated rates would then be the same simple transformation of the quantiles of the simulated means. However, what if we are interested in something not expressed as a simple function of the mean, for instance the median, the coefficient of variation, the quartiles, Q_1 or Q_3, the IQR=$Q_3 - Q_1$ or any other quantile? Well, a very small adaptation of the method above would make that possible for us. To express that we now cover any kind of statistic one could think of, we use the general notation, the Greek letter θ, for a general feature of the distribution. For instance, θ could be the true median of the population distribution, and then $\hat{\theta}$ is the sample median computed from the sample taken.

|||| Method 4.7 Confidence interval for any feature θ by parametric bootstrap

Assume we have actual observations $x_1, \ldots, x_n$ and assume that they stem from some probability distribution with density (pdf) f:

1. Simulate k samples of n observations from the assumed distribution f where the mean is set to $\bar{x}$ [a]
2. Calculate the statistic $\hat{\theta}$ in each of the k samples $\hat{\theta}_1^*, \ldots, \hat{\theta}_k^*$
3. Find the $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ quantiles for these, $q^*_{100(\alpha/2)\%}$ and $q^*_{100(1-\alpha/2)\%}$ as the $100(1-\alpha)\%$ confidence interval: $\left[q^*_{100(\alpha/2)\%},\ q^*_{100(1-\alpha/2)\%}\right]$

[a] *(Footnote: And otherwise chosen to match the data as good as possible: some distributions have more than just a single mean related parameter, e.g. the normal or the log-normal. For these one should use a distribution with a variance that matches the sample variance of the data. Even more generally the approach would be to match the chosen distribution to the data by the so-called maximum likelihood approach)*

Please note again, that you can simply substitute the θ with whatever statistics that you are working with. This then also shows that the method box includes the often occurring situation, where a confidence interval for the mean μ is the aim.

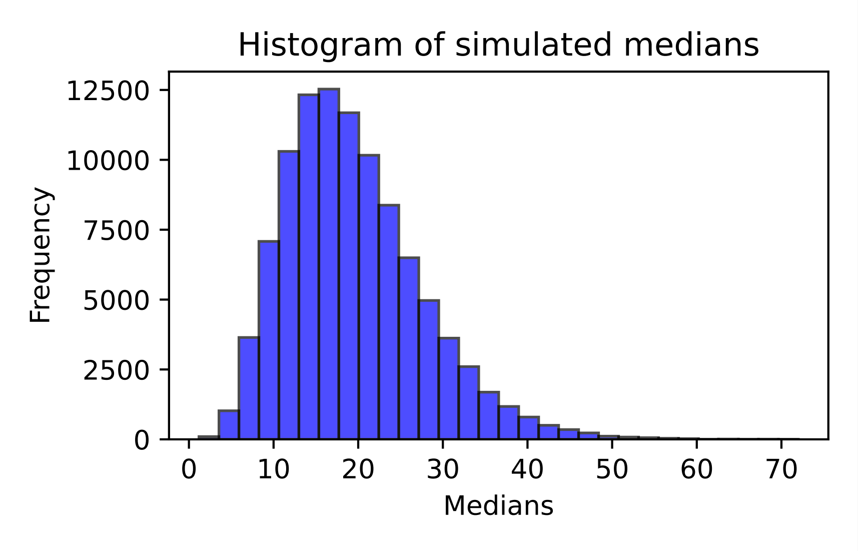
Example 4.8 Confidence interval for the median assuming an exponential distribution

Let us look at the exponential data from the previous section and find the confidence interval for the median:

```
# Read the data
x = np.array([32.6, 1.6, 42.1, 29.2, 53.4, 79.3, 2.3 , 4.7, 13.6, 2.0])
n = len(x)
rate = 1/x.mean()

# Set the number of simulations
k = 100000

# 1. Simulate k samples each with n=10 observations from  an
#    exponential distribution with the estimated rate
simsamples = stats.expon.rvs(scale=1/rate, size=(k,n))

# 2. Compute the median in each of the k samples
simsamples = pd.DataFrame(simsamples)
simmedians = simsamples.median(axis=1)

# 3. Find the two relevant quantiles of the k generated medians
print(np.quantile(simmedians,[0.025,0.975],
method=’averaged_inverted_cdf’))

[ 7.093 38.333]
```

The simulated sampling distribution of medians that we use for our statistical analysis can be studied by the histogram:

```
# Histogram of the simulated medians
plt.hist(simmedians, bins=30, edgecolor='black', color='blue',
alpha=0.7)
plt.xlabel('Medians')
plt.ylabel('Frequency')
plt.title('Histogram of simulated medians')
plt.tight_layout()
plt.show()
```



We see again clearly that the sampling distribution in this case is not a normal nor a t-distribution: it has a clear right skewed shape.

|||| Example 4.9 Confidence interval for Q_3 assuming a normal distribution

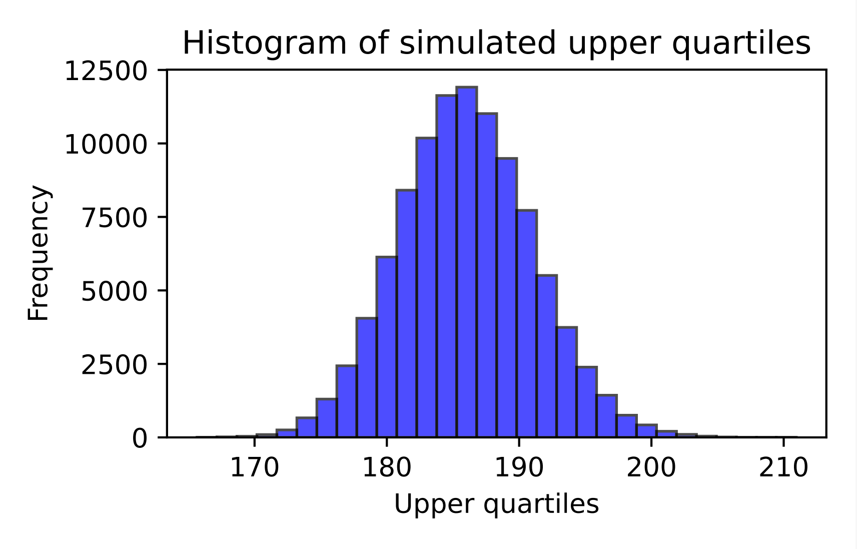
Let us look at the heights data from the previous chapters and find the 99% confidence interval for the upper quartile: (Please note that you will find NO theory nor analytically expressed method boxes in the material to solve this challenge). We proceed like in the previous example:

```
# Read the data
x = np.array([168, 161, 167, 179, 184, 166, 198, 187, 191, 179])
n = len(x)
mu = x.mean()
sd = x.std(ddof=1)

# Set the number of simulations
k = 100000

# 1. Simulate k samples each with n=10 observations from  a
#    normal distribution with the estimated parameters
simsamples = stats.norm.rvs(loc=mu, scale=sd, size=(k,n))

# 2. Compute the upper quartile in each of the k samples
simsamples = pd.DataFrame(simsamples)

# A version using the quantile-function from Pandas.
# Note: Pandas does not use our method of calculating quantiles.
simUQs = simsamples.quantile(0.75, axis=1)

# A version using the quantile-function from NumPy.
simUQs =  np.quantile(simsamples, 0.75, axis=1,
method="averaged_inverted_cdf")

# 3. Find the two relevant quantiles of the k generated medians
print(np.quantile(simUQs,[0.005,0.995],method=’averaged_inverted_cdf’))
```

```
[173.481 199.813]
```

The simulated sampling distribution of upper quartiles that we use for our statistical analysis can be studied by the histogram:

```
# Histogram of the simulated upper quartiles
plt.hist(simUQs, bins=30, edgecolor='black', color='blue', alpha=0.7)
plt.xlabel('Upper quartiles')
plt.ylabel('Frequency')
plt.title('Histogram of simulated upper quartiles')
plt.tight_layout()
plt.show()
```



In this case the Q_3 of $n = 10$ samples of a normal distribution appear to be rather symmetric and nicely distributed, so maybe one could in fact use the normal distribution, also as an approximate sampling distribution in this case.

4.2.4 Two-sample confidence intervals assuming any distributions

In this section we extend what we learned in the two previous sections to the case where the focus is a comparison between two (independent) samples. We present a method box which is the natural extensions of the method box from above, comparing any kind of feature (hence including the mean comparison):

|||| Method 4.10 Two-sample confidence interval for any feature comparison $\theta_1 - \theta_2$ by parametric bootstrap

Assume we have actual observations $x_1, \ldots, x_{n_1}$ and $y_1, \ldots, y_{n_2}$ and assume that they stem from some probability distributions with density f_1 and f_2:

1. Simulate k sets of 2 samples of n_1 and n_2 observations from the assumed distributions setting the means to $\hat{\mu}_1 = \bar{x}$ and $\hat{\mu}_2 = \bar{y}$, respectively [a]

2. Calculate the difference between the features in each of the k samples $\hat{\theta}^*_{x1} - \hat{\theta}^*_{y1}, \ldots, \hat{\theta}^*_{xk} - \hat{\theta}^*_{yk}$

3. Find the $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ quantiles for these, $q^*_{100(\alpha/2)\%}$ and $q^*_{100(1-\alpha/2)\%}$ as the $100(1-\alpha)\%$ confidence interval $\left[q^*_{100(\alpha/2)\%}, \, q^*_{100(1-\alpha/2)\%}\right]$

[a] *(Footnote: And otherwise chosen to match the data as good as possible: some distributions have more than just a single mean related parameter, e.g. the normal or the log-normal. For these one should use a distribution with a variance that matches the sample variance of the data. Even more generally the approach would be to match the chosen distribution to the data by the so-called maximum likelihood approach)*

|||| Example 4.11 CI for the difference of two means from exponential distributed data

Let us look at the exponential data from the previous section and compare that with a second sample of $n = 12$ observations from another day at the call center

$$9.6, \; 22.2, \; 52.5, \; 12.6, \; 33.0, \; 15.2, \; 76.6, \; 36.3, \; 110.2, \; 18.0, \; 62.4, \; 10.3.$$

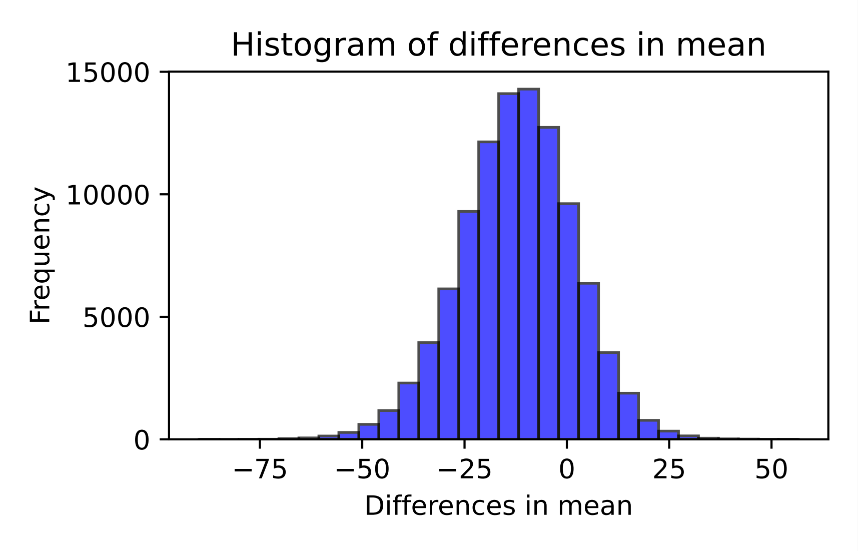
Let us quantify the difference between the two days and conclude whether the call rates and/or means are any different on the two days:

```
# Read the data
x = np.array([32.6, 1.6, 42.1, 29.2, 53.4, 79.3, 2.3 , 4.7, 13.6, 2.0])
y = np.array([9.6, 22.2, 52.5, 12.6, 33.0, 15.2, 76.6, 36.3, 110.2,
18.0, 62.4, 10.3])
n1 = len(x)
n2 = len(y)
rate1 = 1/x.mean()
rate2 = 1/y.mean()

# Set the number of simulations
k = 100000

# 1. Simulate k samples each with n1=10 observations from an
#    exponential distribution with the estimate rate of X
simXsamples = stats.expon.rvs(scale=1/rate1, size=(k,n1))
simXsamples = pd.DataFrame(simXsamples)

# 2. Simulate k samples each with n2=12 observations from  an
#    exponential distribution with the estimated rate of Y
simYsamples = stats.expon.rvs(scale=1/rate2, size=(k,n2))
simYsamples = pd.DataFrame(simYsamples)

# 3. Compute the difference between the two simulated means - k times
simDifmeans = simXsamples.mean(axis=1) - simYsamples.mean(axis=1)

# 4. Find the two relevant quantiles of the k generated differences in
mean
print(np.quantile(simDifmeans,[0.025,0.975],
method=’averaged_inverted_cdf’))

[-40.521  14.028]
```

Thus, although the mean waiting time was higher on the second day ($\bar{y} = 38.24$ s), the range of acceptable values (the confidence interval) for the difference in means is $[-40.5,\ 14.0]$ – a pretty large range and including 0, so we have no evidence of the claim that the two days had different mean waiting times (nor call rates then) based on the current data.

Let us, as in previous examples take a look at the distribution of the simulated samples. In a way, we do not really need this for doing the analysis, but just out of curiosity, and for the future it may give a idea of how far from normality the relevant sampling distribution really is:

```
# Histogram of the simulated differences
plt.hist(simDifmeans, bins=30, edgecolor='black', color='blue',
alpha=0.7)
plt.xlabel('Differences in mean')
plt.ylabel('Frequency')
plt.title('Histogram of differences in mean')
plt.tight_layout()
plt.show()
```



In this case the differences of means of exponential distributions appears to be rather symmetric and nicely distributed, so maybe one could in fact use the normal distribution, also as an approximate sampling distribution in this case.

|||| Example 4.12 Nutrition study: comparing medians assuming normal distributions

Let us compare the median energy levels from the two-sample nutrition data from Example 3.46. And let us do this still assuming the normal distribution as we also assumed in the previous example. First we read in the data:

```
# Read the data
xA = np.array([7.53, 7.48, 8.08, 8.09, 10.15, 8.4, 10.88, 6.13, 7.9])
xB = np.array([9.21, 11.51, 12.79, 11.85, 9.97, 8.79, 9.69, 9.68,
9.19])
nA = len(xA)
nB = len(xB)
```

Then we do the two-sample median comparison by the parametric, normal based, bootstrap:

```
# Set the number of simulations
k = 100000

# 1. Simulate k samples each with nA=9 observations from a
#    normal distribution with the estimate parameters for group A
simAsamples = stats.norm.rvs(loc=xA.mean(), scale=xA.std(ddof=1) ,
size=(k,nA))
simAsamples = pd.DataFrame(simAsamples)

# 2. Simulate k samples each with nB=9 observations from a
#    normal distribution with the estimate parameters for group B
simBsamples = stats.norm.rvs(loc=xB.mean(), scale=xB.std(ddof=1) ,
size=(k,nB))
simBsamples = pd.DataFrame(simBsamples)

# 3. Compute the difference between the two simulated medians - k times
simDifmedians = simAsamples.median(axis=1) - simBsamples.median(axis=1)


# 4. Find the two relevant quantiles of the k generated differences in
medians
print(np.quantile(simDifmedians,[0.025,0.975],
method='averaged_inverted_cdf'))

[-3.617 -0.401]
```

Thus, we accept that the difference between the two medians is somewhere between 0.4 and 3.6, and confirming the group difference that we also found in the means, as the 0 is not included in the interval.

Note the differences in the Python code compared to the previous bootstrapping example: we use the `stats.expon.rvs`-function instead of the `stats.norm.rvs`-function and change the method from `.mean` to `.median`.

|||| Remark 4.13 Hypothesis testing by simulation based confidence intervals

We have also seen that even though the simulation method boxes given are providing confidence intervals: we can also use this for hypothesis testing, by using the basic relation between hypothesis testing and confidence intervals. A confidence interval includes the ’acceptable’ values, and values outside the confidence interval are the ’rejectable’ values.

4.3 The non-parametric bootstrap

4.3.1 Introduction

In the introduction to the parametric bootstrap section above it was discussed that another approach instead of finding the ’right’ distribution to use is to not assume any distribution at all. This can be done, and a way to do this simulation based is called the *non-parametric bootstrap* and is presented in this section. The section is structured as the parametric bootstrap section above – including the similar subsections and similar method boxes. So there will be two method boxes in this section: one for the one-sample analysis and one for the two-sample analysis.

In fact, the non-parametric approach could be seen as the parametric approach but substituting the density/distribution used for the simulation by the observed distribution of the data, that is, the empirical cumulative distribution function (ecdf), cf. Chapter 1. In practice this is carried out by (re)-sampling the data we have again and again: To get the sampling distribution of the mean (or any other feature) based on the n observations that we have in our given sample, we simply again and again take new samples with n observations from the one we have. This is done ”with replacement” such that the ”new” samples, from now on called the *bootstrap samples* would contain some of the original observations in duplicates (or more) and others will not be there.

4.3.2 One-sample confidence interval for μ

We have the sample: $x_1, \ldots, x_n$.

The $100(1-\alpha)\%$ confidence interval for μ determined by the non-parametric bootstrap is first exemplified:

Example 4.14 Women's cigarette consumption

In a study women's cigarette consumption before and after giving birth is explored. The following observations of the number of smoked cigarettes per day were observed:

before	after	before	after
8	5	13	15
24	11	15	19
7	0	11	12
20	15	22	0
6	0	15	6
20	20		

This is a typical paired *t*-test setup, as discussed in Section 3.2.3, which then was handled by finding the 11 differences and thus transforming it into a one-sample setup. First we read the observations into Python and calculate the differences by:

```
# # Read the data and calculate the difference for each woman before
and after
x1 = np.array([8, 24, 7, 20, 6, 20, 13, 15, 11, 22, 15])
x2 = np.array([5, 11, 0, 15, 0, 20, 15, 19, 12, 0, 6])
dif = x1-x2
print(dif)

[ 3 13  7  5  6  0 -2 -4 -1 22  9]
```

There is a random-sampling function in the NumPy package (which again is based on a uniform random number generator): `np.random.choice`. Eg. you can get 5 repeated samples with replacement by: (Note that the argument `replace` is `true` by default. Sampling without replacement can thus be done by specifying `replace=false`.)

```
np.random.choice(dif,size=(5,len(dif)))

array([[-4,  5,  0, -1,  7, -2, -2,  5,  0, -2,  3],
       [ 3, -1,  5, -2,  9, 13, -2, -1,  0, 13,  9],
       [ 6, -2, 22,  7, -4,  7,  7,  5,  9,  5, 22],
       [-1,  5,  6, -1,  5, -1,  7,  3,  3,  6,  6],
       [-4,  7, -2, -2,  3,  7, -2, -2,  9, 13, 22]])
```

Explanation: the first argument, `dif`, defines the sampling space, and the second argument, `size=(5,len(dif))`, defines the number of bootstrap samples, 5, and their respective sizes, `len(dif)`.

One can then run the following to get a 95% confidence interval for μ based on $k = 100000$:

```
# Set the number of simulations
k = 100000

# Simulate k samples each with 11 observations by
# sampling with replacement from the data
simsamples = np.random.choice(dif,size=(k,len(dif)))

# Compute the mean in each of the k samples
simmeans = simsamples.mean(axis=1)

# Find the two relevant quantiles of the k generated meeans
print(np.quantile(simmeans,[0.025,0.975],
method=’averaged_inverted_cdf’))

[1.364 9.818]
```

Explanation: The np.random.choice-function is called 100.000 times and the results collected in an 11 × 100.000 matrix. Then in a single call the 100.000 averages are calculated and subsequently the relevant quantiles found.

Note, that we use the similar three steps as above for the parametric bootstrap, with the only difference that the simulations are carried out by the re-sampling the given data rather than from some probability distribution.

4.3.3 One-sample confidence interval for any feature

What we have just done can be more generally expressed as follows:

|||| Method 4.15 Confidence interval for any feature θ by non-parametric bootstrap

Assume we have actual observations $x_1, \ldots, x_n$:

1. Simulate k samples of size n by randomly sampling among the available data (with replacement)
2. Calculate the statistic $\hat{\theta}$ in each of the k samples $\hat{\theta}_1^*, \ldots, \hat{\theta}_k^*$
3. Find the $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ quantiles for these, $q^*_{100(\alpha/2)\%}$ and $q^*_{100(1-\alpha/2)\%}$ as the $100(1-\alpha)\%$ confidence interval: $\left[q^*_{100(\alpha/2)\%},\ q^*_{100(1-\alpha/2)\%}\right]$

|||| Example 4.16

Let us find the 95% confidence interval for the median cigarette consumption change in the example from above:

```python
# The 95% CI for the median change
k = 100000
simsamples = pd.DataFrame(np.random.choice(dif,size=(k,len(dif))))
simmedians = simsamples.median(axis=1)
print(np.quantile(simmedians,[0.025,0.975],
method='averaged_inverted_cdf'))
```

```
[-1.000  9.000]
```

4.3.4 Two-sample confidence intervals

We now have two random samples: $x_1, \ldots, x_{n_1}$ and $y_1, \ldots, y_{n_2}$. The $100(1-\alpha)\%$ confidence interval for $\theta_1 - \theta_2$ determined by the non-parametric bootstrap is defined as:

|||| Method 4.17 Two-sample confidence interval for $\theta_1 - \theta_2$ by non-parametric bootstrap

Assume we have actual observations $x_1, \ldots, x_{n_1}$ and $y_1, \ldots, y_{n_2}$:

1. Simulate k sets of 2 samples of n_1 and n_2 observations from the respective groups (with replacement)
2. Calculate the difference between the features in each of the k samples $\hat{\theta}^*_{x1} - \hat{\theta}^*_{y1}, \ldots, \hat{\theta}^*_{xk} - \hat{\theta}^*_{yk}$
3. Find the $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ quantiles for these, $q^*_{100(\alpha/2)\%}$ and $q^*_{100(1-\alpha/2)\%}$ as the $100(1-\alpha)\%$ confidence interval: $\left[q^*_{100(\alpha/2)\%}, q^*_{100(1-\alpha/2)\%}\right]$

|||| Example 4.18 Teeth and bottle

In a study it was explored whether children who received milk from bottle as a child had worse or better teeth health conditions than those who had not received milk from the bottle. For 19 randomly selected children it was recorded when they had their first incident of caries:

bottle	age	bottle	age	bottle	Age
no	9	no	10	yes	16
yes	14	no	8	yes	14
yes	15	no	6	yes	9
no	10	yes	12	no	12
no	12	yes	13	yes	12
no	6	no	20		
yes	19	yes	13		

One can then run the following to obtain a 95 % confidence interval for $\mu_1 - \mu_2$ based on $k = 100000$:

```
# Reading in "no bottle" group
x = np.array([9, 10, 12, 6, 10, 8, 6, 20, 12])
# Reading in "yes bottle" group
y = np.array([14,15,19,12,13,13,16,14,9,12])
# Number of simulations
k = 100000
# Simulate each sample k times
simxsamples = pd.DataFrame(np.random.choice(x,size=(k,len(x))))
simysamples = pd.DataFrame(np.random.choice(y,size=(k,len(y))))
# Calculate the sample mean differences
simmeandifs = simxsamples.mean(axis=1) - simysamples.mean(axis=1)
# Quantiles of the differences gives the CI
print(np.quantile(simmeandifs,[0.025,0.975],
method=’averaged_inverted_cdf’))

[-6.211 -0.122]
```

Example 4.19

Let us make a 99% confidence interval for the difference of medians between the two groups in the tooth health example:

```
# CI for the median differences
simmediandifs = simxsamples.median(axis=1) - simysamples.median(axis=1)

print(np.quantile(simmediandifs,[0.005,0.995],
method=’averaged_inverted_cdf’))

[-8.000  0.000]
```

|||| Remark 4.20 Warning: Bootstrapping may not always work well for small sample sizes!

The bootstrapping idea was presented here rather enthusiastically as an almost magic method that can do everything for us in all cases. This is not the case. Some statistics are more easily bootstrapped than others and generally non-parametric bootstrap will not work well for small samples. The inherent lack of information with small samples cannot be removed by any magic trick. Also, there are more conceptually difficult aspects of bootstrapping for various purposes to improve on some of these limitations, see the next section. Some of the "naive bootstrap" CI interval examples introduced in this chapter is likely to not have extremely good properties – the coverage percentages might not in all cases be exactly at the aimed nominal levels.

Chapter 5

Simple Linear regression

5.1 Linear regression and least squares

In engineering applications we are often faced with the problem of determining the best model of some outcome given a known input

$$y = f(x), \tag{5-1}$$

hence x is the input and the function f is the model. The task is now to find the best model given the input variables (x) and the outcome (y). The simplest model, besides just a mean value (covered in Chapters 3 and 4), would be a model where f is a linear function of x

$$y = \beta_0 + \beta_1 x. \tag{5-2}$$

When the outcome y is the result of some experiment, the model will not be perfect, and we need to add an error term

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = \{1, \dots, n\}, \tag{5-3}$$

where ε_i is called the *error* and is a sequenece of independent random variables with expectation equal zero (i.e. the mean $\mathrm{E}(\varepsilon_i) = 0$ and some variance ($\mathrm{V}(\varepsilon_i) = \sigma^2$). The statistical interpretation of (5-2) is therefore that it expresses the expected value of the outcome

$$\mathrm{E}(Y_i) = \beta_0 + \beta_1 x_i, \tag{5-4}$$

also called the *model prediction*.

It is of course a very unusual situation that we actually know the values of β_0 and β_1 and we will have to rely on estimates based on some observations $(y_1, \ldots, y_n)$. As usual we express this by putting a "hat" on the parameters

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i, \tag{5-5}$$

meaning that we expect or predict $\hat{y}_i$ (in mean or average) under the conditions given by x_i.

|||| Example 5.1

A car manufacturer wants to find the relation between speed and fuel consumption, to do so she sets up the following model

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \tag{5-6}$$

here $\mathrm{E}(Y_i)$ is the expected fuel consumption at the speed x_i. Further, there will be uncontrollable variations, e.g. due to differences in weather condition, but also nonlinear effects not included in the model might be present. These variations are captured by the ε_i's. We see that speed is something we control here, and we then observe the outcome (here fuel consumption), at different experimental conditions (speeds).

In this chapter we will deal with estimation and inference of β_0, β_1, and prediction of Y_i given x_i. At some point we will have realizations (or observations) of the outcome, in this case we write

$$y_i = \beta_0 + \beta_1 x_i + e_i, \quad i = \{1, \ldots, n\}. \tag{5-7}$$

Now y_i is a realization and e_i is the deviation between the model prediction and the actual observation: a realization of the error ε_i, it is called a *residual*. Clearly, we want the residuals to be small in some sense, the usual choice (and the one treated in this chapter) is in the Residual Sum of Squares (RSS) sense, i.e. we want to minimize the residual sum of squares

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (Y_i - (\beta_0 + \beta_1 x_i))^2, \tag{5-8}$$

where we have emphasized that the residual sum of squares is a function of the parameters (β_0, β_1). The parameter estimates $(\hat{\beta}_0, \hat{\beta}_1)$ are the values of β_0 and β_1 which minimize RSS. Note, that we use Y_i and ε_i rather than the observed values (y_i and e_i), this is to emphasize that the estimators are random variables, in actual calculations after the experiments are carried out we will just replace Y_i with y_i and ε_i with e_i. Figure 5.1 sketches the linear regression problem.

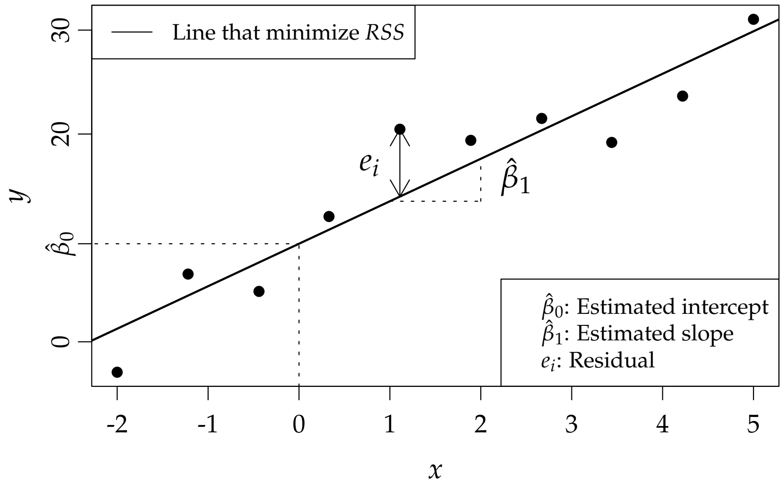


Figure 5.1: Conceptual diagram for the simple linear regression problem.

|||| Remark 5.2 Estimates and estimators

In (5-8) the RSS is a function of the random variables (Y_i), thus making RSS a random variable. If we replace Y_i with the realizations y_i then RSS is also a realization.
In this chapter the result of optimizing RSS with respect to β_0 and β_1 will be denoted $\hat{\beta}_0$ and $\hat{\beta}_1$. Sometimes $\hat{\beta}_0$ and $\hat{\beta}_1$ will be functions of Y_i and sometimes they will be functions of the realizations y_i, they are referred to as:

1. **Estimators**: before the experiment has been carried out, then $\hat{\beta}_0$ and $\hat{\beta}_1$ are functions of Y_i and they are also random variables, and we call them *estimators*.

2. **Estimates**: after the experiment had been carried out, then $\hat{\beta}_0$ and $\hat{\beta}_1$ are functions of y_i and they are also realizations of random variables, and we call them *estimates*.

Remark 5.3 Two types of examples

In this chapter we will use two types of examples, one is labelled "Simulation", which are simulation studies intended to illustrate the consequences of theorems and general results. While the other type of examples (not labelled "Simulation"), are intended to illustrate the use of the theorems on pratical examples.

5.2 Parameter estimates and estimators

When $\hat{\beta}_0$ and $\hat{\beta}_1$ is a result of minimizing the function in Equation (5-8), we refer to the estimators as *least squares estimators*. The least squares estimators are given in the following theorem:

Theorem 5.4 Least squares estimators

The least squares estimators of β_0 and β_1 are given by

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(x_i - \bar{x})}{S_{xx}}, \tag{5-9}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{x}, \tag{5-10}$$

where $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$.

As we can see above the estimators ($\hat{\beta}_1$ and $\hat{\beta}_2$) are functions of random variables (Y_i and $\bar{Y}$), and thus the estimators are themselves random variables. We can therefore talk about the expectation, variance and distribution of the estimators. In analyses with data we will of course only see realizations of Y_i and we just replace Y_i and $\bar{Y}$ with their realizations y_i and $\bar{y}$. In this case we speak about *estimates* of β_0 and β_1.

Before we go on with the proof of Theorem 5.4, the application of the theorem is illustrated in the following example:

|||| Example 5.5 Student height and weight

Consider the student height and weight data presented in Chapter 1,

Heights (x_i)	168	161	167	179	184	166	198	187	191	179
Weights (y_i)	65.5	58.3	68.1	85.7	80.5	63.4	102.6	91.4	86.7	78.9

We want to find the best least squares regression line for these points, this is equivalent to calculating the least squares estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$.

We start by finding the two sample means

$$\begin{aligned}\bar{x} &= \frac{1}{10}(168 + 161 + \ldots + 179) = 178,\\ \bar{y} &= \frac{1}{10}(65.5 + 58.3 + \ldots + 78.9) = 78.11.\end{aligned}$$

The value of S_{xx} is calculated by

$$S_{xx} = (168 - 178)^2 + \ldots + (179 - 178)^2 = 1342.$$

We can now calculate $\hat{\beta}_1$ as

$$\hat{\beta}_1 = \frac{1}{1342}\big((65.5 - 78.11)(168 - 179) + \ldots + (79.9 - 78.11)(179 - 178)\big) = 1.11,$$

and finally, we can calculate $\hat{\beta}_0$ as

$$\hat{\beta}_0 = 78.11 - 1.11 \cdot 178 = -120.$$

The calculations can be implemented by

```python
# Read data
x = np.array([168, 161, 167, 179, 184, 166, 198, 187, 191, 179])
y = np.array([65.5, 58.3, 68.1, 85.7, 80.5, 63.4, 102.6, 91.4, 86.7,
78.9])

# Calculate averages
xbar = x.mean()
ybar = y.mean()

# Parameters estimates
Sxx = sum((x - xbar)**2)
beta1hat = sum((x - xbar)*(y - ybar)) / Sxx
beta0hat = ybar - beta1hat * xbar
print(round(beta0hat.mean(),2), round(beta1hat.mean(),2))
```

```
-119.96 1.11
```

Rather than using “manual” calculations, we can use the buildin functions

```
student = pd.DataFrame({'x': x, 'y': y})
fitStudents = smf.ols(formula = 'y ~ x', data=student).fit()
print(fitStudents.summary(slim=True))

                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.932
Model:                            OLS   Adj. R-squared:                  0.924
No. Observations:                  10   F-statistic:                     110.3
Covariance Type:            nonrobust   Prob (F-statistic):           5.87e-06
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   -119.9581     18.897     -6.348      0.000    -163.535     -76.381
x              1.1127      0.106     10.504      0.000       0.868       1.357
==============================================================================
```

As we can see the two calculations give the same results regarding the parameter estimates. We can also see that the high level calculation gives some more information. How to interpret and calculate these numbers will be treated in the following pages.

Before we go on with the analysis of the results, the proof of Theorem 5.4 is presented:

|||| Proof

Of Theorem 5.4: In order to find the minimum of the function *RSS* we differentiate the residual sum of squares with respect to the parameters

$$\frac{\partial RSS}{\partial \hat{\beta}_0} = -2\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)), \tag{5-11}$$

now equating with zero we get

$$\begin{aligned} 0 &= -2\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) \\ &= -2n\bar{y} + 2n\hat{\beta}_0 + 2n\hat{\beta}_1\bar{x}, \end{aligned} \tag{5-12}$$

solving for $\hat{\beta}_0$ gives

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}, \tag{5-13}$$

and by similar calculations we get

$$\begin{aligned}
\frac{\partial RSS}{\partial \hat{\beta}_1} &= \frac{\partial}{\partial \hat{\beta}_1}\left(\sum_{i=1}^{n}(y_i - (\bar{y} - \hat{\beta}_1\bar{x} + \hat{\beta}_1 x_i))^2\right) \\
&= \frac{\partial}{\partial \hat{\beta}_1}\left(\sum_{i=1}^{n}((y_i - \bar{y}) - \hat{\beta}_1(x_i - \bar{x}))^2\right) \\
&= -2\sum_{i=1}^{n}((y_i - \bar{y}) - \hat{\beta}_1(x_i - \bar{x}))(x_i - \bar{x}) \\
&= -2\left[\sum_{i=1}^{x}(y_i - \bar{y})(x_i - \bar{x}) - \hat{\beta}_1\sum_{i=1}^{x}(x_i - \bar{x})^2\right],
\end{aligned} \tag{5-14}$$

equating with zero and solving for $\hat{\beta}_1$ gives

$$\begin{aligned}
\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\
&= \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{S_{xx}}.
\end{aligned} \tag{5-15}$$

The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are called **least squares estimates**, because they minimize the sum of squared residuals (i.e. *RSS*). Replacing y_i with Y_i give the estimators in the theorem.

■

When we have obtained parameter estimates in the linear regression model above, we would like to make quantitative statements about the uncertainty of the parameters, and in order to design tests we will also need the probability distribution of the parameter estimators. The usual assumption is that the errors are normal random variables

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \text{where } \varepsilon_i \sim N(0, \sigma^2), \tag{5-16}$$

or in other words the errors are independent identically distributed (i.i.d.) normal random variables with zero mean and variance σ^2. When random variables are involved we know that repeating the experiment will result in different values of the response (Y_i), and therefore in different values of the parameter estimates. To illustrate this we can make simulation experiments to analyse the behaviour of the parameter estimates. Recall that the role of simulation examples are to illustrate probabilistic behaviour of e.g. estimators, not how actual data is analysed.

|||| Remark 5.6 How to write a statistical model

In Remark 3.2 it was explained how to write the model behind the t-tests, i.e.

$$X_i \sim N(\mu, \sigma^2) \text{ and i.i.d.} \tag{5-17}$$

Remember, that i.i.d. is short for independently and identically distributed, which essentially means that the observations are selected randomly from population, see the text after Example 1.2.

