Chapter 1

Introduction, descriptive statistics, Python and data visualization

Exercises

Contents

Initilize Python packages

```
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
```

1.1 Infant birth weight

In a study of different occupational groups the infant birth weight was recorded for randomly selected babies born by hairdressers, who had their first child. The following table shows the weight in grams (observations specified in sorted order) for 10 female births and 10 male births:

Females (x)	2474	2547	2830	3219	3429	3448	3677	3872	4001	4116
Males (y)	2844	2863	2963	3239	3379	3449	3582	3926	4151	4356

Solve at least the following questions a)-c) first "manually" and then by the inbuilt functions in Python. It is OK to use Python as alternative to your pocket calculator for the "manual" part, but avoid the inbuilt functions that will produce the results without forcing you to think about how to compute it during the manual part.

a) What is the sample mean, variance and standard deviation of the female births? Express in your own words the story told by these numbers. The idea is to force you to interpret what can be learned from these numbers.

|||| Solution

We have $n = 10$, hence the sample mean is

$$\begin{aligned}\bar{x} &= \frac{1}{10}(2474 + 2547 + 2830 + 3219 + 3429 + 3448 + 3677 + 3872 + 4001 + 4116) \\ &= 3361.3,\end{aligned}$$

and the sample variance

$$\begin{aligned}s^2 = \frac{1}{9}\big((2474 - 3361.3)^2 &+ (2547 - 3361.3)^2 + (2830 - 3361.3)^2 + (3219 - 3361.3)^2 \\ &+ (3429 - 3361.3)^2 + (3448 - 3361.3)^2 + (3677 - 3361.3)^2 + (3872 - 3361.3)^2 \\ &+ (4001 - 3361.3)^2 + (4116 - 3361.3)^2\big) \\ = 344920.5,\end{aligned}$$

and finally the sample standard deviation is

$$s = \sqrt{s^2} = \sqrt{344920.5} = 587.30.$$

```
## In Python we compute it by:
x = np.array([2474, 2547, 2830, 3219, 3429, 3448, 3677, 3872, 4001,
4116])

print(np.mean(x))

3361.3


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

344920.4555555556


print(np.std(x, ddof=1))

587.2992895922449
```

Interpretation: if we consider the 10 female births as a representative sample from the population of all female births, we estimate the population mean weight μ to be $\hat{\mu} = 3361$ g. Individual female births will not be exactly 3361 g each of them, they will typically differ from that value. They are estimated to differ from the mean by $s = 587$ g on average. Since they are expected to differ both above and below the mean, one would expect most female births to be within plus/minus $2 \cdot 587 = 1174$ g of the mean.

An average absolute difference to the mean (i.e. estimated by the sample standard deviation s) somehow matches (on a linear scale) that individual observations distribute from the mean minus $2s$ to the mean plus $2s$! (at least if they are evenly distributed).

b) Compute the same summary statistics of the male births. Compare and explain differences with the results for the female births.

|||| Solution

For the manual computation, the same three formulas as above should be used. Here we show the Python-computations and results:

```
## In Python we compute it by:
y = np.array([2844, 2863, 2963, 3239, 3379, 3449, 3582, 3926, 4151,
4356])

print(np.mean(y))

3475.2


print(np.var(y, ddof=1))

283158.1777777777


print(np.std(y, ddof=1))

532.1260919911537
```

Thus

$$\bar{x} = 3475.2,$$
$$s^2 = 283158,$$
$$s = 532.13.$$

Comparison: the male birth weights are on average a little higher, but the standard deviation is a little smaller.

An important part of the course is to give you methods that would make it possible for you to do a comparison of these numbers in a more elaborate and clever way than above. A concern for the thoughtful reader would be: what might happen if we repeated this study by recording birth weights for another sample of 2×10 births? Would the comparison come out the same way or differently? Actually, it IS possible to answer this question based just on a SINGLE sample, if we include some probability calculations in the statement.

c) Find the five quartiles for each sample — and draw the two box plots with pen and paper (i.e. not using Python.)

