

Appendix A

Collection of formulas and commands

Contents

- A Collection of formulas and commands**
 - A.1 Introduction, descriptive statistics, commands and data visualization 1
 - A.2 Probability and Simulation 3
 - A.2.1 Distributions 5
 - A.3 Statistics for one and two samples 16
 - A.4 Simulation based statistics 18
 - A.5 Simple linear regression 19
 - A.6 Multiple linear regression 21
 - A.7 Inference for proportions 22
 - A.8 Comparing means of multiple groups - ANOVA 23
- Glossaries** 25
- Acronyms** 26

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 included in the same order as in the book, except for the distributions which are listed together.

Before working through this chapter, ensure the required packages are installed. This chapter was developed using scipy version 1.15.3 (check using `scipy.__version__` and upgrade using `pip install --upgrade scipy`). At the beginning of each Python script or notebook, include the following imports:

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

A.1 Introduction, descriptive statistics, commands and data visualization

	Description	Formula	Command
1.4	Sample mean The mean of a sample.	$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$	<code>np.mean(x)</code>
1.5	Sample median The value that divides a sample in two halves with equal number of observations in each.	$Q_2 = \begin{cases} x_{(\frac{n+1}{2})} & \text{for odd } n \\ \frac{x_{(\frac{n}{2})} + x_{(\frac{n+2}{2})}}{2} & \text{for even } n \end{cases}$	<code>np.median(x)</code>
1.7	Sample quantile The value that divide a sample such that p of the observations are less than the value. The 0.5 quantile is the Median.	$q_p = \begin{cases} \frac{x_{(np)} + x_{(np+1)}}{2} & \text{for } pn \text{ integer} \\ x_{(\lceil np \rceil)} & \text{for } pn \text{ non-integer} \end{cases}$	<code>np.quantile(x, p, method='averaged_inverted_cdf')</code> <i>eller</i> <code>np.percentile(x, p, < method='averaged_inverted_cdf')</code>

	Description	Formula	Command
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	$Q_0 = q_0 = \text{"minimum"}$ $Q_1 = q_{0.25} = \text{"lower quartile"}$ $Q_2 = q_{0.5} = \text{"median"}$ $Q_3 = q_{0.75} = \text{"upper quartile"}$ $Q_4 = q_1 = \text{"maximum"}$	<code>np.quantile(x, p, method= 'averaged_inverted_cdf')</code> <i>where</i> <code>p = np.array([0, 0.25, 0.5, 0.75, 1])</code>
1.10	Sample variance The sum of squared differences from the mean divided by $n - 1$.	$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$	<code>np.var(x, ddof=1)</code>
1.11	Sample standard deviation The square root of the sample variance.	$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$	<code>np.std(x, ddof=1)</code>
1.12	Sample coefficient of variation The sample standard deviation seen relative to the sample mean.	$CV = \frac{s}{\bar{x}}$	<code>np.std(x, ddof=1)/ np.mean(x)</code>
1.15	Sample Inter Quartile Range IQR: The middle 50% range of data	$IQR = Q_3 - Q_1$	<code>stats.iqr(x)</code>
1.18	Sample covariance Measure of linear strength of relation between two samples	$s_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$	<code>np.cov(x, y, ddof=1)[0,1]</code>
1.19	Sample correlation Measure of the linear strength of relation between two samples between -1 and 1.	$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right) = \frac{s_{xy}}{s_x \cdot s_y}$	<code>np.corrcoef(x, y)[0,1]</code>

