Advanced Non-Life Insurance

Prof. dr. Tim Verdonck

Academic year 2018-2019

Contents

Chapter 1

Introduction to *R*

1.1 Download *R* and *RStudio* software

We will use statistical software *R* in combination with software program *RStudio*. Both programs are freeware and can be downloaded at

- Windows: `http://cran.r-project.org/bin/windows/`
- Mac: `http://cran.r-project.org/bin/macosx/`
- Linux: `http://cran.r-project.org/bin/linux/`

Although *R* is sufficient, it is advised to also download *RStudio* interface. This nice interface gives extra possibilities. *RStudio* can be downloaded at `http://www.rstudio.com/products/rstudio/download/`.

1.2 Introduction to *R* and *RStudio*

RStudio has 4 command windows as can be seen in 1.1.

- Command window or console is below on the left. All commands are executed in this windows. You can immediately type and run code here, but it is efficienter to create separate script files in the script window.
- Script window. You can open a new script by *File>New> R script*. In this file you can write code and then save the file with *File>Save*.
- Extra window 1. This window has two tabs: "Environment" and "History". History shows all commands that were executed, whereas Environment shows all variables, function, etc. that are currently in memory.
- Extra window 2. This window has 5 tabs. Using "Files" you can scroll to files and open them. Under "Plots" the figures are shown. "Help" is convenient to find information about functions. There is also "An Introduction to R" manual. "Packages" shows the packages, whereas "Viewer" can be used to study local web applications.

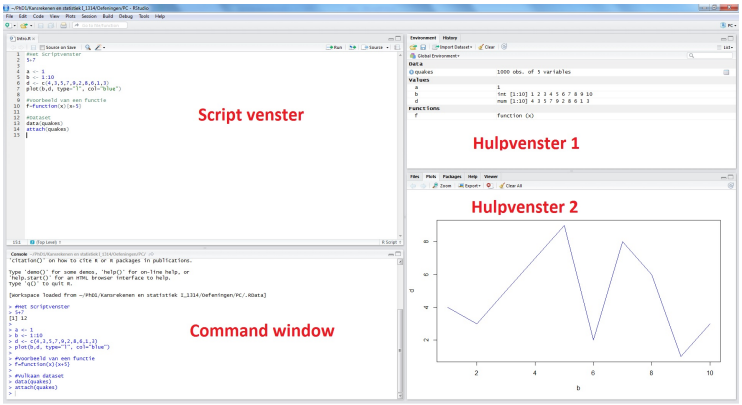


Figure 1.1: Windows in *RStudio*.

Open and type in following commands:

```
5+3
x <- 5+3
y = 9-2
x
x*y
```

To execute code from a script, you go that line and press *Ctrl+Enter* or click on button "Run". You can also select several lines simultaneously. In Environment you notice that variables x and y now contain values 8 and 7.

Remark: Do not use letters c and t for creating variables, since these are names of functions that already exist. Also avoid names as *log*, *exp*, ...

Execute following *RStudio*using a script file:

```
z <- log(x)
?log
a <- sqrt(x)*z^2
b <- exp(a+z)
rm(x,y)
ls()
rm(list=ls())
```

A lot of distributions are already defined in *R*. For every distribution there are typical 4 commands. We illustrate this for the continuous uniform distribution.

- `runif(n,min=a,max=b)` randomly generates n numbers from the uniform distribution between a and b. When a and b are absent, then the interval $[0, 1]$ is taken.
- `dunif(x,min=a,max=b)` computes the probability density function $f(x)$.
- `punif(x,min=a,max=b,lower.tail=TRUE)` computes the cumulative distribution function $F(x) = P(X \leqslant x)$. `lower.tail` has `TRUE` as default value. When using `FALSE`, one computes the survival function $P(X > x)$.
- `qunif(p,min=a,max=b,lower.tail=TRUE)` computes x such that $P(X \leqslant x) = p$ for $0 \leqslant p \leqslant 1$ (the p-th quantile). When using `FALSE` at `lower.tail`, one computes x such that $P(X > x) = p$, i.e. the $1 - p$-th quantile.

The names of the options (e.g. `min`) do not need to be mentioned if one keeps the predetermined order. Hence `runif(n,1,2)` yields same result as `runif(n,min=1,max=2)`. The option `lower.tail` has as default value `TRUE`. Hence when filling in nothing, the value `TRUE` is used.

An overview of distributions in `R` can be found in Table 1.1. As mentioned before: for every distribution, there exist 4 functions in `R` starting with `d`, `p`, `q` and `r` and computing pdf, cdf, quantile and random draws respectively. As an example, the function `pnorm` computes the cdf of a normal distribution.

Distribution	**Function (root)**	**Parameters**	**Defaults**
beta	beta	shape1, shape2	_, _
binomial	binom	size, prob	_, _
Cauchy	cauchy	location, scale	0, 1
chi-squared	chisq	df, ncp	_, 1
F	f	df1, df2	_, _
gamma	gamma	shape, rate, scale	_, 1, 1/rate
geometric	geom	prob	_
hyper-geometric	hyper	m, n, k	_, _, _
log-normal	lnorm	meanlog, sdlog	0, 1
logistic	logis	location, scale	0, 1
negative binomial	nbinom	size, prob, mu	_, _, _
normal	norm	mean, sd	0, 1
Poisson	pois	Lambda	1
Student's t	t	df, ncp	_, 1
uniform	unif	min, max	0, 1
Weibull	weibull	shape, scale	_, 1
Wilcoxon	wilcoxon	m, n	_, _

Table 1.1: Distributions in `R`.

1.3 Exercises

Make sure that you always create a script for solving exercises when using `R`!

1. (a) Create two samples via the `sample` command in the following manner:

```
set.seed(1)
X <- sample(1:6,100,replace = TRUE)
Y <- sample(1:6,10000,replace = TRUE)
```

`set.seed(1)` makes sure (although it is a random sample) that everybody gets the same results.
Note that this is similar to taking a sample from a discrete uniform distribution on $\{1, \ldots, 6\}$.

(b) Create a barplot.

(c) Calculate each of the characteristics on a theoretical manner for the discrete uniform distribution and by using `R` for your samples.

characteristic	X	Y	population
mean			
median			
variance			
standard deviation			
IQR			
MAD			
skewness			
kurtosis			

2. Let $X \sim \mathcal{U}(0, 20)$. Compute with `R`:

   $P(X > 12) =$ .....

   $P(4 < X \leq 12) =$ .....