Using this notation the linear regression model could be written

$$Y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2) \text{ and independent,} \tag{5-18}$$

however we will write models as above in Equation (5-16).
Note, if $\beta_1 = 0$ the model is

$$Y_i = \beta_0 + \varepsilon_i, \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \text{ and i.i.d.,} \tag{5-19}$$

which is exactly the model above in Equation (5-17), and the estimate of the mean of the population, from which the sample (i.e. $(y_1, \ldots, y_n)$) was taken, is then

$$\hat{\mu} = \hat{\beta}_0. \tag{5-20}$$

|||| Example 5.7 Simulation of parameter estimation

Consider the linear model

$$Y_i = 10 + 3x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, 5^2). \tag{5-21}$$

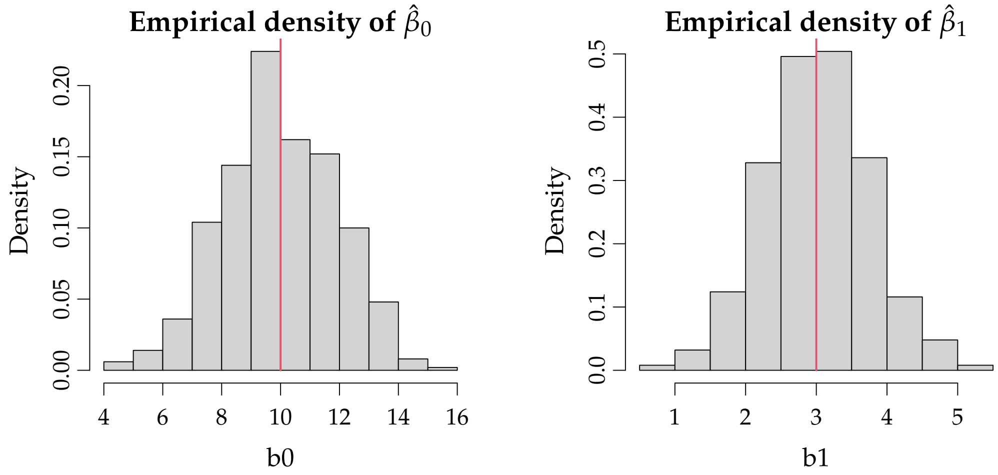
We can make repetitions of this experiment by

```
np.random.seed(124)
n = 10
k = 500
beta0 = 10
beta1 = 3
sigma = 5
x = np.linspace(-2, 5, num=n)
y0 = np.zeros((k,n))
y = y0 + beta0 + beta1*x + np.random.normal(0, sigma, size=(k,n))
```

The variable y now contains n rows and k columns, representing k experiments, for each of the experiment we can calculate the parameter estimates:

```
b0 = np.zeros(k)
b1 = np.zeros(k)
for i in range(k):
    D = pd.DataFrame({'x': x, 'y': y[i,:]})
    result = smf.ols(formula = 'y ~ x', data=D).fit()
    b0[i] = result.params["Intercept"]
    b1[i] = result.params["x"]
print(round(b0.mean(),2), round(b1.mean(),2))

10.09 2.97
```

As we can see the average of the parameter estimates (b0.mean and b1.mean) are very close to the true parameter values ($\beta_0 = 10$ and $\beta_1 = 3$). We can of course also look at the empirical density (the normalized histogram, see Section 1.6.1) of the parameter estimates:



The estimates seem to be rather symmetrically distributed around the true parameter values. It is also clear that there is some variation in the estimates: the estimates of β_0 range from about 4 to about 16 and the estimates of β_1 range from about 1 to 5.

Try changing the code (see the accompanying chapter script):

What happens to the mean value of the estimates if you change the number of data points (n)?

What happens to the empirical density and the scatter plot of the parameter estimates if you change:

- The number of data points (n)?
- The range of x-values?
- The residual variance (σ^2)?
- The values of β_0 and β_1?

In the example above we saw that the average of the parameter estimates were very close to the true values, this is of course a nice property of an estimator. When this is the case in general, i.e. when $\mathrm{E}[\hat{\beta}_i] = \beta_i$ we say that the estimator is central or unbiased. The estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ are in fact central, and we show this in Section 5.2.1 below.

In order to test hypothesis about β_0 and β_1 we will also need to give exact statements about the distribution of the parameters. We saw in Example 5.7 above that the distributions seem to be symmetric around the true values, but we will need more precise statements about the distributions and their variances. This important part will be dealt with in the Sections 5.3 and 5.4.

5.2.1 Estimators are central

In the linear regression model we assume that the observed values of Y_i can be split into two parts: the prediction (the part explained by the regression line ($\beta_0 + \beta_1 x_i$)) and the error (a random part (ε_i)). As usual we view our estimators as functions of random variables (the ε_i's), so it makes sense to calculate the expectation of the estimators. The assumption $\mathrm{E}(\varepsilon_i) = 0$ is central for the presented arguments, and will be used repeatedly.

In order to find the expectation of the parameter estimators we rewrite our estimators as functions of the true parameters (β_0 and β_1)

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(x_i - \bar{x})}{S_{xx}}, \tag{5-22}$$

inserting $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i + \varepsilon_i) = \beta_0 + \beta_1 \bar{x} + \bar{\varepsilon}$ gives

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}\left[(\beta_0 + \beta_1 x_i + \varepsilon_i - (\beta_0 + \beta_1 \bar{x} + \bar{\varepsilon})\right](x_i - \bar{x})}{S_{xx}}, \tag{5-23}$$

now the sum is divided into a part which depends on ε_i (the random part) and a part which is independent of ε_i

$$\begin{aligned}\hat{\beta}_1 &= \frac{\sum_{i=1}^n \beta_1(x_i-\bar{x})^2}{S_{xx}} + \frac{\sum_{i=1}^n (\varepsilon_i-\bar{\varepsilon})(x_i-\bar{x})}{S_{xx}}\\ &= \beta_1 + \frac{\sum_{i=1}^n \varepsilon_i(x_i-\bar{x})}{S_{xx}} - \frac{\bar{\varepsilon}\sum_{i=1}^n (x_i-\bar{x})}{S_{xx}},\end{aligned} \tag{5-24}$$

now observe that $\sum_{i=1}^n (x_i-\bar{x}) = 0$ to get

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^n \varepsilon_i(x_i-\bar{x})}{S_{xx}}, \tag{5-25}$$

for $\hat{\beta}_0$ we get

$$\begin{aligned}\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1\bar{x}\\ &= \frac{1}{n}\sum_{i=1}^n(\beta_0+\beta_1x_i+\varepsilon_i) - \left(\beta_1 + \frac{\sum_{i=1}^n \varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}\\ &= \beta_0 + \beta_1\bar{x}_i + \frac{1}{n}\sum_{i=1}^n \varepsilon_i - \left(\beta_1 + \frac{\sum_{i=1}^n \varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}\\ &= \beta_0 + \frac{1}{n}\sum_{i=1}^n \varepsilon_i - \left(\frac{\sum_{i=1}^n \varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}.\end{aligned} \tag{5-26}$$

Since expectation is a linear operation (see Chapter 2) and the expectation of ε_i is zero we find that $\mathrm{E}[\hat{\beta}_0] = \beta_0$ and $\mathrm{E}[\hat{\beta}_1] = \beta_1$, and we say that $\hat{\beta}_0$, $\hat{\beta}_1$ are central estimators.

5.3 Variance of estimators

In order for us to be able to construct confidence intervals for parameter estimates, talk about uncertainty of predictions and test hypothesis, we will need the variance of the parameter estimates as well as an estimator of the error variance (σ^2).

Parameter variance and covariance of estimators are given in the following theorem:

|||| Theorem 5.8 Variance of estimators

The variance and covariance of the estimators in Theorem 5.4 are given by

$$\mathrm{V}[\hat{\beta}_0] = \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2}{S_{xx}}, \tag{5-27}$$

$$\mathrm{V}[\hat{\beta}_1] = \frac{\sigma^2}{S_{xx}}, \tag{5-28}$$

$$\mathrm{Cov}[\hat{\beta}_0, \hat{\beta}_1] = -\frac{\bar{x}\sigma^2}{S_{xx}}, \tag{5-29}$$

where σ^2 is usually replaced by its estimate ($\hat{\sigma}^2$). The central estimator for σ^2 is

$$\hat{\sigma}^2 = \frac{RSS(\hat{\beta}_0, \hat{\beta}_1)}{n-2}. \tag{5-30}$$

When the estimate of σ^2 is used the variances also become estimates and we'll refer to them as $\hat{\sigma}^2_{\beta_0}$ and $\hat{\sigma}^2_{\beta_1}$.

The variance of $\hat{\beta}_1$ is a function of the true error variance (σ^2) and S_{xx}. For most (all reasonable) choices of the regressors (x), S_{xx} will be an increasing function of n, and the variance of $\hat{\beta}_1$ will therefore decrease as n increases. This expresses that we will be more certain about the estimates as we increase the number of points in our sample. The same is true for the variance of $\hat{\beta}_0$, and the covariance between $\hat{\beta}_1$ and $\hat{\beta}_0$. The error variance estimate ($\hat{\sigma}^2$) is the residual sum of squares divided by $n-2$, the intuitive explanation for the $n-2$ (rather than n or $n-1$) is that if we only have two ($n = 2$) pairs (x_i, y_i), it will not be possible to say anything about the variation (the residuals will be zero). Or another phrasing is that; we have used 2 degrees of freedom to estimate $\hat{\beta}_0$ and $\hat{\beta}_1$.

Before we turn to the proof of Theorem 5.8, we will take a look at a couple of examples.

|||| Example 5.9 (Example 5.5 cont.)

In Example 5.5 we found the parameter estimates

$$\hat{\beta}_0 = -120, \quad \hat{\beta}_1 = 1.11,$$

we can now find predicted values of the dependent variable by

$$\hat{y}_i = -120 + 1.11 \cdot x_i,$$

and the values of the residuals

$$e_i = y_i - \hat{y}_i,$$

and finally the error variance estimate is

$$\hat{\sigma}^2 = \frac{1}{10-2}\sum_{i=1}^{10} e_i^2.$$

We can implement the results by:

```
beta0 = fitStudents.params["Intercept"]
beta1 = fitStudents.params["x"]
e = student["y"] - (beta0 + beta1 * student["x"])
n = len(e)
sigma = math.sqrt(np.sum(e**2) / (n - 2))
sigma_beta0 = math.sqrt(sigma**2 * (1 / n + xbar**2 / Sxx))
sigma_beta1 = math.sqrt(sigma**2 / Sxx)
print(round(sigma,2), round(sigma_beta0,2), round(sigma_beta1,2))

3.88 18.9 0.11
```

As usual we use standard deviations rather than variances, this also means that we can compare with the results from `smf.ols` (see Example 5.5). Again we can find our estimates in the `Python`-output, the parameter standard deviations are given in the second column of the coefficient matrix. The estimated standard deviation is not reported in the summary from `smf.ols`, but can be extracted by

```
round(np.sqrt(fitStudents.scale),2)

np.float64(3.88)
```

The simulation example (Example 5.7) can also be extended to check the equations of Theorem 5.8:

|||| Example 5.10 Simulation continued

In Example 5.7 we looked at simulation from the model

$$Y_i = 10 + 3x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, 5^2)$$

In order to calculate the variance estimates we need to calculate $\bar{x}$ and S_{xx}:

```
np.random.seed(1241)
Sxx = (n-1)*x.var()
print(x.mean(), round(Sxx,2))

1.5 44.92


k = 500
y0 = np.zeros((k,n))
y = y0 + beta0 + beta1*x + np.random.normal(0, 5, size=(k,n))
```

and we would expect to obtain the variance estimates close to

$$
\begin{aligned}
\mathrm{V}[\hat{\beta}_0] &= 5^2\left(\frac{1}{10} + \frac{1.50^2}{49.91}\right) = 3.63\\
\mathrm{V}[\hat{\beta}_1] &= \frac{5^2}{49.91} = 0.501.
\end{aligned}
$$

With simulations we find:

```
b0 = np.zeros(k)
b1 = np.zeros(k)
sigma = np.zeros(k)
for i in range(k):
    D = pd.DataFrame({’x’: x, ’y’: y[i,:]})
    result = smf.ols(formula = ’y ∼ x’, data=D).fit()
    b0[i] = result.params["Intercept"]
    b1[i] = result.params["x"]
    sigma[i] = math.sqrt(result.scale)

print(round(b0.var(),2), round(b1.var(),2), round(sigma.mean(),2))

3.58 0.46 4.81
```

We can see that the simulated values are close to the theoretical values. You are invited to play around with different settings for the simulation, in particular increasing k will increase the accuracy of the estimates of the variances.

The example above shows how Theorem 5.8 can be illustrated by simulation, a formal proof is given by:

|||| Proof

Of Theorem 5.8. Using (5-26) we can write the variance of $\hat{\beta}_0$ as

$$\mathrm{V}(\hat{\beta}_0) = \mathrm{V}\left[\beta_0 + \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i - \left(\frac{\sum_{i=1}^{n}\varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}\right], \tag{5-31}$$

using the definition of the variance ($\mathrm{V}(X) = \mathrm{E}[(X-\mathrm{E}[X])^2]$) and $\mathrm{E}(\varepsilon) = 0$ we get

$$\begin{aligned}\mathrm{V}(\hat{\beta}_0) = \mathrm{V}\left[\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\right] + \mathrm{V}\left[\left(\frac{\sum_{i=1}^{n}\varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}\right] - \\ 2\,\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\left(\frac{\sum_{i=1}^{n}\varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}\right],\end{aligned} \tag{5-32}$$

now use independence between ε_i and ε_j ($i \neq j$) to get

$$\begin{aligned}\mathrm{V}(\hat{\beta}_0) &= \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2\sum_{i=1}^{n}(x_i-\bar{x})^2}{(S_{xx})^2} + \frac{\bar{x}\sigma^2}{nS_{xx}}\sum_{i=1}^{n}(x_i-\bar{x}) \\ &= \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2}{S_{xx}}.\end{aligned} \tag{5-33}$$

Finally, the variance of $\hat{\beta}_1$ is (again using the definition of variance and independence of the ε's)

$$\begin{aligned}\mathrm{V}(\hat{\beta}_1) &= \mathrm{V}\left[\beta_1 + \frac{\sum_{i=1}^{n}\varepsilon_i(x_i-\bar{x})}{S_{xx}}\right] \\ &= \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2\,\mathrm{V}(\varepsilon_i)}{(S_{xx})^2} \\ &= \frac{\sigma^2}{S_{xx}},\end{aligned} \tag{5-34}$$

and the covariance between the parameters estimates becomes

$$\begin{aligned}\mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) &= \mathrm{E}\left[(\hat{\beta}_0-\beta_0)(\hat{\beta}_1-\beta_1)\right] \\ &= \mathrm{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i - \frac{\sum_{i=1}^{n}\varepsilon_i(x_i-\bar{x})}{S_{xx}}\right)\bar{x}\frac{\sum_{i=1}^{n}\varepsilon_i(x_i-\bar{x})}{S_{xx}}\right] \\ &= \frac{\bar{x}}{nS_{xx}}\,\mathrm{E}\left[\sum_{i=1}^{n}\varepsilon_i\sum_{j=1}^{n}\varepsilon_j(x_j-\bar{x})\right] - \frac{\bar{x}}{(S_{xx})^2}\,\mathrm{E}\left[\sum_{i=1}^{n}\varepsilon_i^2(x_i-\bar{x})^2\right] \\ &= \frac{\bar{x}\sigma^2(n\bar{x}-n\bar{x})}{nS_{xx}} - \frac{\bar{x}}{(S_{xx})^2}\sigma^2\sum_{i=1}^{n}(x_i-\bar{x})^2 \\ &= -\frac{\bar{x}\sigma^2}{S_{xx}}.\end{aligned} \tag{5-35}$$

To get an estimate of the residual variance we calculate the expected value of the residual sum of squares

$$\mathrm{E}(RSS) = \mathrm{E}\left[\sum_{i=1}^{n}(Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2\right], \tag{5-36}$$

inserting $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ and rearranging gives

$$\begin{aligned}
\mathrm{E}(RSS) &= \sum_{i=1}^{n} \mathrm{E}\big[(-(\hat{\beta}_0 - \beta_0) - (\hat{\beta}_1 - \beta_1)x_i + \varepsilon_i)^2\big] \\
&= \sum_{i=1}^{n} \{\mathrm{E}\big[(\hat{\beta}_0 - \beta_0)^2\big] + \mathrm{E}\big[(\hat{\beta}_1 - \beta_1)^2\big]\, x_i^2 + \mathrm{E}[\varepsilon_i^2] + \qquad (5\text{-}37) \\
&2\,\mathrm{E}\big[(\hat{\beta}_0 - \beta_0)(\hat{\beta}_1 - \beta_1)\big]\, x_i - 2\,\mathrm{E}\big[(\hat{\beta}_0 - \beta_0)\varepsilon_i\big] - 2\,\mathrm{E}\big[(\hat{\beta}_1 - \beta_1)\varepsilon_i\big]\, x_i\}\,,
\end{aligned}$$

now observe that $\mathrm{E}\big[(\hat{\beta}_0 - \beta_0)^2\big] = \mathrm{V}[\hat{\beta}_0]$, $\mathrm{E}\big[(\hat{\beta}_1 - \beta_1)^2\big] = \mathrm{V}[\hat{\beta}_1]$, $\mathrm{E}(\varepsilon_i^2) = \sigma^2$, and $\mathrm{E}\big[(\hat{\beta}_0 - \beta_0)(\hat{\beta}_1 - \beta_1)\big] = \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1)$, and insert $\hat{\beta}_0$ and $\hat{\beta}_1$ in the last two terms

$$\begin{aligned}
\mathrm{E}(RSS) &= n\,\mathrm{V}(\hat{\beta}_0) + \mathrm{V}(\hat{\beta}_1)\sum_{i=1}^{n} x_i^2 + n\sigma^2 + 2\sum_{i=1}^{n} Cov\,(\hat{\beta}_0, \hat{\beta}_1)\, x_i - \\
&2\sum_{i=1}^{n}\left\{\mathrm{E}\left[\left(\frac{1}{n}\sum_{j=1}^{n}\varepsilon_j - \frac{\sum_{j=1}^{2}\varepsilon_j(x_j - \bar{x}}{S_{xx}}\right)\varepsilon_i\right] - \mathrm{E}\left[\frac{\sum_{j=1}^{n}\varepsilon_j(x_j - \bar{x})}{S_{xx}}\varepsilon_i\right] x_i\right\} \\
&= \sigma^2 + \frac{n\bar{x}^2\sigma^2}{S_{xx}} + \frac{\sigma^2\sum_{i=1}^{n} x_i^2}{S_{xx}} + n\sigma^2 - 2\sum_{i=1}^{n}\frac{\bar{x}\sigma^2}{S_{xx}}x_i - \qquad (5\text{-}38) \\
&2\sum_{i=1}^{n}\left(\frac{\sigma^2}{n} - \frac{\sigma^2(x_i - \bar{x})}{S_{xx}}\right) - 2\sum_{i=1}^{n}\frac{\sigma^2(x_i - \bar{x})x_i}{S_{xx}},
\end{aligned}$$

now collect terms and observe that $\sum x_i = n\bar{x}$

$$\begin{aligned}
\mathrm{E}(RSS) &= \sigma^2(n+1) + \frac{\sigma^2}{S_{xx}}\sum_{i=1}^{n}(x_i^2 + \bar{x}^2) - 2\frac{n\bar{x}^2\sigma^2}{S_{xx}} - 2\sigma^2 - 2\frac{\sigma^2\sum_{i=1}^{n}(x_i^2 - x_i\bar{x})}{S_{xx}} \\
&= \sigma^2(n-1) + \frac{\sigma^2}{S_{xx}}\sum_{i=1}^{n}(-x_i^2 - \bar{x}^2 + 2x_i\bar{x}) \\
&= \sigma^2(n-1) - \frac{\sigma^2}{S_{xx}}S_{xx} \qquad (5\text{-}39) \\
&= \sigma^2(n-2),
\end{aligned}$$

and thus a central estimator for σ^2 is $\hat{\sigma}^2 = \frac{RSS}{n-2}$.

■

Before we continue with parameter distributions and hypothesis testing, the next example illustrates the behaviour of the parameter variance estimates:

|||| Example 5.11 Simulation continued

Consider the following model

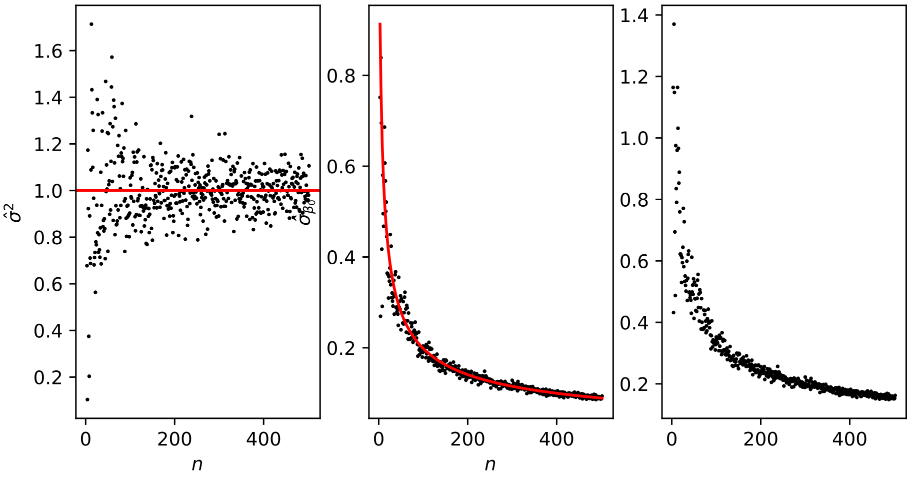
$$Y_i = 1 + x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0,1), \qquad (5\text{-}40)$$

also assume that $x_i = \frac{i-1}{n-1}$, $i = 1, \ldots, n$ where n is the number of pairs (x_i, y_i). We want to make a simulation experiment for increasing number of pairs, and extract the parameter variance, parameter covariance and residual variance estimates. In order to do so we need to extract these numbers from a linear model i `Python`. This can be done by:

```
np.random.seed(134)
x = np.linspace(0, 1, num=10)
y = 1 + x + np.random.normal(0, 1, size=10)
# Fit the model (estimate parameter)
fit = smf.ols(formula = ’y ∼ x’, data=pd.DataFrame({’x’: x, ’y’:
y})).fit()
# Print summary of model fit
print(fit.summary(slim=True))

                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.003
Model:                            OLS   Adj. R-squared:                 -0.122
No. Observations:                  10   F-statistic:                   0.02230
Covariance Type:            nonrobust   Prob (F-statistic):              0.885
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.8541      0.445      4.162      0.003       0.827       2.881
x              0.1122      0.751      0.149      0.885      -1.620       1.844
==============================================================================



# Residual standard deviation
sigma = math.sqrt(fit.scale)
# Estimated standard deviation of parameters
print(round(fit.bse,2))

Intercept    0.45
x            0.75
dtype: float64
```

Now let’s return to the simulation example, the number of independent variables (x) is increased and we draw the residual from the standard normal distribution, in this particular case we can find S_{xx} as a function of n, and compare the expected values (fix $\sigma^2 = 1$) with the simulation results

```
np.random.seed(134)
k = 500
sigma_beta = np.zeros(shape=(k,2))
sigma = np.zeros(k)
n = np.arange(3, k+3)
for i in range(k):
    x = np.linspace(0, 1, num=n[i])
    y = 1 + x + np.random.normal(0, 1, size=n[i])
    D = pd.DataFrame({'x': x, 'y': y})
    fit = smf.ols(formula = 'y ~ x', data=D).fit()
    sigma_beta[i,:] = fit.bse
    sigma[i] = math.sqrt(fit.scale)
```



We see that the residual variance converge to the true value with smaller and smaller variation, while the parameter variances converge to zero. In a plot like this we can therefore see the gain from obtaining more observations of the model.

Again you are encouraged to change some of the specifications of the simulation set up and see what happens.

5.4 Distribution and testing of parameters

The regression model is given by

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{5-41}$$

where the estimators of the parameters and their variances are given by Theorems 5.4 and 5.8. Since the estimators are linear functions of normal random variables (ε_i) they will also be normal random variables. To give the full stochastic model we need to use the estimate of the residual variance, and take the uncertainty of this estimator into account when constructing tests.

As we already saw in Example 5.7 we cannot expect to get the true value of the parameter, but there will be some deviations from the true value due to the stochastic nature of the model/real world application. The purpose of this section is to give the precise description of the parameter distributions. We aim at testing hypothesis of the type

$$H_{0,i}: \quad \beta_i = \beta_{0,i}, \tag{5-42}$$

against some alternatives. The general remarks on hypothesis testing from Chapter 3 still apply, but we will go through the specific construction for linear regression here.

The central estimator of σ^2 (Equation (5-30)) is χ^2-distributed with $n-2$ degrees of freedom. In order to test the hypothesis in Equation (5-42) we need the normalized distance to a null hypothesis (i.e the distance from the observed estimate $\hat{\beta}_{0,i}$ to the value under the null hypothesis $\beta_{0,i}$). From Theorem 5.8 the standard deviations of the parameter estimates are found to

$$\hat{\sigma}_{\beta_0} = \sqrt{\frac{\hat{\sigma}^2}{n} + \frac{\bar{x}^2\hat{\sigma}^2}{S_{xx}}} = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}}, \tag{5-43}$$

$$\hat{\sigma}_{\beta_1} = \sqrt{\frac{\hat{\sigma}^2}{S_{xx}}} = \hat{\sigma}\sqrt{\frac{1}{\sum_{i=1}^{n}(x_i-\bar{x})^2}}, \tag{5-44}$$

under the null hypothesis the normalized (with standard deviations) distance between the estimators and the true values are both t-distributed with $n-2$ degrees of freedom, and hypothesis testing and confidence intervals are based on this t-distribution:

|||| Theorem 5.12 Test statistics

Under the null hypothesis ($\beta_0 = \beta_{0,0}$ and $\beta_1 = \beta_{0,1}$) the statistics

$$T_{\beta_0} = \frac{\hat{\beta}_0 - \beta_{0,0}}{\hat{\sigma}_{\beta_0}}, \tag{5-45}$$

$$T_{\beta_1} = \frac{\hat{\beta}_1 - \beta_{0,1}}{\hat{\sigma}_{\beta_1}}, \tag{5-46}$$

are t-distributed with $n-2$ degrees of freedom, and inference should be based on this distribution.

|||| Proof

The proof is omitted, but rely on the fact that $\hat{\beta}_j$ is normally distributed, $\hat{\sigma}^2_{\beta_j}$ is related to the χ^2-distribution, and (provided independence) a standard normal random variable divided by the square root of a χ^2 distributed random variable is t-distributed.

■

In this material we only test two-sided hypothesis. The hypothesis can be concluded using p-values or critical values, in the same way as we saw for hypothesis regarding mean values in Chapter 3 Section 3.1.7.

|||| Example 5.13 Example 5.9 cont.

We continue with the data from Examples 5.5 and 5.9, where we found the parameter estimates and the variance estimates. We want to test the hypotheses

$$H_{00}: \beta_0 = 0 \quad \text{vs.} \quad H_{10}: \beta_0 \neq 0, \tag{5-47}$$

$$H_{01}: \beta_1 = 1 \quad \text{vs.} \quad H_{11}: \beta_1 \neq 1, \tag{5-48}$$

on confidence level $\alpha = 0.05$. With reference to Examples 5.5 and 5.9, and Theorem 5.12, we can calculate the t-statistics as

$$t_{\text{obs},\beta_0} = \frac{-119.96}{18.897} = -6.35, \tag{5-49}$$

$$t_{\text{obs},\beta_1} = \frac{1.113 - 1}{0.1059} = 1.07. \tag{5-50}$$

H_{00} is rejected if $|t_{\text{obs},\beta_0}| > t_{1-\alpha/2}$, and H_{01} is rejected if $|t_{\text{obs},\beta_1}| > t_{1-\alpha/2}$, as usual we can find the critical values by:

```
round(stats.t.ppf(0.975, df=10-2), 2)

np.float64(2.31)
```

and we see that with significance level $\alpha = 0.05$, then H_{00} is rejected and H_{01} isn't. If we prefer p-values rather than critical values, these can be calculated by:

```
pv0 = 2 * (1 - stats.t.cdf(6.35, df=10-2))
pv1 = 2 * (1 - stats.t.cdf(1.07, df=10-2))
print(round(pv0,5), round(pv1,2))

0.00022 0.32
```

The p-value for the intercept (β_0) is less than 0.05, while the p-value for β_1 is greater than 0.05, hence we conclude that $\beta_0 \neq 0$, but we cannot reject that $\beta_1 = 1$. The summary of linear model buildin function, also give t-statistics and p-values (see Example 5.5). The test statistic and the p-value for H_{01} is different from the one we obtained above. The reason for this is that `smf.ols` tests the default hypothesis $H_{0i} : \beta_i = 0$ against the alternative $H_{1i} : \beta_i \neq 0$. Even though this choice is reasonable in many situations it does not cover all situation, and we need to calculate p-values from the summary statistics ourselves if the hypotheses are different from the default ones.

|||| Method 5.14 Level α t-tests for parameter

1. Formulate the *null hypothesis*: $H_{0,i} : \beta_i = \beta_{0,i}$, and the alternative hypothesis $H_{1,i} : \beta_i \neq \beta_{0,i}$
2. Compute the test statistic $t_{\text{obs},\beta_i} = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\beta_i}}$
3. Compute the evidence against the *null hypothesis*

$$p\text{-value}_i = 2 \cdot P(T > |t_{\text{obs},\beta_i}|) \qquad (5\text{-}51)$$

4. If p-value$_i < \alpha$ reject $H_{0,i}$, otherwise accept $H_{0,i}$

In many situations we will be more interested in quantifying the uncertainty of the parameter estimates rather than testing a specific hypothesis. This is usually given in the form of confidence intervals for the parameters:

|||| Method 5.15 Parameter confidence intervals

$(1 - \alpha)$ confidence intervals for β_0 and β_1 are given by

$$\hat{\beta}_0 \pm t_{1-\alpha/2} \cdot \hat{\sigma}_{\beta_0}, \qquad (5\text{-}52)$$

$$\hat{\beta}_1 \pm t_{1-\alpha/2} \cdot \hat{\sigma}_{\beta_1}, \qquad (5\text{-}53)$$

where $t_{1-\alpha/2}$ is the $(1 - \alpha/2)$-quantile of a t-distribution with $n - 2$ degrees of freedom. Where $\hat{\sigma}_{\beta_0}$ and $\hat{\sigma}_{\beta_1}$ are calculated from the results in Theorem 5.8, and Equations (5-43) and (5-44).

|||| Remark 5.16

We will not show (prove) the results in Method 5.15, but see Remark 3.34.

|||| Example 5.17 Example 5.13 cont.

Based on Method 5.15 we immediately find the 95% confidence intervals for the parameters

$$
\begin{aligned}
I_{\beta_0} &= -119.96 \pm t_{0.975} \cdot 18.9 = [-163.5, -76.4], \\
I_{\beta_1} &= 1.113 \pm t_{0.975} \cdot 0.1059 = [0.869, 1.36],
\end{aligned}
$$

with the degrees of freedom for the t-distribution equal 8, and we say with high confidence that the intervals contain the true parameter values. Of course we can get these directly from the result returned by `ols.smf`:

```
print(round(fitStudents.conf_int(alpha=0.05),2))
```

```
                 0      1
Intercept -163.53 -76.38
x             0.87   1.36
```

5.4.1 Confidence and prediction intervals for the line

It is clearly of interest to predict outcomes of future experiments. Here we need to distinguish between prediction intervals, where we predict the outcome of one single experiment, and confidence intervals, where we predict the mean value of future outcomes. In the latter case we only need to account for the uncertainty in the parameter estimates while in the first case we will also need to account for the uncertainty of the error (the random part ε_i).

If we conduct a new experiment with $x_i = x_{\text{new}}$ the *expected outcome* is

$$\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \tag{5-54}$$

where the only source of variation comes from the variance of the parameter estimates, and we can calculate the variance of $\hat{Y}_{\text{new}}$

$$
\begin{aligned}
\text{V}(\hat{Y}_{\text{new}}) &= \text{V}(\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}}) \\
&= \text{V}(\hat{\beta}_0) + \text{V}(\hat{\beta}_1 x_{\text{new}}) + 2\,\text{Cov}(\hat{\beta}_0, \hat{\beta}_1 x_{\text{new}}),
\end{aligned}
\tag{5-55}
$$

now use the calculation rules for variances and covariances (Section 2.7), and insert the variances and the covariance from Theorem 5.8

$$\begin{aligned} \mathrm{V}(\hat{Y}_{\text{new}}) &= \frac{\sigma^2}{n} + \frac{\sigma^2\bar{x}^2}{S_{xx}} + \frac{\sigma^2 x_{\text{new}}^2}{S_{xx}} - 2\frac{\sigma^2\bar{x}x_{\text{new}}}{S_{xx}} \\ &= \sigma^2\left(\frac{1}{n} + \frac{(x_{\text{new}}-\bar{x})^2}{S_{xx}}\right), \end{aligned} \tag{5-56}$$

to find the variance of a single new point, we are using

$$Y_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} + \varepsilon_{\text{new}}, \tag{5-57}$$

and therefore need to add the variance of the residuals (ε_{new} is independent from $\hat{\beta}_0$ and $\hat{\beta}_1$)

$$\mathrm{V}(Y_{\text{new}}) = \sigma^2\left(1 + \frac{1}{n} + \frac{(x_{\text{new}}-\bar{x})^2}{S_{xx}}\right). \tag{5-58}$$

When we construct confidence and prediction intervals we need to account for the fact that σ^2 is estimated from data and thus use the t-distribution:

|||| Method 5.18 Intervals for the line

The (1-α) **confidence interval** for the line $\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}}$ is

$$\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \pm t_{1-\alpha/2}\hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_{\text{new}}-\bar{x})^2}{S_{xx}}}, \tag{5-59}$$

and the (1-α) **prediction interval** is

$$\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \pm t_{1-\alpha/2}\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}}-\bar{x})^2}{S_{xx}}}, \tag{5-60}$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the t-distribution with $n-2$ degrees of freedom.

|||| Remark 5.19

We will not show the results in Method 5.18, but use Equations (5-54)–(5-58) and Remark 3.34.

As illustrated in Figure 5.2 the confidence interval width will approach zero book-IntroStatistics

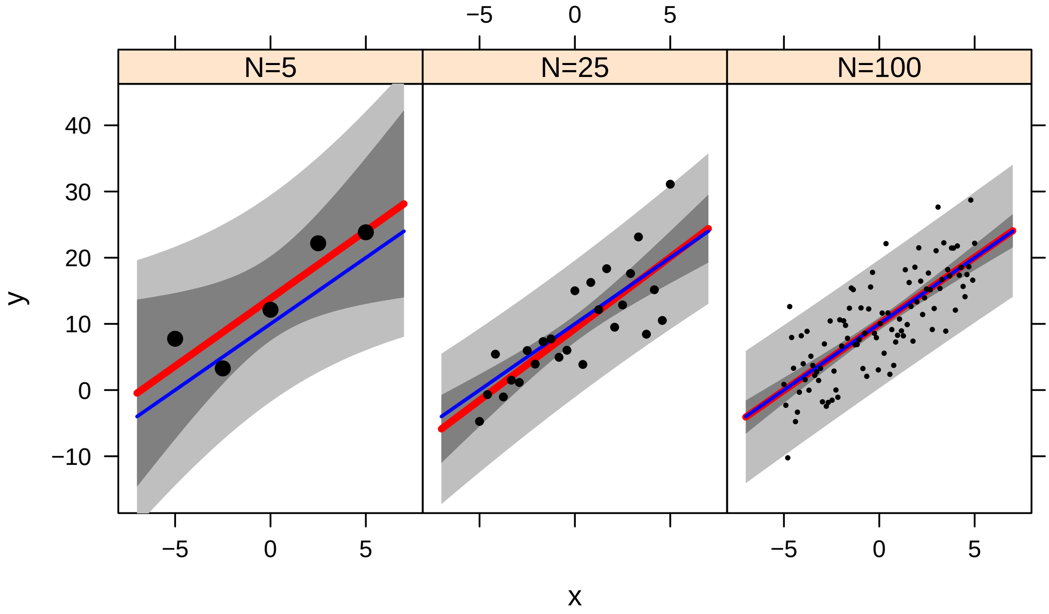


Figure 5.2: Best linear fit (red line), truth (blue line), 95% prediction interval for the points (light grey area), 95 % CI for the line (dark grey area), and observed values (black dots), for simulated data (see Example 5.21).

for an increasing number of data points (n) increase or as S_{xx} increase (actually, in most situations S_{xx} will also increase as n increase). Note also, that the confidence and prediction interval widths are smallest when $x_{\text{new}} = \bar{x}$. The prediction interval width will approach $2z_{1-\alpha/2} \cdot \sigma$ as $n \rightarrow \infty$. The difference between the intervals are that the prediction interval covers a new observation in $(1-\alpha) \cdot 100\%$ of the times, while the confidence interval is expected to cover the true regression line $(1-\alpha) \cdot 100\%$ of the times. One important point is: when we have calculated the prediction interval based on some particular sample, then we actually don't know the probability of this interval covering new observations. What we know is: if we repeat the experiment, then in $(1-\alpha) \cdot 100\%$ of the times a new observation will be covered (we make a new observation each time). Same goes for the confidence interval: we don't know if the true regression line is covered by a particular interval, we only know that if we repeat the experiment, then in $(1-\alpha) \cdot 100\%$ of the times the true regression line will be covered.