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 probability for one x value.	$f(x) = P(X = x)$	<code>stats.norm.pdf()</code> <code>stats.binom.pmf()</code> <code>stats.hypergeom.pmf()</code> <code>stats.poisson.pmf()</code>
2.9	Cumulated distribution function (cdf) gives the probability in a range of x values where $P(a < X \leq b) = F(b) - F(a)$.	$F(x) = P(X \leq x)$	<code>stats.norm.cdf()</code> <code>stats.binom.cdf()</code> <code>stats.hypergeom.cdf()</code> <code>stats.poisson.cdf()</code>
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 $\lim_{x \rightarrow -\infty} F(x) = 0$ and $\lim_{x \rightarrow \infty} F(x) = 1$	$F(x) = P(X \leq x) = \int_{-\infty}^x f(u) du$	
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 .	$E(aX + b) = a E(X) + b$ $V(aX + b) = a^2 V(X)$	
2.56	Mean and variance of a linear combination The mean and variance of a linear combination of random variables.	$E(a_1 X_1 + a_2 X_2 + \dots + a_n X_n) = a_1 E(X_1) + a_2 E(X_2) + \dots + a_n E(X_n)$ $V(a_1 X_1 + a_2 X_2 + \dots + a_n X_n) = a_1^2 V(X_1) + a_2^2 V(X_2) + \dots + a_n^2 V(X_n)$	

	Description	Formula	Command
2.58	Covariance The covariance between be two random variables X and Y .	$\text{Cov}(X,Y) = E [(X - E[X])(Y - E[Y])]$	

A.2.1 Distributions

Here all the included distributions are listed including some important theorems and definitions related specifically with a distribution.

Binomial Probability Distribution (discrete)

$$X \sim B(n, p)$$

$$f(x) = P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}$$

$$\mathbb{E}[X] = \mu = np$$

$$\text{Var}(X) = \sigma^2 = np(1 - p)$$

Notation in book and Python

Book	Python	
n	n	(total number of "draws")
p	p	(probability of success in each event)
x	k	(observed number of success'es, out of n possible)

Python Functions in `scipy.stats.binom`

- `rvs(n, p, size=...)`: Random variates
- `pmf(k, n, p)`: Probability mass function (book: $f(x)$)
- `cdf(k, n, p)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, n, p)`: Percent-point function (quantile)
- `mean(n, p)`: Mean
- `var(n, p)`: Variance
- `std(n, p)`: Standard deviation

Hypergeometric Distribution (discrete)

$$X \sim H(n, a, N)$$

$$f(x) = P(X = x) = \frac{\binom{a}{x} \binom{N-a}{n-x}}{\binom{N}{n}}$$

$$\mathbb{E}[X] = \mu = n \cdot \frac{a}{N}$$

$$\text{Var}(X) = \sigma^2 = n \cdot \frac{a}{N} \cdot \frac{N-a}{N} \cdot \frac{N-n}{N-1}$$

Notation in Book and Python

Book	Python	
N	M	(total number of objects)
a	n	(total number of success objects)
n	N	(total number of "draws")
x	k	(observed number of success'es)

Python Functions in `scipy.stats.hypergeom`

- `rvs(M, n, N, size=...)`: Random variates
- `pmf(k, M, n, N)`: Probability mass function (book: $f(x)$)
- `cdf(k, M, n, N)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, M, n, N)`: Percent-point function (quantile)
- `mean(M, n, N)`: Mean
- `var(M, n, N)`: Variance
- `std(M, n, N)`: Standard deviation

Poisson Distribution (discrete)

$$X \sim \text{Po}(\lambda)$$

$$f(x) = P(X = x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

$$\mathbb{E}[X] = \mu = \lambda$$

$$\text{Var}(X) = \sigma^2 = \lambda$$

Notation in Book and Python

Book	Python	
λ	mu	(average rate)
x	k	(observed number of events)

Python Functions in `scipy.stats.poisson`

- `rvs(mu, size=...)`: Random variates
- `pmf(k, mu)`: Probability mass function (book: $f(x)$)
- `cdf(k, mu)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, mu)`: Percent-point function (quantile)
- `mean(mu)`: Mean
- `var(mu)`: Variance
- `std(mu)`: Standard deviation

Uniform Distribution (continuous)

$$X \sim U(\alpha, \beta)$$

$$f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \alpha \leq x \leq \beta \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{E}[X] = \mu = \frac{\alpha + \beta}{2}$$

$$\text{Var}(X) = \sigma^2 = \frac{(\beta - \alpha)^2}{12}$$

Notation in Book and Python

Book	Python	
α	loc	(lower bound)
β	loc + scale	(upper bound)
x	x	(observed value)