   $f_X(7) =$ .....

   $f_X(21) =$ .....

   $Q_X(0.99) =$ .....

   $Med(X) =$ .....

   The largest of 10 random observations from X: .....

   The largest of 1000 random observations from X: .....

3. Fill in the table below.

distribution	question	answer
$X \sim NB(5, 0.7)$	$Q_X(0.9)$	
$X \sim \mathcal{P}(2)$	$P(X \geq 5)$	
$X \sim \mathcal{N}(10, 16)$	$F_X(5)$	
$X \sim \beta(5, 2)$	$f_X(0.7)$	
$X \sim \mathcal{N}(1, 4)$	$P(exp(X) > 5)$	
$X \sim \Gamma_{0.5,4}$	$P(X < 8)$	
$X \sim F_{5,8}$	$Q_X(0.95)$	
$T \sim t_{500}$	$Q_T(0.99)$	
$Z \sim \mathcal{N}(0, 1)$	$Q_Z(0.99)$	

4. Plot the pdf of standard normal distribution on interval $[-3, 3]$.

5. Plot the cdf of $Y_1 \sim \mathcal{N}(0, 1)$ and $Y_2 \sim \mathcal{N}(0, 2)$ on interval $[-4, 4]$.

6. (a) Generate 5000 observations from the exponential distribution with parameter $\lambda = 2$ by using `rexp` and by using probability integral transformation and `runif`.

   (b) Plot a histogram and the kernel density estimate of the sample.

7. Look at the following code:

```
k <- 2500
gem <- rep(0,k)
set.seed(4)
for(i in 1:k)
{
gem[i] <- mean(rnorm(100,mean=10,sd=3))
}
par(mfrow=c(2,2))
hist(gem)
plot(density(gem))
gemF <- ecdf(gem)
plot(gemF, verticals= TRUE, do.points = FALSE)
qqnorm(gem)
qqline(gem)
par(mfrow=c(1,1))
mean(gem)
sd(gem)
```

   (a) What do we illustrate here?

   (b) Repeat the experiment but now change the mean, sd, number of observations and distribution.

8. Suppose that $Y_1, \ldots, Y_n$ are idd $N(\mu, \sigma^2)$ where μ is known. Calculate the MLE of σ^2.

9. Suppose that $X_1, \ldots, X_n$ are iid Exp(λ). Calculate the MLE of λ.

10. Let X be random variable with mean μ and variance σ^2. Show that the kurtosis of X plus 2 is equal to the variance of $\left(\frac{X-\mu}{\sigma}\right)^2$.

Chapter 2

Exploratory data analysis

2.1 Introduction

When performing a data analysis, a first step should be to look at the data and to plot the data in several ways. This makes it possible

- to catch mistakes
- to see patterns in the data
- to find violations of statistical assumptions
- to generate hypotheses

By performing a visual examination, some problems like bad data, outliers, mislabeling of variables, missing elements, an unsuitable model, ... can often already be detected.

Bad data, i.e. outlying data because of errors, should be corrected when possible and deleted otherwise. However, note that outliers are not always bad data! For example, outliers due to a stock market crash are good data and should be kept in mind (perhaps one has expand the model to accommodate them). It is important to detect atypical observations and to investigate and understand them so that appropriate action can be taken!

Exploratory data analysis refers to methods for describing and summarizing data and is useful in revealing the structure of the data. We will discuss methods that are useful in displaying the distribution of data values, which play the role for data that the probability density function or the cumulative density function plays for a random variable.

We will focus on simpler numerical summaries that indicate a central value of the data or a quantification of the spread, so a more condensed summary than graphical displays. Data summaries (descriptive statistics) will be used to describe certain features of the data, to learn about the unknown probability density function and to capture the observed dependencies in the data.

We distinguish between two types of data:

- Categorical or qualitative data
  - binary
  - nominal
  - ordinal
- Quantitative or numeric data
  - discrete
  - continuous

Data can be of different dimensions too. We can have univariate, bivariate and multivariate data.

To end, we can consider several basic numerical summaries of the data like measures of location or center, measures of dispersion or scale (level of scatteredness), ...

2.2 The empirical cumulative distribution function (EDF) and quantiles

Let $Y_1, \ldots, Y_n$ be a random sample from a probability distribution that has cumulative distribution function F.
A frequently used concept is that of 'order statistics'. Order statistics $Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}$ are nothing more than the values $Y_1, \ldots, Y_n$ ordered from smallest to largest. We can use them for instance to define a sample cumulative distribution function.

Definition 2.2.1. The **sample or empirical cumulative distribution function $\hat{F}(y)$ or $F_n(y)$** is defined to be the proportion of the sample that is less than or equal to y:

$$\hat{F}(y) = \frac{\sum_{i=1}^{n} \mathbb{1}_{Y_i \leq y}}{n}$$

where $\mathbb{1}_{(.)}$ is the indicator function:

- $\mathbb{1}_{Y_i \leq y} = 1$ if $Y_i \leq y$
- $\mathbb{1}_{Y_i \leq y} = 0$ if $Y_i > y$

From Definition 2.2.1 it follows directly that

$$\hat{F}(Y_{(k)}) = \frac{k}{n}$$

In `R` one can use the function `ecdf()`.

Example 2.2.2. We look at F_n for a sample of size 150 from $N(0, 1)$. We see that the sample cumulative distribution function F_n (solid line) differs from the true cumulative distribution function F (dashed line). This is due to random variation.

In the previous chapter we already considered the inverse of the cumulative distribution function. We can extend that now to the empirical case.

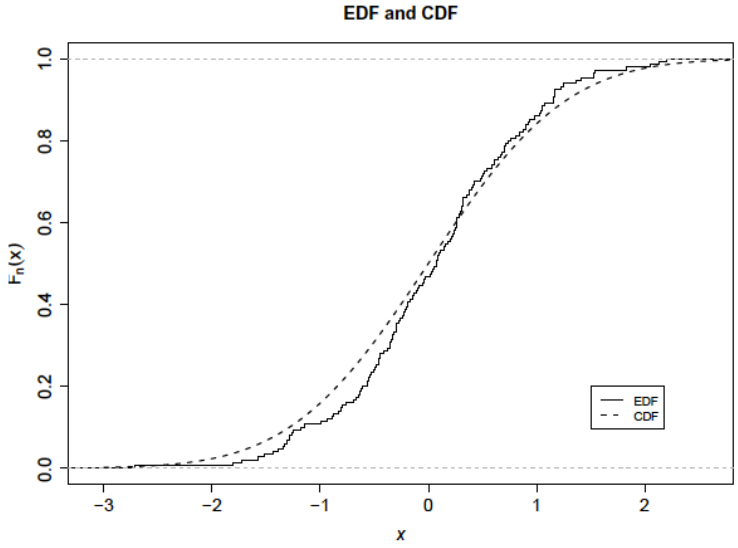


Figure 2.1: Cumulative and empirical distribution function of a normally distributed sample.