In the following: first an example on calculating confidence and prediction intervals, second an example on the width of the intervals, and finally Example 5.22 on the prediction interval coverage, are given.

|||| Example 5.20 Student height and weight Example (5.17 cont.)

With reference to Example 5.17 suppose we want to calculate prediction and confidence intervals for the line for a new student with $x_{\text{new}} = 200$ cm, the prediction is $\hat{y}_{\text{new}} = 102.6$ kg and the 95% confidence and prediction intervals become

$$I_{\text{pred}} = -120 + 1.113 \cdot 200 \pm t_{0.975}(8) \cdot 3.88\sqrt{1 + \frac{1}{10} + \frac{(178-200)^2}{1342}} = [91.8, 113], \tag{5-61}$$

$$I_{\text{conf}} = -120 + 1.113 \cdot 200 \pm t_{0.975}(8) \cdot 3.88\sqrt{\frac{1}{10} + \frac{(178-200)^2}{1342}} = [96.5, 109], \tag{5-62}$$

where $t_{0.975}$ is the 0.975-quantile of a t-distribution with $n-2$ degrees of freedom.

The intervals can be calculated directly by:

```
new_data = pd.DataFrame({'x': [200]})

# Get prediction and confidence intervals
pd.set_option("display.float_format", None) ## unset option
pred = fitStudents.get_prediction(new_data).summary_frame(alpha=0.05)
print(round(pred,2))

     mean  mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  obs_ci_upper
0  102.59     2.63          96.52         108.66         91.77        113.41
```

in the output `mean` refer to the predicted expected value, `mean_se` refer to the standard error of the predicted expected value, `mean_ci_..` refer to confidence intervals while `obs_ci_..` refer to prediction intervals.

|||| Example 5.21 Simulation

Figure 5.2 illustrates the difference between the confidence and prediction intervals for simulated data, with different numbers of observations. The simulated model is

$$y_i = 10 + 2x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, 5^2). \tag{5-63}$$

When n increases the width of the confidence interval for the line narrows and approaches 0, while the prediction interval width does not approach 0, but rather $2z_{1-\alpha/2}\sigma$. Further, the width of the prediction interval will always be larger than the width of the confidence interval.

Example 5.22 Prediction interval coverage

In this example it is illustrated that we actually don’t know the probability that a prediction interval covers new observations, when it is calculated using a sample (i.e. we have a realization of the prediction interval). First, a prediction interval is calculated using a single sample and it is investigated how many of k new observations falls inside it:

```
np.random.seed(129)
n = 30
beta0 = 10
beta1 = 3
sigma = 0.5
# Generate some input values
x = np.random.uniform(-10, 10, n)
# Simulate output values
y = beta0 + beta1*x + np.random.normal(0, sigma, size=n)
# Fit a simple linear regression model to the sample
D = pd.DataFrame({’x’: x, ’y’: y})
fit = smf.ols(formula = ’y ∼ x’, data=D).fit()

# The number of new observations
k = 10000
# Generate k new input values
xnew = pd.DataFrame({’x’: np.random.uniform(-10, 10, k)})
# Calculate the prediction intervals for the new input values
PI = fit.get_prediction(xnew).conf_int(obs=True)
# Simulate new output observations
ynew = beta0 + beta1*xnew[’x’] + np.random.normal(0, sigma, size=k)
# Calculate the fraction of times the prediction interval covered the
# new observation
y_within_ci = (lambda y, ci_low, ci_high: (y > ci_low) & (y <
ci_high))(ynew, PI[:,0], PI[:,1])
print(sum(y_within_ci)/k)

0.8162
```

We see that the interval covered only 81.6% of the new observations, which is quite a bit below 95% (per default `.get_prediction` use $\alpha = 5\%$).

Now, lets repeat the sampling, so we make a new sample k times and each time calculate a new fit and prediction interval, and each time check if a new observation falls inside it:

```
# The number of simulated samples
k = 10000
# Define function to make model and check new y observation
def check_newobs_within_ci():
  # The number of observations and the parameters
  n = 30
  beta0 = 10
  beta1 = 3
  sigma = 0.5
  # Generate some input values
  x = np.random.uniform(-10, 10, n)
  # Simulate output values
  y = beta0 + beta1*x + np.random.normal(0, sigma, size=n)
  # Fit a simple linear regression model to the sample
  D = pd.DataFrame({’x’: x, ’y’: y})
  fit = smf.ols(formula = ’y ∼ x’, data=D).fit()
  # Generate ONE new input value
  xnew = pd.DataFrame({’x’: np.random.uniform(-10, 10, 1)})
  # The prediction interval for the new value
  PI = fit.get_prediction(xnew).conf_int(obs=True)
  # Simulate a single new observations
  ynew = beta0 + beta1*xnew[’x’] + np.random.normal(0, sigma, size=1)
  # Check if the new observation was inside the interval
  y_within_ci = (lambda y, ci_low, ci_high: (y > ci_low) & (y <
ci_high))(ynew, PI[:,0], PI[:,1])
  return y_within_ci

# Replicate the calculation 10000 times
replicated_results = [check_newobs_within_ci() for _ in range(k)]

# The fraction of covered new observations
print(sum(replicated_results)/k)

0    0.9536
Name: x, dtype: float64
```

It is found that coverage is now very close to the expected 95% and this is indeed the way the coverage probability should be interpreted: with repeated sampling the probability is $1 - \alpha$ that a prediction interval will cover a randomly chosen new observation. Same goes for confidence intervals (of any kind): with repeated sampling the probability is $1 - \alpha$ that a confidence interval will cover the true value.

5.5 Matrix formulation of simple linear regression

The simple linear regression problem can be formulated in vector-matrix notation as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I}), \tag{5-64}$$

or

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \quad \varepsilon_i \sim N(0, \sigma^2). \tag{5-65}$$

One of the advantages of the matrix formulation is that the analysis generalize to higher dimensions in a straight forward way (i.e. more xs and parameters as in the following chapter). The residual sum of squares is given by

$$RSS = \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}), \tag{5-66}$$

and the parameter estimators are given by:

|||| Theorem 5.23

The estimators of the parameters in the simple linear regression model are given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}, \tag{5-67}$$

and the covariance matrix of the estimates is

$$\mathrm{V}[\hat{\boldsymbol{\beta}}] = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}, \tag{5-68}$$

and central estimate for the error variance is

$$\hat{\sigma}^2 = \frac{RSS}{n-2}. \tag{5-69}$$

Here $\mathrm{V}[\hat{\boldsymbol{\beta}}]$ is a matrix with elements $(\mathrm{V}[\hat{\boldsymbol{\beta}}])_{11} = \mathrm{V}[\hat{\beta}_0]$, $(\mathrm{V}[\hat{\boldsymbol{\beta}}])_{22} = \mathrm{V}[\hat{\beta}_1]$, and $(\mathrm{V}[\hat{\boldsymbol{\beta}}])_{12} = (\mathrm{V}[\hat{\boldsymbol{\beta}}])_{21} = \mathrm{Cov}[\hat{\beta}_0, \hat{\beta}_1]$.

When we want to find the minimum of RSS, we again need to differentiate RSS

with respect to the parameters

$$\begin{aligned}\frac{\partial RSS}{\partial \boldsymbol{\beta}} &= -2\boldsymbol{X}^T(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta}) \\ &= -2(\boldsymbol{X}^T\boldsymbol{Y}-\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}).\end{aligned} \tag{5-70}$$

Solving for $\boldsymbol{\beta}$ gives

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}, \tag{5-71}$$

taking the expectation of $\hat{\boldsymbol{\beta}}$ we get

$$\begin{aligned}\mathrm{E}[\hat{\boldsymbol{\beta}}] &= \mathrm{E}[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}] \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\,\mathrm{E}[\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}] \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}.\end{aligned} \tag{5-72}$$

The variance of the parameters estimates are given by

$$\begin{aligned}\mathrm{V}[\hat{\boldsymbol{\beta}}] &= \mathrm{V}[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}] \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\,\mathrm{V}[\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}]\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-T} \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\mathrm{V}[\boldsymbol{X}\boldsymbol{\beta}]+\mathrm{V}[\boldsymbol{\varepsilon}])\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-T} \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\sigma^2\boldsymbol{I}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-T} \\ &= \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \\ &= \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}.\end{aligned} \tag{5-73}$$

Again a central estimate for σ^2 is

$$\hat{\sigma}^2 = \frac{RSS(\hat{\boldsymbol{\beta}})}{n-2}, \tag{5-74}$$

and the estimate of the parameter covariance matrix is

$$\hat{\boldsymbol{\Sigma}}_{\beta} = \hat{\sigma}^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}. \tag{5-75}$$

Marginal tests ($H_0 : \beta_i = \beta_{i,0}$) are constructed by observing that

$$\frac{\hat{\beta}_i - \beta_{i,0}}{\sqrt{(\hat{\boldsymbol{\Sigma}}_{\beta})_{ii}}} \sim t(n-2). \tag{5-76}$$

The matrix calculations are illustrated in the next example.

|||| Example 5.24 Student height and weight

To illustrate how the matrix formulation works in student height and weight data is worked through below:

```
# Data
y = student[’y’]
n = len(student[’y’])
X = np.array([np.repeat(1,n),student[’x’]]).T

# Parameter estimates and variance
beta = np.linalg.inv(X.T@X)@X.T@y
e        = y - X@beta
s        = math.sqrt(np.sum(e**2) / (n - 2))
Vbeta    = s**2 * np.linalg.inv(X.T@X)
sbeta    = np.sqrt(Vbeta.diagonal())
Tstat   = beta / sbeta
pval    = 2 * (1 - stats.t.cdf(np.abs(Tstat), df = n-2))

# Print the results
coefMat = np.array([beta, sbeta, Tstat, pval]).T
col_names = ["Estimates","Std.Error","t.value","p.value"]
row_names = ["beta0", "beta1"]
coefMat = pd.DataFrame(coefMat,columns = col_names, index = row_names)
pd.set_eng_float_format(accuracy=3)
print(coefMat)
```

```
         Estimates    Std.Error     t.value      p.value
beta0 -119.958E+00   18.897E+00  -6.348E+00  221.088E-06
beta1    1.113E+00  105.939E-03  10.504E+00    5.875E-06
```

```
print(round(s,2))
```

```
3.88
```

```
pd.set_option("display.float_format", None) ## unset option
```

```
# Prediction and confidence interval
xnew  = np.array([1, 200]).T
ynew  = xnew@beta
Vconf = xnew@Vbeta@xnew.T
Vpred = Vconf + s**2
se_pred = np.sqrt(np.array([Vconf, Vpred]))
np.round(se_pred, 2)

array([2.630, 4.690])
```

5.6 Correlation

In the analysis above we focus on situations where we are interested in one variable (y) as a function of another variable (x). In other situations we might be more interested in how x and y vary together. Examples could be ecosystems, where the number of predators is a function of the number of preys, but the reverse relation is also true, further both of these numbers are affected by random variations and knowledge of one only gives partial knowledge of the other. Another example is individual student grade in 2 different courses, before any grade has been given we will expect that a high grade in one course will imply a high grade in the other course, but none of them is controlled or known in advance.

In the cases above we talk about correlation analysis and to this end we will need the sample correlation coefficient, as defined in Section 1.4.3

$$\hat{\rho} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s_x}\right)\left(\frac{y_i-\bar{y}}{s_y}\right). \tag{5-77}$$

In Section 1.4.3 we notated sample correlation with r, but here we use $\hat{\rho}$, since it is an estimate for the correlation ρ (see Section 2.8), and imply that there is a meaningful interpretation of the ρ.

5.6.1 Inference on the sample correlation coefficient

In order to answer the question: are X and Y correlated? We will be interested in constructing a test of the type

$$H_0 : \rho = 0, \quad H_1 : \rho \neq 0. \tag{5-78}$$

Consider the model

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{5-79}$$

in this case we can rewrite the sample correlation as

$$\begin{aligned}\hat{\rho} &= \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s_x}\right)\left(\frac{y_i-\bar{y}}{s_y}\right)\\ &= \frac{S_{xx}}{n-1}\frac{1}{S_{xx}}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s_x}\right)\left(\frac{y_i-\bar{y}}{s_y}\right)\\ &= \frac{S_{xx}}{n-1}\frac{1}{s_x s_y}\hat{\beta}_1\\ &= \frac{s_x}{s_y}\hat{\beta}_1,\end{aligned} \tag{5-80}$$

implying that the hypothesis (5-78) can be tested by testing the hypothesis

$$H_0: \beta_1 = 0; \quad H_1: \beta_1 \neq 0. \tag{5-81}$$

since clearly the relationship in Equation (5-79) can be reversed. It should be noted that we cannot use the test to construct a confidence interval for ρ.

It should be stressed that correlation does not imply causality, it just implies that the variables x and y vary together. As an example consider the number of beers sold at the university bar and the number of students attending the introductory course in statistics. Let's say that both numbers have increased and therefore have a high correlation coefficient, but it does not seem reasonable to conclude that students are more interested in statistics when drinking beers. A closer look might reveal that the number of enrolled students have actually increased and this can indeed explain the increase in both numbers.

5.6.2 Correlation and regression

In the linear regression models we would like to measure how much of the variation in the outcome (Y) is explained by the input (x). A commonly used measure for this is the coefficient of determination (explanation) or R^2-value (see also the `Python` summary in Example 5.5).

|||| Definition 5.25 Coefficient of determination R^2

The coefficient of determination expresses the proportion of variation in the outcome (Y) explained by the regression line

$$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}. \tag{5-82}$$

In order to find this we will split the variance of y into a component due to the regression line and a component due to the residual variation

$$\begin{aligned}
s_y^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i + e_i - \frac{1}{n}\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_1 x_i + e_i))^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n}(\hat{\beta}_1(x_i - \bar{x}) + e_i)^2 \\
&= \hat{\beta}_1^2 s_x^2 + \frac{n-2}{n-1}\hat{\sigma}^2,
\end{aligned} \tag{5-83}$$

where the first term on the right hand side is the variability explained by the regression line and the second term is the residual variation. Dividing with the variance of Y gives a splitting in the relative variation from each of the terms. If we write out the variation explained by the regression line we get

$$\begin{aligned}
\frac{\hat{\beta}_1^2 s_x^2}{s_y^2} &= \left(\frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^2 \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}\frac{n-1}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \\
&= \left(\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})\right)^2 \frac{n-1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\frac{n-1}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \\
&= \left(\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)\right)^2 \\
&= \hat{\rho}^2.
\end{aligned} \tag{5-84}$$

We can therefore conclude that the proportion of variability (R^2) in Y explained by the regression line is equal to the squared sample correlation coefficient ($\hat{\rho}^2$).

|||| Example 5.26 Student weight and height (Example 5.20 cont.)

With reference to Example 5.20 above we can calculate the correlation coefficient by:

```
np.round(np.corrcoef(student[’x’], student[’y’])[0, 1]**2,3)

np.float64(0.932)
```

or we can base our calculations on the estimated slope:

```
round(fitStudents.params["x"]**2 * student[’x’].var() /
student[’y’].var(),3)

np.float64(0.932)
```

or we can find it directly in the summary of the regression model (see Example 5.5): where the number is called `R-squared`.

5.7 Model validation

So far we have discussed how to estimate parameters, predict future values, make inference etc. in the model

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2). \tag{5-85}$$

In all we have done so far the basic assumption is that the residuals are normally distributed with zero mean and constant variance, and further the residuals are mutually independent. These are assumptions which should be checked and if the assumptions are not fulfilled some actions should be taken in order to fix this. This is called *model validation* or *residual analysis* and is exactly the same idea behind the validation needed for the mean model used for t-tests in Section 3.1.8, though here including a few more steps.

The normality assumption can be checked by a normal q-q plot, and the constant variance assumption may be checked by plotting the residuals as a function of the fitted values. The normal q-q plot have been treated in Section 3.1.8 and should be applied equivalently. Plotting the residuals as a function of the fitted values should not show a systematic behaviour, this means that the range should be constant and the mean value should be constant, as illustrated in the following example:

|||| Example 5.27 Residuals analysis

We consider data generated from the following three models

$$Y_{1,i} = \beta_0 + \beta_1 x_{1,i} + \varepsilon_i, \quad \varepsilon_i \sim N(0,1), \tag{5-86}$$

$$Y_{2,i} = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \varepsilon_i, \quad \varepsilon_i \sim N(0,1), \tag{5-87}$$

$$Y_{3,i} = e^{\beta_0 + \beta_1 x_{1,i} + \varepsilon_i}, \quad \varepsilon_i \sim N(0,1) \tag{5-88}$$

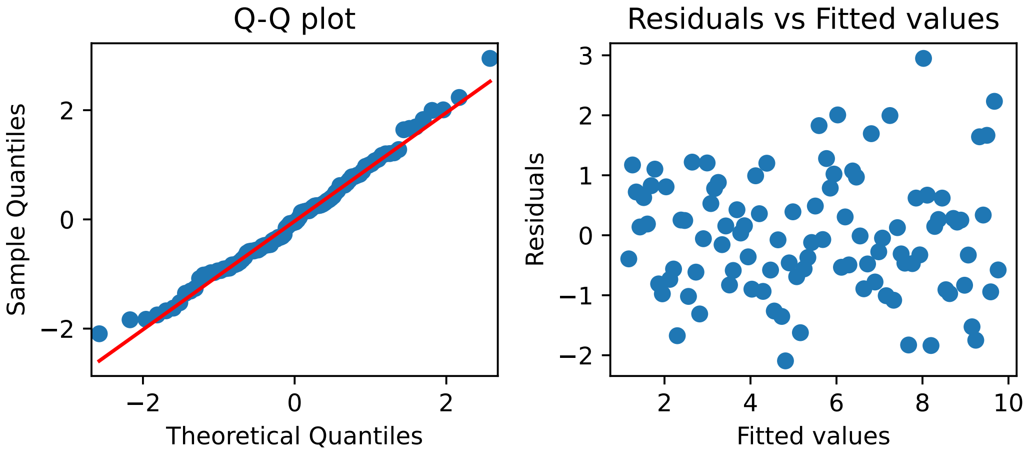
In all cases we fit the model

$$Y_i = \beta_0 + \beta_1 x_{1,i} + \varepsilon_i, \quad \varepsilon_i \sim N(0,\sigma^2), \tag{5-89}$$

to the data: from the first model we would expect that the residual analysis do not show any problems, for the second model we have a linear dependence which is not included in the model and we should see this in the residual analysis, and the third is a non-linear function of the residuals as well as the regressors and one way to handle this will be discussed.

The first model is simulated, estimated and analysed by ($\beta_0 = 0, \beta_1 = 1$, and $\sigma^2 = 1$):

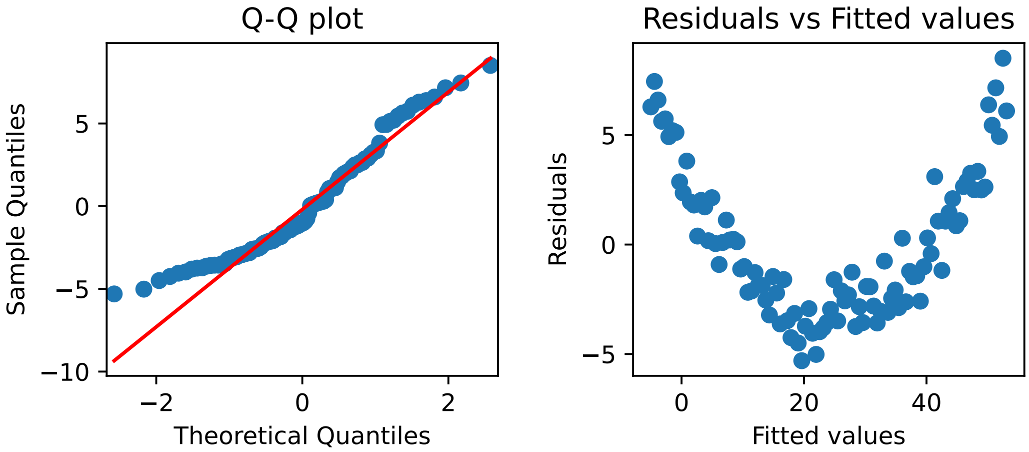
```python
np.random.seed(134)
n = 100
x1 = np.linspace(1, 10, n)
y = x1 + np.random.normal(0, 1, size=n)
D = pd.DataFrame({’x’: x1, ’y’: y})
fit = smf.ols(formula = ’y ∼ x’, data=D).fit()
# Get the predictions (fitted values)
ypred = fit.predict(D) # or fit.fittedvalues
# Plot the Q-Q plot and the residuals vs. fitted values
fig, ax = plt.subplots(1,2)
# Q-Q plot of the residuals
sm.qqplot(y - ypred, line="q", a=1/2, ax=ax[0])
ax[0].set_title("Q-Q plot")
# Scatter plot
ax[1].scatter(ypred, y - ypred)
ax[1].set_xlabel("Fitted values")
ax[1].set_ylabel("Residuals")
ax[1].set_title("Residuals vs Fitted values")
plt.tight_layout()
plt.show()
```



As we can see there is no serious departure from normality and there are no patterns in the residuals as a function of the fitted values.

The second model (with $\beta_0 = 0$, $\beta_1 = 1$, $\beta_2 = 0.5$ and $\sigma^2 = 1$) is simulated, estimated and analysed by (plot functions omitted):

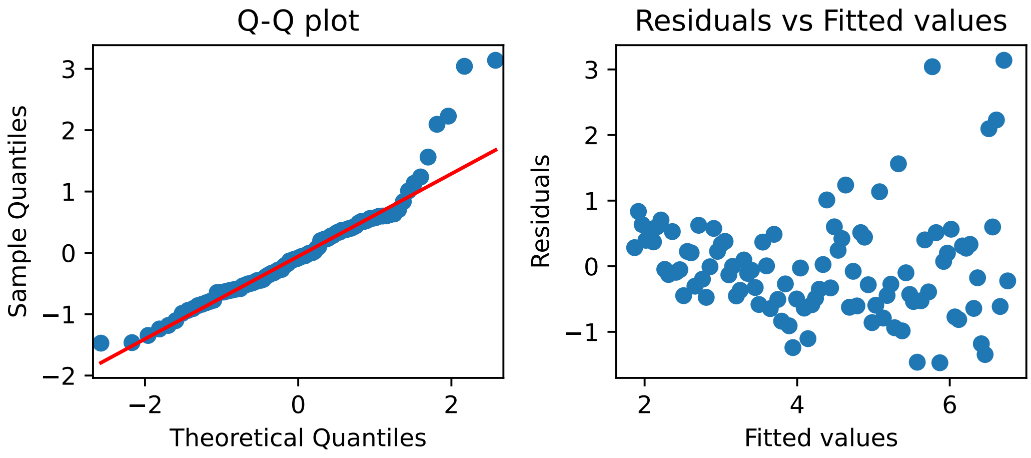
```
np.random.seed(134)
n = 100
x1 = np.linspace(1, 10, n)
x2 = x1**2
y = x1 + 0.5 * x2 + stats.norm.rvs(loc = 0, scale = 1, size=n)
D = pd.DataFrame({'x': x1, 'y': y})
fit = smf.ols(formula = 'y ~ x', data=D).fit()
```



We see some departure from normality, but also that the residuals are related to the fitted values with a clear pattern. In the next chapter we will learn that we should find the hidden dependence (x_2) and include it in the model.

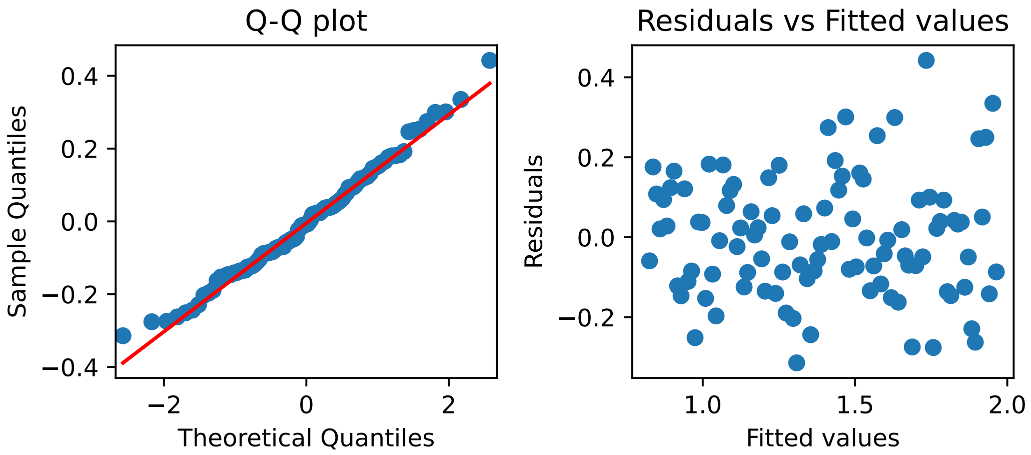
The third model (with $\beta_0 = 0$, $\beta_1 = 1$, $\beta_2 = 0.5$ and $\sigma^2 = 1$) is simulated, estimated and analysed by (plot function omitted):

```
np.random.seed(134)
n = 100
x1 = np.linspace(4, 10, n)
y = np.exp( 0.2 * x1 + np.random.normal(0, 0.15, size=n))
D = pd.DataFrame({'x': x1, 'y': y})
fit = smf.ols(formula = 'y ~ x', data=D).fit()
```



We see some departure from normality, and also that the variance increases as a function of the fitted values. When the variance is clearly related with the fitted values one should try to transform the dependent variable. The following do the analysis based in log-transformed data:

```
np.random.seed(134)
n = 100
x1 = np.linspace(4, 10, n)
y = np.exp( 0.2 * x1 + np.random.normal(0, 0.15, size=n))
y = np.log(y)
D = pd.DataFrame({'x': x1, 'y': y})
fit = smf.ols(formula = 'y ~ x', data=D).fit()
```



From the q-q plot it is found that the distribution is now very close to a normal distribution compared to previous q-q plot. And, as we can see the residuals are no longer related clearly to the fitted values.

|||| Method 5.28 Model validation (or residual analysis)

1. Check the normality assumption with a q-q plot of the residuals
2. Check the systematic behaviour by plotting the residuals e_i as a function of fitted values $\hat{y}_i$

|||| Remark 5.29 Independence

In general independence should also be checked, while there are ways to do this we will not discuss them here.

Chapter 6

Multiple Linear Regression

In Chapter 5 we described the linear regression model, when the outcome (Y) is a linear function of *one* regressor (x). It is natural to extend this model to include more than one regressor, in general we can write

$$Y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_p x_{p,i} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-1}$$

where as usual we assume that the residuals (ε_i) are independent and identically distributed (i.i.d.) normal random variables with zero mean and some unknown constant variance (σ^2). Note, that this is the assumption for all random variable error terms in models presented in this chapter, however it is not noted for every model.

The model in Equation (6-1) is referred to as the *General Linear Model* (GLM), and is closely related to the ANOVA covered in a later chapter. As we will see in Section 6.2, we can also use the approach to approximate non-linear functions of the regressors, i.e.

$$Y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2). \tag{6-2}$$

The optimal set of parameters for the multiple linear regression model is found by minimising the residual sum of squares

$$RSS(\beta_0, \ldots, \beta_p) = \sum_{i=1}^{n} \left[Y_i - (\beta_0 + \beta_1 x_{1,i} + \cdots + \beta_p x_{p,i}) \right]^2, \tag{6-3}$$

where n is the number of observations. The general problem is illustrated in Figure 6.1, where the black dots represent the observations (y_i), the blue and red lines represent errors (e_i) (the ones we minimize), and the surface represented by the grey lines is the optimal estimate (with $p = 2$)

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \hat{\beta}_2 x_{2,i}, \tag{6-4}$$

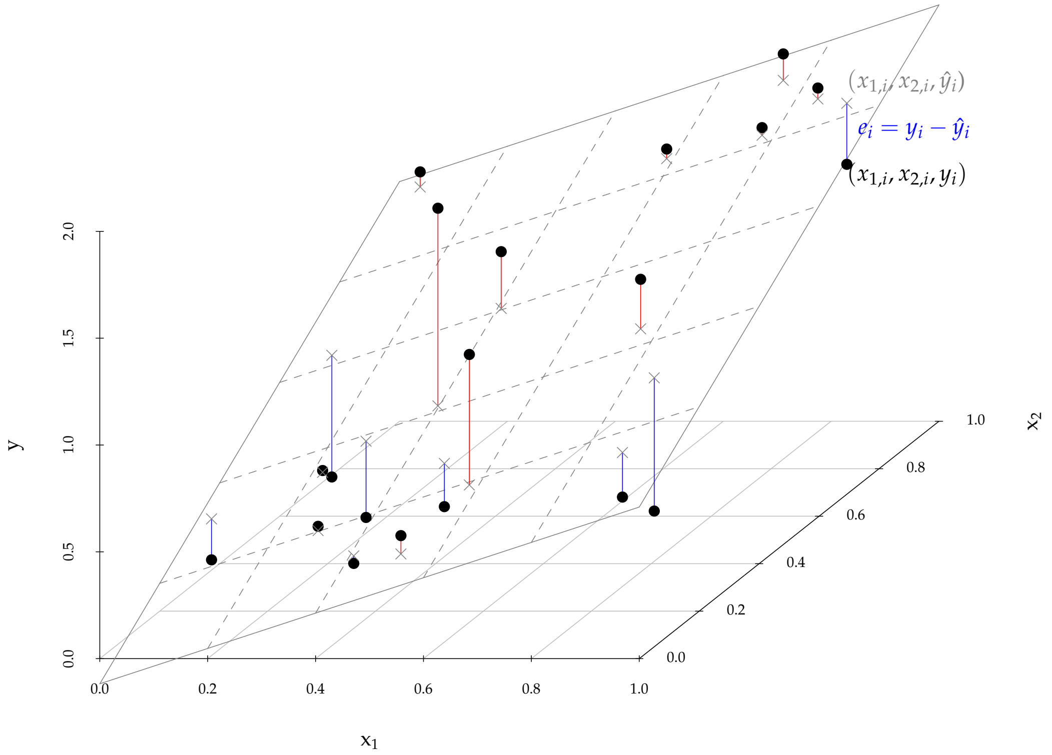


Figure 6.1: Conceptual plot for the multiple linear regression problem (red lines, $e_i > 0$, blue lines ($e_i < 0$).

or

$$y_i = \hat{y}_i + e_i, \tag{6-5}$$

again we put a "hat" on the parameters to emphasize that we are dealing with parameter estimates (or estimators), as a result of minimising Equation (6-3) with respect to $\beta_0, \ldots, \beta_p$.

Let's have a look at a small example:

Example 6.1

The car manufacture in Example 5.1 in Chapter 5 constructed a linear model for fuel consumption as a function of speed, now a residual analysis revealed that the residuals were not independent of the fitted values and therefore the model should be extended. It is realized that the fuel consumption is a function of wind speed as well as the speed of the car, and a new model could be formulated as

$$Y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \varepsilon_i \tag{6-6}$$

where $x_{1,i}$ is the speed, and $x_{2,i}$ is the wind speed (relative to the car). Another possibility is that the model should in fact not be linear in the speed, but rather quadratic

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{1,i}^2 + \varepsilon_i && (6\text{-}7)\\ &= \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \varepsilon_i, && (6\text{-}8) \end{aligned}$$

where $x_{2,i}$ is now the squared speed. Both models ((6-6) and (6-7)) are linear in the parameters $(\beta_0, \beta_1, \beta_2)$.

The example above illustrate that linearity refers to linearity in the parameters, not the regressors. E.g. the model

$$Y_i = \beta_0 + \beta_2 \log(x_i) + \varepsilon_i, \tag{6-9}$$

is a linear model, while

$$Y_i = \beta_0 + \log(x_i + \beta_2) + \varepsilon_i, \tag{6-10}$$

is not a linear model.

6.1 Parameter estimation

Just as in the case of simple linear regression the optimal parameters are the parameters that minimize the residual sum of squares (RSS), this is equivalent to equating the partial derivatives of RSS (Equation (6-3)) with zero, i.e.

$$\frac{\partial RSS}{\partial \beta_j} = 0; \quad j = 0, 1, \ldots, p, \tag{6-11}$$

which will give us $p+1$ equations (the partial derivatives) in $p+1$ unknowns (the parameters)

$$\begin{aligned} 2\sum_{i=1}^{n} \left[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \cdots + \hat{\beta}_p x_{p,i}) \right] &= 0, && (6\text{-}12)\\ 2\sum_{i=1}^{n} \left[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \cdots + \hat{\beta}_p x_{p,i}) x_{1,i} \right] &= 0, && (6\text{-}13)\\ &\vdots \\ 2\sum_{i=1}^{n} \left[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \cdots + \hat{\beta}_p x_{p,i}) x_{p,i} \right] &= 0, && (6\text{-}14) \end{aligned}$$

the Equations (6-12)-(6-14) are referred to as the normal equations, and as we can see these are a system of linear equations and thus best solved by methods of linear algebra. The matrix formulation is covered in Section 6.6, but for now we will just assume that `Python` is able to solve the normal equations and give the correct parameter estimates, standard errors for the parameter estimates, etc.

When the ε_i's are independent identically normally distributed, we can construct tests for the individual parameters, assuming we know the parameter estimates and their standard errors:

|||| Theorem 6.2 Hypothesis tests and confidence intervals

Suppose the we are given parameter estimates $(\hat{\beta}_0, \ldots, \hat{\beta}_p)$ and their corresponding standard errors $(\hat{\sigma}_{\beta_0}, \ldots, \hat{\sigma}_{\beta_p})$, then under the null hypothesis

$$H_{0,i}: \quad \beta_i = \beta_{0,i}, \tag{6-15}$$

the t-statistic

$$T_i = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\beta_i}}, \tag{6-16}$$

will follow the t-distribution with $n - (p+1)$ degrees of freedom, and hypothesis testing and confidence intervals should be based on this distribution. Further, a central estimate for the residual variance is

$$\hat{\sigma}^2 = \frac{RSS(\hat{\beta}_0, \ldots, \hat{\beta}_p)}{n - (p+1)}. \tag{6-17}$$

The interpretation of multiple linear regression in `Python` is illustrated in the following example:

|||| Example 6.3

The data used for Figure 6.1 is given in the table below

x_1	0.083	0.409	0.515	0.397	0.223	0.292	0.584	0.491	0.923	0.280	0.772	0.857	0.758	0.850	0.409	0.055	0.578	0.745	0.886	0.031
x_2	0.625	0.604	0.077	0.414	0.343	0.202	0.840	0.266	0.831	0.385	0.821	0.308	0.440	0.865	0.111	0.970	0.192	0.939	0.149	0.318
y	0.156	1.234	0.490	1.649	0.500	0.395	1.452	0.416	1.390	0.234	1.574	0.349	1.287	1.709	0.323	1.201	1.210	1.787	0.591	0.110

We assume the model

$$Y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2). \tag{6-18}$$

In order to estimate parameters we would write:

```
# Read data
data = {
’x1’ : [0.083, 0.409, 0.515, 0.397, 0.223, 0.292, 0.584, 0.491, 0.923,
        0.280, 0.772, 0.857, 0.758, 0.850, 0.409, 0.055, 0.578, 0.745,
        0.886, 0.031],
’x2’ : [0.625, 0.604, 0.077, 0.414, 0.343, 0.202, 0.840, 0.266, 0.831,
        0.385, 0.821, 0.308, 0.440, 0.865, 0.111, 0.970, 0.192, 0.939,
        0.149, 0.318],
’y’  : [0.156, 1.234, 0.490, 1.649, 0.500, 0.395, 1.452, 0.416, 1.390,
        0.234, 1.574, 0.349, 1.287, 1.709, 0.323, 1.201, 1.210, 1.787,
        0.591, 0.110]
}

df = pd.DataFrame(data)
```

```
# Parameter estimation
fit = smf.ols(formula = ’y ∼ x1 + x2’, data = df).fit()

# Summary of fit (parameter estimates, standard error, p-values, etc.)
print(fit.summary(slim=True))

                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.632
Model:                            OLS   Adj. R-squared:                  0.589
No. Observations:                  20   F-statistic:                     14.62
Covariance Type:            nonrobust   Prob (F-statistic):           0.000203
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1176      0.212     -0.556      0.586      -0.564       0.329
x1             0.8274      0.304      2.719      0.015       0.185       1.470
x2             1.2393      0.293      4.236      0.001       0.622       1.857
==============================================================================
```

The interpretation of the output is exactly the same as in the simple linear regression. The first column gives the parameter estimates $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$, second column gives the standard error of the parameter estimates $(\hat{\sigma}_{\beta_0}, \hat{\sigma}_{\beta_1}, \hat{\sigma}_{\beta_2})$, third column gives the t-statistics for the standard hypothesis $H_{0,i} : \beta_i = 0$, column four gives the p-value for the two-sided alternative, and finally columns 5-6 give 95% confidence intervals. We can therefore conclude that the effect of x_1 and x_2 are both significant on a 5% confidence level.

|||| Method 6.4 Level α t-tests for parameters

1. Formulate the *null hypothesis*: $H_{0,i} : \beta_i = \beta_{0,i}$, and the alternative hypothesis $H_{1,i} : \beta_i \neq \beta_{0,i}$
2. Compute the test statistic $t_{\text{obs},\beta_i} = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\beta_i}}$
3. Compute the evidence against the *null hypothesis*

$$p\text{-value}_i = 2P(T > |t_{\text{obs},\beta_i}|) \tag{6-19}$$

4. If the p-value$_i < \alpha$ reject $H_{0,i}$, otherwise accept $H_{0,i}$

In many situations we will be more interested in quantifying the uncertainty of the parameter estimates rather than testing a specific hypothesis. This is usually given in the form of confidence intervals for the parameters:

|||| Method 6.5 Parameter confidence intervals

$(1 - \alpha)$ confidence interval for β_i is given by

$$\hat{\beta}_i \pm t_{1-\alpha/2}\, \hat{\sigma}_{\beta_i}, \tag{6-20}$$

where $t_{1-\alpha/2}$ is the $(1 - \alpha/2)$-quantile of a t-distribution with $n - (p + 1)$ degrees of freedom.