Solution

Note that the 10 weights are already ordered in the data table, so the first step of finding the quartiles have been carried out for us. With $n = 10$ we get the following values for np:

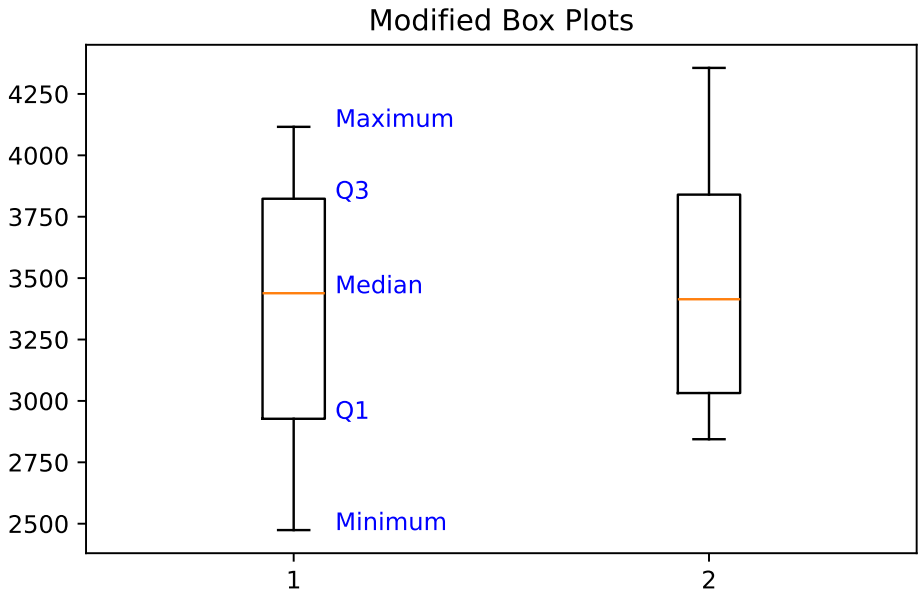
	$p = 0$	$p = 0.25$	$p = 0.5$	$p = 0.75$	$p = 1$
np	0	2.5	5	7.5	10

This means that we according to the definition of quantiles (or percentiles) can read off the Q_1 and Q_3 as the 3rd and the 8th observation and the median as the average of the 5th and 6th observation:

	$p = 0$	$p = 0.25$	$p = 0.5$	$p = 0.75$	$p = 1$
np	0	2.5	5	7.5	10
Quartile	Min	Q_1	Median	Q_3	Max
Females	2474	2830	(3429+3448)/2	3872	4116
Males	2844	2963	(3379+3449)/2	3926	4356

Now the two basic box plots could be made from these 2×5 numbers:

```
## In Python we make the modified box plots by
plt.boxplot([x,y])
plt.title('Modified Box Plots')
plt.show()
```



d) Are there any "extreme" observations in the two samples (use the *modified box plot* definition of extremeness)?

|||| Solution

As the modified box plot is the default choice in Python, and no individual observations are seen beyond the whiskers, there are no extreme observations (which by the way is defined as an observation further than $1.5 \cdot \text{IQR}$) away from the box.

e) What are the coefficient of variations in the two groups?

|||| Solution

The coefficient of variation (CV) is the standard deviation seen relative to the mean, thus for the females it is

$$CV_{\text{female}} = \frac{s_x}{\bar{x}} \cdot 100\% = \frac{587.2993}{3361.3} \cdot 100\% = 17.5\%,$$

and for males it is

$$CV_{\text{male}} = \frac{s_y}{\bar{y}} \cdot 100\% = \frac{532.1261}{3475.2} \cdot 100\% = 15.3\%.$$

1.2 Course Grades

|||| Exercise 1.1 Course grades

To compare the difficulty of 2 different courses at a university the following grades distributions (given as number of pupils who achieved the grades) were registered:

	Course 1	Course 2	Total
Grade 12	20	14	34
Grade 10	14	14	28
Grade 7	16	27	43
Grade 4	20	22	42
Grade 2	12	27	39
Grade 0	16	17	33
Grade -3	10	22	32
Total	108	143	251

a) What is the median of the 251 achieved grades?

|||| Solution

We look at the 251 grades seen from the Total column of the table. Seen from below, these 251 grades are already ordered, so to find the median we should find the 126th ordered observation from below. Since there are 104 grades in the -3, 0, and 2 Grade categories and 42 in the Grade 4 category, the 126th ordered observation from below is a 4, so the answer is: the median is 4.