Python Functions in `scipy.stats.uniform`

- `rvs(loc, scale, size=...)`: Random variates
- `pdf(x, loc, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, loc, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, loc, scale)`: Percent-point function (quantile)
- `mean(loc, scale)`: Mean
- `var(loc, scale)`: Variance
- `std(loc, scale)`: Standard deviation

Normal Distribution (continuous)

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$\mathbb{E}[X] = \mu$$

$$\text{Var}(X) = \sigma^2$$

Notation in Book and Python

Book	Python	
μ	<code>loc</code>	(mean)
σ	<code>scale</code>	(std deviation)
x	<code>x</code>	(observed value)

Python Functions in `scipy.stats.norm`

- `rvs(loc, scale, size=...)`: Random variates
- `pdf(x, loc, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, loc, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, loc, scale)`: Percent-point function (quantile)
- `mean(loc, scale)`: Mean
- `var(loc, scale)`: Variance
- `std(loc, scale)`: Standard deviation

Lognormal Distribution (continuous)

$$X \sim \text{LN}(\alpha, \beta)$$

$$f(x) = \begin{cases} \frac{1}{x\sqrt{2\pi\beta^2}} \exp\left(-\frac{(\ln(x)-\alpha)^2}{2\beta^2}\right) & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$$\mathbb{E}[X] = e^{\alpha + \frac{\beta^2}{2}}$$

$$\text{Var}(X) = (e^{\beta^2} - 1) e^{2\alpha + \beta^2}$$

Notation in Book and Python

Book	Python	
α		(mean of $\ln(X)$)
β	s	(std of $\ln(X)$)
	loc	(only use to shift x-axis)
$\exp(\alpha)$	scale	
x	x	(observed value)

Use: $\text{loc} = 0$, $\text{scale} = \exp(\alpha)$ and $\text{s} = \beta$

Python Functions in `scipy.stats.lognorm`

- `rvs(s, loc, scale, size=...)`: Random variates (s corresponds to β)
- `pdf(x, s, loc, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, s, loc, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, s, loc, scale)`: Percent-point function (quantile)
- `mean(s, loc, scale)`: Mean
- `var(s, loc, scale)`: Variance
- `std(s, loc, scale)`: Standard deviation

Exponential Distribution (continuous)

$$X \sim \text{Exp}(\lambda)$$

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$\mathbb{E}[X] = \mu = \frac{1}{\lambda}$$

$$\text{Var}(X) = \sigma^2 = \frac{1}{\lambda^2}$$

Notation in Book and Python

Book	Python	
λ	1/scale	(rate)
	loc	(only use to shift x-axis)
$\mu = 1/\lambda$	scale	(average waiting time)
x	x	(observed value/waiting time between events)

Use: loc = 0 and scale = $1/\lambda$ or scale = μ

Python Functions in `scipy.stats.expon`

- `rvs(scale, size=...)`: Random variates
- `pdf(x, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, scale)`: Percent-point function (quantile)
- `mean(scale)`: Mean
- `var(scale)`: Variance
- `std(scale)`: Standard deviation

Chi-Square Distribution (continuous)

$$X \sim \chi^2(\nu)$$

$$f(x) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\frac{\nu}{2}-1} e^{-x/2}, \quad x \geq 0$$

Notation in Book and Python

Book	Python	
ν	df	(degrees of freedom)
x	x	(observed value)

Python Functions in `scipy.stats.chi2`

- `rvs(df, size=...)`: Random variates
- `pdf(x, df)`: Probability density function (book: $f(x)$)
- `cdf(x, df)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, df)`: Percent-point function (quantile)
- `mean(df)`: Mean
- `var(df)`: Variance
- `std(df)`: Standard deviation

t-Distribution (continuous)

$$X \sim t(\nu)$$

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Notation in Book and Python

Book	Python	
ν	df	(degrees of freedom)
x	x	(observed value)

Python Functions in `scipy.stats.t`

- `rvs(df, size=...)`: Random variates
- `pdf(x, df)`: Probability density function (book: $f(x)$)
- `cdf(x, df)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, df)`: Percent-point function (quantile)
- `mean(df)`: Mean