Remark 6.6 (On finding $\hat{\beta}_i$ and $\sigma_{\hat{\beta}_i}$ in methods 6.4 and 6.5)

In Chapter 5 we were able to formulate the exact formulas for $\hat{\beta}_i$ and $\hat{\sigma}_{\hat{\beta}_i}$, in a multiple linear regression setting we simply use `Python` (`smf.ols`), to find these values.
The explicit formulas are however given in the matrix formulation of the linear regression problem in Section 6.6.

Example 6.7

For our example the 95% confidence intervals become ($t_{1-\alpha/2} = 2.110$)

$$I_{\beta_0} = -0.118 \pm 2.110 \cdot 0.212, \tag{6-21}$$

$$I_{\beta_1} = 0.827 \pm 2.110 \cdot 0.304, \tag{6-22}$$

$$I_{\beta_2} = 1.239 \pm 2.110 \cdot 0.293, \tag{6-23}$$

or using the software (for β_0):

```
# Calculations
CI = - 0.118 + np.array([-1,1]) * stats.t.ppf(0.975, df = 17) * 0.212
print(CI)

[-0.565  0.329]
```

or directly using the highlevel method (for β_0, β_1, and β_2):

```
print(fit.conf_int(alpha=0.05))

                  0         1
Intercept -0.564307  0.329042
x1         0.185371  1.469529
x2         0.621989  1.856559
```

The examples above illustrates how we can construct confidence intervals for the parameters and test hypotheses without having to implement the actual estimation ourselves.

6.1.1 Confidence and prediction intervals for the line

Just as for the simple linear regression model we will often be interested in prediction of future outcome of an experiment, and as usual we will be interested in quantifying the uncertainty of such an experiment. The expected value of a new experiment (with $x_1 = x_{1,\text{new}}, \ldots, x_p = x_{p,\text{new}}$) is

$$\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{1,\text{new}} + \ldots + \hat{\beta}_p x_{p,\text{new}}. \tag{6-24}$$

In order to quantify the uncertainty of this estimate we need to calculate the variance of $\hat{y}_{\text{new}}$, in Section 5.3 we saw that this variance is a function of: 1) the variance of the parameters, 2) the covariance between the parameters, and 3) x_{new}. This is also true in the multiple linear regression case, except that x_{new} is now a vector and we need to account for pairwise covariance between all parameter estimators. This analysis is most elegantly done with matrix formulation and is covered in Section 6.6. We can however do this using `Python` without dealing with the covariances explicitly.

This is illustrated in the following example:

|||| Example 6.8

With reference to Example 6.3 suppose we want to predict the expected value of Y at $(x_{1,\text{new}}, x_{2,\text{new}}) = (0.5, 0.5)$ and at $(x_{1,\text{new}}, x_{2,\text{new}}) = (1, 1)$, we would also like to know the standard error of the prediction and further the confidence and the prediction intervals. The standard error of the prediction can be calculated by:

```
# # New data
new_data = pd.DataFrame({'x1': [0.5, 1],'x2': [0.5, 1]})

# # Prediction and confidence interval
pred = fit.get_prediction(new_data).summary_frame(alpha=0.05)
print(round(pred,3))

    mean  mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  obs_ci_upper
0  0.916    0.085          0.737          1.095         0.098         1.734
1  1.949    0.214          1.497          2.401         1.032         2.867
```

The data-frame "`new_data`" is the points where we want to predict the outcome, the object "`pred`" has the fitted values (`mean`) at the points in "`new_data`", the standard errors for the predictions (`mean_se`), the upper and lower limits of the confidence intervals (`mean_ci_upper` and `mean_ci_lower`), and the upper and lower limits of the prediction intervals (`obs_ci_upper` and `obs_ci_lower`).

Notice that the standard error for $\hat{y}_{\text{new}}$ is much larger for the point $(x_{1,\text{new}}, x_{2,\text{new}}) = (1,1)$ than for the point $(x_{1,\text{new}}, x_{2,\text{new}}) = (0.5, 0.5)$, this is because the (1,1) point is far from the average of the regressors, while the point (0.5,0.5) is close to the average value of the regressors.

Now, we are actually able to calculate confidence and prediction intervals for the two points, the confidence intervals become

$$CI_1 = 0.9157 \pm t_{1-\alpha/2} \cdot 0.08477, \tag{6-25}$$

$$CI_2 = 1.9491 \pm t_{1-\alpha/2} \cdot 0.21426, \tag{6-26}$$

and the prediction intervals become (add the variance of $\hat{Y}_{\text{new}}$ and $\hat{\sigma}^2$)

$$PI_1 = 0.9157 \pm t_{1-\alpha/2} \cdot \sqrt{0.08477^2 + 0.3784^2}, \tag{6-27}$$

$$PI_2 = 1.9491 \pm t_{1-\alpha/2} \cdot \sqrt{0.21426^2 + 0.3784^2}, \tag{6-28}$$

where $t_{1-\alpha/2}$ is obtained from a t-distribution with 17 degrees of freedom.

The calculations in Python is exemplified for the first prediction interval below

```
p_i = 0.9157 + np.array([-1,1]) * stats.t.ppf(0.975, df=17) *
np.sqrt(0.08477**2 + 0.3784**2)
np.round(p_i, 3)

array([0.098, 1.734])
```

We saw in the example above that the standard error for the fit is large when we are far from the center of mass for the regressors, this is illustrated in Figure 6.2.

|||| Method 6.9 Intervals for the line (by Python)

The (1-α) **confidence and prediction intervals** for the line $\hat{\beta}_0 + \hat{\beta}_1 x_{1,\text{new}} + \cdots + \hat{\beta}_p x_{p,\text{new}}$ are calculated in Python by

```
# Confidence and Prediction interval
fit.get_prediction(new_data).summary_frame(alpha=alpha)
```

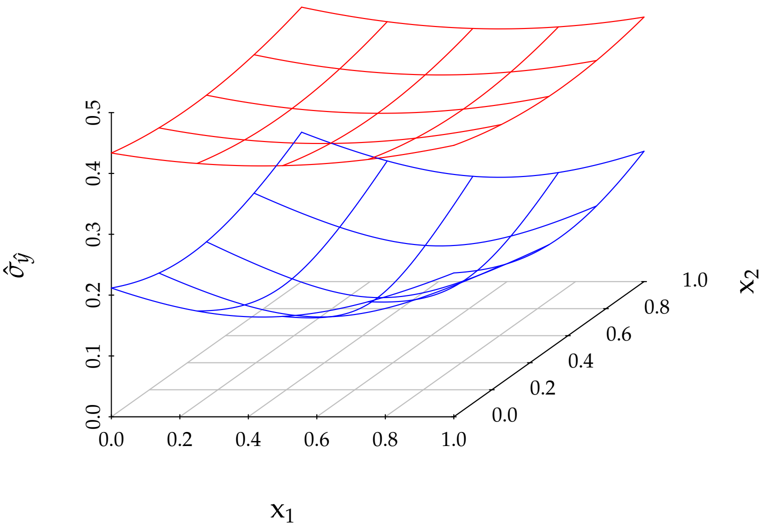


Figure 6.2: Standard error for $\hat{y}_{\text{new}}$ (blue surface) and standard error for y_{new} (red surface).

|||| **Remark 6.10**

Explicit formulas for confidence and prediction intervals are given in Section 6.6.

6.2 Curvilinear regression

Suppose we are given pairs of values of x and y and there seems to be information in x about y, but the relation is clearly non-linear

$$Y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-29}$$

and the non-linear function $f(x)$ is unknown to us. The methods we have discussed don’t apply for non-linear functions, and even if we could do non-linear regression we would not know which function to insert. We do however know from elementary calculus that any function can be approximated by its Taylor series expansion

$$f(x) \approx f(0) + f'(0) \cdot x + \frac{f''(0)}{2}x^2 + \cdots + \frac{f^{(p)}(0)}{p!}x^p, \tag{6-30}$$

now replace the Taylor series coefficients $\left(\frac{f^{(j)}(0)}{j!}\right)$ by β_j and insert (6-30) in (6-29) to get

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p + \varepsilon_i \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon_i, \end{aligned} \tag{6-31}$$

where $x_j = x^j$, we refer to this method as curvilinear regression. The method is illustrated in the following example:

|||| Example 6.11 Simulation of non-linear model

We simulate the following model

$$Y_i = \sin(\pi x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, 0.1^2), \tag{6-32}$$

with $x \in [0, 1]$ by:

```
np.random.seed(12657)
n = 200
x = np.random.uniform(size = n)
y = np.sin(np.pi * x) + np.random.normal(0, 0.1,size=n)
df_sim = pd.DataFrame({’y’: y,’x1’ : x, ’x2’ : x**2})
```

Y_i is a non-linear function of x but lets try to estimate parameters in the simple linear regression model

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-33}$$

and find the 95% confidence interval for the parameters:

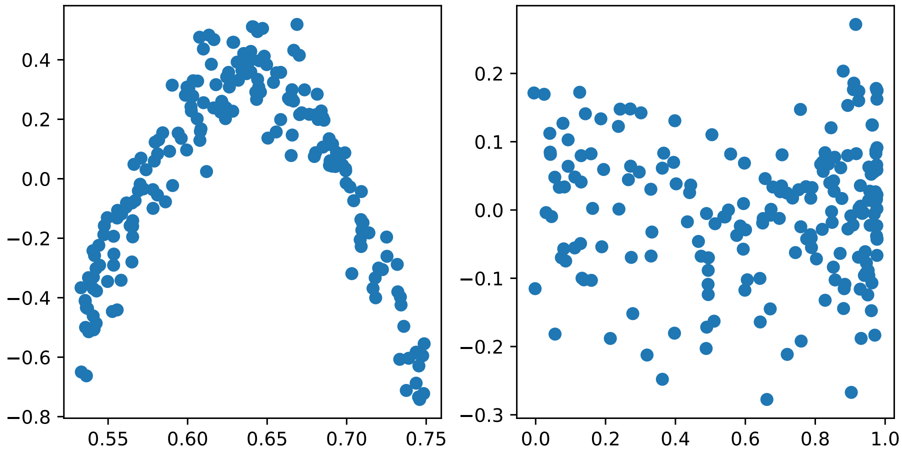
```
fit_sim = smf.ols(formula = ’y ∼ x1’, data = df_sim).fit()
print(round(fit_sim.conf_int(alpha=0.05),3))

               0      1
Intercept  0.510  0.690
x1        -0.097  0.211
```

We see that the 95% confidence interval for β_1 covers zero, and we can therefore not reject the null hypothesis that β_1 is zero. Now include a quadratic term in x_1 to approximate the non-linear function by the model

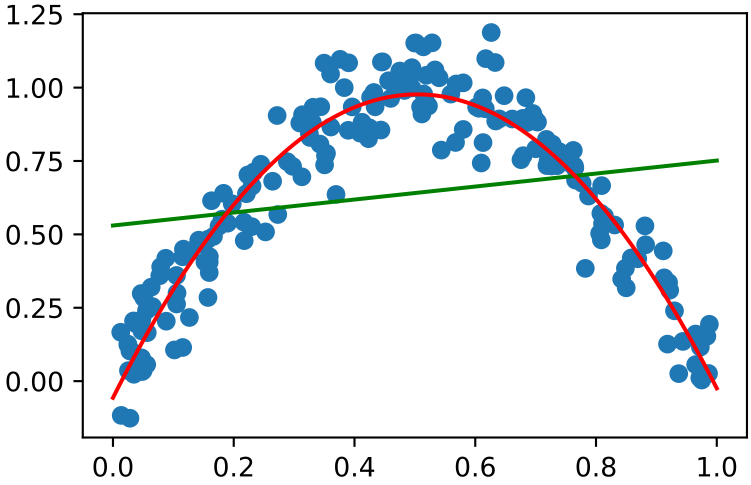
$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-34}$$

```
fit_sim2 = smf.ols(formula = ’y ∼ x1 + x2’, data = df_sim).fit()
print(round(fit_sim2.conf_int(alpha=0.05),3))

               0      1
Intercept -0.095 -0.006
x1         3.885  4.303
x2        -4.292 -3.883
```

Now we see that all parameters are significantly different from zero on a 5% confidence level. The plot below shows the residuals for the two models as a function of the fitted values:



It is clear that including the second order term removed most, if not all, systematic dependence in the residuals. Also looking at the fitted values together with the actual values shows that we have a much better model when including the second order term (red line):



|||| Remark 6.12

In general one should be careful when extrapolation models into areas where there is no data, and this is in particular true when we use curvilinear regression.

6.3 Collinearity

In statistics collinearity refers to situations where the sample correlation between the independent variables is high. If this is the case we should be careful with interpretation of parameter estimates, and often we should actually reduce the model. Now consider the model

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-35}$$

and assume that the sample correlation between x_1 and x_2 is exactly equal 1, this implies that we can write $x_2 = a + bx_1$, inserting in (6-35) gives

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_1 + \beta_2 (a + bx_1) + \varepsilon_i & \text{(6-36)} \\ &= \beta_0 + \beta_2 a + (\beta_1 + \beta_2 b) x_1 + \varepsilon_i, & \text{(6-37)} \end{aligned}$$

which shows that we can only identify $\beta_0 + \beta_2 a$ and $(\beta_1 + \beta_2 b)$, so the model is essentially a simple linear regression model. It could also have been the other way around, i.e. $x_1 = a + bx_2$, and thus it seems that it is not possible to distinguish between x_1 and x_2. In real life application the correlation between the regressors is rarely 1, but rather close to 1 and we need to handle this case as well. In actual practice a simple way to handle this is, by adding or removing one parameter at the time. Other procedures exist, e.g. using the average of the regressors, or using principle component regression, we will not discuss these approaches further here.

A small example illustrates the principle:

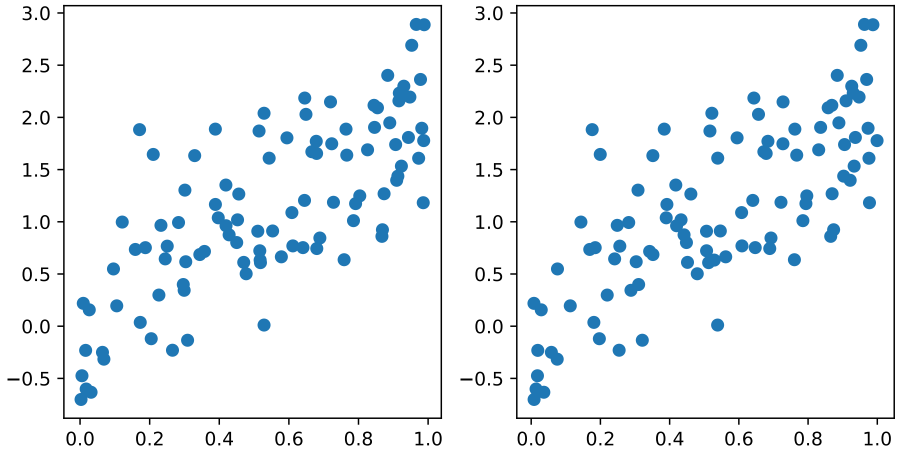
|||| Example 6.13 Simulation

Consider the model

$$Y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-38}$$

with data generated from the following code:

```
np.random.seed(200)
n = 100
x1 = np.random.uniform(size = n)
x2 = x1 + np.random.normal(0, 0.01,size=n)
y = x1 + x2 + np.random.normal(0, 0.5,size=n)
df_sim = pd.DataFrame({’y’: y,’x1’ : x1, ’x2’ : x2})

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.scatter(x1,y)
ax2.scatter(x2,y)
```



Clearly, both x_1 and x_2 contain information about y, but our usual linear regression gives:

```
fit_sim = smf.ols(formula = 'y ∼ x1 + x2', data = df_sim).fit()
print(round(fit_sim.conf_int(alpha=0.05),3))

                 0       1
Intercept  -0.197   0.247
x1        -14.847  10.061
x2         -8.057  16.898
```

we see that none of the parameters are significant (on a 5% level), but if we remove x_1 (this is the one with the highest p-value) from the model we get:

```
fit_sim2 = smf.ols(formula = 'y ∼ x2', data = df_sim).fit()
print(fit_sim2.summary(slim=True))

                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.567
Model:                            OLS   Adj. R-squared:                  0.562
No. Observations:                 100   F-statistic:                     128.2
Covariance Type:            nonrobust   Prob (F-statistic):           1.69e-19
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0283      0.111      0.255      0.799      -0.192       0.249
x2             2.0240      0.179     11.322      0.000       1.669       2.379
==============================================================================
```

and the slope is now highly significant.

The lesson learned from the example above is that we should always try to reduce the model before concluding that individual parameters are zero. Model development is a partly manual process, where the end result might depend on the selection strategy. The usual strategies are: *backward selection*, where we start by the most complicated model we can think of and remove one term at a time (this is what we did in the example above), and *forward selection* where we start by a simple model and include new terms one by one.

|||| Remark 6.14 Interpretation of parameters

In general we can interpret the parameters of a multiple linear regression model as the effect of the variable given the other variables. E.g. β_j is the effect of x_j when we have accounted for other effects (x_i, $i \neq j$). This interpretation is however problematic when we have strong collinearity, because the true effects are hidden by the correlation.

An additional comment on the interpretation of parameters in the example above is: since the data is simulated, we know that the true parameters are $\beta_1 = \beta_2 = 1$. In the full model we got $\hat{\beta}_1 \approx$-2.40 and $\hat{\beta}_2 \approx 4.42$. Both of these numbers are clearly completely off, the net effect is however $\hat{\beta}_1 + \hat{\beta}_2 \approx$2.02 (because $x_1 \approx x_2$). In the reduced model we got $\hat{\beta}_2 =$2.02, which is of course also wrong, but nearly the same level, and only holds because $x_1 \approx x_2$.

6.4 Residual analysis

Just as for the simple linear regression model we will need to justify that the assumptions in the linear regression model holds. This is handled by q-q plots, and considering the relation between the residuals and the fitted values. This analysis is exactly the same as for the simple linear regression in Section 5.7.

We saw that plotting the residuals as a function of fitted values could reveal systematic dependence, which imply there are un-modelled effects that should be included in the model. The question is of course how we can identify such effects. One way is to plot the residuals as a function of potential regressors, which are not included. Plotting the residuals as a function of the included regressors might reveal non-linear effects. Again we illustrate this method by an example:

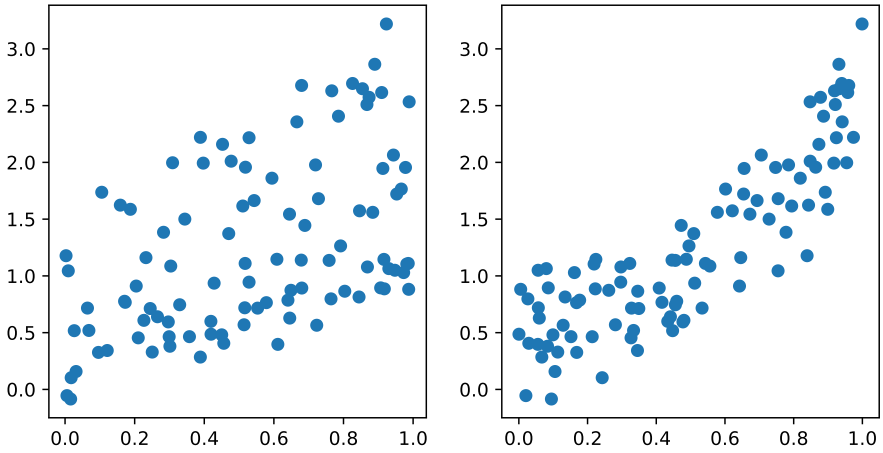
|||| Example 6.15 Residuals analysis

Consider the model in the `Python` script below, the true model is

$$y_i = x_1 + 2x_2^2 + \varepsilon_i, \quad \varepsilon_i \sim N(0, 0.125^2) \tag{6-39}$$

in a real application the true model is of course hidden to us and we would start by a multiple linear model with the two effects x_1 and x_2. Looking at the plots below also suggests that this might be a good model:

```
np.random.seed(200)
n = 100
x1 = np.random.uniform(size = n)
x2 = np.random.uniform(size = n)
y = x1 + 2*x2**2 + + np.random.normal(0, 0.125,size=n)
df_sim = pd.DataFrame({’y’: y,’x1’ : x1, ’x2’ : x2})

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8,4))
ax1.scatter(x1,y)
ax2.scatter(x2,y)
```

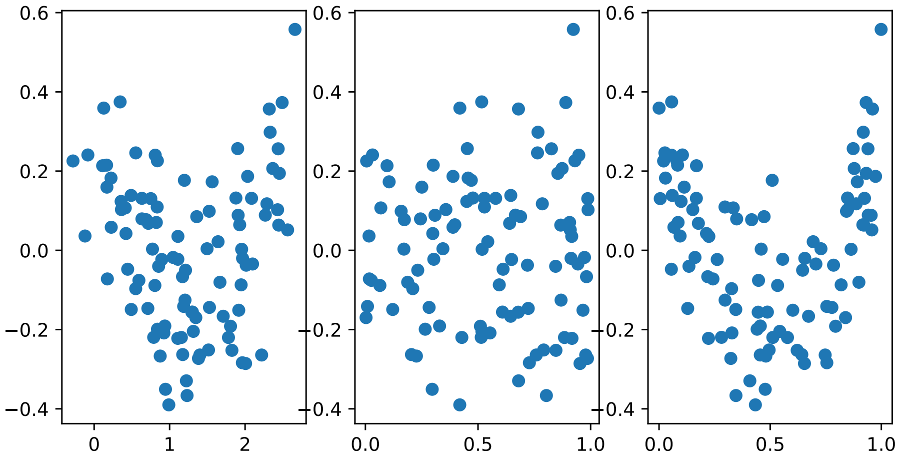


Now we fit the model

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \tag{6-40}$$

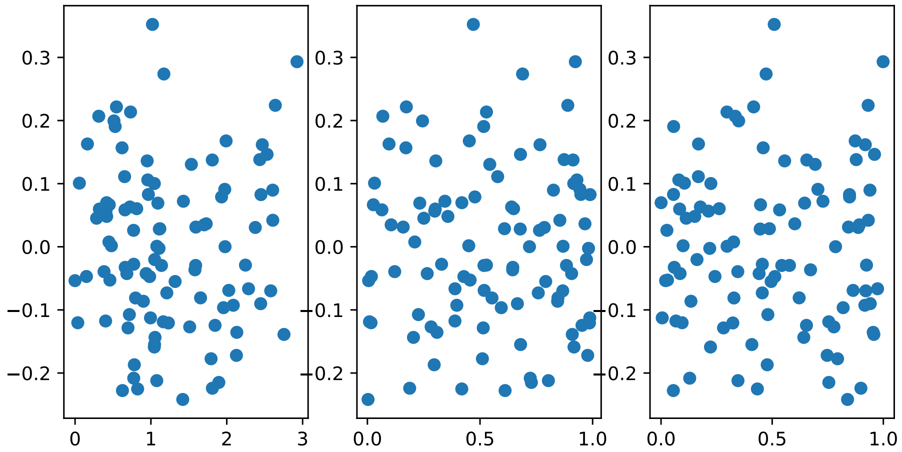
and plot the resulting residuals as a function of the fitted values, and the independent variables (x_1 and x_2). There seems to be a systematic dependence between the fitted values and the residuals (left plot):

```
fit_sim = smf.ols(formula = ’y ∼ x1 + x2’, data = df_sim).fit()

res = y - fit_sim.fittedvalues
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(8,4))

ax1.scatter(fit_sim.fittedvalues,res)
ax2.scatter(x1,res)
ax3.scatter(x2,res)
```



The left plot does however not suggest where the dependence comes from. Now looking at the residuals as a function of x_1 and x_2 (centre and left plot) reveal that the residuals seem to be quadratic in x_2, and we should therefore include x_2^2 in the model:

```
x3 = x2**2
df_sim = pd.DataFrame({’y’: y,’x1’ : x1, ’x2’ : x2,’x3’: x3})
fit_sim = smf.ols(formula = ’y ∼ x1 + x2 +x3’, data = df_sim).fit()

res = y - fit_sim.fittedvalues

fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(8,4))
ax1.scatter(fit_sim.fittedvalues,res)
ax2.scatter(x1,res)
ax3.scatter(x2,res)
```



We now see that there is no systematic dependence in the residuals and we can report the final result.

```
print(fit_sim.summary(slim=True))

                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.971
Model:                            OLS   Adj. R-squared:                  0.970
No. Observations:                 100   F-statistic:                     1057.
Covariance Type:            nonrobust   Prob (F-statistic):           2.33e-73
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0100      0.044     -0.230      0.819      -0.097       0.077
x1             1.0166      0.044     23.150      0.000       0.929       1.104
x2             0.1342      0.173      0.774      0.441      -0.210       0.478
x3             1.8668      0.169     11.056      0.000       1.532       2.202
==============================================================================
```

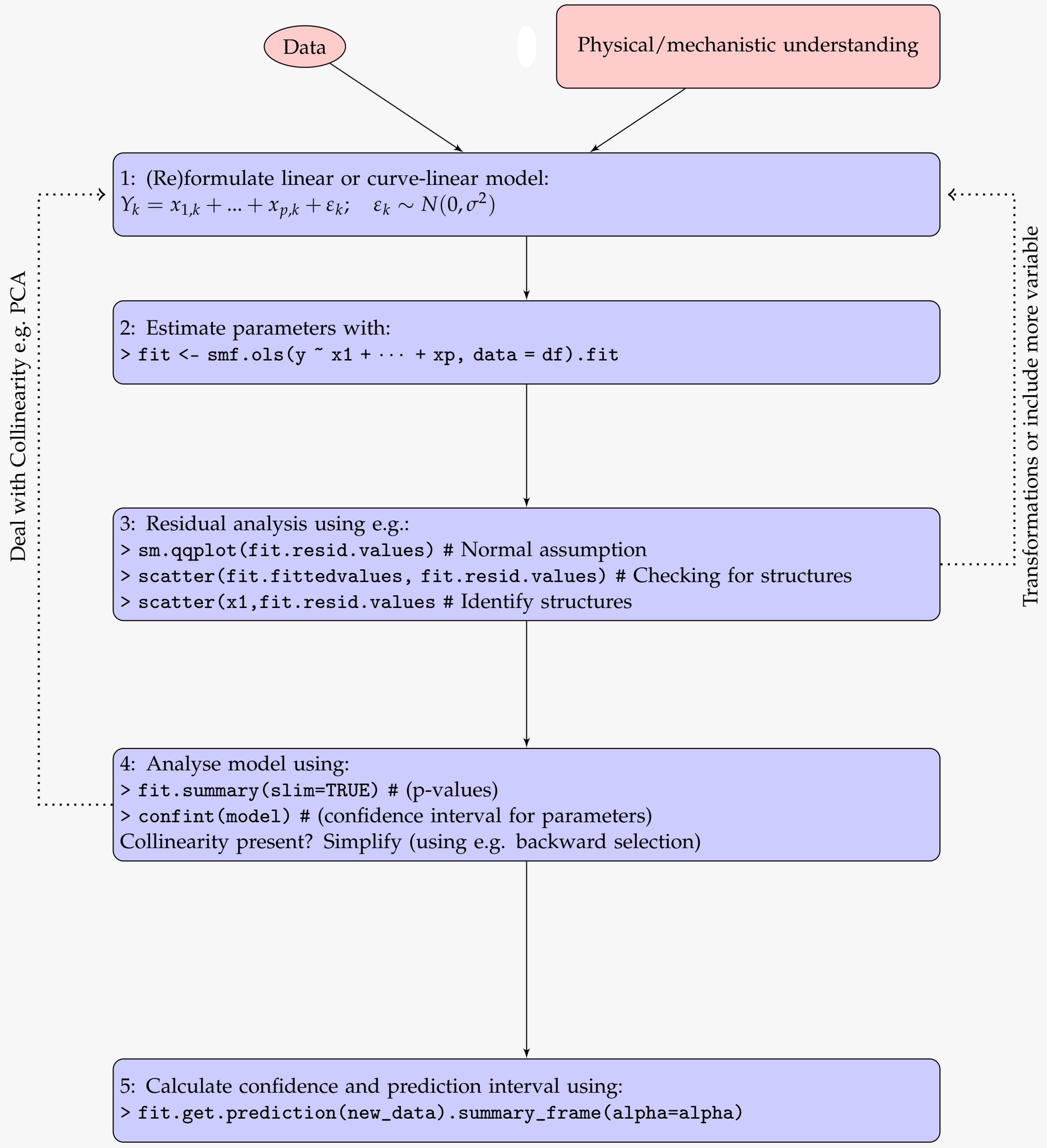
Now we can actually see that we find parameter values close to the true ones, further the slope related to x_2 and the intercept is not significant, however usually when x_2^2 have a significant parameter we would also keep x_2 in the model, the same comment apply to the intercept, that we would usually always include in the model.

6.5 Linear regression in Python

Method 6.16 below gives a practical summary of Chapter 5 and 6 with references to the applied R-functions.

|||| Method 6.16

This method box is a very short guide to Python and linear regression.



Data
Physical/mechanistic understanding
Deal with Collinearity e.g. PCA
Transformations or include more variable

1: (Re)formulate linear or curve-linear model:
$Y_k = x_{1,k} + ... + x_{p,k} + \varepsilon_k; \quad \varepsilon_k \sim N(0, \sigma^2)$

2: Estimate parameters with:

```
> fit <- smf.ols(y ˜ x1 + ··· + xp, data = df).fit
```

3: Residual analysis using e.g.:

```
> sm.qqplot(fit.resid.values) # Normal assumption
> scatter(fit.fittedvalues, fit.resid.values) # Checking for structures
> scatter(x1,fit.resid.values # Identify structures
```

4: Analyse model using:

```
> fit.summary(slim=TRUE) # (p-values)
> confint(model) # (confidence interval for parameters)
```

Collinearity present? Simplify (using e.g. backward selection)

5: Calculate confidence and prediction interval using:

```
> fit.get.prediction(new_data).summary_frame(alpha=alpha)
```

6.6 Matrix formulation

The multiple linear regression problem can be formulated in vector-matrix notation as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \tag{6-41}$$

or

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{p,1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{1,n} & \cdots & x_{p,n} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \qquad \varepsilon_i \sim N(0, \sigma^2). \tag{6-42}$$

Notice, that the formulation in (6-41) is exactly the same as we saw in Section 5.5.

The residual sum of squares are calculated by

$$RSS = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \tag{6-43}$$

and the parameter estimates are given by:

|||| Theorem 6.17

The estimators of the parameters in the simple linear regression model are given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}, \tag{6-44}$$

and the covariance matrix of the estimates is

$$V[\hat{\boldsymbol{\beta}}] = \sigma^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1}, \tag{6-45}$$

and central estimate for the residual variance is

$$\hat{\sigma}^2 = \frac{RSS}{n - (p+1)}. \tag{6-46}$$

The proof of this theorem follows the exact same arguments as the matrix formulation of the simple linear regression model in Chapter 5 and hence it is omitted here.

Marginal tests ($H_0 : \beta_i = \beta_{i,0}$) can also in the multiple linear regression case be constructed by

$$\frac{\hat{\beta}_i - \beta_{i,0}}{\sqrt{(\hat{\mathbf{\Sigma}}_\beta)_{ii}}} \sim t(n-(p+1)). \tag{6-47}$$

6.6.1 Confidence and prediction intervals for the line

Now suppose that we want to make a prediction at a new point

$$\boldsymbol{x}_{\text{new}} = [1, x_{1,\text{new}}, \ldots, x_{p,\text{new}}],$$

in order to construct confidence and prediction intervals we calculate the variance of $\hat{Y}_{\text{new}}$

$$\begin{aligned} \mathrm{V}(\hat{Y}_{\text{new}}) &= \mathrm{V}(\boldsymbol{x}_{\text{new}}\hat{\boldsymbol{\beta}}) \\ &= \boldsymbol{x}_{\text{new}}\,\mathrm{V}(\hat{\boldsymbol{\beta}})\boldsymbol{x}_{\text{new}}^T \\ &= \sigma^2\boldsymbol{x}_{\text{new}}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_{\text{new}}^T, \end{aligned} \tag{6-48}$$

in practice we will of course replace σ^2 with its estimate ($\hat{\sigma}^2$), and hence use quantile of the appropriate t-distribution (and standard errors rather than variances) to calculate confidence intervals. The variance of a single prediction is calculated by

$$\begin{aligned} \mathrm{V}(Y_{\text{new}}) &= \mathrm{V}(\boldsymbol{x}_{\text{new}}\hat{\boldsymbol{\beta}} + \varepsilon_{\text{new}}) \\ &= \boldsymbol{x}_{\text{new}}\,\mathrm{V}(\hat{\boldsymbol{\beta}})\boldsymbol{x}_{\text{new}}^T + \sigma^2 \\ &= \sigma^2(1 + \boldsymbol{x}_{\text{new}}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_{\text{new}}^T). \end{aligned} \tag{6-49}$$

The calculations above illustrate that the derivations of variances are relatively simple, when we formulate our model in the matrix-vector notation.

```python
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.stats.proportion as smprop
```

Chapter 7

Inference for Proportions

7.1 Categorical data

Until now we have mainly focused on continuous outcomes such as the height of students. In many applications the outcome that we wish to study is categorical (7.1). For example, one could want to study the *proportion* of defective components in a sample, hence the outcome has two categories: "defect" and "non-defect". Another example could be a study of the caffeine consumption among different groups of university students, where the consumption could be measured via a questionnaire in levels: none, 1-3 cups per day, more than 3 cups per day. Hence the categorical variable describing the outcome has three categories.

In both examples the key is to describe the *proportion* of outcomes in each category.

Remark 7.1

A variable is categorical if each outcome belongs to a category, which is one of a set of categories.

7.2 Estimation of single proportions

We want to be able to find estimates of the population category proportions (i.e. the "true" proportions). We sometimes refer to such a proportion as the probability of belonging to the category. This is simply because the probability that a randomly sampled observation from the population belongs to the category, is the proportion of the category in the population.

|||| Example 7.2

In a survey in the US in 2000, 1154 people answered the question whether they would be willing to pay more for petrol to help the environment. Of the 1154 participants 518 answered that they would be willing to do so.

Our best estimate of the proportion of people willing to pay more (p) is the observed proportion of positive answers

$$\hat{p} = \frac{\text{"Number of positive answers"}}{\text{"Total number of participants"}} = \frac{518}{1154} = 0.4489.$$

This means that our best estimate of the proportion of people willing to pay more for petrol to help the environment is 44.89%.