Definition 2.2.3. The **sample quantile function** is the inverse of the empirical cumulative distribution function. It provides us the value x such that $P(X \leq x) = \alpha$ for a given α in the following way:

$$\hat{Q}(\alpha) = \hat{F}^{-1}(\alpha) = Y_{(k)}$$

where k is αn rounded to an integer.

For $\alpha \in [0, 1]$, the α sample quantile is the data value $\hat{q}_\alpha$ such that $\alpha 100\%$ of the data are less than $\hat{q}_\alpha$. Further we call the qth quantile often the '$100q$th percentile.

Some quantiles are more special than others and therefore have a specific name:

- 0.5 sample quantile: median
- 0.25 sample quantile: first quartile
- 0.75 sample quantile: third quartile
- $\{0.2, 0.4, 0.6, 0.8\}$ sample quantiles: quintiles

If we choose $\alpha = 1/n, \ldots, (n-1)/n, n$ for a data sample $x_1, \ldots, x_n$, it holds that

$$\hat{Q}(i/n) = x_{(i)}$$

Remark 2.2.4. In practice it is better to use for example $\frac{i}{n+1}$ or $\frac{i-0.5}{n}$ instead of i/n.

Indeed, consider now some increasing

	$x_{(1)}$	$x_{(2)}$	$x_{(3)}$	$x_{(4)}$
	1.4	2.2	2.8	3.5
$\hat{F}(x_{(i)})$	25%	50%	75%	100%

and some decreasing sample order statistics

	$x^*_{(1)}$	$x^*_{(2)}$	$x^*_{(3)}$	$x^*_{(4)}$
	3.5	2.8	2.2	1.4
$\hat{F}^*(x_{(i)})$	25%	50%	75%	100%

The goal is now to have

$$\hat{F}(x_{(i)}) = 1 - \hat{F}^*(x_{(i)})$$

Therefore

$$\hat{F}(x_{(i)}) = \frac{i-a}{n+1-2a}$$

where a is just a random number.

For $a = 0.5$ we obtain the special case of the Hazen plot position:

$$\hat{Q}\left(\frac{i-0.5}{n}\right) = x_{(i)}$$

Intuitively, $i-0.5$ corrects for the fact that the values of $x_{(i)}$ are rounded and that the true value can be a little bit smaller or larger. Alternatively, one could also use the correction of Weibull:

$$\hat{Q}\left(\frac{i}{n+1}\right) = x_{(i)}$$

To conclude this section we give an overview of some useful functions in `R`:

function	description
`sort()`	sort elements of data vector
`min()`	compute minimum value of data vector
`max()`	compute maximum value of data vector
`range()`	compute min and max of a data vector
`quantile()`	compute empirical quantiles
`median()`	compute median
`IQR()`	compute inter-quartile range
`summary()`	compute summary statistics

2.3 Sample statistics

2.3.1 Sample statistics of location

We give here an overview of the most common sample statistics of location.

Sample statistics of location

- **Sample mean**

$$\hat{\mu} = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- **Sample median**: if the sample size is an odd number, then the median is defined to be the middle value of the ordered observations; if the sample size is even, then the median is the average of the two middle values.

$$med = \begin{cases} X_{(n+1)/2} & n \text{ is odd} \\ X_{(n/2)} + X_{((n/2)+1)} & n \text{ is even} \end{cases}$$

- **100α% Trimmed mean**: order the data, discard the lowest $100\alpha\%$ and the highest $100\alpha\%$ and take the arithmetic mean of the remaining data:

$$\overline{X}_\alpha = \frac{X_{([n\alpha]+1)} + \ldots + X_{(n-[n\alpha])}}{n - 2[n\alpha]}$$

where $[n\alpha]$ denotes greatest integer less than or equal to $n\alpha$.

Remark 2.3.1. If the underlying distribution is symmetric, the sample mean, the sample median and the trimmed mean all estimate the center of symmetry.

Remark 2.3.2. In case of a Gaussian distribution the sample mean is optimal, but other measures are more robust.

2.3.2 Sample statistics of dispersion

Below you find a list of sample statistics of dispersion. Note that we encountered the non-sample version already in the previous chapter.

Sample statistics of dispersion

- **Sample variance**

$$\hat{\sigma}^2 = S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

- **Mean absolute deviation**

$$\text{MeAD}_n = \frac{1}{n}\sum_{i=1}^{n}|X_i - \overline{X}|$$

- **Median absolute deviation**

$$\text{MAD}_n = \text{med}(|X_i - \text{med}(X)|)$$

- **InterQuartile Range (IQR)**

$$\text{IQR}_n = \hat{Q}(0.75) - \hat{Q}(0.25)$$

- **Sample covariance**

$$\hat{\text{cov}} = S_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})$$

- **Sample correlation**

$$\hat{\rho} = r = \frac{S_{XY}}{S_X S_Y}$$

- **Sample skewness**

$$\hat{\gamma_3} = \frac{1}{n-1}\sum_{i=1}^{n}\frac{(X_i - \overline{X})^3}{S^3}$$

- **Sample kurtosis**

$$\hat{\gamma_4} = \frac{1}{n-1}\sum_{i=1}^{n}\frac{(X_i - \overline{X})^4}{S^4} - 3$$