Just a note about that it is actually the *sample median* which is asked for, however as noted in Remark 1.3, the *sample* is left out. Further, it is noticed that the *sample median* can be used as an estimate of the *population median* in the same way as illustrated for the mean in Figure 1.1, same goes for quantiles, quartiles, IQR, and all other statistics.

b) What are the quartiles and the IQR (Inter Quartile Range)?

|||| Solution

Since $n \cdot 0.25 = 251 \cdot 0.25 = 62.75$ and $n \cdot 0.75 = 251 \cdot 0.75 = 188.25$ we must find the lower and upper quartiles Q_1 and Q_3 as the 63rd and 189th observation from below. Let's look at the accumulated (from below) numbers:

	Total	Acccum. (from below)
Grade 12	34	251
Grade 10	28	217
Grade 7	43	189
Grade 4	42	146
Grade 2	39	104
Grade 0	33	65
Grade -3	32	32

So it becomes clear that

$$\begin{aligned} Q_1 &= 0, \\ Q_3 &= 7, \\ \text{IQR} &= 7 - 0 = 7. \end{aligned}$$

Finally, a notice about that here the quartiles are the actually the *sample quartiles* and they can be thought of as estimates for the *population quartiles*, as illustrated for the *mean* in Figure 1.1. Actually to be consistent in notation we should use a 'hat' to indicate this, e.g. the *first sample quartile* $\hat{Q}_1$ is an estimate of the *first population quartile*, however to simplify and due to tradition this is not done.

1.3 Cholesterol

|||| Exercise 1.2 Cholesterol

In a clinical trial of a cholesterol-lowering agent, 15 patients' cholesterol (in mmol L^{-1}) was measured before treatment and 3 weeks after starting treatment. Data is listed in the following table:

Patient	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Before	9.1	8.0	7.7	10.0	9.6	7.9	9.0	7.1	8.3	9.6	8.2	9.2	7.3	8.5	9.5
After	8.2	6.4	6.6	8.5	8.0	5.8	7.8	7.2	6.7	9.8	7.1	7.7	6.0	6.6	8.4

a) What is the median of the cholesterol measurements for the patients before treatment, and similarly after treatment?

|||| Solution

To find the medians we need to order both data sets, and then, since $n = 15$, an odd number, the median is the 8th observation $x_{(8)}$ in the ordered set. This is done "manually" (or call it step by step) by Python in the following way:

```
## Reading the data into Python
before = np.array([9.1, 8.0, 7.7, 10.0, 9.6, 7.9, 9.0, 7.1, 8.3, 9.6,
            8.2, 9.2, 7.3, 8.5, 9.5])
after = np.array([8.2, 6.4, 6.6, 8.5, 8.0, 5.8, 7.8, 7.2, 6.7, 9.8,
                  7.1, 7.7, 6.0, 6.6, 8.4])

## Making ordered vectors using numpy's sort function
before_sorted = np.sort(before)
after_sorted = np.sort(after)
## Printing the ordered vectors
print(before_sorted)
```

```
[ 7.100  7.300  7.700  7.900  8.000  8.200  8.300  8.500  9.000  9.100
  9.200  9.500  9.600  9.600 10.000]
```

```
print(after_sorted)
```

```
[5.800 6.000 6.400 6.600 6.600 6.700 7.100 7.200 7.700 7.800 8.000 8.200
 8.400 8.500 9.800]
```

```
## Printing the 8th observation in these vectors. Remember Python is
0-indexed
print(before_sorted[7])
```

```
8.5
```

```
print(after_sorted[7])
```

```
7.2
```

Giving the results

$$\begin{aligned} \text{'median before'} &= 8.5, \\ \text{'median after'} &= 7.2. \end{aligned}$$

Using the Python-function `.describe()` (from pandas) one would get them directly, together with more info:

```
## Get a summary using the pandas library in Python
before_series = pd.Series(before)
after_series = pd.Series(after)
print(before_series.describe())

count    15.000000
mean      8.600000
std       0.902378
min       7.100000
25%       7.950000
50%       8.500000
75%       9.350000
max      10.000000
dtype: float64

print(after_series.describe())

count    15.000000
mean      7.386667
std       1.090129
min       5.800000
25%       6.600000
50%       7.200000
75%       8.100000
max       9.800000
dtype: float64
```

We have also learned that we can use the Python-function `np.quantile` to get the quartiles, and to use the percentile definition given in Definition 1.7, we should use the `method='averaged_inverted_cdf'` argument:

```
## Now using the quantile function in numpy
quartiles_before = np.quantile(before,[0,0.25,0.5,0.75,1],method =
'averaged_inverted_cdf')
quartiles_after = np.quantile(after,[0,0.25,0.5,0.75,1],method =
'averaged_inverted_cdf')
## Printing the Quartiles
print(quartiles_before)

[ 7.100  7.900  8.500  9.500 10.000]


print(quartiles_after)

[5.800 6.600 7.200 8.200 9.800]
```