In the above example we can think of $n = 1154$ trials, where we each time have a binary outcome (yes or no), occurring with the unknown probability p. The random variable X counts the number of times we get a yes to the question, hence X follows a binomial distribution $B(n, p)$ with the probability of observing x successes given by

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}. \tag{7-1}$$

As mentioned in Example 7.2, our best estimate of the unknown p is the proportion

$$\hat{p} = \frac{x}{n}, \quad \hat{p} \in [0, 1]. \tag{7-2}$$

From Chapter 2 we know that if $X \sim B(n, p)$, then

$$\mathrm{E}(X) = np, \tag{7-3}$$
$$\mathrm{V}(X) = np(1-p). \tag{7-4}$$

This means that

$$\mathrm{E}(\hat{p}) = \mathrm{E}\left(\frac{X}{n}\right) = \frac{np}{n} = p, \tag{7-5}$$
$$\mathrm{V}(\hat{p}) = \mathrm{V}\left(\frac{X}{n}\right) = \frac{1}{n^2}\mathrm{V}(X) = \frac{p(1-p)}{n}. \tag{7-6}$$

From Equation (7-5) we see that $\hat{p}$ is an unbiased estimator of the unknown p and from Equation (7-6) that the standard error (the (sampling) standard deviation) of $\hat{p}$ is $\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$. It is important to quantify the uncertainty of the calculated estimate using confidence intervals. For large samples, the Central Limit Theorem gives us that the sample proportion $\hat{p}$ is well approximated by a normal distribution, and thus a $(1-\alpha)100\%$ confidence interval for the population proportion p is

$$\hat{p} \pm z_{1-\alpha/2}\,\sigma_{\hat{p}}. \tag{7-7}$$

However, $\sigma_{\hat{p}}$ depends on the unknown p, which we do not know. In practice we will have to estimate the standard error by substituting the unknown p by the estimate $\hat{p}$.

|||| Method 7.3 Proportion estimate and confidence interval

The best estimate of the probability p of belonging to a category (the population proportion) is the sample proportion

$$\hat{p} = \frac{x}{n}, \tag{7-8}$$

where x is the number of observations in the category and n is the total number of observations.

A large sample $(1-\alpha)100\%$ confidence interval for p is given as

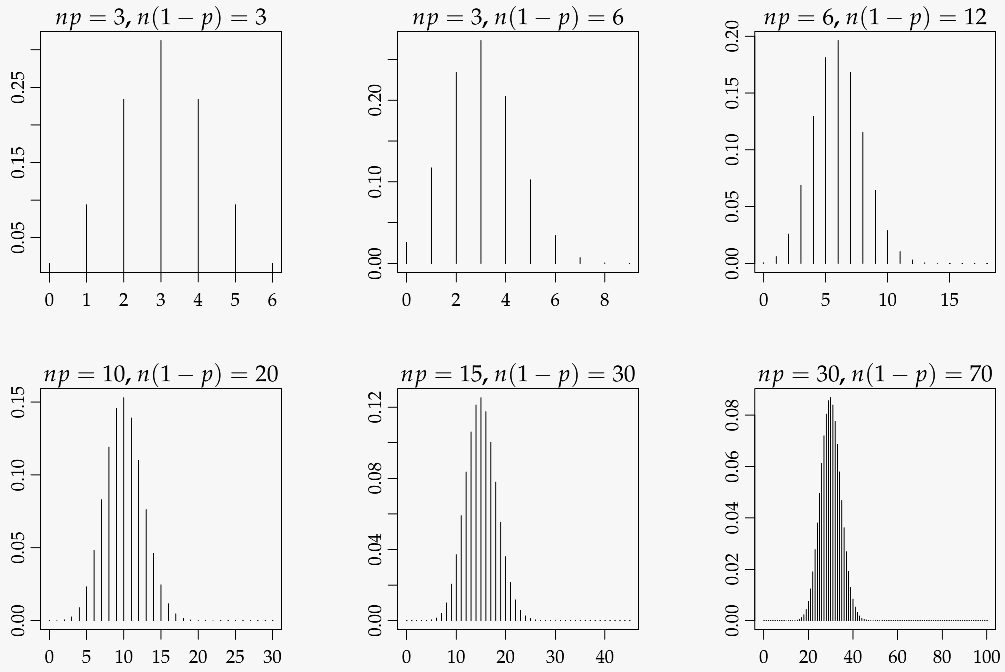
$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}. \tag{7-9}$$

|||| Remark 7.4

As a rule of thumb the normal distribution is a good approximation of the binomial distribution if np and $n(1-p)$ are both greater than 15.

|||| Example 7.5

In the figure below we have some examples of binomial distributions. When we reach a size where $np \geq 15$ and $n(1-p) \geq 15$ it seems reasonable that the bell-shaped normal distribution will be a good approximation.



|||| Example 7.6

If we return to the survey in Example 7.2, we can now calculate the 95% confidence interval for the probability (i.e. the proportion willing to pay more for petrol to help the environment).

We found the estimate of p by the observed proportion to $\hat{p} = \frac{518}{1154} = 0.45$. The standard error of the proportion estimate is

$$\hat{\sigma}_{\hat{p}} = \sqrt{\hat{p}(1-\hat{p})/n} = \sqrt{0.45 \cdot 0.55/1154} = 0.0146.$$

Since we have $n\hat{p} = 1154 \cdot 0.45 = 519.3$ and $n(1-\hat{p}) = 1154 \cdot 0.55 = 634.7$, both greater than 15, we can use the expression from Method 7.3 to get the 95% confidence interval

$$\hat{p} \pm 1.96 \cdot \hat{\sigma}_{\hat{p}} = 0.45 \pm 1.96 \cdot 0.0146 = [0.42, 0.48].$$

From this we can now conclude that our best estimate of the proportion willing to pay more for petrol to protect the environment is 0.45, and that the true proportion with 95% certainty is between 0.42 and 0.48. We see that 0.5 is not included in the confidence interval, hence we can conclude that the proportion willing to pay more for petrol is less than 0.5 (using the usual $\alpha = 0.05$ significance level). We will cover hypothesis testing for proportions more formally below.

|||| Remark 7.7 What about small samples then?

There exist several ways of expressing a valid confidence interval for p in small sample cases, that is, when either $np \leq 15$ or $n(1-p) \leq 15$. We mention three of these here - only for the last one we give the explicit formula:

Continuity correction

The so-called *continuity correction* is a general approach to making the best approximation of discrete probabilities (in this case the binomial probabilities) using a continuous distribution, (in this case the normal distribution). We do not give any details here.

Exact intervals

Probably the most well known of such small sample ways of obtaining a valid confidence interval for a proportion is the so-called *exact* method based on actual binomial probabilities rather than a normal approximation. It is not possible to give a simple formula for these confidence limits, and we will not explain the details here, but simply note that they can be obtained by the Python function `stats.binomtest`. These will be valid no matter the size of n and p.

"Plus 2"-approach

Finally, a simple approach to a good small sample confidence interval for a proportion, will be to us the simple formula given above in Method 7.3, but applied to $\tilde{x} = x + 2$ and $\tilde{n} = n + 4$.

|||| Remark 7.8 Confidence intervals for single proportions in Python

In Python we can either use the function `smprop.proportions_ztest` or `stats.binomtest` to find the confidence interval of a single proportion (and some hypothesis testing information to be described below).

The `stats.binomtest` function uses the exact approach. The `smprop.proportions_ztest` does not use continuity correction, but assumes normality.
Therefore: none of these intervals calculated by Python coincides exactly with the formula given in Method 7.3, neither applied to x and n nor applied to $\tilde{x} = x + 2$ and $\tilde{n} = n + 4$. And vice versa: the exact computational details of the different intervals calculated by Python are not given in the text here.

7.2.1 Testing hypotheses

Hypothesis testing for a single proportion (or probability) p is presented in this section.

The first step is to formulate the null hypothesis and the alternative as well as choosing the level of significance α. The null hypothesis for a proportion has the form

$$H_0 : p = p_0 \tag{7-10}$$

where p_0 is a chosen value between 0 and 1. In Example 7.2, we could be interested in testing whether half of the population, from which the sample was taken, would be willing to pay more for petrol, hence $p_0 = 0.5$.

The alternative hypothesis is the two-sided alternative

$$H_1 : p \neq p_0. \tag{7-11}$$

|||| **Remark 7.9**

As for the t-tests presented in Chapter 3, we can also have one-sided tests for proportions, i.e. the "less than" alternative

$$H_0 : p \geq p_0 \tag{7-12}$$
$$H_1 : p < p_0, \tag{7-13}$$

and the "greater than" alternative

$$H_0 : p \leq p_0 \tag{7-14}$$
$$H_1 : p > p_0, \tag{7-15}$$

however these are not included further in the material, see the discussion in Section 3.1.7 (from page 145 in the book), which applies similarly here.

The next step is to calculate a test statistic as a measure of how well our data fits the null hypothesis. The test statistic measures how far our estimate $\hat{p}$ is from the value p_0 relative to the uncertainty – under the scenario that H_0 is true.

So, under H_0 the true proportion is p_0 and the standard error is $\sqrt{p_0(1-p_0)/n}$, thus to measure the distance between $\hat{p}$ and p_0 in standard deviations we calculate the test statistic

$$z_{\text{obs}} = \frac{x - np_0}{\sqrt{np_0(1-p_0)}}. \tag{7-16}$$

When H_0 is true, the test statistic seen as a random variable is

$$Z = \frac{\hat{p} - p_0}{\sqrt{p_0(1-p_0)/n}} = \frac{X - np_0}{\sqrt{np_0(1-p_0)}}, \tag{7-17}$$

and follows approximately a standard normal distribution $Z \sim N(0,1)$, when n is large enough:

|||| Theorem 7.10

In the large sample case the random variable Z follows approximately a standard normal distribution

$$Z = \frac{X - np_0}{\sqrt{np_0(1-p_0)}} \sim N(0,1), \tag{7-18}$$

when the null hypothesis is true. As a rule of thumb, the result will be valid when both $np_0 > 15$ and $n(1-p_0) > 15$.

We can use this to make the obvious explicit method for the hypothesis test:

|||| Method 7.11 One sample proportion hypothesis test

1. Compute the test statistic using Equation (7-16)

$$z_{\text{obs}} = \frac{x - np_0}{\sqrt{np_0(1-p_0)}}$$

2. Compute evidence against the *null hypothesis*

$$H_0 : p = p_0, \tag{7-19}$$

vs. the *the alternative hypothesis*

$$H_1 : p \neq p_0, \tag{7-20}$$

by the

$$p\text{-value} = 2 \cdot P(Z > |z_{\text{obs}}|). \tag{7-21}$$

where the standard normal distribution $Z \sim N(0, 1^2)$ is used

3. If the p-value $< \alpha$ we reject H_0, otherwise we accept H_0,

or

The rejection/acceptance conclusion can equivalently be based on the critical value(s) $\pm z_{1-\alpha/2}$:
if $|z_{\text{obs}}| > z_{1-\alpha/2}$ we reject H_0, otherwise we accept H_0

|||| Example 7.12

To conclude Example 7.2 we want to test the null hypothesis

$$H_0 : p = 0.5,$$

against the alternative

$$H_1 : p \neq 0.5.$$

We have chosen $\alpha = 0.05$, hence the critical value is the 0.975 quantile in the standard normal distribution $z_{1-\alpha/2} = 1.96$. Thus we get the observed value of the test statistic by

$$z_{\text{obs}} = \frac{518 - 577}{\sqrt{1154 \cdot 0.5 \cdot (1 - 0.5)}} = -3.47.$$

Since $z = -3.47 < -1.96$ then we reject H_0. The p-value is calculated as the probability of observing z_{obs} or more extreme under the null hypothesis

$$2 \cdot P(Z \geq 3.47) = 0.0005.$$

We can get this directly using Python:

```
# Testing the probability = 0.5 with a two-sided alternative
# We have observed 518 out of 1154
# Do it without continuity corrections
z_obs,p_value = smprop.proportions_ztest(518, 1154, value=0.5,
prop_var=0.5)
print(z_obs)

-3.473594375515837


print(p_value)

0.0005135367279608199
```

Note that the results are exactly the same as when calculated by hand even though the test statistic used is actually $Z^2 \sim \chi^2$ with one degree of freedom, since this is the same as saying $Z \sim N(0,1)$. This is explained in detail later in the chapter.

7.2.2 Sample size determination

Before conducting a study, it is important to consider the sample size needed to achieve a wanted precision. In the case with a single probability to estimate, we see that the error we make when using the estimator $\hat{p} = \frac{x}{n}$ is given by $\left|\frac{x}{n} - p\right|$. Using the normal approximation (see Theorem 7.3) we can conclude that the error will be bounded by

$$\left|\frac{x}{n} - p\right| < z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}}, \tag{7-22}$$

with probability $1-\alpha$. Thus the *Margin of Error* (ME) of the estimate becomes

$$ME = z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}}. \tag{7-23}$$

Similar to the method given for quantitative data in Method 3.63, we can use

Equation (7-23) to determine the needed sample size in a single proportions setup. Solving for n we get:

|||| Method 7.13 Sample size formula for the CI of a proportion

Given some "guess" (scenario) of the size of the unknown p, and given some requirement to the ME-value (required expected precision) the necessary sample size is then

$$n = p(1-p)\left(\frac{z_{1-\alpha/2}}{ME}\right)^2. \tag{7-24}$$

If p is unknown, a worst case scenario with $p = 1/2$ is applied and necessary sample size is

$$n = \frac{1}{4}\left(\frac{z_{1-\alpha/2}}{ME}\right)^2. \tag{7-25}$$

The expression in Equation (7-25) for n when no information about p is available is due to the fact that $p(1-p)$ is largest for $p = 1/2$, so the required sample size will be largest when $p = 1/2$.

Method 7.13 can be used to calculate the sample size for a given choice of ME.

7.3 Comparing proportions in two populations

For categorical variables we sometimes want to compare the proportions in two populations (groups). Let p_1 denote the proportion in group 1 and p_2 the proportion in group 2. We will compare the groups by looking at the difference in proportions $p_1 - p_2$, which is estimated by $\hat{p}_1 - \hat{p}_2$.

|||| Example 7.14

In a study in the US (1975) the relation between intake of contraceptive pills (birth control pills) and the risk of blood clot in the heart was investigated. The following data were collected from a participating hospital:

	Contraceptive pill	No pill
Blood clot	23	35
No blood clot	34	132
Total	57	167

We have a binary outcome blood clot (yes or no) and two groups (pill or no pill). As in Section 7.2 we find that the best estimates of the unknown probabilities are the observed proportions

$$\hat{p}_1 = \frac{\text{"Number of blood clots in the pill group"}}{\text{"Number of women in the pill group"}} = \frac{23}{57} = 0.4035, \tag{7-26}$$

$$\hat{p}_2 = \frac{\text{"Number of blood clots in the no pill group"}}{\text{"Number of women in the no pill group"}} = \frac{35}{167} = 0.2096. \tag{7-27}$$

The difference in probabilities is estimated to be

$$\hat{p}_1 - \hat{p}_2 = 0.4035 - 0.2096 = 0.1939. \tag{7-28}$$

Thus the observed probability of getting a blood clot, was 0.1939 higher in the contraceptive pill group than in the no pill group.

We have the estimate $\hat{p}_1 - \hat{p}_2$ of the difference in probabilities $p_1 - p_2$ and the uncertainty of this estimate can be calculated by:

|||| Method 7.15

An estimate of the standard error of the estimator $\hat{p}_1 - \hat{p}_2$ is

$$\hat{\sigma}_{\hat{p}_1-\hat{p}_2} = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}. \tag{7-29}$$

The $(1-\alpha)100\%$ confidence interval for the difference $p_1 - p_2$ is

$$(\hat{p}_1 - \hat{p}_2) \pm z_{1-\alpha/2} \cdot \hat{\sigma}_{\hat{p}_1-\hat{p}_2}. \tag{7-30}$$

This confidence interval requires independent random samples for the two groups and large enough sample sizes n_1 and n_2. A rule of thumb is that $n_i p_i \geq 10$ and $n_i(1-p_i) \geq 10$ for $i = 1, 2$, must be satisfied.

|||| Remark 7.16

The standard error in Method 7.15 can be calculated by

$$\mathrm{V}(\hat{p}_1 - \hat{p}_2) = \mathrm{V}(\hat{p}_1) + \mathrm{V}(\hat{p}_2) = \hat{\sigma}^2_{\hat{p}_1} + \hat{\sigma}^2_{\hat{p}_2}, \tag{7-31}$$

$$\hat{\sigma}_{\hat{p}_1 - \hat{p}_2} = \sqrt{\mathrm{V}(\hat{p}_1 - \hat{p}_2)} = \sqrt{\hat{\sigma}^2_{\hat{p}_1} + \hat{\sigma}^2_{\hat{p}_2}}. \tag{7-32}$$

Notice, that the standard errors are added (before the square root) such that the standard error of the difference is larger than the standard error for the observed proportions alone. Therefore in practice the estimate of the difference $\hat{p}_1 - \hat{p}_2$ will often be further from the true difference $p_1 - p_2$ than $\hat{p}_1$ will be from p_1 or $\hat{p}_2$ will be from p_2.

|||| Example 7.17

Returning to Example 7.14 where we found the estimated difference in probability to be

$$\hat{p}_1 - \hat{p}_2 = 0.4035 - 0.2096 = 0.1939. \tag{7-33}$$

The estimated standard error of the estimate is

$$\hat{\sigma}_{\hat{p}_1 - \hat{p}_2} = \sqrt{\frac{0.4035(1 - 0.4035)}{57} + \frac{0.2096(1 - 0.2096)}{167}} = 0.0722. \tag{7-34}$$

A 99% confidence interval for this difference is then

$$(\hat{p}_1 - \hat{p}_2) \pm z_{0.995} \cdot \hat{\sigma}_{\hat{p}_1 - \hat{p}_2} = 0.1939 \pm 2.5758 \cdot 0.0722 = [0.0079, 0.3799]. \tag{7-35}$$

Hence our best estimate of the difference is 0.19 and with very high confidence the true difference is between 0.008 and 0.38.

We find that 0 is not included in the confidence interval, so 0 is not a plausible value for the difference $p_1 - p_2$. The values in the confidence interval are all positive and therefore we can conclude that $(p_1 - p_2) > 0$, that is $p_1 > p_2$, i.e. the probability of blood clot is larger in the contraceptive pill group than in the no pill group.

We can also compare the two proportions p_1 and p_2 using a hypothesis test. As in Method 7.11, there are four steps when we want to carry out the test. The first step is to formulate the hypothesis and the alternative.

The null hypothesis is $H_0 : p_1 = p_2$ and we will denote the common proportion p, and choose a two-sided alternative $H_1 : p_1 \neq p_2$.

In the second step we calculate a test statistic measuring how far $\hat{p}_1 - \hat{p}_2$ falls from 0, which is the value of $p_1 - p_2$ under H_0.

Under H_0, we only have one proportion p (since $p_1 = p_2 = p$). The best estimator for this common proportion is the overall observed proportion

$$\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}. \tag{7-36}$$

When the two sample sizes n_1 and n_2 are similar, this pooled estimate of the overall proportion will be approximately half way between $\hat{p}_1$ and $\hat{p}_2$, but otherwise the pooled estimate will be closest to the estimate from the largest sample size.

|||| Method 7.18 Two sample proportions hypothesis test

The two-sample hypothesis test for comparing two proportions is given by the following procedure:

1. Compute, with $\hat{p} = \frac{x_1+x_2}{n_1+n_2}$, the test statistic

$$z_{\text{obs}} = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{7-37}$$

2. Compute evidence against the *null hypothesis*

$$H_0 : p_1 = p_2, \tag{7-38}$$

vs. the *the alternative hypothesis*

$$H_1 : p_1 \neq p_2, \tag{7-39}$$

by the

$$p\text{-value} = 2 \cdot P(Z > |z_{\text{obs}}|). \tag{7-40}$$

where the standard normal distribution $Z \sim N(0, 1^2)$ is used

3. If the p-value $< \alpha$ we reject H_0, otherwise we accept H_0,

or

The rejection/acceptance conclusion can equivalently be based on the critical value(s) $\pm z_{1-\alpha/2}$:
if $|z_{\text{obs}}| > z_{1-\alpha/2}$ we reject H_0, otherwise we accept H_0

Example 7.19

In Example 7.17 we tested whether the probability of blood clot is the same for the group taking the pills as for the group without pills using the CI. The null hypothesis and alternative are

$$H_0 : p_1 = p_2,$$
$$H_1 : p_1 \neq p_2.$$

This time we will test on a 1% significance level ($\alpha = 0.01$).

The pooled estimate of the probability of blood clot under H_0 is

$$\hat{p} = \frac{23 + 35}{57 + 167} = 0.259,$$

which is closest to the estimate from the largest group $\hat{p}_2 = 0.210$.

According to Method 7.15 the test statistic is

$$z_{\text{obs}} = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}} = \frac{0.194}{\sqrt{0.259(1-0.259)(\frac{1}{57} + \frac{1}{167})}} = 2.89.$$

The p-value is calculated by looking up z_{obs} in a standard normal distribution (i.e. $N(0,1)$)

$$2P(Z \geq 2.89) = 0.0039 < 0.01.$$

As the p-value is less than 0.01 we can reject the null hypothesis of equal probabilities in the two groups.

Instead of doing all the calculations in steps, we can use the function `smprop.proportions_ztest()` to test the hypothesis.

```
# Testing that the probabilities for the two groups are equal
z_obs, p_val = smprop.proportions_ztest([23, 35], [57, 167], value=0,
prop_var=0)
print(z_obs)

2.8859712586466184


print(p_val)

0.003902077897925702
```

7.4 Comparing several proportions

In the previous Section 7.3, we were interested in comparing proportions from two groups. In some cases we might be interested in proportions from two or more groups, or in other words if several binomial distributions share the same parameter p. The data can be setup in a $2 \times c$ table, where "Success" is the response we are studying (e.g. a blood clot occurs) and "Failure" is when the response does not occur (e.g. no blood clot).

	Group 1	Group 2	...	Group c	*Total*
Success	x_1	x_2	...	x_c	x
Failure	$n_1 - x_1$	$n_2 - x_2$	...	$n_c - x_c$	$n - x$
Total	n_1	n_2	...	n_c	n

We are then interested in testing the null hypothesis

$$H_0 : p_1 = p_2 = \ldots = p_c = p \tag{7-41}$$

against the alternative hypothesis: that the probabilities are not equal (or more precisely: that that at least one of the probabilities is different from the others).

Under H_0 the best estimator for the common p is the overall observed proportion

$$\hat{p} = \frac{x}{n}. \tag{7-42}$$

To test the null hypothesis, we need to measure how likely it is to obtain the observed data (or more extreme) under the null hypothesis. So, under the scenario that the null hypothesis is true, we can calculate the expected number of successes in the jth group as

$$e_{1j} = n_j \cdot \hat{p} = n_j \cdot \frac{x}{n}, \tag{7-43}$$

and the expected number of failures is

$$e_{2j} = n_j \cdot (1 - \hat{p}) = n_j \cdot \frac{(n-x)}{n}. \tag{7-44}$$

Notice, that the expected number for a cell is calculated by multiplying the row and column totals for the row and column, where the cell belongs and then dividing by the grand total n.

|||| Method 7.20 The multi-sample proportions χ^2-test

The hypothesis

$$H_0: p_1 = p_2 = \ldots = p_c = p, \tag{7-45}$$

can be tested using the test statistic

$$\chi^2_{\text{obs}} = \sum_{i=1}^{2} \sum_{j=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}, \tag{7-46}$$

where o_{ij} is the observed number in cell (i, j) and e_{ij} is the expected number in cell (i, j).

The test statistic χ^2_{obs} should be compared with the χ^2-distribution with $c-1$ degrees of freedom.

The χ^2-distribution is approximately the sampling distribution of the statistics under the null hypothesis. The rule of thumb is that it is valid when all the computed expected values are at least 5: $e_{ij} \geq 5$.

The test statistic in Method 7.20 measures the distance between the observed number in a cell and what we would expect if the null hypothesis is true. If the hypothesis is true then χ^2 has a relatively small value, as most of the cell counts will be close to the expected values. If H_0 is false, some of the observed values will be far from the expected resulting in a larger χ^2.

|||| Example 7.21

Returning to Example 7.19 we can consider a 2×2 table as a case of a $2 \times c$ table. We can organize our table with "Success" and "Failure" in the rows and groups as the columns.

	Contraceptive pill	No pill	*Total*
Blood clot	23	35	58
No blood clot	34	132	166
Total	57	167	224

Here $x = 23 + 35 = 58$ and $n = 224$

For each cell we can now calculate the expected number if H_0 is true. For the pill and blood clot cell we get

$$e_{1,1} = \frac{58 \cdot 57}{224} = 14.76, \tag{7-47}$$

but we only observed 23 cases.

For the no pill and blood clot cell we get

$$e_{1,2} = \frac{58 \cdot 167}{224} = 43.24, \tag{7-48}$$

which is more than the observed 35 cases.

In the following table we have both the observed and expected values.

	Birth control pill	No birth control pill	*Total*
Blood clot	$o_{11} = 23$ $e_{11} = 14.76$	$o_{12} = 35$ $e_{12} = 43.24$	$x = 58$
No blood clot	$o_{21} = 34$ $e_{21} = 42.24$	$o_{22} = 132$ $o_{22} = 123.8$	$(n - x) = 166$
Total	$n_1 = 57$	$n_2 = 167$	$n = 224$

The observed χ^2 test statistic can be calculated

$$\chi^2_{\text{obs}} = \frac{(23 - 14.76)^2}{14.76} + \frac{(35 - 43.24)^2}{43.24} + \frac{(34 - 42.24)^2}{42.24} + \frac{(132 - 123.8)^2}{123.8} = 8.33. \tag{7-49}$$

We then find the p-value, by calculating how likely it is to get 8.33 or more extreme if the null hypothesis is true, using the χ^2 distribution with $c - 1 = 2 - 1 = 1$ degrees of freedom

$$p\text{-value} = P(\chi^2 \geq 8.33) = 0.0039, \tag{7-50}$$

which is exactly the same as the result in Example 7.14. Do the same with the `stats.chi2_contingency()` function in Python:

```
# Reading the data into Python
pill_study = np.array([[23, 35], [34, 132]])
# Using Pandas
pill_study = pd.DataFrame(pill_study, index=[’Blood Clot’, ’No Clot’],
columns=[’Pill’, ’No pill’])
print(pill_study)

            Pill  No pill
Blood Clot    23       35
No Clot       34      132


# Chi^2 test for testing that the distribution for the two groups are
equal
chi2, p_val, dof, expected = stats.chi2_contingency(pill_study,
correction=False)
# Test Statistic
print(chi2)

8.328830105734347


# P value
print(p_val)

0.0039020778979257016


# Degrees of freedom
print(dof)

1


# Expected frequencies under the null hypothesis
# Output will not be pandas DataFrame, but we can use pandas to display
it nicely
print(pd.DataFrame(expected, index=[’Blood Clot’, ’No Clot’],
columns=[’Pill’, ’No pill’]))

                 Pill     No pill
Blood Clot  14.758929   43.241071
No Clot     42.241071  123.758929
```

In Section 7.3 we presented a z-test for the hypothesis $H_0 : p_1 = p_2$, where

$$z_{\text{obs}} = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}},$$

and in this section we have just seen a χ^2 test that can also be used for 2 × 2 tables. Using some algebra it turns out that the two tests are equivalent

$$\chi^2_{\text{obs}} = z^2_{\text{obs}}, \tag{7-51}$$

and they give exactly the same p-value for testing $H_0 : p_1 = p_2$ against $H_1 : p_1 \neq p_2$.

7.5 Analysis of Contingency Tables

Until now we have been looking at $2 \times c$ tables, but we can also have a more general setup with $r \times c$ tables that arise when two categorical variables are cross-tabulated. Such tables usually arise from two kinds of studies. First, we could have samples from several groups (as in Section 7.4), but allowing for more than two outcome categories. An example of this could be an opinion poll, where three samples were taken at different time points by asking randomly selected people whether they supported either: Candidate 1, Candidate 2 or were undecided. Here we want to compare the distribution of votes for the three groups (i.e. over time).

The other setup giving rise to an $r \times c$ table is when we have samples with two paired categorical variables with same categories (i.e. both variables are measured on each observational unit). This might happen if we had a sample of students and categorized them equivalently according to their results in English and mathematics (e.g. good, medium, poor). These tables are also called contingency tables.

The main difference between the two setups is: in the first setup the column totals are the size of each sample (i.e. fixed to the sample sizes), whereas in the second setup the column totals are not fixed (i.e. they count outcomes and the grand total is fixed to the sample size). However, it turns out that both setups are analysed in the same way.

7.5.1 Comparing several groups

In the situation comparing several groups, the hypothesis is that the distribution is the same in each group

$$H_0 : p_{i1} = p_{i2} = \ldots = p_{ic} = p_i, \quad \text{for all rows } i = 1, 2, \ldots, r. \tag{7-52}$$

So the hypothesis is that the probability of obtaining an outcome in a row category does not depend on the given column.

As in Section 7.4 we need to calculate the expected number in each cell under H_0

$$e_{ij} = \text{"}j\text{th column total"} \cdot \frac{\text{"}i\text{th row total"}}{\text{"grand total"}} = n_j \cdot \frac{x_i}{n}. \tag{7-53}$$

|||| Method 7.22 The $r \times c$ frequency table χ^2-test

For an $r \times c$ table the hypothesis

$$H_0 : p_{i1} = p_{i2} = \ldots = p_{ic} = p_i, \text{ for all rows } i = 1, 2, \ldots, r, \tag{7-54}$$

is tested using the test statistic

$$\chi^2_{\text{obs}} = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}. \tag{7-55}$$

where o_{ij} is the observed number in cell (i, j) and e_{ij} is the expected number in cell (i, j). This test statistic should be compared with the χ^2-distribution with $(r-1)(c-1)$ degrees of freedom and the hypothesis is rejected at significance level α if

$$\chi^2_{\text{obs}} > \chi^2_{1-\alpha}\big((r-1)(c-1)\big). \tag{7-56}$$

From Method 7.22, we see that we use the same test statistic as for $2 \times c$ tables measuring the distance between the observed and expected cell counts. The degrees of freedom $(r-1)(c-1)$ occurs because only $(r-1)(c-1)$ of the expected values e_{ij} need to be calculated – the rest can be found by subtraction from the relevant row or column totals.

|||| Example 7.23

An opinion poll has been made at three time points (4 weeks, 2 weeks and 1 week before the election) each time 200 participants was asked who they would vote for: Candidate 1, Candidate 2 or were undecided. The following data was obtained:

	4 weeks before	2 weeks before	1 week before	*Row total*
Candidate 1	79	91	93	263
Candidate 2	84	66	60	210
Undecided	37	43	47	127
Column total	200	200	200	600

Note, that in this poll example the sample sizes are equal (i.e. $n_1 = n_2 = n_3 = 200$), however that is not a requirement.

We want to test the hypothesis that the votes are equally distributed in each of the three polls

$$H_0 : p_{i1} = p_{i2} = p_{i3}, \text{ for all rows } i = 1,2,3. \tag{7-57}$$

The expected number of votes under H_0 is calculated for the "Candidate 2" - "2 weeks before" cell of the table

$$e_{22} = \text{"2'nd column total"} \cdot \frac{\text{"2'nd row total"}}{\text{"grand total"}} = \frac{210 \cdot 200}{600} = 70. \tag{7-58}$$

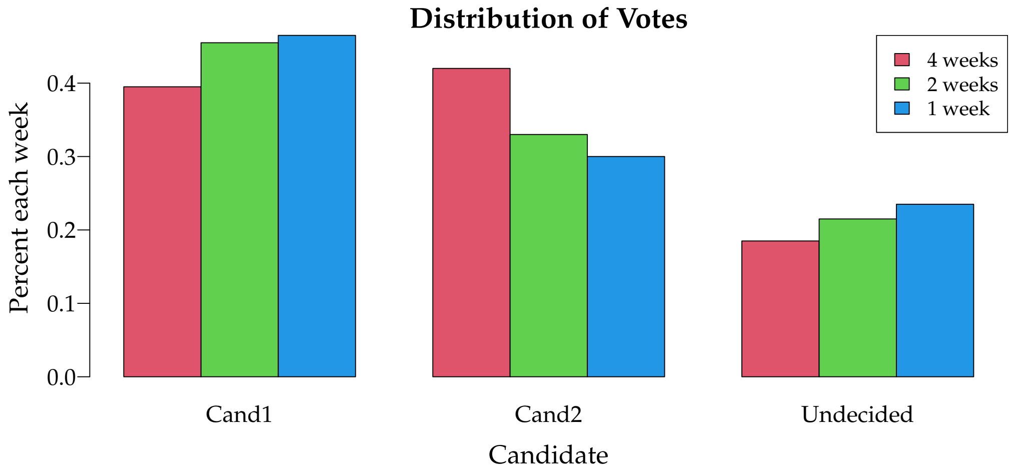
Continuing in the same way we can calculate all the expected cell counts:

	4 weeks before	2 weeks before	1 week before
Candidate 1	$o_{11} = 79$ $e_{11} = 87.67$	$o_{12} = 91$ $e_{12} = 87.67$	$o_{13} = 93$ $e_{13} = 87.67$
Candidate 2	$o_{21} = 84$ $e_{21} = 70.00$	$o_{22} = 66$ $e_{22} = 70.00$	$o_{23} = 60$ $e_{23} = 70.00$
Undecided	$o_{31} = 37$ $e_{31} = 42.33$	$o_{32} = 43$ $e_{32} = 42.33$	$o_{33} = 47$ $e_{33} = 42.33$

Looking at this table, it seems that 4 weeks before, Candidate 1 has less votes than expected while Candidate 2 has more, but we need to test whether these differences are statistically significant.

We can test the hypothesis in Equation (7-52) using a χ^2 test with $(3-1)(3-1) = 4$ degrees of freedom.

However, first we will calculate the observed column percentages and plot them:



From the bar plot it could seem that the support for Candidate 2 decreases as the election approaches, but we need to test whether this is significant. In the following Python code the hypothesis, stating that the distribution at each time point is the same, is tested:

```
# Reading the data into Python
poll = np.array([[79, 91, 93], [84, 66, 60], [37, 43, 47]])
poll = pd.DataFrame(poll, index=[’Cand1’, ’Cand2’, ’Undecided’],
                    columns=[’4 weeks’, ’2 weeks’, ’1 week’])

# testing same distribution in the three populations
chi2, p_val, dof, expected = stats.chi2_contingency(poll,
correction=False)
# Test statistic
print(chi2)

6.961978041718169


# p-value
print(p_val)

0.1379112060673381


# Degrees of Freedom
print(dof)

4


# Expected frequencies under the null hypothesis
print(pd.DataFrame(expected, index=[’Cand1’, ’Cand2’, ’Undecided’],
                    columns=[’4 weeks’, ’2 weeks’, ’1 week’]))

             4 weeks    2 weeks     1 week
Cand1      87.666667  87.666667  87.666667
Cand2      70.000000  70.000000  70.000000
Undecided  42.333333  42.333333  42.333333
```

From the χ^2 test we get an observed test statistic of 6.96, and we must now calculate how likely it is to obtain this value or more extreme from a χ^2-distribution with 4 degrees of freedom. It leads to a p-value of 0.14, so we accept the null hypothesis and find that there is no evidence showing a change in distribution among the three polls.

7.5.2 Independence between the two categorical variables

When the only fixed value is the grand *total*, then the hypothesis we are interested in concerns independence between the two categorical variables

$$\begin{aligned} H_0 &: \text{"The two variables are independent"}, \\ H_1 &: \text{"The two variables are not independent (they are associated)"}. \end{aligned} \tag{7-59}$$

Using the cell proportions p_{ij} the null hypothesis can be written as:

|||| Theorem 7.24

To test if two categorical variables are independent the null hypothesis

$$H_0 : p_{ij} = p_{i.}p_{.j} \text{ for all } i, j, \tag{7-60}$$

where $p_{i.} = \sum_{j=1}^{c} p_{ij}$ is the proportion of row i and $p_{.j} = \sum_{i=1}^{r} p_{ij}$ is the proportion of column j, is tested.

The p-value for the observed result under this null hypothesis is calculated using the χ^2 test statistic from Method 7.22.

|||| Example 7.25

A group of 400 students have had an English test and a mathematics test. The results of each test a categorized as either bad, average or good.