- `var(df)`: Variance
- `std(df)`: Standard deviation

F-Distribution (continuous)

$$X \sim F(\nu_1, \nu_2)$$

(see book def 2.95)

Notation in Book and Python

Book	Python	
ν_1	<code>dfn</code>	(numerator df)
ν_2	<code>dfd</code>	(denominator df)
x	x	(observed value)

Python Functions in `scipy.stats.f`

- `rvs(dfn, dfd, size=...)`: Random variates
- `pdf(x, dfn, dfd)`: Probability density function (book: $f(x)$)
- `cdf(x, dfn, dfd)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, dfn, dfd)`: Percent-point function (quantile)
- `mean(dfn, dfd)`: Mean
- `var(dfn, dfd)`: Variance
- `std(dfn, dfd)`: Standard deviation

	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 successes.	$f(x; n, p) = P(X = x)$ $= \binom{n}{x} p^x (1-p)^{n-x}$ where $\binom{n}{x} = \frac{n!}{x!(n-x)!}$	<code>stats.binom.pmf(x,n,p)</code> <code>stats.binom.cdf(q,n,p)</code> <code>stats.binom.ppf(q,n,p)</code> <code>stats.binom.rvs(n,p,size)</code>
2.21	Mean and variance of a binomial distributed random variable.	$\mu = np$ $\sigma^2 = np(1-p)$	

	Description	Formula	Command
2.24	Hypergeometric distribution n is the number of draws without replacement, a is number of successes 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 $\binom{a}{b} = \frac{a!}{b!(a-b)!}$	Use stats. in front of the following commands hypergeom.pmf(x,N,a,n) hypergeom.cdf(x,N,a,n) hypergeom.ppf(p,N,a,n) hypergeom.rvs(N,a,n,size)
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.	$f(x; \lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$	stats.poisson.pmf(x,l) stats.poisson.cdf(q,l) stats.poisson.ppf(p,l) stats.poisson.rvs(l,size) where l= 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.	$f(x; \alpha, \beta) = \begin{cases} 0 & \text{for } x < \alpha \\ \frac{1}{\beta - \alpha} & \text{for } x \in [\alpha, \beta] \\ 0 & \text{for } x > \beta \end{cases}$ $F(x; \alpha, \beta) = \begin{cases} 0 & \text{for } x < \alpha \\ \frac{x - \alpha}{\beta - \alpha} & \text{for } x \in [\alpha, \beta] \\ 1 & \text{for } x > \beta \end{cases}$	Use stats. in front of the following commands uniform.pdf(x,min,dif) uniform.cdf(q,min,dif) uniform.ppf(p,min,dif) uniform.rvs(min,dif,size) where min= α , dif= $\beta - \alpha$
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.	$f(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	stats.norm.pdf(x,mu,sd) stats.norm.cdf(q,mu,sd) stats.norm.ppf(p,mu,sd) stats.norm.rvs(mu,sd, size) where mu= μ , sd= σ .
2.38	Mean and variance of a normal distributed random variable.	μ σ^2	

	Description	Formula	Command
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 .	$f(x) = \frac{1}{x\sqrt{2\pi}\beta} e^{-\frac{(\ln x - \alpha)^2}{2\beta^2}}$	Use <code>stats.lognorm.</code> in front of the following commands <code>pdf(x,sdlog,scale=mu)</code> <code>cdf(x,sdlog,scale=mu)</code> <code>ppf(p,sdlog,scale=mu)</code> <code>rvs(sdlog, scale=mu, size=size)</code> <i>where</i> <code>mu=e^α, sdlog=β.</code>
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.	$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$	Use <code>stats.expon.</code> in front of the following commands <code>pdf(x,scale=1/lambda)</code> <code>cdf(q,scale=1/lambda)</code> <code>ppf(p,scale=1/lambda)</code> <code>rvs(scale=1/lambda, size=size)</code>
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 $\Gamma(\frac{\nu}{2})$ is the Γ -function and ν is the degrees of freedom.	$f(x) = \frac{1}{2^{\frac{\nu}{2}}\Gamma(\frac{\nu}{2})} x^{\frac{\nu}{2}-1} e^{-\frac{x}{2}}; \quad x \geq 0$	<code>stats.chi2.pdf(x, df=df)</code> <code>stats.chi2.cdf(q, df=df)</code> <code>stats.chi2.ppf(p, df=df)</code> <code>stats.chi2.rvs(df=df, size=size)</code> <i>where</i> <code>df=ν.</code>
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 $\nu = n - 1$.	$\chi^2 = \frac{(n-1)S^2}{\sigma^2}$	