It can be noted that some of the quartiles given here are not exactly the same as those given by the `.describe()` function. This is due to the fact that `.describe()` from pandas uses the default setting of the `np.quantile` function, so NOT the `method='averaged_inverted_cdf` option. We will live with this little difference, which will not cause any problems. We consider both results just as valid, just only one of them are defined in the material.

b) Find the standard deviations of the cholesterol measurements of the patients before and after treatment.

|||| Solution

We should use the defining formulae for the sample mean (Def. 1.4) and sample standard deviation (Def. 1.11) for each sample

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

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

In Python we find these as:

```
print(np.mean(before))

8.6

print(np.mean(after))

7.386666666666667

print(np.std(before, ddof=1))

0.9023778112773574

print(np.std(after, ddof=1))

1.0901288696209053
```

c) Find the sample covariance between cholesterol measurements of the patients before and after treatment.

Solution

Define the "before treatment" sample as $x_1, x_2, \ldots, x_{15}$ and the "after treatment" sample $y_1, y_2, \ldots, y_{15}$, then the sample covariance is found using Definition 1.18 as

$$s_{xy} = \frac{1}{14}\sum_{i=1}^{15}(x_i - 8.6)(y_i - 7.3867) = 11.15/14 = 0.79643.$$

In Python we find this as:

```
## Calculate the sample covariance 'manually'
cov_manual = np.sum((before - np.mean(before)) * (after -
np.mean(after))) / 14
print(cov_manual)

0.7964285714285715


## or use the inbuilt function
print(np.cov(before,after, ddof=1))

[[0.814 0.796]
 [0.796 1.188]]
```

d) Find the sample correlation between cholesterol measurements of the patients before and after treatment.

Solution

This is Definition 1.19 and simply

$$r = \frac{s_{xy}}{s_x \cdot s_y} = \frac{0.79643}{0.90238 \cdot 1.0901} = 0.8096.$$

In Python we find this by:

```
## 'Manually'
print(0.79643/(0.90238*1.0901))

0.8096397271662439


## or
cor = np.cov(before, after)[0,1]/(np.std(before,ddof = 1) *
np.std(after,ddof = 1))
print(cor)

0.809618797174745


## or correlation directly in numpy
print(np.corrcoef(before,after))

[[1.000 0.810]
 [0.810 1.000]]
```

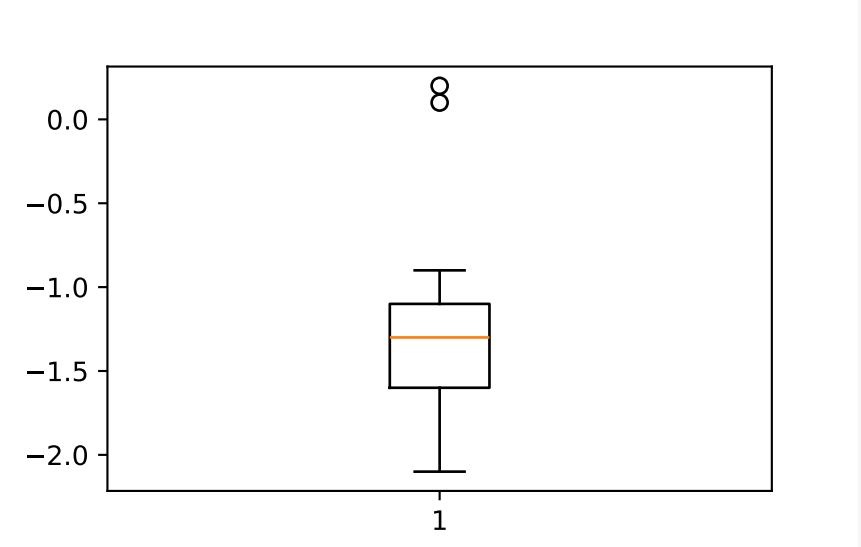
e) Compute the 15 differences (Dif = Before − After) and do various summary statistics and plotting of these: sample mean, sample variance, sample standard deviation, boxplot etc.

|||| Solution

The differences are:

Patient	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Before	9.1	8.0	7.7	10.0	9.6	7.9	9.0	7.1	8.3	9.6	8.2	9.2	7.3	8.5	9.5
After	8.2	6.4	6.6	8.5	8.0	5.8	7.8	7.2	6.7	9.8	7.1	7.7	6.0	6.6	8.4
Dif	0.9	1.6	1.1	1.5	1.6	2.1	1.2	-0.1	1.6	-0.2	1.1	1.5	1.3	1.9	1.1

```
## Analysis of differences
dif = after-before
## Quartiles
quartiles_dif = np.quantile(dif,[0,0.25,0.5,0.75,1],method =
'averaged_inverted_cdf')
print(quartiles_dif)

[-2.100 -1.600 -1.300 -1.100  0.200]

## Sample variance
print(np.var(dif, ddof=1))

0.4098095238095241

## Sample standard deviation
print(np.std(dif, ddof=1))

0.6401636695482836

## Boxplot
fig = plt.boxplot(dif)
plt.show()
```