- **Sample moment**

$$a_k = \frac{\sum X_i^k}{n}$$

- **Sample central moment**

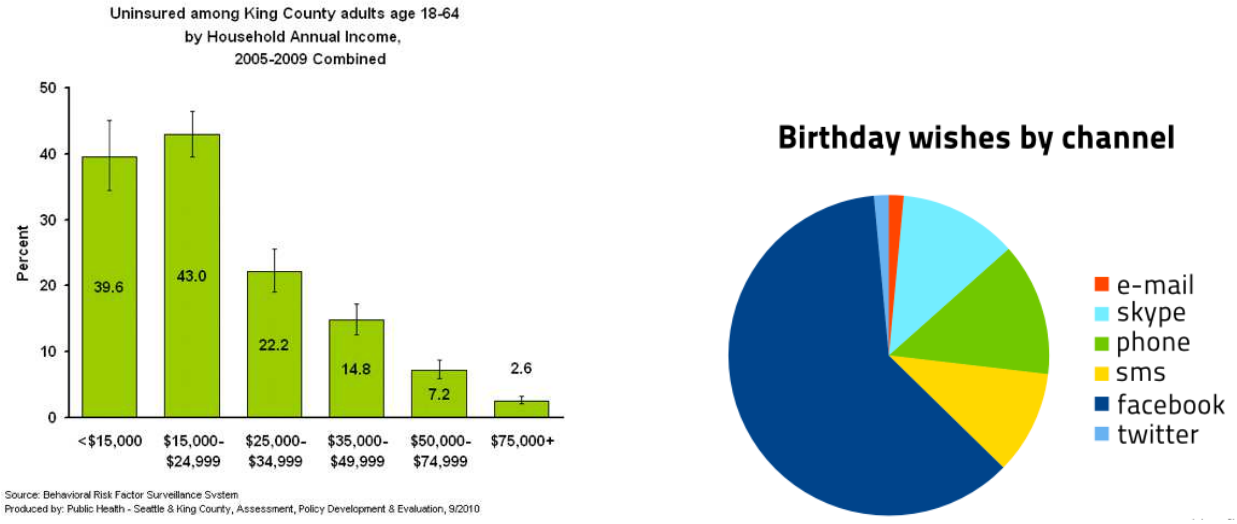
$$m_k = \frac{\sum (X_i - \overline{X})^k}{n}$$

Some useful `R` functions can be found in the table below

function	package	description
`mean()`	base	compute sample mean
`colMeans()`	base	compute column means of matrix
`var()`	stats	compute sample variance
`sd()`	stats	compute sample standard deviation
`skewness()`	PerformanceAnalytics	compute sample skewness
`kurtosis()`	PerformanceAnalytics	compute sample excess kurtosis
`apply()`		apply functions over rows or columns of a matrix or data.frame

The package `fBasics` includes some more useful functions too.

2.4 Bar plot and histogram



(a) Bar chart.

(b) Pie chart.

Figure 2.2: Pictorial representation of qualitative data.

In order to get an idea about qualitative data, a bar chart or pie chart can be used. Such a chart is a pictorial representation that shows the frequency or proportion in each category of the dataset.

To create such charts, the `R` functions `table()`, `prop.table()`, `barplot()`, `pie()`,... can be used.

Assume that the marginal cumulative distribution function F has a probability density function f and that our goal is to describe the shape of the distribution of the data. A very simple and well-known estimator for f is a histogram, which divides the real numbers in classes or cells that have a limited length.

In order to construct such a histogram, one should perform the following steps:

1. Order the data from the smallest to the largest values.
2. Divide the range into N equally spaced cells or bins.
3. Count the number of observations in each cell.
4. Create a bar chart to show the counts or the proportion in each category.

In `R` one needs to use the command `hist()` with options `freq` and `breaks`. One needs to look for a balance in order to obtain a reasonable histogram. If the bin width is too small, the histogram is too ragged. On the other hand, if the bin is too wide, its shape is over-smoothed and obscured. We now have a look at some examples which show that the bin width is quite important.

Example 2.4.1. The following figure shows histograms of the S&P 500 log returns. For instance in Figure 2.3(a) 30 cells are used. However, outliers (at least one due to Black Monday) are difficult to see.

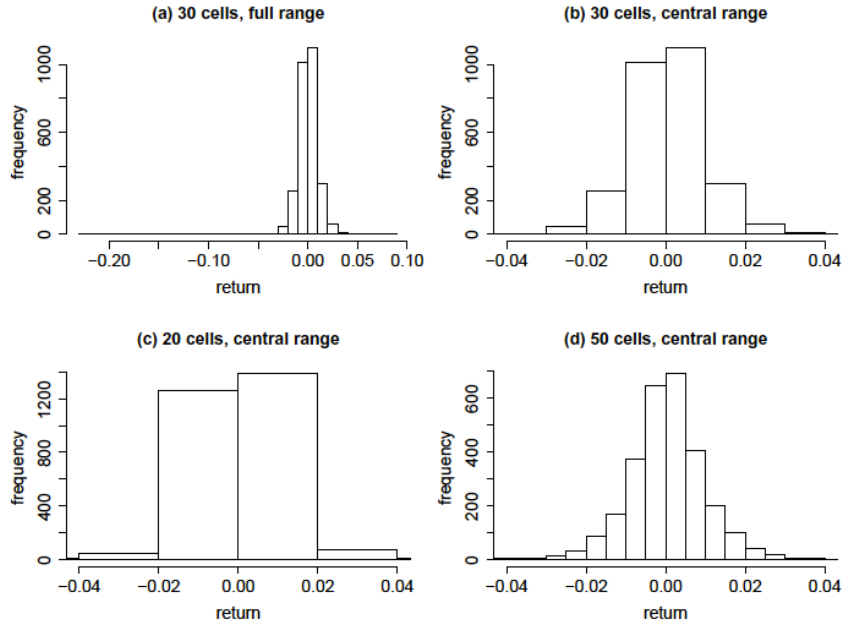


Figure 2.3: Histograms of S&P 500 log returns, using a different number of cells.

Example 2.4.2. A random sample of n = 400 people in a small town were asked to state their most recent annual income.

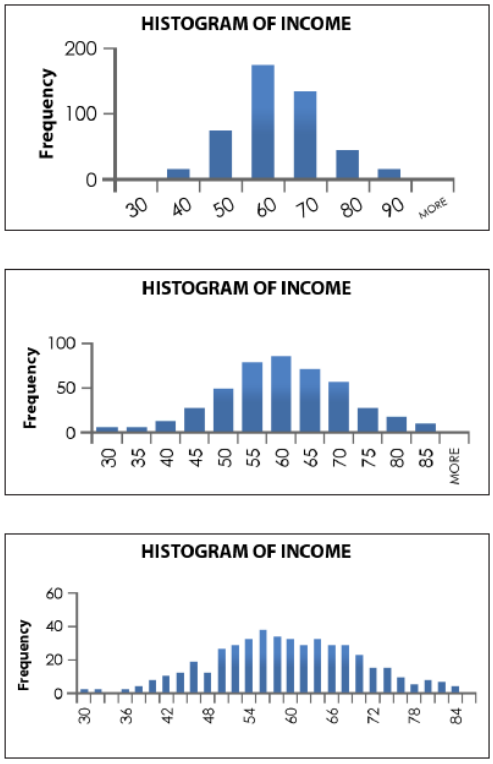


Figure 2.4: Histograms of income with a varying number of cells.