	Description	Formula	Command
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 $\Gamma(\cdot)$ is the Gamma function.	$f_T(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	
2.87	Relation between normal random variables and χ^2 -distributed random variables. $Z \sim N(0, 1)$ and $Y \sim \chi^2(\nu)$.	$X = \frac{Z}{\sqrt{Y/\nu}} \sim t(\nu)$	<code>stats.t.pdf(x,df)</code> <code>stats.t.cdf(q,df)</code> <code>stats.t.ppf(p,df)</code> <code>stats.t.rvs(df,size=size)</code> where <code>df=ν</code> .
2.89	For normal distributed random variables $X_1, \dots, X_n$, the random variable follows the t -distribution, where $\bar{X}$ is the sample mean, μ is the mean of X , n is the sample size and S is the sample standard deviation.	$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1)$	
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 and ν_2 are the degrees of freedom and $B(\cdot, \cdot)$ is the Beta function.	$f_F(x) = \frac{1}{B(\frac{\nu_1}{2}, \frac{\nu_2}{2})} \left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}} \cdot x^{\frac{\nu_1}{2}-1} \left(1 + \frac{\nu_1}{\nu_2}x\right)^{-\frac{\nu_1+\nu_2}{2}}$	<code>stats.f.pdf(x,df1,df2)</code> <code>stats.f.cdf(q,df1,df2)</code> <code>stats.f.ppf(p,df1,df2)</code> <code>stats.f.rvs(df1,df2, size=size)</code> where <code>df1=ν₁, df2=ν₂</code> .
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 $V \sim \chi^2(\nu_2)$.	$\frac{U/\nu_1}{V/\nu_2} \sim F(\nu_1, \nu_2)$	
2.98	$X_1, \dots, X_{n_1}$ and $Y_1, \dots, Y_{n_2}$ 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	$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1^2)$	
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	$\bar{x} \pm t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}$	where $t_{1-\alpha/2}$ can be found in python with <code>stats.t.ppf(1-alpha/2,df)</code>
3.14	Central Limit Theorem (CLT)	$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$	
3.19	Confidence interval for the variance and standard deviation	$\sigma^2 : \left[\frac{(n-1)s^2}{\chi_{1-\alpha/2}^2}, \frac{(n-1)s^2}{\chi_{\alpha/2}^2} \right]$ $\sigma : \left[\sqrt{\frac{(n-1)s^2}{\chi_{1-\alpha/2}^2}}, \sqrt{\frac{(n-1)s^2}{\chi_{\alpha/2}^2}} \right]$	where $\chi_{1-\alpha/2}^2$ and $\chi_{\alpha/2}^2$ can be found in python with respectively <code>stats.chi2.ppf(1-alpha/2,df)</code> and <code>stats.chi2.ppf(alpha/2,df)</code>
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.	<code>pval=2*(1-stats.t.cdf(tobs,df))</code>
3.23	The one-sample t -test statistic and p -value	$p\text{-value} = 2 \cdot P(T > t_{\text{obs}})$ $t_{\text{obs}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$ $H_0 : \mu = \mu_0$	
3.24	The hypothesis test	Rejected: $p\text{-value} < \alpha$ Accepted: <i>otherwise</i>	
3.29	Significant effect	An effect is significant if the $p\text{-value} < \alpha$	