The mean effect (decrease of cholesterol due to treatment) would be estimated at 1.2 nMol/l. But clearly there is also a high degree of differences in what the effect is: the standard deviation of (all) the differences is 0.64. Looking at the boxplot, we find two patients with values identified as extreme, which from the data table is

seen to be patient no 8 and 10. The better way, maybe, here to tell the story would be the following: for 2 out of 15 patients (13% of patients) the treatment clearly had no effect. For the remaining 13 out of 15 (87% of patients) the treatment had the following average effect and standard deviation (recomputing the mean and standard deviation for the 13 patients):

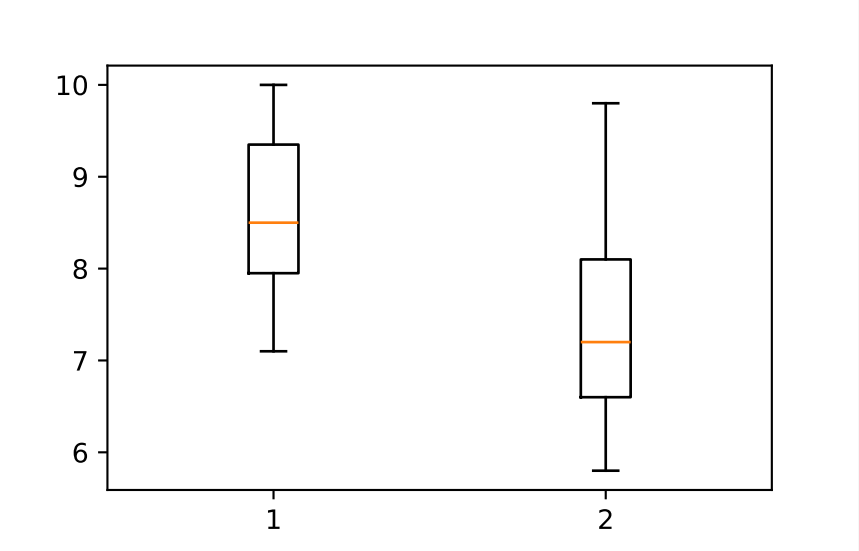
```
## Analysis of 13 non-extreme differences
## Take out observation 8 and 10
dif13 = np.delete(dif,[7,9])
## Mean of the 13 differences
print(np.mean(dif13))

-1.4230769230769231


## Standard deviation of the 13 differences
print(np.std(dif13, ddof=1))

0.3467800603500874
```

f) Observing such data the big question is whether an average decrease in cholesterol level can be "shown statistically". How to formally answer this question is presented in Chapter 3, but consider now which summary statistics and/or plots would you look at to have some idea of what the answer will be?

Solution

In the previous question we were studying the differences in the attempt to answer this question. One could also, as we did initially look at the data separately, and e.g. supplement by the grouped boxplot:

```
fig = plt.boxplot([before,after])
plt.show()
```



And we would conclude: the average effect is 1.2 (we see no extreme patients in this plot!), and the standard deviation within each group of data is around 1 (see above: $s_{\text{before}} = 0.9$ and $s_{\text{after}} = 1.1$).

Which of the two approaches do you prefer - the "difference"-approach or the "separate"-approach?

We would definitely recommend the "difference"-approach, or as we will call it later, the "paired" approach, since this match the setup of the study, and in the most correct way uses the relevant information. Note how the difference-approach identifies the outliers/extremes and also ends up with much smaller standard deviations, also seen by the range and/or box-widths(IQR) in the box-plots. The point is that in the differences we have removed the variability stemming from the characteristics of each patient (e.g. body mass, genes, etc.). One phrase used is that in such an experiment like this, a patient acts as his own control, and hence the fact the patients are different does not blur the important effect signal.

1.4 Project start

|||| Exercise 1.3 Project start

a) Go to Learn or the website and take a look at the first project. Read the project page on the website for more information (02323.compute.dtu.dk/projects or 02402.compute.dtu.dk/projects). Choose a project and read the project description. Follow the steps to import the data into Python and get started with the explorative data analysis.

|||| Solution

There is no results for this exercise - you have to do it as a project.