Example 2.4.3. This example looks at histograms of the fiscal incomes in 2006 in Flanders. It shows us that a (frequency) histogram can be misleading if the bins are not of equal size.

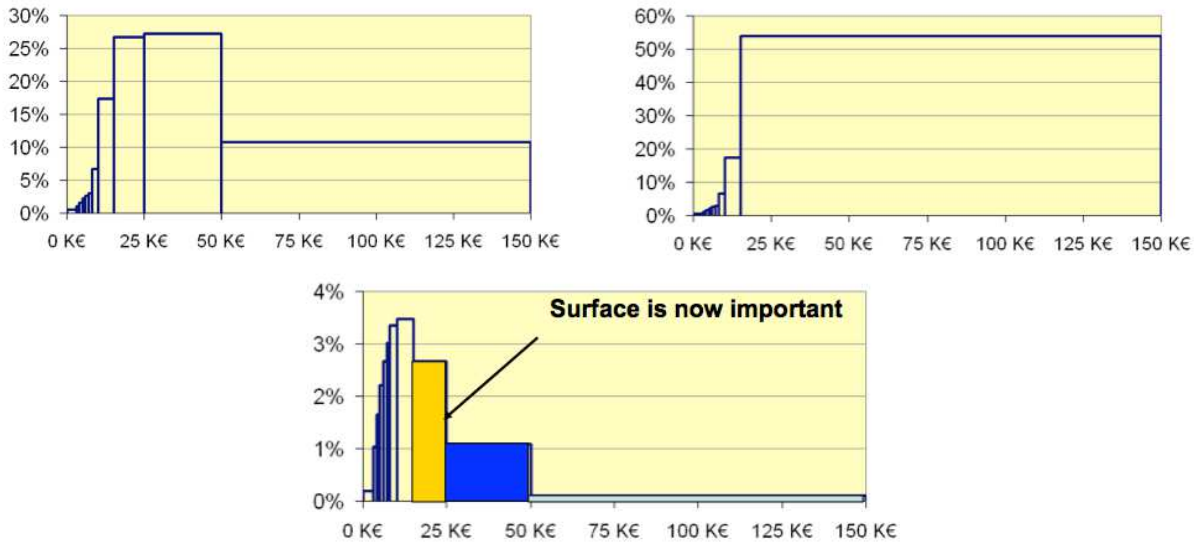


Figure 2.5: Histograms of the fiscal incomes in 2006 in Flanders.

The area under the density histogram is 1 and the relative frequencies are now given by the surface of the bars, not by the heights.

2.5 Kernel density estimator

A histogram is easy to construct but is often a crude density estimator, it is not really smooth. Moreover it is very sensitive to the number and the locations of the cells.

As an alternative, we can look for a kernel density estimator, which takes its name from the so-called 'kernel function' K. This is a probability density function that is symmetric around 0.

Definition 2.5.1. The **kernel density estimator** based on $Y_1, \ldots, Y_n$ (sample from pdf f) is

$$\hat{f}(y) = \frac{1}{nb} \sum_{i=1}^{n} K\left(\frac{y - Y_i}{b}\right)$$

where b is the bandwidth and determines the resolution of the estimator. The kernel density estimator consists of the superposition of the *hills* centered on the observations. The result for specific data is a kernel density estimate.

We will use here the standard normal density function, which is a common choice for K. In `R` one can use the function `density()`.

Example 2.5.2. Consider a small simulated data set of 6 observations from the standard normal distribution. This is a small sample size just for visual clarity, it does not give an accurate estimate. The 6 data points are shown at bottom of the figures as vertical lines.

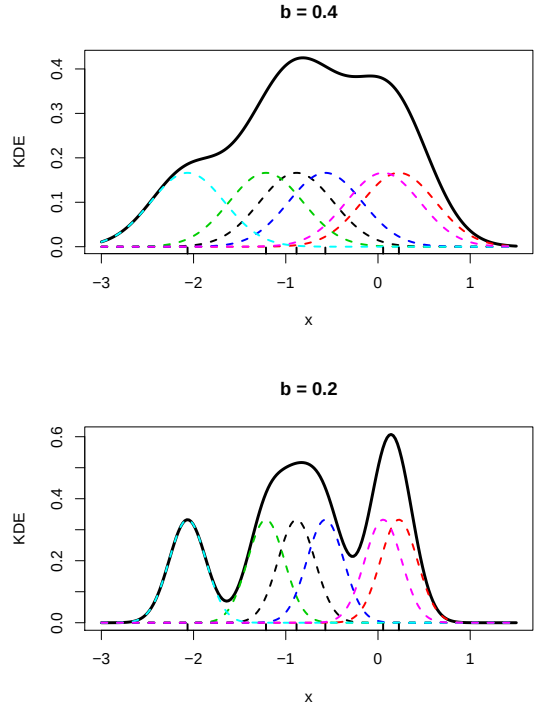


Figure 2.6: Kernel density estimates for different values of bandwidth b.

One can see that a small b results in high variance and low bias, whereas a large b gives low variance and high bias.

Appropriate values of the bandwidth b in Definition 2.5.1 depend on both the sample size n and the true (unknown) density. Fortunately, large amount of research has been devoted to automatic selection of b, which, in effect, estimates the roughness of the true density. Many methods have been proposed for the selection of the bandwidth parameter in kernel density estimation and several choices of calculated bandwidths are available in `R`, where the default is `nrd0` for historical and compatibility reasons (rather than as a general recommendation).

`nrd0` implements a rule-of-thumb (Silverman, 1986) for choosing the bandwidth of the Gaussian kernel density estimator:

$$0.9 \min(s_n, IQR_n/1.34) n^{-1/5}$$

Another common method is `SJ`, which refers to the method by Sheather and Jones (1991) and which selects the bandwidth by using pilot estimation of derivatives. However, automatic bandwidth selectors are very useful and often used as a starting point for further fine-tuning as the next example illustrates.

Example 2.5.3. The solid curve shows the result by using the default bandwidth from the `R` function `density()`. The dashed curve shows the kernel density estimate when the default bandwidth is multiplied by 1/3. We see that this curve is wiggly (too much random variability). The dotted curve is the resulting kernel density estimate when the default bandwidth is multiplied by 3. This curve is very smooth but underestimates the peak near 0, which is a sign of bias.