English	Mathematics			
	Bad	Average	Good	*Row total*
Bad	23	60	29	112
Average	28	79	60	167
Good	9	49	63	121
Column total	60	188	152	400

We want to test the hypothesis of independence between results in English and mathematics. First we read the data into Python and calculate proportions and totals:

```
# Reading the data into Python
results = np.array([[23, 60, 29], [28, 79, 60], [9, 49, 63]])
results_df = pd.DataFrame(results, index=['EngBad', 'EngAve',
'EngGood'],
                            columns=['MathBad', 'MathAve',
'MathGood'])
```

```
# Percentages
prop = results_df/results_df.sum().sum()
print(prop)

         MathBad  MathAve  MathGood
EngBad    0.0575   0.1500    0.0725
EngAve    0.0700   0.1975    0.1500
EngGood   0.0225   0.1225    0.1575

# Row totals
print(results_df.sum(axis=1))

EngBad     112
EngAve     167
EngGood    121
dtype: int64

# Column totals
print(results_df.sum(axis=0))

MathBad      60
MathAve     188
MathGood    152
dtype: int64
```

We want to calculate the expected cell count if H_0 is true. Consider the events "good English result" and "good mathematics result" corresponding to cell $(3,3)$. Under the hypothesis of independence, we have

$$p_{33} = P(\text{"Good English and Good Maths"}) = P(\text{"Good English"}) \cdot P(\text{"Good Maths"}) \tag{7-61}$$

From the calculated row and column totals, we would estimate

$$\hat{p}_{33} = \left(\frac{121}{400}\right) \cdot \left(\frac{152}{400}\right), \tag{7-62}$$

and out of 400 students we would expect

$$e_{33} = 400 \cdot \hat{p}_{33} = 400 \cdot \left(\frac{121}{400}\right) \cdot \left(\frac{152}{400}\right) = 121 \cdot \frac{152}{400} = 45.98. \tag{7-63}$$

The method of calculating the expected cell counts is exactly as before. For the "Good English and Good Mathematics" cell the expected value is less than the observed 63. Continuing in this way, we can calculate all the expected cell counts:

English	Mathematics		
	Bad	Average	Good
Bad	$o_{11} = 23$ $e_{11} = 16.80$	$o_{12} = 60$ $e_{12} = 52.64$	$o_{13} = 29$ $e_{13} = 42.56$
Average	$o_{21} = 28$ $e_{21} = 25.05$	$o_{22} = 79$ $e_{22} = 78.49$	$o_{23} = 60$ $e_{23} = 63.46$
Good	$o_{31} = 9$ $e_{31} = 18.15$	$o_{32} = 49$ $e_{32} = 56.87$	$o_{33} = 63$ $e_{33} = 45.98$

We can see that we have more students than expected in the Good - Good cell and less than expected in the two Bad - Good cells. We can now test the hypothesis of independence between English and mathematics results:

```
# Testing independence between english and maths results
chi2, p, dof, expected = stats.chi2_contingency(results,
correction=False)
# Test statistic
print(chi2)

20.178903582087926


# p-value
print(p)

0.00046038041384262443


# Degrees of Freedom
print(dof)

4


# Expected frequencies under the null hypothesis
print(pd.DataFrame(expected, index=['EngBad', 'EngAve', 'EngGood'],
                    columns=['MathBad', 'MathAve', 'MathGood']))

         MathBad  MathAve  MathGood
EngBad     16.80    52.64     42.56
EngAve     25.05    78.49     63.46
EngGood    18.15    56.87     45.98
```

The χ^2-test gives a test statistic of 20.18, which under H_0 follows a χ^2-distribution with 4 degrees of freedom leading to a p-value of 0.0005. This means that the hypothesis of independence between English and mathematics results is rejected.

Even though the hypothesis were formulated differently in the first setup when *comparing several groups*, compared to the second setup with the hypothesis on *independence of two categorical variables*, it turns out that the first hypothesis (7-52) is also about independence. Two events are independent if

$$P(A \text{ and } B) = P(A) \cdot P(B), \tag{7-64}$$

which expresses: the probability of both event A and event B occurring is equal to the probability of event A occurring times the probability of event B occuring.

Another way of defining independence of two variables is through conditioning. Two events are independent if

$$P(A|B) = P(A), \tag{7-65}$$

which states: the probability of event A does not change if we have information about B. In the first Example 7.23 the probability of voting for Candidate 1 is the same irrespective of week and therefore the distribution in one week is independent of the results from the other weeks.

Chapter 8

Comparing means of multiple groups - ANOVA

8.1 Introduction

In Chapter 3 the test of difference in mean of two groups was introduced

$$H_0: \quad \mu_1 - \mu_2 = \delta_0. \tag{8-1}$$

Often we are interested in testing if the mean of the two groups are different ($H_0 : \mu_1 = \mu_2$), against the alternative ($\mu_1 \neq \mu_2$). Often we will face a situation where we have data in multiple (more than two) groups leading to the natural extension of the two-sample situation to a multi-sample situation. The hypothesis of k groups having the same means can then be expressed as

$$H_0: \quad \mu_1 = \mu_2 = \cdots = \mu_k. \tag{8-2}$$

Or in words we have k groups (often referred to as treatments) and we want to test if they all have the same mean against the alternative that at least one group is different from the other groups. Note, that the hypothesis is not expressing any particular values for the means, but just that they are all the same.

The purpose of the data analysis in such a multi-group situation can be expressed as a two-fold purpose:

1. Answer the question: are the group means (significantly) different (hypothesis test)?

2. Tell the story about (or "quantify") the groups and their potential differences (estimates and confidence intervals)

The statistical analysis used for such an analysis is called one-way Analysis of Variance (ANOVA). Though there is an initial contradiction in the name, as ANOVA is used to compare the means of populations and not their variances, the name should not be met with confusion. An ANOVA expresses how different the means of k populations are by measuring how much of the variance in data is explained by grouping the observations (in other words: the variance explained by fitting a model with a mean for each population). If enough of the variation is explained, then a significant difference in population means can be concluded.

The one-way ANOVA is the natural multi-sample extension of the independent two-sample setup covered in Chapter 3. We will also present a natural multi-sample extension of the two paired-sample situation from Chapter 3. This generalization, where the k samples are somehow dependent, e.g. if the same individuals are used in each of the groups, is called two-way ANOVA.

8.2 One-way ANOVA

8.2.1 Data structure and model

As mentioned above we assume that we have data from k groups, also assume n_i repetitions in group (i), this imply that we can order data in a table like:

Tr_1	y_{11}	$\dots$	y_{1,n_1}
$\vdots$	$\vdots$	$\dots$	
Tr_k	$y_{k,1}$	$\dots$	y_{k,n_k}

The total number of observations is $n = \sum_{i=1}^{k} n_i$, note that there does not have to be the same number of observations within each group (treatment).

As for the two-sample case in Chapter 3 there are some standard assumptions that are usually made in order for the methods to come to be 100% valid. In the case of one-way ANOVA, these assumptions are expressed by formulating a "model" much like how regression models in Chapters 5 and 6 are expressed

$$Y_{ij} = \mu_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2). \tag{8-3}$$

The model is expressing that the observations come from a normal distribution within each group, that each group (i) has a specific mean, and that the variance is the same (σ^2) for all groups. Further, we see explicitly that we have a number of observations (n_i) within each group $(j = 1, \dots, n_i)$.

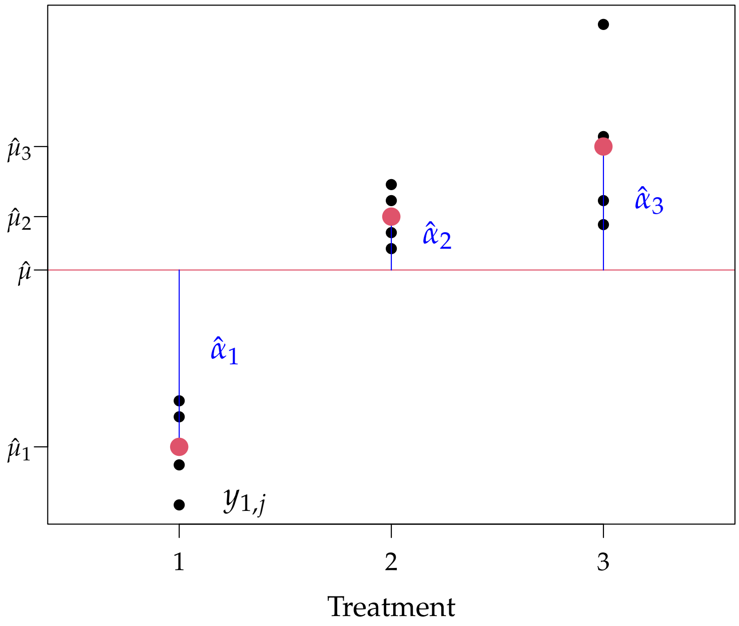


Figure 8.1: Conceptual plot for the ANOVA problem.

As noted above the relevant hypothesis to fulfil the first purpose of the analysis is that of equal group means (8-2). It turns out that a slight modification of (8-3) is convenient

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2). \tag{8-4}$$

Now, the situation is described with a μ that corresponds to the overall mean (across all groups), and then $\alpha_i = \mu_i - \mu$ is the difference between each group mean and the overall mean. The individual group mean is then $\mu_i = \mu + \alpha_i$, and the null hypothesis is expressed as

$$H_0 : \alpha_1 = \cdots = \alpha_k = 0, \tag{8-5}$$

with the alternative $H_1 : \alpha_i \neq 0$ for at least one i. The concept is illustrated in Figure 8.1 (for $k = 3$), the black dots are the measurements y_{ij}, the red line is the overall average, red dots are the average within each group, and the blue lines are the difference between group average and the overall average ($\hat{\alpha}_i$).

Let's have a look at an example, before we discuss the analysis in further details.

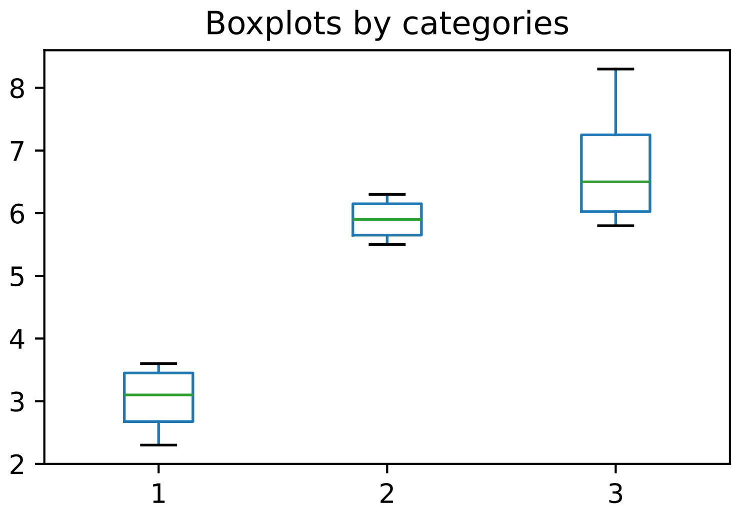
Example 8.1 Basic example

The data used for Figure 8.1 is given by:

Group A	Group B	Group C
2.8	5.5	5.8
3.6	6.3	8.3
3.4	6.1	6.9
2.3	5.7	6.1

The question is of course: is there a difference in the means of the groups (A, B and C)? We start by having a look at the observations:

```
y = np.array([2.8, 3.6, 3.4, 2.3,
              5.5, 6.3, 6.1, 5.7,
              5.8, 8.3, 6.9, 6.1])
treatm = pd.Categorical([1, 1, 1, 1,
                     2, 2, 2, 2,
                     3, 3, 3, 3])

D = pd.DataFrame({'y': y, 'treatm': treatm})

D.boxplot(by='treatm', grid=False)
plt.title('Boxplots by categories')
plt.suptitle('') # Removing automatic titles
plt.xlabel('')
plt.show()
```



By using pd.Categorical the treatments are not considered as numerical values by Python, but rather as factors (or grouping variables), and we can get the boxplot of the within group variation. This plot gives information about the location of data and variance homogeneity (the model assumption), of course with only 4 observations in each group it is difficult to asses this assumption.

Now we can calculate the parameter estimates ($\hat{\mu}$ and $\hat{\alpha}_i$) by:

```
mu = np.mean(y)
muis = D.groupby(’treatm’,observed=True)[’y’].mean()
alpha = muis - mu
print(mu)

5.233333333333333


print(muis)

treatm
1    3.025
2    5.900
3    6.775
Name: y, dtype: float64


print(alpha)

treatm
1   -2.208333
2    0.666667
3    1.541667
Name: y, dtype: float64
```

So our estimate of the overall mean is $\hat{\mu} = 5.23$, and the group levels (offsets from the overall sample mean) are $\hat{\alpha}_1 = -2.21$, $\hat{\alpha}_2 = 0.67$ and $\hat{\alpha}_3 = 1.54$. The question we need to answer is: how likely is it that the observed differences in group means are random variation? If this is very unlikely, then it can be concluded that at least one of them is significantly different from zero.

The shown use of the pandas function `groupby` function is a convenient way of finding the `mean` of `y` for each level of the factor `treatm`. By the way if the `mean` is substituted by any other function, e.g. the `variance`, we could similarly find the sample variance within each group (we will have a closer look at these later):

```
D.groupby('treatm',observed=True)['y'].var()

treatm
1    0.349167
2    0.133333
3    1.249167
Name: y, dtype: float64
```

8.2.2 Decomposition of variability, the ANOVA table

A characteristic of ANOVA in general and one-way ANOVA specifically is the fact that the overall variability (measured by the total variation) decomposes into interpretable components – it is these components which are used for hypothesis testing and more. For the one-way ANOVA presented in this section the total variation, that is, the variation calculated across all the data completely ignoring the fact that the data falls in different groups, can be decomposed into two components: a component expressing the group differences and a component expressing the (average) variation within the groups:

|||| Theorem 8.2 Variability decomposition

The total sum of squares (SST) can be decomposed into sum of squared errors (SSE) and treatment sum of squares ($SS(Tr)$)

$$\underbrace{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y})^2}_{\text{SST}} = \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2}_{\text{SSE}} + \underbrace{\sum_{i=1}^{k} n_i(\bar{y}_i-\bar{y})^2}_{\text{SS(Tr)}}, \tag{8-6}$$

where

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{k}\sum_{j=1}^{n_i} y_{ij}, \quad \bar{y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij}. \tag{8-7}$$

Expressed in short form

$$SST = SS(Tr) + SSE. \tag{8-8}$$

Before we turn to the proof of the theorem, we will briefly discuss some interpretations and implications of this. First we look at each of the three terms separately.

The *SST* expresses the *total variation*. Let us compare with Equation (1-6) the formula for sample variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2. \tag{8-9}$$

We can see that if the sample variance formula is applied to the the y_{ij}s joined into a single sample (i.e. a single index counts through all the n observations), then the sample variance is simply SST divided by $n-1$. The sample variance expresses then the *average variation per observation*. Therefore, we have

$$SST = (n-1) \cdot s_y^2, \tag{8-10}$$

where s_y^2 is the sample variance for all the y_{ij}s seen as a single sample (i.e. a sample from single population).

The group mean differences are quantified by the $SS(Tr)$ component, which can basically be seen directly from the definition, where the overall mean is subtracted from each group mean. As discussed above it can alternatively be expressed by deviations $\hat{\alpha}_i$

$$SS(Tr) = \sum_{i=1}^{k} n_i(\bar{y}_i - \bar{y})^2 = \sum_{i=1}^{k} n_i\hat{\alpha}_i^2, \tag{8-11}$$

so $SS(Tr)$ is the sum of squared α_i's multiplied by the number of observations in group n_i.

|||| Remark 8.3

$SS(Tr)$ is also the key expression to get the idea of why we call the whole thing “analysis of variance”: if we, for a second, assume that we have the same number of observations in each group: $n_1 = \ldots = n_k$, and let us call this common number m. Then we can express $SS(Tr)$ directly in terms of the variance of the k means

$$SS(Tr) = (k-1) \cdot m \cdot s^2_{\text{means}}, \tag{8-12}$$

where

$$s^2_{\text{means}} = \frac{1}{k-1} \sum_{i=1}^{k} (\bar{y}_i - \bar{y})^2. \tag{8-13}$$

Let us emphasize that the formulas of this remark is not thought to be formulas that we use for practical purposes, but they are expressed to show explicitly that “$SS(Tr)$ quantifies the group differences by variation”. Another way of thinking of $SS(Tr)$ is that it quantifies the “the variance explained by grouping the observations“, i.e. the variance explained by fitting a model with a mean for each group.

Finally, SSE expresses the average variability within each group, as each individual observation y_{ij} is compared with the mean in the group to which it belongs. In Figure 8.1 these are the differences between each of the black dots with the relevant read dot. Again we can link the formula given above to basic uses of the sample variance formula:

Theorem 8.4 Within group variability

The sum of squared errors SSE divided by $n-k$, also called the residual mean square $MSE = SSE/(n-k)$ is the weighted average of the sample variances from each group

$$MSE = \frac{SSE}{n-k} = \frac{(n_1-1)s_1^2 + \cdots + (n_k-1)s_k^2}{n-k}, \tag{8-14}$$

where s_i^2 is the variance within the ith group

$$s_i^2 = \frac{1}{n_i-1}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2. \tag{8-15}$$

When $k=2$, that is, we are in the two-sample case presented in Section 3.2, the result here is a copy of the pooled variance expression in Method 3.52

$$\text{For } k=2: \; MSE = s_p^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n-2}. \tag{8-16}$$

Intuitively, we would say that if some of the $\hat{\alpha}_i$'s are large (in absolute terms), then it is evidence against the null hypothesis of equal means. So a large $SS(Tr)$ value is evidence against the null hypothesis. It is also natural that "large" should be relative to some variation. SSE is the within group variation, and it also seems reasonable that if $\hat{\alpha}_i$ is large and the variation around $\hat{\mu}_i$ is small then this is evidence against the null hypothesis. It does therefore seem natural to compare $SS(Tr)$ and SSE, and we will get back to the question of exactly how to do this after the proof of Theorem 8.2:

Proof

Add and subtract $\bar{y}_i$ in SST to get

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y})^2 &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i+\bar{y}_i-\bar{y})^2 \qquad (8\text{-}17)\\ &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}\left[(y_{ij}-\bar{y}_i)^2+(\bar{y}_i-\bar{y})^2+2(y_{ij}-\bar{y}_i)(\bar{y}_i-\bar{y})\right]\\ &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2+\sum_{i=1}^{k}\sum_{j=1}^{n_i}(\bar{y}_i-\bar{y})^2+2\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)(\bar{y}_i-\bar{y})\\ &= \sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2+\sum_{i=1}^{k}n_i(\bar{y}_i-\bar{y})^2+2\sum_{i=1}^{k}(\bar{y}_i-\bar{y})\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i),\end{aligned}$$

now observe that $\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)=0$, and the proof is completed.

■

Example 8.5

We can now continue our example and calculate SST, SSE, and $SS(Tr)$:

```
muis = muis.values # Coverting to numpy array
alpha = muis - mu
SST = np.sum((y - mu)**2)
SSE = (np.sum((y[treatm == 1] - muis[0])**2) +
       np.sum((y[treatm == 2] - muis[1])**2) +
       np.sum((y[treatm == 3] - muis[2])**2))
SSTr = 4 * np.sum(alpha**2)
print(np.round([SST, SSE, SSTr],3))

[35.987  5.195 30.792]
```

For these data we have that $n_1 = n_2 = n_3 = 4$, so according to Theorem 8.2 we could also find SSE from the average of the variances within each group:

```
vars = D.groupby(’treatm’,observed=True)[’y’].var()
print((12 - 3) * np.mean(vars))

5.195000000000002
```

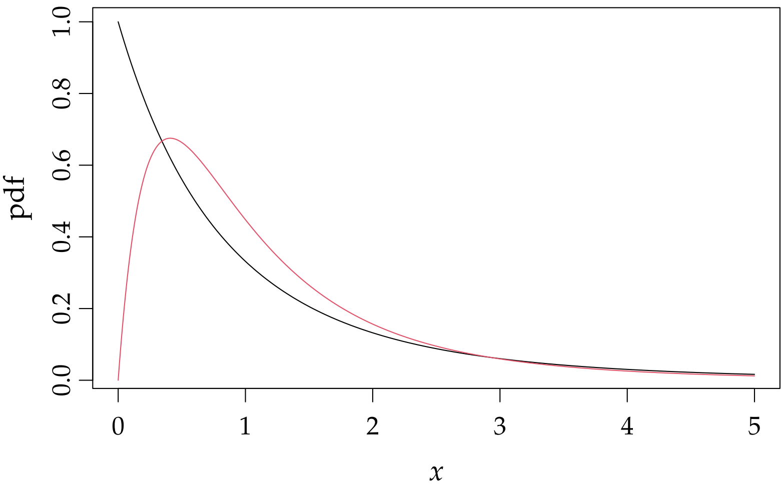


Figure 8.2: pdf of the F-distribution with 2 and 9 degrees of freedom (black line), and with 4 and 9 degrees of freedom (red line).

Now we have established that we should compare $SS(Tr)$ and SSE in some way, it should of course be quantified exactly in which way they should be compared. Now it turns out that the numbers $SS(Tr)/(k-1)$ and $SSE/(n-k)$ are both central estimators for σ^2, when the null hypothesis is true, and we can state the following theorem:

|||| Theorem 8.6

Under the null hypothesis

$$H_0: \quad \alpha_i = 0, \quad i = 1, 2, \ldots, k, \tag{8-18}$$

the test statistic

$$F = \frac{SS(Tr)/(k-1)}{SSE/(n-k)}, \tag{8-19}$$

follows an F-distribution with $k-1$ and $n-k$ degrees of freedom.

The F-distribution is generated by the ratio between independent χ^2 distributed random variables, and the shape is shown in Figure 8.2 for two particular choices of degrees of freedom.

As we have discussed before, the null hypothesis should be rejected if $SS(Tr)$ is large and SSE is small. This implies that we should reject the null hypothesis when the test statistic (F) is large in the sense of Theorem 8.6 (compare with $F_{1-\alpha}$). The statistics are usually collected in an ANOVA table like this:

Source of variation	Degrees of freedom	Sums of squares	Mean sum of squares	Test-statistic F	p-value
Treatment	$k-1$	$SS(Tr)$	$MS(Tr) = \frac{SS(Tr)}{k-1}$	$F_{\text{obs}} = \frac{MS(Tr)}{MSE}$	$P(F > F_{\text{obs}})$
Residual	$n-k$	SSE	$MSE = \frac{SSE}{n-k}$		
Total	$n-1$	SST			

|||| Example 8.7

We can now continue with our example and find the F-statistic and the p-value:

```
F = (SSTr / (3 - 1)) / (SSE / (12 - 3))
pv = 1 - stats.f.cdf(F, 3 - 1, 12 - 3)
print(F, pv)

26.67228103946101 0.0001650052218172826
```

So we have a test statistic $F = 26.7$ and a p-value equal to 0.000165 and we reject the null hypothesis on e.g. level $\alpha = 0.05$. The calculations can of course also be done directly in Python, by:

```
fit = smf.ols('y ~ treatm', data=D).fit()
anova = sm.stats.anova_lm(fit)
print(anova)

           df     sum_sq    mean_sq          F    PR(>F)
treatm    2.0  30.791667  15.395833  26.672281  0.000165
Residual  9.0   5.195000   0.577222        NaN       NaN
```

Note, that in the direct Python calculation it is very important to include `treatm` as a factor (categorical), in order to get the correct analysis.

If we reject the null hypothesis, it implies that the observations can be finally described by the initial model re-stated here

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2), \tag{8-20}$$

and the estimate of the error variance σ^2 is $\hat{\sigma}^2 = SSE/(n-k) = MSE$.

|||| **Remark 8.8 When multiple groups = 2 groups**

When $k = 2$, that is, we are in the two-sample case studied in Chapter 3, we already saw above in Theorem 8.4 that $MSE = s_p^2$. Actually, this means that the analysis we get from a one-way ANOVA when we apply it for only $k = 2$ groups, which could be perfectly fine - nothing in the ANOVA approach really relies on k having to be larger than 2 - corresponds to the pooled t-test given as Method 3.53. More exact

$$\text{for } k = 2: \; F_{\text{obs}} = t_{\text{obs}}^2, \tag{8-21}$$

where t_{obs} is the pooled version coming from Methods 3.52 and 3.53. Thus the p-value obtained from the $k = 2$ group ANOVA equals exactly the p-value obtained from the pooled t-test given in Method 3.53.

8.2.3 Post hoc comparisons

If we reject the overall null hypothesis above, and hence conclude that $\alpha_i \neq 0$ for at least one i it makes sense to ask which of the treatments are actually different. That is, trying to meet the second of the two major purposes indicated in the beginning. This can be done by pairwise comparison of the treatments. We have already seen in Chapter 3, that such comparison could be based on the t-distribution. We can construct confidence interval with similar formulas except that we should use $MSE = SSE/(n-k)$ as the estimate of the error variance and hence also $n - k$ degrees of freedom in the t-distribution:

|||| Method 8.9 Post hoc pairwise confidence intervals

A single pre-planned $(1-\alpha)\cdot 100\%$ confidence interval for the difference between treatment i and j is found as

$$\bar{y}_i - \bar{y}_j \pm t_{1-\alpha/2}\sqrt{MSE\left(\frac{1}{n_i}+\frac{1}{n_j}\right)}, \tag{8-22}$$

where $t_{1-\alpha/2}$ is based on the t-distribution with $n-k$ degrees of freedom.

If all $M = k(k-1)/2$ combinations of pairwise confidence intervals are calculated using the formula M times, but each time with $\alpha_{\text{Bonferroni}} = \alpha/M$ (see Remark 8.14 below).

Similarly one could do pairwise hypothesis tests:

|||| Method 8.10 Post hoc pairwise hypothesis tests

A single pre-planned level α hypothesis tests

$$H_0:\ \mu_i = \mu_j,\ \ H_1:\ \mu_i \neq \mu_j, \tag{8-23}$$

is carried out by

$$t_{\text{obs}} = \frac{\bar{y}_i - \bar{y}_j}{\sqrt{MSE\left(\frac{1}{n_i}+\frac{1}{n_j}\right)}}, \tag{8-24}$$

and

$$p\text{-value} = 2\cdot P(T > |t_{\text{obs}}|), \tag{8-25}$$

where the t-distribution with $n-k$ degrees of freedom is used.

If all $M = k(k-1)/2$ combinations of pairwise hypothesis tests are carried out use the approach M times but each time with test level $\alpha_{\text{Bonferroni}} = \alpha/M$ (see Remark 8.14 below).

Example 8.11

Returning to our small example we get the pairwise confidence intervals. If the comparison of A and B was specifically planned before the experiment was carried out, we would find the 95%-confidence interval as:

```
print(muis[0] - muis[1] + np.array([-1, 1]) *
       stats.t.ppf(1 - 0.05 / 2, 12 - 3) * np.sqrt(SSE / (12 - 3) *
(1/4 + 1/4)))

[-4.090 -1.660]
```

and we can hence also conclude that treatment A is different from B. The p-value supporting this claim is found as:

```
tobs = (muis[0] - muis[1]) / np.sqrt(SSE / (12 - 3) * (1/4 + 1/4))
print(2 * (1 - stats.t.cdf(np.abs(tobs), 12 - 3)))

0.0004613963065729365
```

If we do all three possible comparisons, $M = 3 \cdot 2/2 = 3$, and we will use an overall $\alpha = 0.05$, we do the above three times, but using each time $\alpha_{\text{Bonferroni}} = 0.05/3 = 0.016667$:

```
alpha_bonf = 0.05 / 3
# A-B
print(alpha[0] - alpha[1] + np.array([-1, 1]) *
      stats.t.ppf(1-alpha_bonf/2, 12 - 3) * np.sqrt(SSE/(12 - 3) * (1/4
+ 1/4)))
```

```
[-4.451 -1.299]
```

```
# A-C
print(alpha[0] - alpha[2] + np.array([-1, 1]) *
      stats.t.ppf(1-alpha_bonf/2, 12 - 3) * np.sqrt(SSE/(12 - 3) * (1/4
+ 1/4)))
```

```
[-5.326 -2.174]
```

```
# B-C
print(alpha[1] - alpha[2] + np.array([-1, 1]) *
      stats.t.ppf(1-alpha_bonf/2, 12 - 3) * np.sqrt(SSE/(12 - 3) * (1/4
+ 1/4)))
```

```
[-2.451  0.701]
```

and we conclude that treatment A is different from B and C, while we cannot reject that B and C are equal. The *p*-values for the last two comparisons could also be found, but we skip that now.

The so-called Bonferroni correction done above, when we do all possible post hoc comparisons, has the effect that it becomes more difficult (than without the correction) to claim that two treatments have different means.

|||| Example 8.12

The 0.05/3-critical value with 9 degrees of freedom is $t_{0.9917} = 2.933$ whereas the 0.05-critical value is $t_{0.975} = 2.262$:

```
print(stats.t.ppf(1 - alpha_bonf / 2, 12 - 3), stats.t.ppf(1 - 0.05 /
2, 12 - 3))
```

```
2.9333240883739897 2.2621571628540993
```

So two treatment means would be claimed different WITH the Bonferroni correction if they differ by more than half the width of the confidence interval

$$2.933 \cdot \sqrt{SSE/9 \cdot (1/4 + 1/4)} = 1.576 \tag{8-26}$$

whereas without the Bonferroni correction should only differ by more than

$$2.262 \cdot \sqrt{SSE/9 \cdot (1/4 + 1/4)} = 1.215 \tag{8-27}$$

to be claimed significantly different.

|||| Remark 8.13 Least Significant Difference (LSD) values

If there is the same number of observations in each treatment group $m = n_1 = \ldots = n_k$ the LSD value for a particular significance level

$$LSD_\alpha = t_{1-\alpha/2}\sqrt{2 \cdot MSE/m} \tag{8-28}$$

will have the same value for all the possible comparisons made.
The LSD value is particularly useful as a "measuring stick" with which we can go and compare all the observed means directly: the observed means with difference higher than the LSD are significantly different on the α-level. When used for all of the comparisons, as suggested, one should as level use the Bonferroni corrected version $LSD_{\alpha_{\text{Bonferroni}}}$ (see Remark 8.14 below for an elaborated explanation).

|||| Remark 8.14 Significance by chance in multiple testings!

Imagine that we performed an ANOVA in a situation with $k = 15$ groups. And then we do all the $M = 15 \cdot 14/2 = 105$ possible pairwise hypothesis tests. Assume for a moment that the overall null hypothesis is true, that is, there really are no mean differences between any of the 15 groups. And think about what would happen if we still performed all the 105 tests with $\alpha = 0.05$! How many significant results would we expect among the 105 hypothesis tests? The answer is that we expect $\alpha \cdot 105 = 0.05 \cdot 105 = 5.25$, that is, approximately 5 significant tests are expected. And what would the probability be of getting at least one significant test out of the 105? The answer to this question can be found using the binomial distribution

$$\begin{aligned} P(\text{"At least one significant result in 105 independent tests"}) \\ &= 1 - 0.95^{105} \\ &= 0.9954. \end{aligned} \tag{8-29}$$

So whereas we, when performing a single test, have a probability of $\alpha = 0.05$ of getting a significant result, when we shouldn't, we now have an overall Type I error probability of seeing at least one significant result, when we shouldn't, of 0.9954! This is an extremely high (overall) Type 1 risk. This is also sometimes called the "family wise" Type 1 risk. In other words, we will basically always with $k = 15$ see at least one significant pairwise difference, if we use $\alpha = 0.05$. This is why we recommend to use a correction method when doing multiple testings like this. The Bonferroni correction approach is one out of several different possible approaches for this.

Using the Bonferroni corrected $\alpha_{\text{Bonferroni}} = 0.05/105$ in this case for each of the 105 tests would give the family wise Type 1 risk

$$\begin{aligned} P(\text{"At least one significant result in 105 independent tests"}) \\ &= 1 - (1 - 0.05/105)^{105} \\ &= 0.049 \end{aligned} \tag{8-30}$$

8.2.4 Model control

The assumptions for the analysis we have applied in the one-way ANOVA model are more verbally expressed as:

1. The data comes from a normal distribution in each group
2. The variances from each group are the same

The homogeneous variances assumption can be controlled by simply looking at the distributions within each sample, most conveniently for this purpose by the group-wise box plot already used in the example above.

The normality within groups assumption could in principle also be investigated by looking at the distributions within each group - a direct generalization of what was suggested in Chapter 3 for the two-group setting. That is, one could do a q-q plot within each group. It is not uncommon though, that the amount of data within a single group is too limited for a meaningful q-q plot investigation. Indeed for the example here, we have only 4 observations in each group, and q-q plots for 4 observations do not make much sense.

There is an alternative, where the information from all the groups are pooled together to a single q-q plot. If we pool together the 12 residuals, that is, within group deviations, they should all follow the same zero-mean normal distribution, given by

$$\varepsilon_{ij} \sim N(0, \sigma^2). \tag{8-31}$$

|||| Method 8.15 Normality control in one-way ANOVA

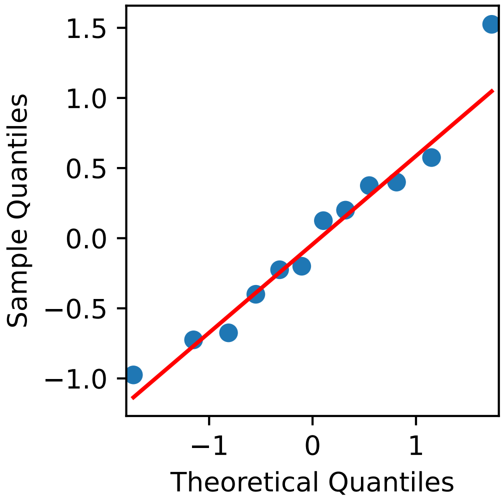
To control for the normality assumptions in one-way ANOVA we perform a q-q plot on the pooled set of n estimated residuals

$$e_{ij} = y_{ij} - \bar{y}_i, \; j = 1, \ldots, n_i, \; i = 1 \ldots, k. \tag{8-32}$$

|||| Example 8.16

For the basic example we get the normal q-q plot of the residuals by

```
sm.qqplot(fit.resid.values, line='q',a=1/2)
plt.tight_layout()
plt.show()
```



```
print(fit.resid.values)

[-0.225  0.575  0.375 -0.725 -0.400  0.400  0.200 -0.200 -0.975  1.525
  0.125 -0.675]
```

8.2.5 A complete worked through example: plastic types for lamps

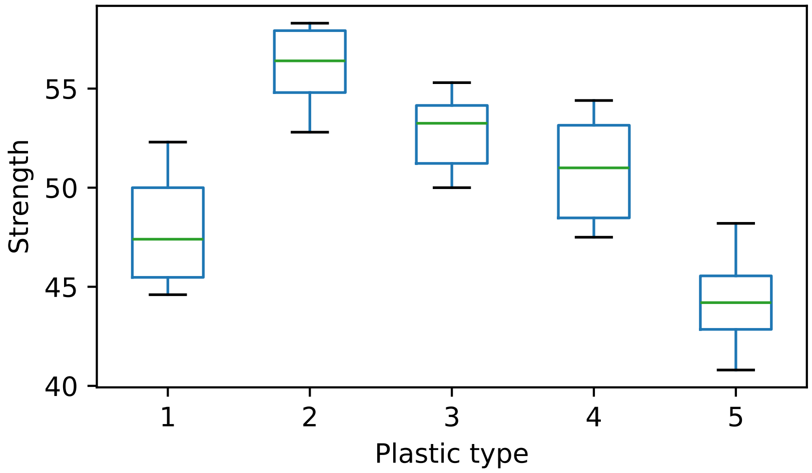
Example 8.17 Plastic types for lamps

On a lamp two plastic screens are to be mounted. It is essential that these plastic screens have a good impact strength. Therefore an experiment is carried out for 5 different types of plastic. 6 samples in each plastic type are tested. The strengths of these items are determined. The following measurement data was found (strength in kJ/m^2):

Type of plastic				
I	II	III	IV	V
44.6	52.8	53.1	51.5	48.2
50.5	58.3	50.0	53.7	40.8
46.3	55.4	54.4	50.5	44.5
48.5	57.4	55.3	54.4	43.9
45.2	58.1	50.6	47.5	45.9
52.3	54.6	53.4	47.8	42.5

We run the following in Python:

```
D = pd.DataFrame({
       ’strength’: [44.6, 52.8, 53.1, 51.5, 48.2, 50.5, 58.3, 50.0,
53.7, 40.8,
                     46.3, 55.4, 54.4, 50.5, 44.5, 48.5, 57.4, 55.3,
54.4, 43.9,
                     45.2, 58.1, 50.6, 47.5, 45.9, 52.3, 54.6, 53.4,
47.8, 42.5],
       ’plastictype’: pd.Categorical(np.tile(np.arange(1, 6), 6))
})

D.boxplot(by=’plastictype’, grid=False)
plt.suptitle(”) # Removing automatic titles
plt.title(”)
plt.xlabel(’Plastic type’)
plt.ylabel(’Strength’)
plt.tight_layout()
plt.show()
```



```
fit = smf.ols(’strength ∼ plastictype’, data=D).fit()
anova = sm.stats.anova_lm(fit)
print(anova)

               df  sum_sq   mean_sq          F        PR(>F)
plastictype   4.0  491.76  122.9400  18.233863  3.987701e-07
Residual     25.0  168.56    6.7424        NaN           NaN
```

The ANOVA results are more nicely put in a table here:

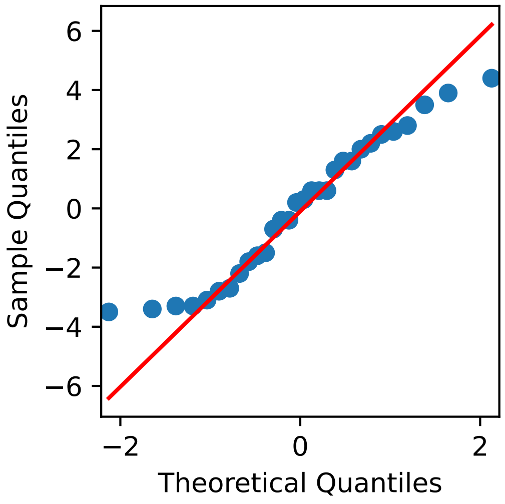
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Plastictype	4	491.76	122.94	18.23	$4 \cdot 10^{-7}$
Residuals	25	168.56	6.74		

From the box plot we see that there appears to be group mean differences and extremely low p-value in the ANOVA table confirms this: there is very strong evidence against the null hypothesis of the five means being the same

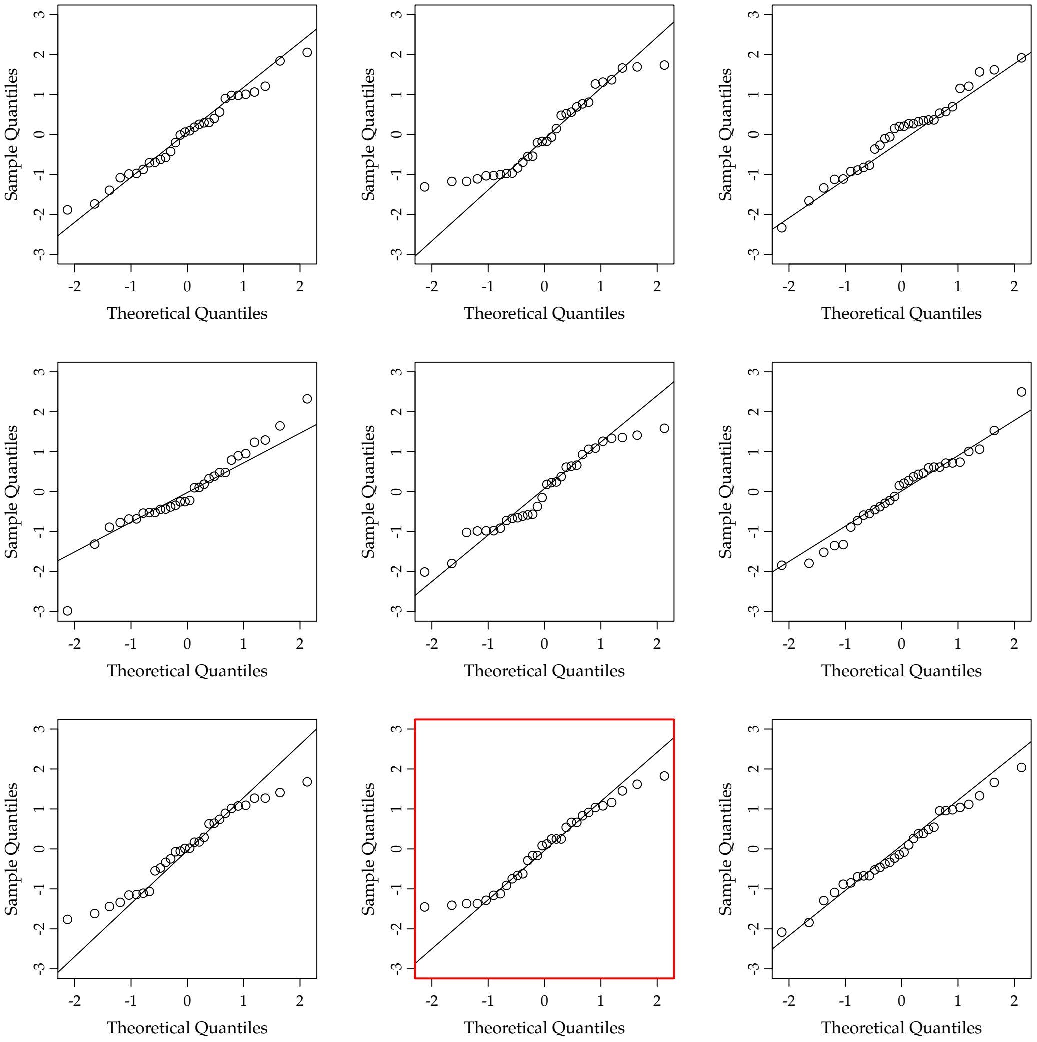
$$H_0: \ \mu_1 = \cdots = \mu_5. \tag{8-33}$$

Model assumptions: the box plots do not indicate clear variance differences (although it can be a bit difficult to know exactly how different such patterns should be for it to be a problem. Statistical tests exist for such varicance comparisons, but they are not included here). Let us check for the normality by doing a normal q-q plot on the residuals:

```
sm.qqplot(fit.resid.values, line='q', a=1/2)
plt.tight_layout()
plt.show()
```



Or using the idea of comparing with repeated plots on the standardized residuals: (See Section 3.1.8)



There appears to be no important deviation from normality.