	Description	Formula	Command
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 $t_{1-\alpha/2}$	
3.32	The one-sample hypothesis test by the critical value	Reject: $ t_{\text{obs}} > t_{1-\alpha/2}$ accept: <i>otherwise</i>	
3.33	Confidence interval for μ	$\bar{x} \pm t_{1-\alpha/2} \cdot \frac{s}{\sqrt{n}}$ acceptance region/CI: $H_0 : \mu = \mu_0$	
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 = $2 \cdot P(T > t_{\text{obs}})$ Reject: p -value $< \alpha$ or $ t_{\text{obs}} > t_{1-\alpha/2}$ Accept: <i>Otherwise</i>	
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: $p_i = \frac{i}{n}, i = 1, \dots, n$ commonly approach: $p_i = \frac{i-0.5}{n+1}, i = 1, \dots, n$	
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 = $2 \cdot P(T > t_{\text{obs}})$ Reject: p -value $< \alpha$ or $ t_{\text{obs}} > t_{1-\alpha/2}$ Accept: <i>Otherwise</i>	
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	$T = \frac{(\bar{X}_1 - \bar{X}_2) - \delta_0}{\sqrt{S_p^2/n_1 + S_p^2/n_2}}$	

	Description	Formula	Command
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	$\sigma_{f(X_1, \dots, X_n)}^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i}\right)^2 \sigma_i^2$	
4.4	Non-linear error propagation by simulation	1. Simulate k outcomes 2. Calculate the standard deviation by $s_{f(X_1, \dots, X_n)}^{\text{sim}} = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (f_j - \bar{f})^2}$	
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 $S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2$	<pre>D=pd.DataFrame({'x': x, 'y': y}) linfit=smf.ols(formula = 'y ~ x', data=D).fit() print(linfit.summary(slim=True))</pre> where the row Intercept refers to values related to β_0, and x refers to values related to β_1
5.8	Variance of estimators	$V[\hat{\beta}_0] = \frac{\sigma^2}{n} + \frac{\bar{x}^2 \sigma^2}{S_{xx}}$ $V[\hat{\beta}_1] = \frac{\sigma^2}{S_{xx}}$ $\text{Cov}[\hat{\beta}_0, \hat{\beta}_1] = -\frac{\bar{x} \sigma^2}{S_{xx}}$	
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 $p\text{-value} = 2 \cdot P(T > t_{\text{obs},\beta_i})$ where $t_{\text{obs},\beta_i} = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\beta_i}}$. If $p\text{-value} < \alpha$ then reject H_0, otherwise accept H_0	
5.15	Parameter confidence intervals	$\hat{\beta}_0 \pm t_{1-\alpha/2} \hat{\sigma}_{\beta_0}$ $\hat{\beta}_1 \pm t_{1-\alpha/2} \hat{\sigma}_{\beta_1}$	<code>linfit.conf_int(0.05)</code>
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: $\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}} \pm t_{1-\alpha/2} \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(x_{\text{new}} - \bar{x})^2}{S_{xx}}}$	<pre>Dn=pd.DataFrame({'x': xn}) tab=linfit.get_prediction(Dn).summary_frame(alpha=0.05) ci=tab[['mean_ci_lower', 'mean_ci_upper']] pi=tab[['obs_ci_lower', 'obs_ci_upper']]</pre>

	Description	Formula	Command
5.23	The matrix formulation of the parameter estimators in the simple linear regression model	$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$ $V[\hat{\beta}] = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$ $\hat{\sigma}^2 = \frac{RSS}{n-2}$	
5.25	Coefficient of determination R^2	$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$	
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 $\hat{y}_i$	<pre>res=linfit.resid yfit=linfit.fittedvalues fig, ax=plt.subplots(2) stats.probplot(res, dist="norm", plot=ax[0]) ax[1].scatter(yfit, res) ax[1].axhline(y=0) plt.tight_layout() plt.show()</pre> (Remember to set title)