Over- and underfitting means a poor bias-variance tradeoff where an overfitted curve thus has too much variance and an underfitted curve has too much bias.

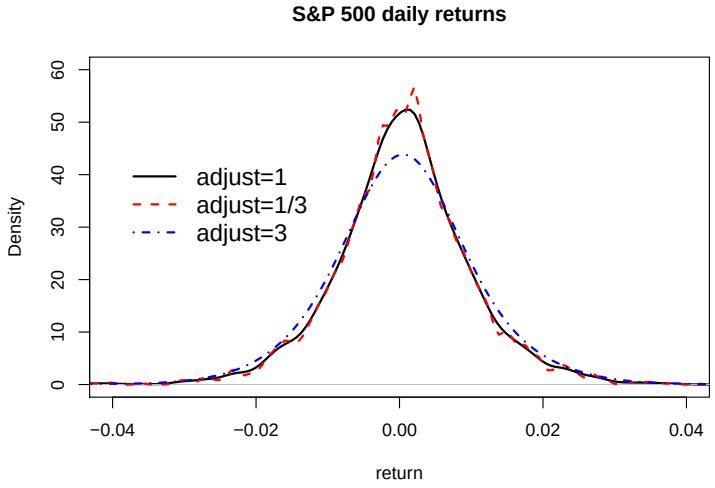


Figure 2.7: Kernel density estimates with different values for b.

One can see in Figure 2.7 that the density estimate is bell-shaped, suggesting that a normal distribution might be suitable. We can then compare the kernel density estimates with normal densities

Figure 2.8(a) by using sample mean and standard deviation of the returns

Figure 2.8(b) by using robust estimators sample median and MAD.

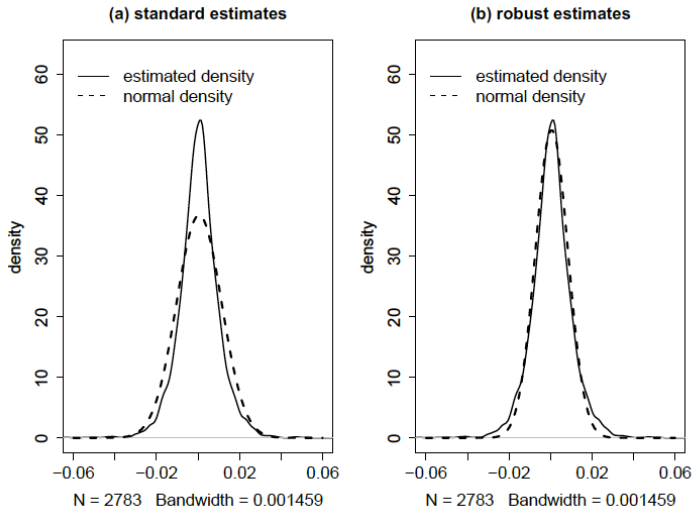


Figure 2.8: Normal densities based on standard, resp. robust estimates.

The results are dissimilar. The reason is that the outlying returns inflate the classical estimators and cause the normal density to be too dispersed in the middle of the data. The need for robust estimators is itself a sign of non-normality. (The t-distribution provides a better model).

2.6 Boxplot

A boxplot is a useful graphical tool for comparing several samples. Although they are most commonly used to compare distributions among groups, boxplots can also be drawn to summarize a single sample, providing a quick check of symmetry and the presence of atypical observations.

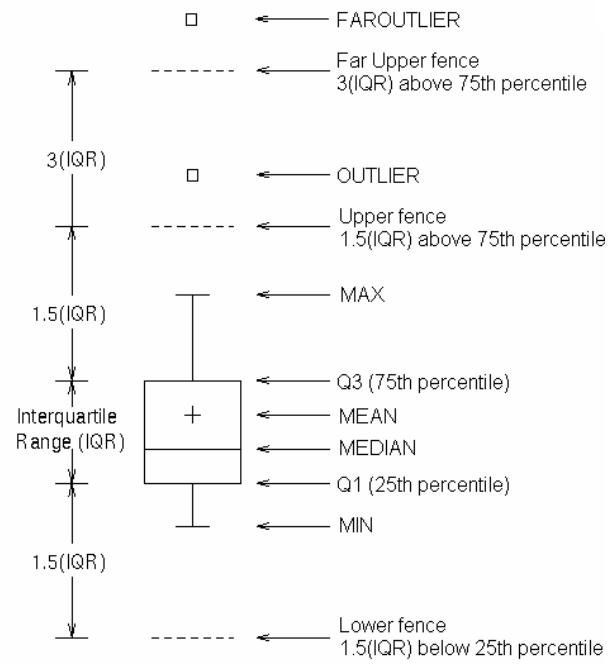


Figure 2.9: General boxplot.

The box in the middle extends from the first to the third quartile and thus gives the range of the middle half of the data (IQR). Further, the line in the middle of the box is at the median and the whiskers are vertical dashed lines extending from the top and the bottom of each box. They extend to the smallest and largest data points whose distance from the bottom or top of the box is at most 1.5 times the IQR. Observations beyond whiskers are plotted separately. In `R` the command `boxplot()` can be used, Figure 2.9 shows how a boxplot looks like in general.

Example 2.6.1. We focus on data from Ornstein (1976) on interlocking directorates among 248 major Canadian corporations:

- assets of corporation in millions of dollars
- corporation's sector of operation (categorical variable with 10 levels)
- nation in which the firm is controlled (CAN, UK, US or OTHER)
- number of interlocking directorate and executive positions maintained between each company and others in the data set

We plot a boxplot of the number of interlocks for each level of nation.

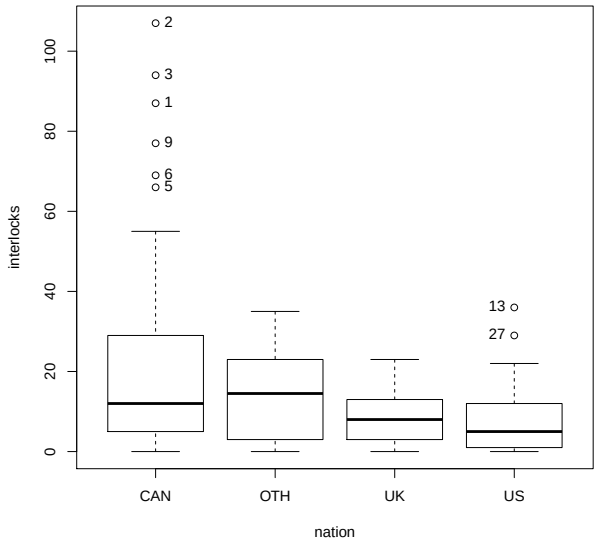


Figure 2.10: Boxplots of the number of interlocks.