To complete the story about (quantifying) the five plastic types, we first compute the five means:

```
print(D.groupby('plastictype',observed=True)['strength'].mean())

plastictype
1    47.9
2    56.1
3    52.8
4    50.9
5    44.3
Name: strength, dtype: float64
```

And then we want to construct the $M = 5 \cdot 4/2 = 10$ different confidence intervals using Method 8.9. As all n_is equal 6 in this case, all 10 confidence intervals will have the same width, and we can use Remark 8.13 and compute the (half) width of the confidence intervals, the *LSD*-value. And since there are 10 multiple comparisons we will use $\alpha_{\text{Bonferroni}} = 0.05/10 = 0.005$:

```
LSD_0_005 = stats.t.ppf(1 - 0.005 / 2, 25) * np.sqrt(2 * 6.74 / 6)
print(LSD_0_005)

4.61387770149341
```

So Plastictypes are significantly different from each other if they differ by more than 4.61. A convenient way to collect the information about the 10 comparisons is by ordering the means from smallest to largest and then using the so-called compact letter display:

Plastictype	Mean	
5	44.3	a
1	47.9	ab
4	50.9	bc
3	52.8	cd
2	56.1	d

Plastic types with a mean difference less than the *LSD*-value, hence not significantly different share letters. Plastic types not sharing letters are significantly different. We can hence read off all the 10 comparisons from this table.

One could also add the compact letter information to the box plot for a nice plotting - it is allowed to be creative (while not changing the basic information and the results!) in order to communicate the results.

8.3 Two-way ANOVA

8.3.1 Data structure and model

The one-way ANOVA is the natural multi-sample extension of the independent two-sample situation covered in Section 3.2. The k samples are hence completely independent from each other, which e.g. in a clinical experiment would mean that different patients get different treatments – and hence each patient only tries a single treatment. Often this would be the only possible way to do a comparison of treatments.

However, sometimes it will be possible to apply multiple treatments to the same patient (with some time in between). This could then lead to a multi-treatment setup, where the sample within each treatment consists of the same patients. This is the natural extension of the paired-design setup covered in Section 3.2.3, where we "pair" if there is exactly 2 treatments. With more than two treatments we use the phrase "block". A block would then be the patient in this case - and the same blocks then appear in all treatment samples. The "block" name comes from the historical background of these methods coming from agricultural field trials, where a block would be an actual piece of land within which all treatments are applied.

|||| Remark 8.18 Design: independent sampling or blocking?

For the project manager who is in charge of designing the study there will be a choice to make in cases where both approaches are practicable feasible: should the independent samples approach or the blocked approach be used? Should we use, say, 4 groups of 20 patients each, that is 80 patients all together, or should we use the same 20 patients in each of the four groups? The costs would probably be more or less the same. It sounds nice with 80 patients rather than 20? However, the answer is actually pretty clear if whatever we are going to measure will vary importantly from person to person. And most things in medical studies do vary a lot from person to person due to many things: gender, age, weight, BMI, or simply due to genetic differences that means that our bodies will respond differently to the medicine. Then the blocked design would definitely be the better choice! This is so, as we will see below, in the analysis of the blocked design the block-main-variability is accounted for and will not "blur" the treatment difference signal. In the independent design the person-to-person variability may be the main part of the "within group" variability used for the statistical analysis. Or differently put: in a block design each patient would act as his/her own control, the treatment comparison is carried out "within the block".

For the actual study design it would in both cases be recommended to randomize the allocation of patients as much as possible: In the independent design patients should be allocated to treatments by randomization. In the block design all patients receive all treatments but then one would randomize the order in which they receive the treatments. For this reason these two types of experimental designs are usually called the *Completely Randomized Design* and the *Randomized Block Design*.

We looked above in the one-way part at an example with 3 treatments with 4 observations for each. If the observations were on 4 different persons (and not 12) it would make sense and would be important to include this person variability in the model. The resulting model becomes

$$Y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2), \tag{8-34}$$

so there is an overall mean μ, a treatment effect α_i and a block effect β_j and our usual random error term ε_{ij}.

The design is illustrated in the table below, so we have k treatments (A$_1$,...,A$_k$) and l blocks (B$_1$,...,B$_l$):

	B$_1$	...	B$_l$
A$_1$	y_{11}	...	$y_{1,l}$
⋮	⋮	...	⋮
A$_k$	$y_{k,1}$	...	$y_{k,l}$

We can now find the parameters in the model above by

$$\hat{\mu} = \frac{1}{k \cdot l} \sum_{i=1}^{k} \sum_{j=1}^{l} y_{ij}, \tag{8-35}$$

$$\hat{\alpha}_i = \left(\frac{1}{l} \sum_{j=1}^{l} y_{ij} \right) - \hat{\mu}, \tag{8-36}$$

$$\hat{\beta}_j = \left(\frac{1}{k} \sum_{i=1}^{k} y_{ij} \right) - \hat{\mu}. \tag{8-37}$$

Or expressed more compact, with the definitions of the terms obvious from the above

$$\hat{\mu} = \bar{\bar{y}}, \tag{8-38}$$

$$\hat{\alpha}_i = \bar{y}_{i\cdot} - \bar{\bar{y}}, \tag{8-39}$$

$$\hat{\beta}_j = \bar{y}_{\cdot j} - \bar{\bar{y}}. \tag{8-40}$$

In a way, these means are the essential information in these data. All the rest we do is just all the statistics to distinguish signal from noise. It does not change the fact, that these means contain the core story. It also shows explicitly how we now compute means, not only across one way in the data table, but also across the other way. We compute means both row-wise and column-wise. Hence the name: two-way ANOVA.

Example 8.19

Lets assume that the data we used in the previous section was actually a result of a randomized block design and we could therefore write:

	Group A	Group B	Group C
Block 1	2.8	5.5	5.8
Block 2	3.6	6.3	8.3
Block 3	3.4	6.1	6.9
Block 4	2.3	5.7	6.1

In this case we should of course keep track of the blocks as well as the treatments:

```
y = np.array([2.8, 3.6, 3.4, 2.3,
              5.5, 6.3, 6.1, 5.7,
              5.8, 8.3, 6.9, 6.1])

treatm = pd.Categorical([1, 1, 1, 1,
                         2, 2, 2, 2,
                         3, 3, 3, 3])
block = pd.Categorical([1, 2, 3, 4,
                        1, 2, 3, 4,
                        1, 2, 3, 4])

D = pd.DataFrame({’y’: y, ’treatm’: treatm, ’block’: block})
```

Now we can calculate the parameter estimates ($\hat{\mu}$ and $\hat{\alpha}_i$, and $\hat{\beta}_j$):

```
mu = np.mean(y)
alpha = D.groupby('treatm',observed=True)['y'].mean() - mu
beta = D.groupby('block',observed=True)['y'].mean() - mu
print(mu)

5.233333333333333


print(alpha)

treatm
1   -2.208333
2    0.666667
3    1.541667
Name: y, dtype: float64


print(beta)

block
1   -0.533333
2    0.833333
3    0.233333
4   -0.533333
Name: y, dtype: float64
```

so our estimates of the overall mean (μ) and α_i remain the same while the estimated block effects are $\hat{\beta}_1 = -0.53$, $\hat{\beta}_2 = 0.83$, $\hat{\beta}_3 = 0.23$ and $\hat{\beta}_4 = -0.53$.

8.3.2 Decomposition of variability and the ANOVA table

In the same way as we saw for the one-way ANOVA, there exists a decomposition of variation for the two-way ANOVA:

|||| Theorem 8.20 Variation decomposition

The total sum of squares (SST) can be decomposed into sum of squared errors (SSE), treatment sum of squares ($SS(Tr)$), and a block sum of squares ($SS(Bl)$)

$$\begin{aligned}\underbrace{\sum_{i=1}^{k}\sum_{j=1}^{l}(y_{ij}-\hat{\mu})^2}_{\text{SST}} &= \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{l}(y_{ij}-\hat{\alpha}_i-\hat{\beta}_j-\hat{\mu})^2}_{\text{SSE}} + l\cdot\underbrace{\sum_{i=1}^{k}\hat{\alpha}_i^2}_{\text{SS(Tr)}} + k\cdot\underbrace{\sum_{j=1}^{l}\hat{\beta}_j^2}_{\text{SS(Bl)}} \\ &= \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{l}(y_{ij}-\bar{y}_{i\cdot}-\bar{y}_{\cdot j}+\bar{\bar{y}})^2}_{\text{SSE}} + l\cdot\underbrace{\sum_{i=1}^{k}(\bar{y}_{i\cdot}-\bar{\bar{y}})^2}_{\text{SS(Tr)}} + k\cdot\underbrace{\sum_{j=1}^{l}(\bar{y}_{\cdot j}-\bar{\bar{y}})^2}_{SS(Bl)},\end{aligned} \tag{8-41}$$

Expressed in short form

$$SST = SS(Tr) + SS(Bl) + SSE. \tag{8-42}$$

Note, how the SST and $SS(Tr)$ are found exactly as in the one-way ANOVA. If one ignores the block-way of the table, the two-way data has exactly the same structure as one-way data (with the same number of observations in each group). Further, note how $SS(Bl)$ corresponds to finding a "one-way $SS(Tr)$", but on the other way of the table (and ignoring the treatment-way of the data table). So from a computational point of view, finding these three, that is, finding SST, $SS(Tr)$ and $SS(Bl)$ is done by known one-way methodology. And then the last one, SSE, could then be found from the decomposition theorem as

$$SSE = SST - SS(Tr) - SS(Bl). \tag{8-43}$$

|||| Example 8.21

Returning to the example we get (SST and $SS(Tr)$ remain unchanged):

```
beta = beta.values # Converting to numpy array
SSBl = 3 * np.sum(beta**2)
SSE = SST - SSTr - SSBl
print(np.round([SST, SSE, SSTr, SSBl],3))
```

```
[35.987  1.242 30.792  3.953]
```

Again, tests for treatment effects and block effects are done by comparing $SS(Tr)$ or $SS(Bl)$ with SSE:

Theorem 8.22

Under the null hypothesis

$$H_{0,Tr}: \quad \alpha_i = 0, \quad i = 1, 2, \ldots, k, \tag{8-44}$$

the test statistic

$$F_{Tr} = \frac{SS(Tr)/(k-1)}{SSE/((k-1)(l-1))}, \tag{8-45}$$

follows an F-distribution with $k-1$ and $(k-1)(l-1)$ degrees of freedom. Further, under the null hypothesis

$$H_{0,Bl}: \quad \beta_j = 0, \quad j = 1, 2, \ldots, l, \tag{8-46}$$

the test statistic

$$F_{Bl} = \frac{SS(Bl)/(l-1)}{SSE/((k-1)(l-1))}, \tag{8-47}$$

follows an F-distribution with $l-1$ and $(k-1)(l-1)$ degrees of freedom.

Example 8.23

Returning to our example we get:

```
# Test statistics
Ftr = SSTr / (3-1) / ( SSE / ((3-1) * (4-1)))
Fbl = SSBl / (4-1) / ( SSE / ((3-1) * (4-1)))
print(Ftr, Fbl)

74.39597315436248 6.367785234899335
```

```
# p-values
pv_tr = 1 - stats.f.cdf(Ftr, 3 - 1, (3 - 1) * (4 - 1))
pv_bl = 1 - stats.f.cdf(Fbl, 4 - 1, (3 - 1) * (4 - 1))
print(pv_tr, pv_bl)

5.823829718287765e-05 0.027048337827318747
```

or directly in Python:

```
fit = smf.ols(’y ∼ treatm + block’, data=D).fit()
anova = sm.stats.anova_lm(fit)
print(anova)

           df     sum_sq    mean_sq          F    PR(>F)
treatm    2.0  30.791667  15.395833  74.395973  0.000058
block     3.0   3.953333   1.317778   6.367785  0.027048
Residual  6.0   1.241667   0.206944        NaN       NaN
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
treatm	2	30.79	15.40	74.40	0.0001
block	3	3.95	1.32	6.37	0.0270
Residuals	6	1.24	0.21		

we see that the block effect is actually significant on a 5% confidence level, and also that the p-value for the treatment effect is changed (the evidence against $H_{0,Tr}$ is stronger) when we accounted for the block effect.

The test statistics and p-values are often collected in an analysis of variance table as already shown above:

Source of variation	Degrees of freedom	Sums of squares	Mean sums of squares	Test statistic F	p-value
Treatment	$k-1$	$SS(Tr)$	$MS(Tr) = \frac{SS(Tr)}{k-1}$	$F_{Tr} = \frac{MS(Tr)}{MSE}$	$P(F > F_{Tr})$
Block	$l-1$	$SS(Bl)$	$MS(Bl) = \frac{SS(Bl)}{l-1}$	$F_{Bl} = \frac{MS(Bl)}{MSE}$	$P(F > F_{Bl})$
Residual	$(l-1)(k-1)$	SSE	$MSE = \frac{SSE}{(k-1)(l-1)}$		
Total	$n-1$	SST			

8.3.3 Post hoc comparisons

The post hoc investigation is done following the same approach and principles as for one-way ANOVA with the following differences:

1. Use the MSE and/or SSE from the two-way analysis instead of the MSE and/or SSE from the one-way analysis
2. Use $(l-1)(k-1)$ instead of $n-k$ as degrees of freedom and as denominator for SSE

With these changes the Method boxes 8.9 and 8.10 and the Remark 8.13 can be used for post hoc investigation of treatment differences in a two-way ANOVA.

Example 8.24

Returning to our small example we now find the pairwise treatment confidence intervals within the two-way analysis. If the comparison of A and B was specifically planned before the experiment was carried out, we would find the 95%-confidence interval as:

```
print(muis[0] - muis[1] + np.array([-1,1]) *
      stats.t.ppf(0.975, (4-1)*(3-1)) * np.sqrt(SSE/((4-1) *
(3-1))*(1/4 + 1/4)))

[-3.662 -2.088]
```

and we can hence also conclude that treatment A is different from B. The p-value supporting this claim is found as:

```
tobs = (muis[0] - muis[1]) / np.sqrt(SSE/6 * (1/4 + 1/4))
print(2 * (1 - stats.t.cdf(abs(tobs), 6)))

0.00010947341433942227
```

If we do all three possible comparisons, $M = 3 \cdot 2/2 = 3$, and we will use an overall $\alpha = 0.05$, we do the above three times, but using each time $\alpha_{\text{Bonferroni}} = 0.05/3 = 0.017$:

```
alpha = alpha.values
alpha_bonf = 0.05 / 3
# A vs. B
print(alpha[0] - alpha[1] + np.array([-1, 1]) *
      stats.t.ppf(1 - alpha_bonf/2, 6) * np.sqrt(SSE/6 * (1/4 + 1/4)))


[-3.932 -1.818]


# A vs. C
print(alpha[0] - alpha[2] + np.array([-1, 1]) *
      stats.t.ppf(1 - alpha_bonf/2, 6) * np.sqrt(SSE/6 * (1/4 + 1/4)))


[-4.807 -2.693]


# B vs. C
print(alpha[1] - alpha[2] + np.array([-1, 1]) *
      stats.t.ppf(1 - alpha_bonf/2, 6) * np.sqrt(SSE/6 * (1/4 + 1/4)))


[-1.932  0.182]
```

and we conclude that treatment A is different from B and C, while we cannot reject that B and C are equal. The p-values for the last two comparisons could also be found, but we skip that.

8.3.4 Model control

Also model control runs almost exactly the same way for two-way ANOVA as for one-way:

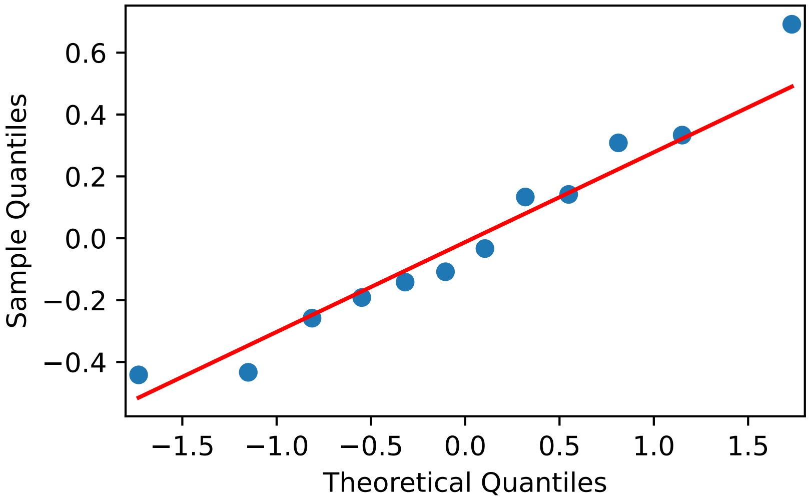
- Use a q-q plot on residuals to check for the normality assumption
- Check variance homegenity by categorized box plots

The only difference is that the box plotting to investigate variance homogeneity should be done on the residuals - NOT on the actual data. And that we can investigate both potential treatment heterogeneity as block heterogeneity.

|||| Example 8.25

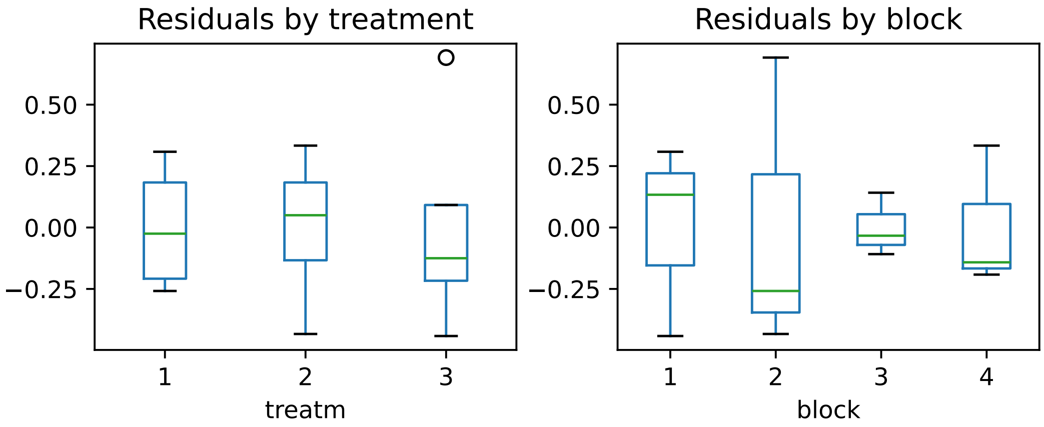
First the residual normality plot:

```
sm.qqplot(fit.resid.values, line='q', a=1/2)
plt.tight_layout()
plt.show()
```



Then the investigation of variance homogeneity:

```
D[’residuals’] = fit.resid.values # Add residuals to DataFrame
fig, ax = plt.subplots(ncols=2)
D.boxplot(column=’residuals’, by=’treatm’, ax=ax[0], grid=False)
ax[0].set_title(’Residuals by treatment’)
D.boxplot(column=’residuals’, by=’block’, ax=ax[1], grid=False,)
ax[1].set_title(’Residuals by block’)
plt.suptitle(’’)
plt.tight_layout()
plt.show()
```



Actually, if we’ve had data with a higher number of observations for each block, we might have had a problem here as blocks 2 and 3 appears to be quite different on their variability, however since there are very few observations (3 in each block) it is not unlikely to get this difference in variance when there is no difference (but again: it is not very easy to know, exactly where the limit is between what is OK and what is not OK in a situation like this. It is important information to present and take into the evaluation of the results, and in the process of drawing conclusions).

8.3.5 A complete worked through example: Car tires

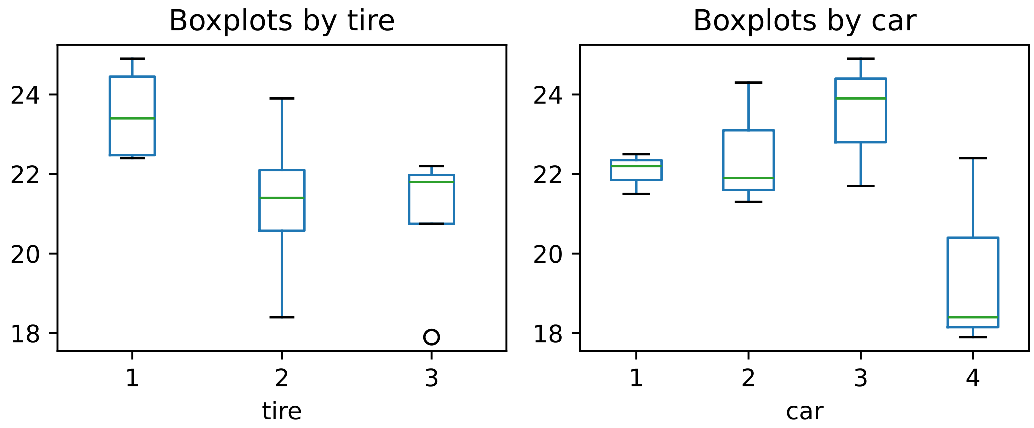
|||| Example 8.26 Car tires

In a study of 3 different types of tires (“treatment”) effect on the fuel economy, drives of 1000 km in 4 different cars ("blocks") were carried out. The results are listed in the following table in km/l.

	Car 1	Car 2	Car 3	Car 4	*Mean*
Tire 1	22.5	24.3	24.9	22.4	22.525
Tire 2	21.5	21.3	23.9	18.4	21.275
Tire 3	22.2	21.9	21.7	17.9	20.925
Mean	21.400	22.167	23.167	19.567	21.575

Let us analyse these data with a two-way ANOVA model, but first some explorative plotting:

```
# Collecting the data in a data frame
D = pd.DataFrame({
       'y': [22.5, 24.3, 24.9, 22.4,
              21.5, 21.3, 23.9, 18.4,
              22.2, 21.9, 21.7, 17.9],
       'car': pd.Categorical([1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]),
       'tire': pd.Categorical([1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3]),
})

fig, ax = plt.subplots(ncols=2)
D.boxplot(column='y', by='tire', ax=ax[0],grid=False)
ax[0].set_title('Boxplots by tire')
D.boxplot(column='y', by='car', ax=ax[1],grid=False)
ax[1].set_title('Boxplots by car')
plt.suptitle('')
plt.tight_layout()
plt.show()
```



Then the actual two-way ANOVA:

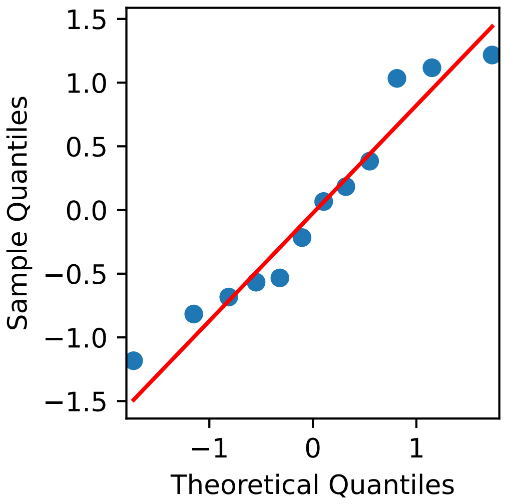
```
fit = smf.ols(’y ~ car + tire’, data=D).fit()
anova = sm.stats.anova_lm(fit)
print(anova)

           df     sum_sq   mean_sq         F    PR(>F)
car       3.0  25.175833  8.391944  7.025814  0.021726
tire      2.0  15.926667  7.963333  6.666977  0.029888
Residual  6.0   7.166667  1.194444       NaN       NaN
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
car	3	25.18	8.39	7.03	0.0217
tire	2	15.93	7.96	6.67	0.0299
Residuals	6	7.17	1.19		

Conclusion: Tires (treatments) are significantly different and Cars (blocks) are significantly different.

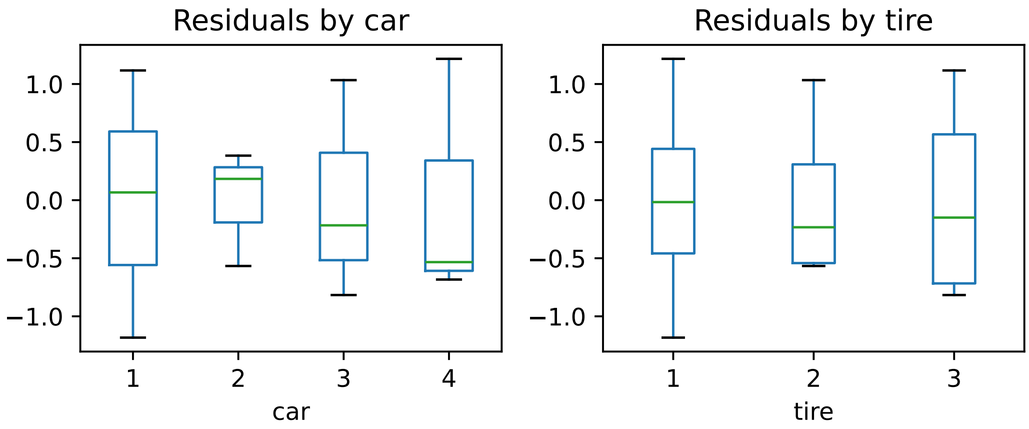
And the model control (for the conclusions to be validated). First the residual normality plot:

```
sm.qqplot(fit.resid.values, line=’q’, a=1/2)
plt.tight_layout()
plt.show()
```



Then the investigation of variance homogeneity:

```
D['residuals'] = fit.resid.values # Add residuals to DataFrame
fig, ax = plt.subplots(ncols=2)
D.boxplot(column='residuals', by='car', ax=ax[0],grid=False)
ax[0].set_title('Residuals by car')
D.boxplot(column='residuals', by='tire', ax=ax[1],grid=False)
ax[1].set_title('Residuals by tire')
plt.suptitle('')
plt.tight_layout()
plt.show()
```



It seems like the variance for Car 2 and Car 3 is difference, however, as in the previous example, there are very few observations (only 3) for each car, hence this difference in variation is not unlikely if there is no difference. Thus we find that there do not see any important deviations from the model assumptions.

Finally, the post hoc analysis, first the treatment means:

```
print(D.groupby('tire',observed=True)['y'].mean())

tire
1    23.525
2    21.275
3    20.925
Name: y, dtype: float64
```

We can find the 0.05/3 (Bonferroni-corrected) *LSD*-value from the two-way version of Remark 8.13:

```
LSD_bonf = stats.t.ppf(1-0.05/6, 6) * np.sqrt(2*1.19/4)
print(LSD_bonf)

2.5358194018640283
```

So tires are significantly different from each other if they differ by more than 2.54. A convenient way to collect the information about the 3 comparisons is by ordering the means from smallest to largest and then using the so-called compact letter display:

Tire	Mean		
3	20.925	a	
2	21.275	a	b
1	23.525		b

There is no significant difference between mean of Tire 2 and 3, and no significant difference between mean of 2 and 1, but there is significant difference between mean of 1 and 3.

8.4 Perspective

We have already seen how the R-version of the ANOVA, both one-way and two-way, are carried out by the R-function `lm`. We also used `lm` for simple and multiple linear regression (MLR) analysis in Chapters 5 and 6. “lm” stands for “linear model”, and in fact from a mathematical perspective all these models are what can be termed *linear models*, or sometimes *general linear models*. So differently put, the ANOVA models can in fact be expressed as multiple linear regression models, and the theory and matrix notation etc. from MLR can be used to also work with ANOVA models.

This becomes convenient to understand if one moves on to situations, models and statistical analysis going beyond the current course. An example of this would be situations where we have as well factors as quantitative (continuous) regression input in the same data set.

Important to know also is that the two basic ANOVA versions presented in this material is just the start to be able to handle more general situations. An example could be that, a two-way ANOVA could also occur in a different way than shown here: if we perform what would be a completely randomized study, that is, we have independent sampled groups, but with the groups being represented by a two-way treatment factor structure, say, factor A with 5 levels and factor B with 3 levels. Hence, we have all 15 groups consisting of all combinations of the two treatments, but with several observations within each of the 15 groups. This would sometimes be called a two-way ANOVA with replications, whereas the randomized block setting covered above then would be the two-way ANOVA without replication (there is only and exactly one observation for each combination of treatment and block).

And then the next step could be even more than two treatment factors, and maybe such a multi-factorial setting could even be combined with blocking and maybe some quantitative x-input (then often called covariates) calling for extensions of all this.

Another important extension direction are situations with different levels of observations/variability: there could be hierarchical structures in the data, e.g. repeated measurement on an individual animal, but having also many animals in the study, and animals might come from different farms, that lies in different regions within different countries. This calls for so-called hierarchical models, multi-level models, variance components models or models, where both treatment factors and such hierarchical random effects are present – the so-called mixed models.

All of this and many other good things can be learned in statistics courses building further on the methods presented in this material!