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 $= 2 \cdot P(T > t_{\text{obs},\beta_i})$ where $t_{\text{obs},\beta_i} = \frac{\hat{\beta}_i - \beta_{0,i}}{\hat{\sigma}_{\hat{\beta}_i}}$. If p-value $< \alpha$ the <i>reject</i> H_0, otherwise <i>accept</i> H_0	<pre>D=pd.DataFrame({'x1': x1, 'x2': x2, 'y': y}) linfit=smf.ols(formula='y ~ x1+x2', data=D).fit() print(linfit.summary(slim=True))</pre>
6.5	Parameter confidence intervals	$\hat{\beta}_i \pm t_{1-\alpha/2} \hat{\sigma}_{\hat{\beta}_i}$	<code>linfit.conf_int(0.05)</code>
6.9	Confident and prediction interval (in R)	Confident interval for the line $\hat{\beta}_0 + \hat{\beta}_1 x_{1,\text{new}} + \dots + \hat{\beta}_p x_{p,\text{new}}$ Interval for a new point prediction $\hat{\beta}_0 + \hat{\beta}_1 x_{1,\text{new}} + \dots + \hat{\beta}_p x_{p,\text{new}} + \varepsilon_{\text{new}}$	<pre>Dn=pd.DataFrame({'x1': x1n, 'x2': x2n}) tab=linfit.get_prediction(Dn).summary_frame(alpha=0.05) ci=tab[['mean_ci_lower', 'mean_ci_upper']] pi=tab[['obs_ci_lower', 'obs_ci_upper']]</pre>
6.17	The matrix formulation of the parameter estimators in the multiple linear regression model	$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$ $V[\hat{\beta}] = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$ $\hat{\sigma}^2 = \frac{RSS}{n - (p + 1)}$	
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	$\hat{p} = \frac{x}{n}$ $\hat{p} \pm z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$	
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 = $2 \cdot P(Z > z_{\text{obs}})$ where $Z \sim N(0, 1^2)$ If p -value $< \alpha$ the <i>reject</i> H_0 , otherwise <i>accept</i> H_0	<code>zobs, pval=smpop.\</code> <code>proportions_ztest(x, n,</code> <code>value=0.5, prop_var=0.5)</code>
7.13	Sample size formula for the CI of a proportion	Guessed p (with prior knowledge): $n = p(1-p) \left(\frac{z_{1-\alpha/2}}{ME} \right)^2$ Unknown p : $n = \frac{1}{4} \left(\frac{z_{1-\alpha/2}}{ME} \right)^2$	
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 = $2 \cdot P(Z > z_{\text{obs}})$ where $Z \sim N(0, 1^2)$ If p -value $< \alpha$ the <i>reject</i> H_0 , otherwise <i>accept</i> H_0	
7.20	The multi-sample proportions χ^2 -test	Test: $H_0 : p_1 = p_2 = \dots = p_c = p$ by $\chi^2_{\text{obs}} = \sum_{i=1}^2 \sum_{j=1}^c \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$	<code>chi2, p, dof, expected=\</code> <code>stats.chi2_contingency(</code> <code>X, correction=False)</code>
7.22	The $r \times c$ frequency table χ^2 -test	Test: $H_0 : p_{i1} = p_{i2} = \dots = p_{ic} = p_i$ for all rows $i = 1, 2, \dots, r$ by $\chi^2_{\text{obs}} = \sum_{i=1}^r \sum_{j=1}^c \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$ Reject if $\chi^2_{\text{obs}} > \chi^2_{1-\alpha}((r-1)(c-1))$ Otherwise accept	

A.8 Comparing means of multiple groups - ANOVA

	Description	Formula	Command
8.2	One-way ANOVA variation decomposition	$\underbrace{\sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2}_{SST} = \underbrace{\sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2}_{SSE} + \underbrace{\sum_{i=1}^k n_i (\bar{y}_i - \bar{y})^2}_{SS(Tr)}$	
8.4	One-way within group variability	$MSE = \frac{SSE}{n-k} = \frac{(n_1-1)s_1^2 + \dots + (n_k-1)s_k^2}{n-k}$ $s_i^2 = \frac{1}{n_i-1} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2$	
8.6	One-way test for difference in mean for k groups	$H_0: \alpha_i = 0; \quad i = 1, 2, \dots, k,$ $F = \frac{SS(Tr)/(k-1)}{SSE/(n-k)}$ F-distribution with $k-1$ and $n-k$ degrees of freedom	<pre>D=pd.DataFrame({'y': y, 'group': group}) model=smf.ols('y ~ C(group)', data=D).fit() anova_results=sm.stats.\ anova_lm(model, typ=2)</pre>
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 $\alpha_{Bonferroni} = \alpha/M$	
8.10	Post hoc pairwise hypothesis tests	Test: $H_0: \mu_i = \mu_j$ vs. $H_1: \mu_i \neq \mu_j$ by $p\text{-value} = 2 \cdot P(T > t_{\text{obs}})$ 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 $\alpha_{Bonferroni} = \alpha/M$	
8.13	Least Significant Difference (LSD) values	$LSD_{\alpha} = t_{1-\alpha/2} \sqrt{2 \cdot MSE / m}$	