2.7 Tools to check for normality

Many statistical models assume that a random sample comes from a normal distribution. This is fine in many domains of application where data are well approximated by a normal distribution. For example, biological variables such as blood pressure, serum cholesterol, height and weight naturally follow a normal distribution.

In practice, data are often not normal or the sample size is not large enough to gain the benefit of the central limit theorem (certainly with skewed data). In finance, stock returns, changes in interest rates, changes in foreign exchange rates, and other data of interest have often many more outliers

than would occur under normality. For modelling financial markets, the normal distribution is not normal and it is more suitable to consider heavy-tailed distributions, such as the t-distribution. Therefore it is vital to check if the normality assumption is justified!

2.7.1 Normal probability plot

In order to check the assumption of normality, normal probability plots can be used. If the assumption is false, those plots may be used to investigate how the distribution differs from the normal distribution.

Recall from before that if $X \sim N(\mu, \sigma^2)$, then for quantile function $Q(.)$ it holds that

$$Q(\alpha) = \mu + \sigma\Phi^{-1}(\alpha) \qquad \forall 0 < \alpha < 1$$

where Q is the inverse of the cumulative distribution function F ($P(X \leq Q(\alpha)) = \alpha$) and where $\Phi^{-1}(.)$ is the quantile function of the standard normal distribution.

In case of a normal distribution, the points $(\Phi^{-1}(\alpha), Q(\alpha))$ lie on a perfect line for different values of α. However, in practice we have to approximate the population quantile function $Q(.)$ by the empirical quantile function $\hat{Q}(.)$. If the sample $x_1, \ldots, x_n$ is normally distributed, then the points $(\Phi^{-1}(\alpha), \hat{Q}(\alpha))$ lie approximately on a straight line. A systematic deviation of the plot from a straight line is evidence of non-normality.

Remark 2.7.1. Since $\hat{Q}(\frac{i-0.5}{n}) = x_{(i)}$, we choose $\alpha = (1-0.5)/n, \ldots, (n-1.5)/n, (n-0.5)/n$.

In case the pattern in a normal plot is non-linear, one checks where the plot is convex and where it is concave to interpret the pattern.

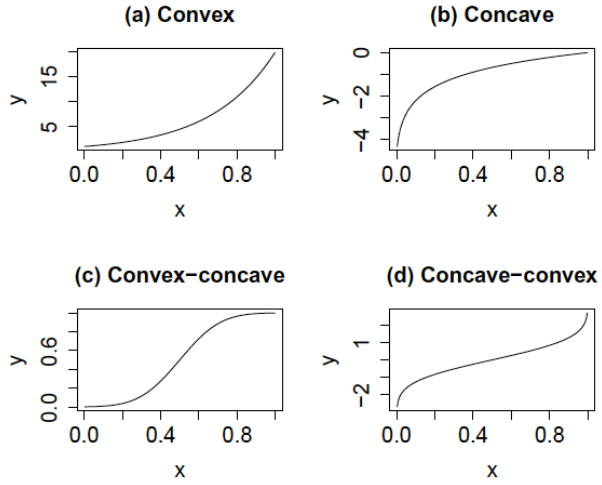
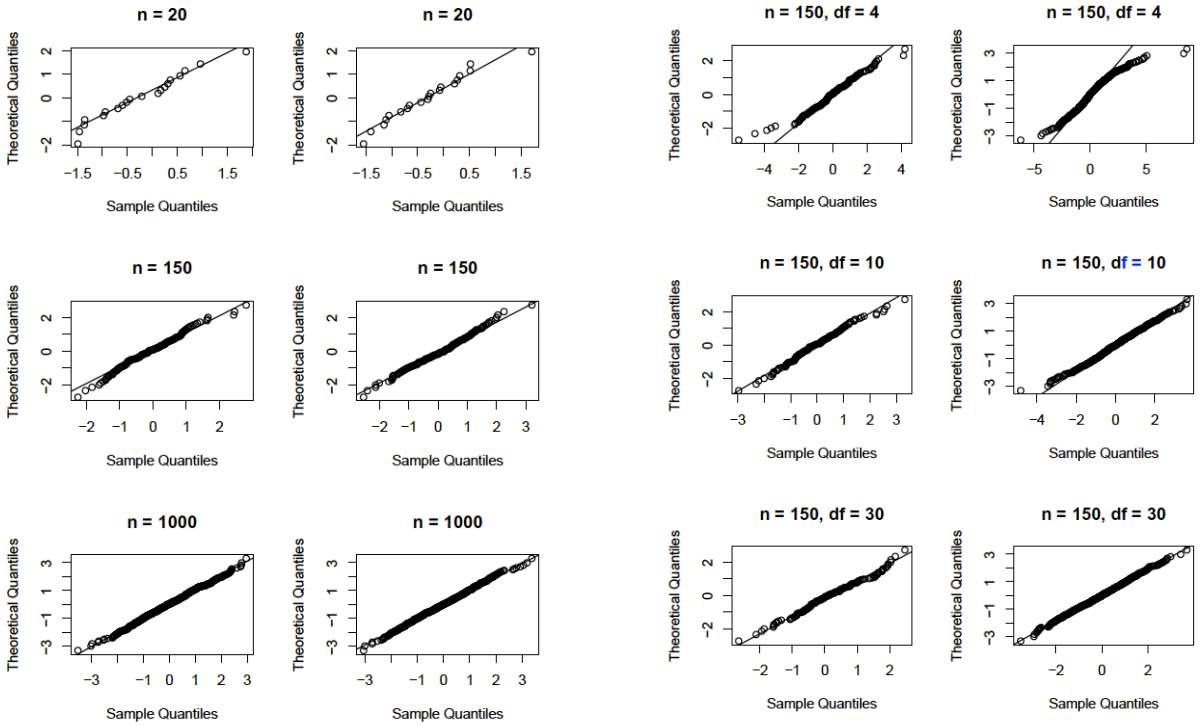


Figure 2.11: Different patterns in a normal probability plot.

Figure 2.11 (a),(b),(c) and (d) respectively indicate left skewness, right skewness, heavier tails than normal and lighter tails than normal. Be aware that it is essential to know which axis contains the data! In `R` the option `datax=TRUE` means that the data is on the x-axis and the theoretical quantiles are on the y-axis.