Glossaries

Alternative hypothesis [Alternativ hypotese] The alternative hypothesis (H_1) is oftern the negation of the null hypothesis 143, 145, 146, 163, 179, 257, 279, 288

Binomial distribution [Binomial fordeling] If an experiment has two possible outcomes (e.g. failure or success, no or yes, 0 or 1) and is repeated more than one time, then the number of successes is binomial distributed 57, 59, 60, 275, 276, 288, 343

Block [Blok] The block name comes from the historical background of agricultural field trials, where a block would be an actual piece of land within which all treatments are applied 327, 328

Box plot [Box plot] The so-called boxplot in its basic form depicts the five quartiles (min, Q1 , median, Q3 , max) with a box from Q1 to Q3 emphasizing the Inter Quartile Range (IQR) 27, 30–32, 36

Categorical data [Kategorisk data] A variable is called categorical if each observation belongs to one of a set of categories 1, 27

cumulated distribution function [Fordelingsfunktion]The cdf is the function which determines the probability of observing an outcome of a random variable below a given value **350**,

χ^2**-distribution** [χ^2-fordeling (udtales: chi-i-anden fordeling)] 96, 98, 132, 133, 293

confidence interval [Konfidensinterval] The confidence interval is a way to handle the uncertainty by the use of probability theory. The confidence interval represents those values of the unknown population mean μ that we believe is based on the data. Thus we believe the true mean in the statistics class is in this interval 122,

Class The frequency distribution of the data for a certain grouping of the data 27, 29

Central Limit Theorem [Centrale grænseværdisætning] The Central Limit Theorem (CLT) states that the sample mean of independent identically distributed outcomes converges to a normal distribution 126

Continuity correction The so-called Continuity correction is a general approach to make the best approximation of discrete probabilities 278

Continuous random variable [Kontinuert stokastisk variabel] If an outcome of an experiment takes a continuous value, for example: a distance, a temperature, a weight, etc., then it is represented by a continuous random variable 42, 66, 68, 92, 350

Correlation [Korrelation] The sample correlation coefficient are a summary statistic that can be calculated for two (related) sets of observations. It quantifies the (linear) strength of the relation between the two. See also: Covariance 16–20, 23, 87, 88, 244–246, 265, 267, 343, 349

Covariance [Kovarians] The sample covariance coefficient are a summary statistic that can be calculated for two (related) sets of observations. It quantifies the (linear) strength of the relation between the two. See also: Correlation 16–20, 23, 87–89, 224, 225, 228, 230, 236, 241, 242, 259, 272, 343, 349, 351

Critical value *Kritisk værdi* As an alternative to the p-value one can use the so-called critical values, that is the values of the test-statistic which matches exactly the significance level 141–143, 145, 163, 169, 233, 281, 286

Degrees of freedom [Frihedsgrader] The number of "observations" in the data that are free to vary when estimating statistical parameters often defined as $n-1$ 96, 98, 101, 108, 118, 119, 122, 130, 132, 133, 137, 141, 145, 161–163, 165, 167, 169, 177, 225, 232, 234, 236, 238, 255, 257, 260, 289, 290, 293, 294, 296, 300

Descriptive statistics [Beskrivende statistik] Descriptive statistics, or explorative statistics, is an important part of statistics, where the data is summarized and described 1, 4, 8

Discrete random variable [Diskret stokastisk variabel] A discrete random variable has discrete outcomes and follows a discrete distribution 43, 50, 53, 90, 91

Distribution [Fordeling] Defines how the data is distributed such as, normal distribution, cumulated distribution function, probability density function exponential distribution, log-normal distribution, Poisson distribution, uniform distribution, hypergeometric distribution, binomial distribution, t-distribution, F-distribution 42

Empirical cumulative distribution [Empirisk fordeling] The empirical cumulative distribution function F_n is a step function with jumps i/n at observation values, where i is the number of identical observations at that value 29, 30, 150, 207

Expectation [Forventningsværdi] A function for calculating the mean. The value we expect for a random variable (or function of random variables), hence of the population 51, 108, 214, 217, 223, 224, 242

Exponential distribution [Eksponential fordelingen] The usual application of the exponential distribution is for describing the length (usually time) between events which, when counted, follows a Poisson distribution 77, 78, 82, 193, 194, 196, 204, 343

***F*-distribution** [*F*-fordelingen] The *F*-distribution appears as the ratio between two independent χ^2-distributed random variables 108, 343, 363

Frequency [Frekvens] How frequent data is observed. The frequency distribution of the data for a certain grouping is nicely depicted by the histogram, which is a barplot of either raw frequencies or for some number of classes 27–29, 38

Histogram [Histogram] The default histogram uses the same width for all classes and depicts the raw frequencies/counts in each class. By dividing the raw counts by n times the class width the density histogram is found where the area of all bars sum to 1 27–29, 32, 46, 76, 114, 149, 150, 186, 195, 198, 200, 222

Hypergeometric distribution [Hypergeometrisk fordeling] 60, 61, 343

Independence [Uafhængighed] 87, 90–92, 126, 149, 228, 251, 297–301

Independent samples [Uafhængige stikprøver] 172–174

(Statistical) Inference [Statistisk inferens (følgeslutninger baseret på data)] 5, 95, 113, 124, 127, 215, 232, 247

Interval [Interval] Data in a specified range 62–64, 79, 81, 114

Inter Quartile Range [Interkvartil bredde] The Inter Quartile Range (IQR) is the middle 50% range of data 15, 349

Least squares [Mindste kvadraters (metode)] 217, 218, 220

Linear regression [Lineær regression (-sanalyse)] 1, 20, 215, 220, 223, 232, 241, 245, 254, 257, 259, 263, 266, 267, 272

Log-normal distribution [Lognormal fordeling] 77, 343

Maximum likelihood [Estimator baseret på maximum likelihood metoden] 194, 197, 202

Median [Median, stikprøvemedian] The median of population or sample (note, in text no distinguishment between *population median* and *sample median*) 8, 10, 11, 24, 157, 193, 196, 197, 348

Multiple linear regression [Multipel lineær regression (-sanalyse)] 1, 252, 255, 258, 259, 272

Non-parametric (test) [Ikke-parametriske (tests)] 186, 187, 193, 207, 210

Normal distribution [Normal fordeling] 66, 70, 73, 74, 77, 93, 96, 99, 101, 106, 107, 110, 114–119, 121, 124, 127, 129, 131, 149–156, 177, 180, 185–187, 193, 194, 201, 204, 230, 343

Null hypothesis [Nulhypotese (H_0)] 134–137, **138**, 139–143, 145, 146, 148, 160, 162–165, 167, 169, 232, 234, 255, 257, 263, 279–281, 285–290, 296, 297

One-sample t-test Missing description 141, 144, 145, 179

One-sided (test) [Énsidet test] Is also called directional (test) 179, 279

***P*-value** [p-værdi (for faktisk udfald af en teststørrelse)] 99, 135–139, 141, 144, 148, 160, 167, 171, 175, 233, 234, 257, 266, 282, 287, 290, 292, 296, 297, 300, 317, 332, 334, 343

probability density function The pdf is the function which determines the probability of every possible outcome of a random variable **350**,

Poisson distribution [Poisson fordeling] 343, 344

Quantile [Fraktil, stikprøvefraktil] The quantiles of population or sample (note, in text no distinguishment between *population quantile* and *sample quantile*) 11, 348

Quartile [Fraktil, stikprøvefraktil] The quartiles of population or sample (note, in text no distinguishment between *population quartile* and *sample quartile*) 12, 348

Sample variance [Empirisk varians, stikprøvevarians] 13, 349

Sample mean [Stikprøvegennemsnit] The average of a sample 9, 10, 14, 24, 49–52, 54, 87, 101, 103, 105, 106, 113–118, 120, 121, 123, 126, 127, 131, 134, 159, 178, 348

Significance level A number α (often 0.05) which is used to quantify precision or uncertainty 343

Standard deviation [Standard afvigelse] 349

Standard normal distribution [Standardiseret normalfordeling ($N(0,1)$)] 281

t-distribution [t-fordeling] 101, 343

Two-sided (test) [Tosidet test (test med tosidet alternativ)] Is also called non-directional (test) 168, 233, 279, 285

Uniform distribution [Uniform (rektangulær) fordeling] 343

Acronyms

ANOVA Analysis of Variance 164, 172, 303, 307, 313, 314, 319, 320, 323, 325, 327, 329, 330, 333, 335, 337, 340, *Glossary:* Analysis of Variance

cdf cumulated distribution function 44, 343, 350, *Glossary:* cumulated distribution function

CI confidence interval 114, 115, 121–124, 129, 131, 133, 134, 139, 142, 143, 148, 155, 157, 159–161, 168–172, 175, 177, 178, 193, 194, 196–199, 202, 206, 207, 210, 224, 232, 234–236, 238, 245, 255, 257, 258, 260, 263, 273, 276–278, 284, 285, 314, 316, 318, 324, 325, 333, 357–359, 362, *Glossary:* confidence interval

CLT Central Limit Theorem 126, 127, 129, 196, *Glossary:* Central Limit Theorem

IQR Inter Quartile Range 8, 15, 16, 30, 31, 197, 342, 349, *Glossary:* Inter Quartile Range

LSD Least Significant Difference *Glossary:* Least Significant Difference

pdf probability density function 343, 350, *Glossary:* probability density function

Appendix A

Collection of formulas and commands

This appendix chapter holds a collection of formulas. All the relevant equations from definitions, methods and theorems are included – along with associated Python commands. All are in included in the same order as in the book, except for the distributions which are listed together.

A.1 Introduction, descriptive statistics, commands and data visualization

	Description	Formula	Command
1.4	**Sample mean** The mean of a sample.	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$	`x.mean()`
1.5	**Sample median** The value that divides a sample in two halves with equal number of observations in each.	$Q_2 = \begin{cases} x_{(\frac{n+1}{2})} & \text{for odd } n \\ \frac{x_{(\frac{n}{2})}+x_{(\frac{n+2}{2})}}{2} & \text{for even } n \end{cases}$	`median(x)`
1.7	**Sample quantile** The value that divide a sample such that p of the observations are less that the value. The 0.5 quantile is the Median.	$q_p = \begin{cases} \frac{x_{(np)}+x_{(np+1)}}{2} & \text{for } pn \text{ integer} \\ x_{(\lceil np \rceil)} & \text{for } pn \text{ non-integer} \end{cases}$	`quantile(x,p,type=2),`

1.8	**Sample quartiles**
	The quartiles are the five quantiles dividing the sample in four parts, such that each part holds an equal number of observations
		$Q_0 = q_0 =$ "minimum"
		$Q_1 = q_{0.25} =$ "lower quartile"
		$Q_2 = q_{0.5} =$ "median"
		$Q_3 = q_{0.75} =$ "upper quartile"
		$Q_4 = q_1 =$ "maximum"
			`quantile(x,`
			`probs,type=2)`
			where
			`probs=`p

	Description	Formula	Command
1.10	**Sample variance** The sum of squared differences from the mean divided by $n-1$.	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$	`var(x)`
1.11	**Sample standard deviation** The square root of the sample variance.	$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$	`sd(x)`
1.12	**Sample coefficient of variance** The sample standard deviation seen relative to the sample mean.	$V = \frac{s}{\bar{x}}$	`sd(x)/mean(x)`
1.15	**Sample Inter Quartile Range** IQR: The middle 50% range of data	$IQR = Q_3 - Q_1$	`IQR(x, type=2)`
1.18	**Sample covariance** Measure of linear strength of relation between two samples	$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$	`cov(x,y)`
1.19	**Sample correlation** Measure of the linear strength of relation between two samples between -1 and 1.	$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s_x}\right)\left(\frac{y_i-\bar{y}}{s_y}\right) = \frac{s_{xy}}{s_x \cdot s_y}$	`cor(x,y)`

A.2 Probability and Simulation

	Description	Formula	Command

2.6	**Probability density function (pdf) for a discrete variable**
	fulfills two conditions: $f(x) \geq 0$ and $\sum_{\text{all } x} f(x) = 1$ and finds the probality for one x value.
		$f(x) = P(X = x)$	`dnorm,dbinom,dhyper,`
			`dpois`

2.9	**Cumulated distribution function (cdf)**
	gives the probability in a range of x values where $P(a < X \leq b) = F(b) - F(a)$.
		$F(x) = P(X \leq x)$	`pnorm,pbinom,phyper,`
			`ppois`

2.13	**Mean of a discrete random variable**	$\mu = E(X) = \sum_{i=1}^{\infty} x_i f(x_i)$	
2.16	**Variance of a discrete random variable X**	$\sigma^2 = \text{Var}(X) = E[(X - \mu)^2]$	
2.32	**Pdf of a continuous random variable** is a non-negative function for all possible outcomes and has an area below the function of one	$P(a < X \leq b) = \int_a^b f(x)dx$	
2.33	**Cdf of a continuous random variable** is non-decreasing and $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$	$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du$	
2.34	**Mean and variance for a continuous random variable X**	$\mu = E(X) = \int_{-\infty}^{\infty} x f(x)dx$ $\sigma^2 = E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 f(x)dx$	
2.54	**Mean and variance of a linear function** The mean and variance of a linear function of a random variable X.	$\text{E}(aX + b) = a\,\text{E}(X) + b$ $\text{V}(aX + b) = a^2\,\text{V}(X)$	
2.56	**Mean and variance of a linear combination** The mean and variance of a linear combination of random variables.	$\text{E}(a_1 X_1 + a_2 X_2 + \cdots + a_n X_n) =$ $a_1\,\text{E}(X_1) + a_2\,\text{E}(X_2) + \cdots + a_n\,\text{E}(X_n)$ $\text{V}(a_1 X_1 + a_2 X_2 + \ldots + a_n X_n) =$ $a_1^2\,\text{V}(X_1) + a_2^2\,\text{V}(X_2) + \cdots + a_n^2\,\text{V}(X_n)$	

	Description	Formula	Command
2.58	**Covariance** The covariance between be two random variables X and Y.	$\mathrm{Cov}(X,Y) = E\left[(X - E[X])(Y - E[Y])\right]$	

A.2.1 Distributions

Here all the included distributions are listed including some important theorems and definitions related specifically with a distribution.

	Description	Formula	Command

2.20	**Binominal distribution**
	n is the number of independent draws and p is the probability of a success in each draw. The Binominal pdf describes the probability of x succeses.
		$f(x;n,p) = P(X=x)$
		$= \binom{n}{x} p^x (1-p)^{n-x}$
		where $\binom{n}{x} = \frac{n!}{x!(n-x)!}$
			`dbinom(x, size, prob)`
			`pbinom(q, size, prob)`
			`qbinom(p, size, prob)`
			`rbinom(n, size, prob)`
			where
			`size=`n`, prob=`p

2.21	Mean and variance of a binomial distributed random variable.	$\mu = np$ $\sigma^2 = np(1-p)$	

2.24	**Hypergeometric distribution**
	n is the number of draws without replacement, a is number of succeses and N is the population size.
		$f(x;n,a,N) = P(X=x)$
		$= \frac{\binom{a}{x}\binom{N-a}{n-x}}{\binom{N}{n}}$
		where $\binom{a}{b} = \frac{a!}{b!(a-b)!}$
			`dhyper(x,m,n,k)`
			`phyper(q,m,n,k)`
			`qhyper(p,m,n,k)`
			`rhyper(nn,m,n,k)`
			where
			`m=`a`, n=`$N-a$`, k=`n

2.25	Mean and variance of a hypergeometric distributed random variable.	$\mu = n\frac{a}{N}$ $\sigma^2 = n\frac{a(N-a)}{N^2}\frac{N-n}{N-1}$	

2.27	**Poisson distribution**
	λ is the rate (or intensity) i.e. the average number of events per interval. The Poisson pdf describes the probability of x events in an interval.
		$f(x;\lambda) = \frac{\lambda^x}{x!}e^{-\lambda}$	`dpois(x,lambda)`
			`ppois(q,lambda)`
			`qpois(p,lambda)`
			`rpois(n,lambda)`
			where
			`lambda=`λ

2.28	Mean and variance of a Poisson distributed random variable.	$\mu = \lambda$ $\sigma^2 = \lambda$	

2.35	**Uniform distribution**
	α and β defines the range of possible outcomes. random variable following the uniform distribution has equal density at any value within a defined range.
		$f(x;\alpha,\beta) = \begin{cases} 0 & \text{for } x < \alpha \\ \frac{1}{\beta-\alpha} & \text{for } x \in [\alpha,\beta] \\ 0 & \text{for } x > \beta \end{cases}$
		$F(x;\alpha,\beta) = \begin{cases} 0 & \text{for } x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \text{for } x \in [\alpha,\beta] \\ 0 & \text{for } x > \beta \end{cases}$
			`dunif(x,min,max)`
			`punif(q,min,max)`
			`qunif(p,min,max)`
			`runif(n,min,max)`
			where
			`min=`α`, max=`β

	Description	Formula	Command
2.36	Mean and variance of a uniform distributed random variable X.	$\mu = \frac{1}{2}(\alpha + \beta)$ $\sigma^2 = \frac{1}{12}(\beta - \alpha)^2$	

2.37	**Normal distribution**
	Often also called the Gaussian distribution.
		$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	`dnorm(x,mean,sd)`
			`pnorm(q,mean,sd)`
			`qnorm(p,mean,sd)`
			`rnorm(n,mean,sd)`
			where
			`mean=`μ`, sd=`σ.

2.38	Mean and variance of a normal distributed random variable.	μ σ^2	
2.43	Transformation of a normal distributed random variable X into a standardized normal random variable.	$Z = \frac{X - \mu}{\sigma}$	

2.46	**Log-normal distribution**
	α is the mean and β^2 is the variance of the normal distribution obtained when taking the natural logarithm to X.
		$f(x) = \frac{1}{x\sqrt{2\pi}\beta} e^{-\frac{(\ln x - \alpha)^2}{2\beta^2}}$	`dlnorm(x,meanlog,sdlog)`
			`plnorm(q,meanlog,sdlog)`
			`qlnorm(p,meanlog,sdlog)`
			`rlnorm(n,meanlog,sdlog)`
			where
			`meanlog=`α`, sdlog=`β.

2.47	Mean and variance of a log-normal distributed random variable.	$\mu = e^{\alpha + \beta^2/2}$ $\sigma^2 = e^{2\alpha + \beta^2}(e^{\beta^2} - 1)$	

2.48	**Exponential distribution**
	λ is the mean rate of events.
		$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$	`dexp(x,rate)`
			`pexp(q,rate)`
			`qexp(p,rate)`
			`rexp(n,rate)`
			where
			`rate=`λ.

2.49	Mean and variance of a exponential distributed random variable.	$\mu = \frac{1}{\lambda}$ $\sigma^2 = \frac{1}{\lambda^2}$	

2.78	**χ^2-distribution**
	$\Gamma\left(\frac{\nu}{2}\right)$ is the Γ-function and ν is the degrees of freedom.
		$f(x) = \frac{1}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)} x^{\frac{\nu}{2}-1} e^{-\frac{x}{2}}; \quad x \geq 0$	`dchisq(x,df)`
			`pchisq(q,df)`
			`qchisq(p,df)`
			`rchisq(n,df)`
			where
			`df=`ν.

	Description	Formula	Command
2.81	Given a sample of size n from the normal distributed random variables X_i with variance σ^2, then the sample variance S^2 (viewed as random variable) can be transformed to follow the χ^2 distribution with the degrees of freedom $\nu = n - 1$.	$\chi^2 = \frac{(n-1)S^2}{\sigma^2}$	
2.83	Mean and variance of a χ^2 distributed random variable.	$E(X) = \nu$ $V(X) = 2\nu$	
2.86	***t*-distribution** ν is the degrees of freedom and $\Gamma()$ is the Gamma function.	$f_T(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	

2.87	Relation between normal random variables and χ^2-distributed random variables. $Z \sim N(0,1)$ and $Y \sim \chi^2(\nu)$.	$X = \frac{Z}{\sqrt{Y/\nu}} \sim t(\nu)$	`dt(x,df)`
			`pt(q,df)`
			`qt(p,df)`
			`rt(n,df)`
			where
			`df=`ν.

2.89	For normal distributed random variables $X_1, \ldots, X_n$, the random variable follows the t-distribution, where $\overline{X}$ is the sample mean, μ is the mean of X, n is the sample size and S is the sample standard deviation.	$T = \frac{\overline{X} - \mu}{S/\sqrt{n}} \sim t(n-1)$	
2.93	Mean and variance of a t-distributed variable X.	$\mu = 0; \quad \nu > 1$ $\sigma^2 = \frac{\nu}{\nu - 2}; \quad \nu > 2$	

2.95	***F*-distribution**
	ν_1 an ν_2 are the degrees of freedom and $B(\cdot,\cdot)$ is the Beta function.
		$f_F(x) = \frac{1}{B\left(\frac{\nu_1}{2}, \frac{\nu_2}{2}\right)} \left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}} \cdot x^{\frac{\nu_1}{2}-1} \left(1 + \frac{\nu_1}{\nu_2}x\right)^{-\frac{\nu_1+\nu_2}{2}}$	`df(x,df1,df2)`
			`pf(q,df1,df2)`
			`qf(p,df1,df2)`
			`rf(n,df1,df2)`
			where
			`df1=`ν_1`,df2=`μ_2.

2.96	The F-distribution appears as the ratio between two independent χ^2-distributed random variables with $U \sim \chi^2(\nu_1)$ and $V \sim \chi^2(\nu_2)$.	$\frac{U/\nu_1}{V/\nu_2} \sim F(\nu_1, \nu_2)$	

	Description	Formula	Command
2.98	$X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ with the mean μ_1 and μ_2 and the variance σ_1^2 and σ_2^2 is independent and sampled from a normal distribution.	$\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \sim F(n_1 - 1, n_2 - 1)$	
2.101	Mean and variance of a F-distributed variable X.	$\mu = \frac{\nu_2}{\nu_2 - 2}; \quad \nu_2 > 2$ $\sigma = \frac{2\nu_2^2(\nu_1 + \nu_2 - 2)}{\nu_1(\nu_2 - 2)^2(\nu_2 - 4)}; \quad \nu_2 > 4$	

A.3 Statistics for one and two samples

	Description	Formula	Command
3.3	The distribution of the mean of normal random variables.	$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \sim N\left(\mu, \frac{\sigma^2}{n}\right)$	
3.5	The distribution of the σ-*standardized* mean of normal random variables	$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N\left(0, 1^2\right)$	
3.5	The distribution of the S-*standardized* mean of normal random variables	$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1)$	
3.7	Standard Error of the mean	$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$	
3.9	The one sample confidence interval for μ	$\bar{x} \pm t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}$	
3.14	Central Limit Theorem (CLT)	$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$	
3.19	Confidence interval for the variance and standard deviation	$\sigma^2 : \left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}; \frac{(n-1)s^2}{\chi^2_{\alpha/2}}\right]$ $\sigma : \left[\sqrt{\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}}; \sqrt{\frac{(n-1)s^2}{\chi^2_{\alpha/2}}}\right]$	
3.22	The p-value	**The p-value** is the probability of obtaining a test statistic that is at least as extreme as the test statistic that was actually observed. This probability is calculated under the assumption that the null hypothesis is true.	`P(T>x)=2(1-pt(x,n-1))`
3.23	The one-sample t-test statistic and p-value	$p\text{-value} = 2 \cdot P(T > \lvert t_{\text{obs}} \rvert)$ $t_{\text{obs}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$ $H_0 : \mu = \mu_0$	
3.24	The hypothesis test	Rejected: p-value $< \alpha$ Accepted: *otherwise*	
3.29	Significant effect	An effect is significant if the p-value$< \alpha$	
3.31	The critical values: $\alpha/2$- and $1-\alpha/2$-quantiles of the t-distribution with $n-1$ degrees of freedom	$t_{\alpha/2}$ and $t_{1-\alpha/2}$	

	Description	Formula	Command
3.32	The one-sample hypothesis test by the critical value	Reject: $\lvert t_{\text{obs}}\rvert > t_{1-\alpha/2}$ accept: *otherwise*	
3.33	Confidence interval for μ	$\bar{x} \pm t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}$ acceptance region/CI: $H_0 : \quad \mu = \mu_0$	
3.36	The level α one-sample t-test	Test: $H_0 : \mu = \mu_0$ and $H_1 : \mu \neq \mu_0$ by p-value $= 2 \cdot P(T > \lvert t_{\text{obs}}\rvert)$ Reject: p-value $< \alpha$ or $\lvert t_{\text{obs}}\rvert > t_{1-\alpha/2}$ Accept: *Otherwise*	
3.63	The one-sample confidence interval (CI) sample size formula	$n = \left(\frac{z_{1-\alpha/2}\cdot\sigma}{ME}\right)^2$	
3.65	The one-sample sample size formula	$n = \left(\sigma\frac{z_{1-\beta}+z_{1-\alpha/2}}{(\mu_0-\mu_1)}\right)^2$	
3.42	The Normal q-q plot with $n > 10$	naive approach: $p_i = \frac{i}{n},\ i = 1,\ldots,n$ commonly aproach: $p_i = \frac{i-0.5}{n+1},\ i = 1,\ldots,n$	
3.49	The (Welch) two-sample t-test statistic	$\delta = \mu_2 - \mu_1$ $H_0 :\ \delta = \delta_0$ $t_{\text{obs}} = \frac{(\bar{x}_1-\bar{x}_2)-\delta_0}{\sqrt{s_1^2/n_1+s_2^2/n_2}}$	
3.50	The distribution of the (Welch) two-sample statistic	$T = \frac{(\bar{X}_1-\bar{X}_2)-\delta_0}{\sqrt{S_1^2/n_1+S_2^2/n_2}}$ $\nu = \frac{\left(\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1-1}+\frac{(s_2^2/n_2)^2}{n_2-1}}$	
3.51	The level α two-sample t-test	Test: $H_0 : \mu_1 - \mu_2 = \delta_0$ and $H_1 : \mu_1 - \mu_2 \neq \delta_0$ by p-value $= 2 \cdot P(T > \lvert t_{\text{obs}}\rvert)$ Reject: p-value $< \alpha$ or $\lvert t_{\text{obs}}\rvert > t_{1-\alpha/2}$ Accept: *Otherwise*	
3.52	The pooled two-sample estimate of variance	$s_p^2 = \frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}$	
3.53	The pooled two-sample t-test statistic	$\delta = \mu_1 - \mu_2$ $H_0 :\ \delta = \delta_0$ $t_{\text{obs}} = \frac{(\bar{x}_1-\bar{x}_2)-\delta_0}{\sqrt{s_p^2/n_1+s_p^2/n_2}}$	
3.54	The distribution of the pooled two-sample t-test statistic	$T = \frac{(\bar{X}_1-\bar{X}_2)-\delta_0}{\sqrt{S_p^2/n_1+S_p^2/n_2}}$	
3.47	The two-sample confidence interval for $\mu_1 - \mu_2$	$\bar{x} - \bar{y} \pm t_{1-\alpha/2} \cdot \sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}$ $\nu = \frac{\left(\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1-1}+\frac{(s_2^2/n_2)^2}{n_2-1}}$	

A.4 Simulation based statistics

	Description	Formula	Command
4.3	The non-linear approximative error propagation rule	$\sigma^2_{f(X_1,\ldots,X_n)} = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_i^2$	
4.4	Non-linear error propagation by simulation	1. Simulate k outcomes 2. Calculate the standard deviation by $s^{\text{sim}}_{f(X_1,\ldots,X_n)} = \sqrt{\frac{1}{k-1}\sum_{i=1}^k (f_j - \bar{f})^2}$	
4.7	Confidence interval for any feature θ by parametric bootstrap	1.Simulate k samples 2.Calculate the statistic $\hat{\theta}$ 3.Calculate CI: $\left[q^*_{100(\alpha/2)\%}, \, q^*_{100(1-\alpha/2)\%}\right]$	
4.10	Two-sample confidence interval for any feature comparison $\theta_1 - \theta_2$ by parametric bootstrap	1.Simulate k sets of 2 samples 2.Calculate the statistic $\hat{\theta}^*_{xk} - \hat{\theta}^*_{yk}$ 3.Calculate CI: $\left[q^*_{100(\alpha/2)\%}, \, q^*_{100(1-\alpha/2)\%}\right]$	

A.5 Simple linear regression

	Description	Formula	Command
5.4	Least square estimators	$\hat{\beta}_1 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})(x_i - \bar{x})}{S_{xx}}$ $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$ where $S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2$	
5.8	Variance of estimators	$\mathrm{V}[\hat{\beta}_0] = \frac{\sigma^2}{n} + \frac{\bar{x}^2\sigma^2}{S_{xx}}$ $\mathrm{V}[\hat{\beta}_1] = \frac{\sigma^2}{S_{xx}}$ $\mathrm{Cov}[\hat{\beta}_0, \hat{\beta}_1] = -\frac{\bar{x}\sigma^2}{S_{xx}}$	
5.12	Tests statistics for $H_0 : \beta_0 = 0$ and $H_0 : \beta_1 = 0$	$T_{\beta_0} = \frac{\hat{\beta}_0 - \beta_{0,0}}{\hat{\sigma}_{\beta_0}}$ $T_{\beta_1} = \frac{\hat{\beta}_1 - \beta_{0,1}}{\hat{\sigma}_{\beta_1}}$	

5.14	Level α t-tests for parameter	Test $H_{0,i} : \beta_i = \beta_{0,i}$ vs. $H_{1,i} : \beta_i \neq \beta_{0,i}$ with p-value $= 2 \cdot P(T > \lvert t_{\text{obs},\beta_i} \rvert)$ where $t_{\text{obs},\beta_i} = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\beta_i}}$.
		If p-value $< \alpha$ then *reject* H_0, otherwise *accept* H_0
			`D <- data.frame(`
			`x=c(), y=c())`
			`fit <- lm(y~x, data=D)`
			`summary(fit)`

5.15	Parameter confidence intervals	$\hat{\beta}_0 \pm t_{1-\alpha/2}\, \hat{\sigma}_{\beta_0}$ $\hat{\beta}_1 \pm t_{1-\alpha/2}\, \hat{\sigma}_{\beta_1}$	`confint(fit,level=0.95)`

5.18	Confident and prediction interval	Confidence interval for the line:
		$\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \pm t_{1-\alpha/2}\hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}}$
		Interval for a new point prediction:
		$\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \pm t_{1-\alpha/2}\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}}$
			`predict(fit,`
			`newdata=data.frame(),`
			`interval="confidence",`
			`level=0.95)`
			`predict(fit,`
			`newdata=data.frame(),`
			`interval="prediction",`
			`level=0.95)`

5.23	The matrix formulation of the parameter estimators in the simple linear regression model	$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ $V[\hat{\boldsymbol{\beta}}] = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ $\hat{\sigma}^2 = \frac{RSS}{n-2}$	
5.25	Coefficient of determination R^2	$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$	

	Description	Formula	Command

5.7	Model validation of assumptions	> Check the normality assumption with a q-q plot of the residuals.
		> Check the systematic behavior by plotting the residuals e_i as a function of fitted values $\hat{y}_i$
			`qqnorm(fit$residuals)`
			`qqline(fit$residuals)`
			`plot(fit$fitted.values, fit$residuals)`

A.6 Multiple linear regression

	Description	**Formula**	**Command**

6.2	Level α t-tests for parameter	Test $H_{0,i} : \beta_i = \beta_{0,i}$ vs. $H_{1,i} : \beta_i \neq \beta_{0,i}$ with p-value $= 2 \cdot P(T > \lvert t_{\text{obs},\beta_i} \rvert)$ where $t_{\text{obs},\beta_i} = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\beta_i}}$.
		If p-value $< \alpha$ the *reject* H_0,
		otherwise *accept* H_0
			`D<-data.frame(x1=c(),`
			`  x2=c(),y=c())`
			`fit <- lm(y~x1+x2,`
			`  data=D)`
			`summary(fit)`

6.5	Parameter confidence intervals	$\hat{\beta}_i \pm t_{1-\alpha/2}\, \hat{\sigma}_{\beta_i}$	`confint(fit,level=0.95)`

6.9	Confident and prediction interval (in R)	Confident interval for the line
		$\hat{\beta}_0 + \hat{\beta}_1 x_{1,\text{new}} + \cdots + \hat{\beta}_p x_{p,\text{new}}$
		
		Interval for a new point prediction
		$\hat{\beta}_0 + \hat{\beta}_1 x_{1,\text{new}} + \cdots + \hat{\beta}_p x_{p,\text{new}} + \varepsilon_{\text{new}}$
			`predict(fit,`
			`  newdata=data.frame(),`
			`  interval="confidence",`
			`  level=0.95)`
			`predict(fit,`
			`  newdata=data.frame(),`
			`  interval="prediction",`
			`  level=0.95)`

6.17	The matrix formulation of the parameter estimators in the multiple linear regression model	$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ $V[\hat{\boldsymbol{\beta}}] = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ $\hat{\sigma}^2 = \dfrac{RSS}{n-(p+1)}$	
6.16	Model selection procedure	Backward selection: start with full model and stepwise remove insignificant terms	

A.7 Inference for proportions

	Description	**Formula**	**Command**

7.3	Proportion estimate and confidence interval	$\hat{p} = \frac{x}{n}$
		$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$
			`prop.test(x=, n=,`
			`correct=FALSE)`

7.10	Approximate proportion with Z	$Z = \frac{X-np_0}{\sqrt{np_0(1-p_0)}} \sim N(0,1)$	

7.11	The level α one-sample proportion hypothesis test	Test: $H_0 : p = p_0$, vs. $H_1 : p \neq p_0$
		by p-value $= 2 \cdot P(Z > \lvert z_{\text{obs}} \rvert)$
		where $Z \sim N(0, 1^2)$
		If p-value $< \alpha$ the *reject* H_0,
		otherwise *accept* H_0
			`prop.test(x=, n=,`
			`correct=FALSE)`

7.13	Sample size formula for the CI of a proportion	Guessed p (with prior knowledge): $n = p(1-p)(\frac{z_{1-\alpha/2}}{ME})^2$ Unknown p: $n = \frac{1}{4}(\frac{z_{1-\alpha/2}}{ME})^2$	
7.15	Difference of two proportions estimator $\hat{p}_1 - \hat{p}_2$ and confidence interval for the difference	$\hat{\sigma}_{\hat{p}_1-\hat{p}_2} = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$ $(\hat{p}_1 - \hat{p}_2) \pm z_{1-\alpha/2} \cdot \hat{\sigma}_{\hat{p}_1-\hat{p}_2}$	

7.18	The level α one-sample t-test	Test: $H_0 : p_1 = p_2$, vs. $H_1 : p_1 \neq p_2$
		by p-value $= 2 \cdot P(Z > \lvert z_{\text{obs}} \rvert)$
		where $Z \sim N(0, 1^2)$
		If p-value $< \alpha$ the *reject* H_0,
		otherwise *accept* H_0
			`prop.test(x=, n=,`
			`correct=FALSE)`

7.20	The multi-sample proportions χ^2-test	Test: $H_0 : p_1 = p_2 = \ldots = p_c = p$
		by $\chi^2_{\text{obs}} = \sum_{i=1}^{2}\sum_{j=1}^{c}\frac{(o_{ij}-e_{ij})^2}{e_{ij}}$
			`chisq.test(X,`
			`correct = FALSE)`

7.22	The $r \times c$ frequency table χ^2-test	Test: $H_0 : p_{i1} = p_{i2} = \ldots = p_{ic} = p_i$
		for all rows $i = 1, 2, \ldots, r$
		by $\chi^2_{\text{obs}} = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(o_{ij}-e_{ij})^2}{e_{ij}}$
		Reject if $\chi^2_{\text{obs}} > \chi^2_{1-\alpha}\big((r-1)(c-1)\big)$
		Otherwise accept
			`chisq.test(X,`
			`correct = FALSE)`

A.8 Comparing means of multiple groups - ANOVA

	Description	**Formula**	**Command**
8.2	One-way ANOVA variation decomposition	$\underbrace{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y})^2}_{\text{SST}} = \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\bar{y}_i)^2}_{\text{SSE}} + \underbrace{\sum_{i=1}^{k} n_i(\bar{y}_i-\bar{y})^2}_{\text{SS(Tr)}}$	
8.4	One-way within group variability	$MSE = \frac{SSE}{n-k} = \frac{(n_1-1)s_1^2+\cdots+(n_k-1)s_k^2}{n-k}$ $s_i^2 = \frac{1}{n_i-1}\sum_{i=1}^{n_i}(y_{ij}-\bar{y}_i)^2$	
8.6	One-way test for difference in mean for k groups	$H_0: \quad \alpha_i = 0; \quad i=1,2,\ldots,k,$ $F = \frac{SS(Tr)/(k-1)}{SSE/(n-k)}$ F-distribution with $k-1$ and $n-k$ degrees of freedom	`anova(lm(y~treatm))`
8.9	Post hoc pairwise confidence intervals	$\bar{y}_i - \bar{y}_j \pm t_{1-\alpha/2}\sqrt{\frac{SSE}{n-k}\left(\frac{1}{n_i}+\frac{1}{n_j}\right)}$ If all $M = k(k-1)/2$ combinations, then use $\alpha_{\text{Bonferroni}} = \alpha/M$	
8.10	Post hoc pairwise hypothesis tests	Test: $H_0: \mu_i = \mu_j$ vs. $H_1: \mu_i \neq \mu_j$ by p-value $= 2\cdot P(T > \lvert t_{\text{obs}}\rvert)$ where $t_{\text{obs}} = \frac{\bar{y}_i-\bar{y}_j}{\sqrt{MSE\left(\frac{1}{n_i}+\frac{1}{n_j}\right)}}$ Test $M = k(k-1)/2$ times, but each time with $\alpha_{\text{Bonferroni}} = \alpha/M$	
8.13	Least Significant Difference (LSD) values	$LSD_\alpha = t_{1-\alpha/2}\sqrt{2\cdot MSE/m}$	
8.20	Two-way ANOVA variation decomposition	$\underbrace{\sum_{i=1}^{k}\sum_{j=1}^{l}(y_{ij}-\hat{\mu})^2}_{\text{SST}} = \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{l}(y_{ij}-\hat{\alpha}_i-\hat{\beta}_j-\hat{\mu})^2}_{\text{SSE}} + \underbrace{l\cdot\sum_{i=1}^{k}\hat{\alpha}_i^2}_{\text{SS(Tr)}} + \underbrace{k\cdot\sum_{j=1}^{l}\hat{\beta}_j^2}_{\text{SS(Bl)}}$	

	Description	**Formula**	**Command**

8.22	Test for difference in means in two-way ANOVA grouped in treatments and in blocks	$H_{0,Tr}: \quad \alpha_i = 0, \quad i = 1,2,\ldots,k$
		$F_{Tr} = \frac{SS(Tr)/(k-1)}{SSE/((k-1)(l-1))}$
		$H_{0,Bl}: \quad \beta_j = 0, \quad j = 1,2,\ldots,l$
		$F_{Bl} = \frac{SS(Bl)/(l-1)}{SSE/((k-1)(l-1))}$
			`fit<-lm(y~treatm+block)`
			`anova(fit)`

One-way ANOVA

Source of variation	Degrees of freedom	Sums of squares	Mean sum of squares	Test-statistic F	p-value
Treatment	$k-1$	$SS(Tr)$	$MS(Tr) = \frac{SS(Tr)}{k-1}$	$F_{\text{obs}} = \frac{MS(Tr)}{MSE}$	$P(F > F_{\text{obs}})$
Residual	$n-k$	SSE	$MSE = \frac{SSE}{n-k}$		
Total	$n-1$	SST			

Two-way ANOVA

Source of variation	Degrees of freedom	Sums of squares	Mean sums of squares	Test statistic F	p-value
Treatment	$k-1$	$SS(Tr)$	$MS(Tr) = \frac{SS(Tr)}{k-1}$	$F_{Tr} = \frac{MS(Tr)}{MSE}$	$P(F > F_{Tr})$
Block	$l-1$	$SS(Bl)$	$MS(Bl) = \frac{SS(Bl)}{l-1}$	$F_{Bl} = \frac{MS(Bl)}{MSE}$	$P(F > F_{Bl})$
Residual	$(l-1)(k-1)$	SSE	$MSE = \frac{SSE}{(k-1)(l-1)}$		
Total	$n-1$	SST			