	Description	Formula	Command
8.20	Two-way ANOVA variation decomposition	$\underbrace{\sum_{i=1}^k \sum_{j=1}^l (y_{ij} - \hat{\mu})^2}_{SST} =$ $\underbrace{\sum_{i=1}^k \sum_{j=1}^l (y_{ij} - \hat{\alpha}_i - \hat{\beta}_j - \hat{\mu})^2}_{SSE} +$ $\underbrace{l \cdot \sum_{i=1}^k \hat{\alpha}_i^2}_{SS(Tr)} + \underbrace{k \cdot \sum_{j=1}^l \hat{\beta}_j^2}_{SS(Bl)}$	
8.22	Test for difference in means in two-way ANOVA grouped in treatments and in blocks	$H_{0,Tr}: \alpha_i = 0, \quad i = 1, 2, \dots, k$ $F_{Tr} = \frac{SS(Tr)/(k-1)}{SSE/((k-1)(l-1))}$ $H_{0,Bl}: \beta_j = 0, \quad j = 1, 2, \dots, l$ $F_{Bl} = \frac{SS(Bl)/(l-1)}{SSE/((k-1)(l-1))}$	<pre>D=pd.DataFrame({'y': y, 'g1': g1, 'g2': g2}) model=smf.ols('y ~ C(g1) + C(g2)', data=D).fit() anova_results=sm.stats.\ anova_lm(model, typ=2)</pre>

One-way ANOVA

Source of variation	Degrees of freedom	Sums of squares	Mean sum of squares	Test-statistic F	p -value
Treatment	$k - 1$	$SS(Tr)$	$MS(Tr) = \frac{SS(Tr)}{k-1}$	$F_{obs} = \frac{MS(Tr)}{MSE}$	$P(F > F_{obs})$
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			

Glossaries

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 [3](#)

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 [3](#)

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 [2](#)

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 [2](#), [4](#)

F-distribution [F-fordelingen] The F -distribution appears as the ratio between two independent χ^2 -distributed random variables [23](#)

Inter Quartile Range [Interkvartil bredde] The Inter Quartile Range (IQR) is the middle 50% range of data [2](#)

Median [Median, stikprøvedmedian] The median of population or sample (note, in text no distinction between *population median* and *sample median*) [1](#)

probability density function The pdf is the function which determines the probability of every possible outcome of a random variable [3](#)

Quantile [Fraktil, stikprøvefraktil] The quantiles of population or sample (note, in text no distinction between *population quantile* and *sample quantile*) [1](#)

Quartile [Fraktil, stikprøvefraktil] The quartiles of population or sample (note, in text no distinction between *population quartile* and *sample quartile*) [2](#)

Sample variance [Empirisk varians, stikprøvevariens] [2](#)

Sample mean [Stikprøvegennemsnit] The average of a sample [1](#)

Standard deviation [Standard afvigelse] [2](#)

Acronyms

ANOVA Analysis of Variance *Glossary*: Analysis of Variance

cdf cumulated distribution function [3](#), *Glossary*: cumulated distribution function

CI confidence interval [17–19](#), [22](#), *Glossary*: confidence interval

CLT Central Limit Theorem *Glossary*: Central Limit Theorem

IQR Inter Quartile Range [2](#), *Glossary*: Inter Quartile Range

LSD Least Significant Difference *Glossary*: Least Significant Difference

pdf probability density function [3](#), *Glossary*: probability density function