Example 2.7.2. This example shows on the one hand normal probability plots of samples of size 20, 150 and 1000 from a normal distribution. To show the typical amount of random variation in normal plots, two independent samples are shown for each sample size. On the other hand we consider normal probability plots of samples of size 150 from a t distribution with 4, 10 and 30 degrees of freedom (df).



(a) Samples from a normal distribution.

(b) Samples from a t distribution.

Figure 2.12: Normal probability plots.

We see that even for normally distributed data, some deviation from linearity is to be expected, especially for smaller sample sizes. Often a reference line is added to the normal probability plot to help the viewer determine whether the plot is reasonably linear.

For the samples of the t distribution we see heavy tails (extreme observations on both the left and right sides are significantly more extreme than they would be for a normal distribution), but none of the samples is skewed.

Sometimes a normal probability plot has a more complex behaviour, as Figure 2.13 shows. Here the kernel density estimator will reveal trimodality (often mixture of several distinct groups of data). Note that a reference line is added to help us to determine whether the plot is reasonably linear. By using the command `qqline()` in `R`, we obtain a line that goes through the pair of the first quartiles and the pair of the third quartiles.

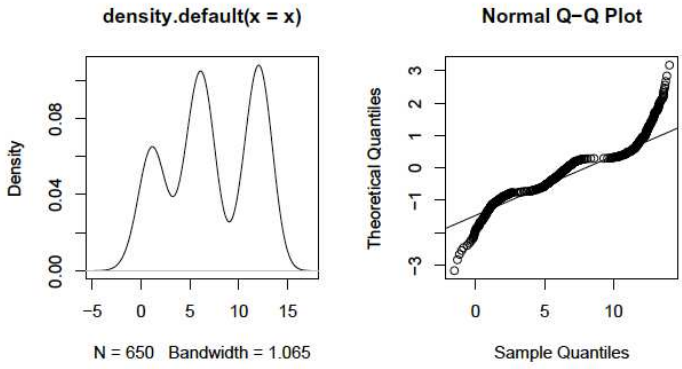


Figure 2.13: Kernel density estimate and corresponding normal probability plot.

2.7.2 Quantile-quantile plots

Normal probability plots are in fact just a special case of QQ plots. A QQ plot is in fact nothing more than a plot of quantiles of one sample/distribution against quantiles of another sample/distribution. For instance, we can plot sample quantiles against quantiles of the $t_6(0, 1)$ distribution. In `R` we can use the function `qqplot()` to do this.

A QQ plot is useful for comparing our data with the quantiles of a distribution that we think is appropriate for our data. Therefore we should interpret such a plot in the following way:

- If the points fall close to a straight line, the conjectured distribution is appropriate.
- If the points do not fall close to a straight line, the conjectured distribution is not appropriate and we should consider a different distribution.

Further, a QQ plot is also useful for comparing two samples by plotting order statistics against each other, where the samples should have the same size.

The general approach to construct a QQplot is as follows:

(i) Start from characterizing a linear relationship between the theoretical quantiles $Q(p)$ from the proposed distribution and the computable quantiles $Q_s(p)$ from a standard distribution from the parametric model.

(ii) Replace the theoretical quantiles $Q(p)$ by the corresponding empirical quantiles $\hat{Q}_n(p)$.

(iii) Plot the empirical quantiles $\hat{Q}_n\left(\frac{i}{n+1}\right) = x_{i,n}$ against the corresponding specific quantiles $Q_s\left(\frac{i}{n+1}\right)$.

(iv) Inspect the linearity in the plot: strong linearity implies a good fit.

Hence QQplots can be used in cases that are more general than the normal distribution, e.g. exponential, lognormal, Pareto, Weibull, ...

We now look for a moment at the special case of the lognormal quantile plot, which is a graphical technique to verify whether data follows a lognormal distribution. In this case we look at

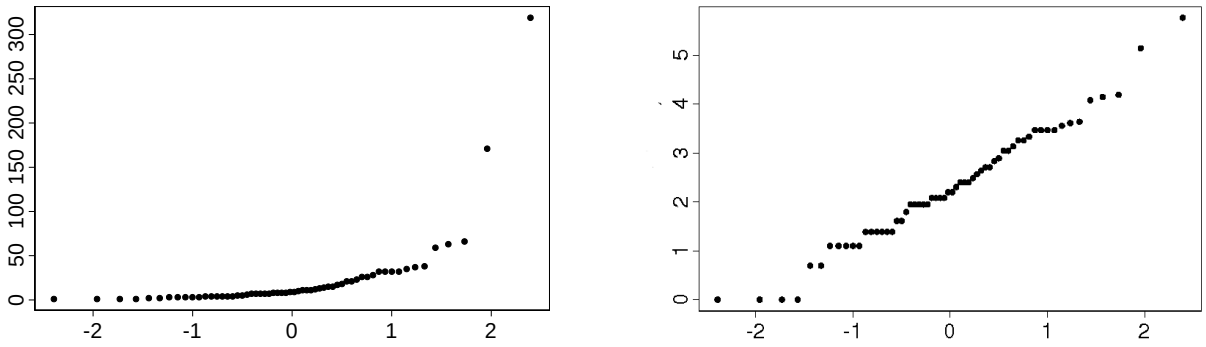
$$\ln(x_{i,n}) \qquad vs \qquad \Phi_X^{-1}\left(\frac{i}{n+1}\right)$$

For the lognormal probability density function we have

$$F_X(x) = \Phi\left(\frac{\ln(x) - \mu_{\ln(X)}}{\sigma_{\ln(X)}}\right) \Rightarrow \ln(x_{i,n}) = \mu_{\ln(X)} + \sigma_{\ln(X)}\Phi^{-1}(p_{i,n})$$

where $\Phi^{-1}(.)$ is the quantile function of the standard normal distribution.

Figure 2.14 shows a normal and a lognormal QQ plot.



(a) Normal QQ plot.

(b) Lognormal QQ plot.

Figure 2.14: QQ plots.

We see that X is skewed but $Y = \log(X)$ is normally distributed.

Example 2.7.3. We have a look at S&P500 log returns. We consider a normal plot in Figure 2.15(a), and t-plots with df (degrees of freedom) 1, 2, 4, 8 and 15 in the rest of Figure 2.15. None of these plots looks exactly linear, but the t-plot with df 4 is rather straight.

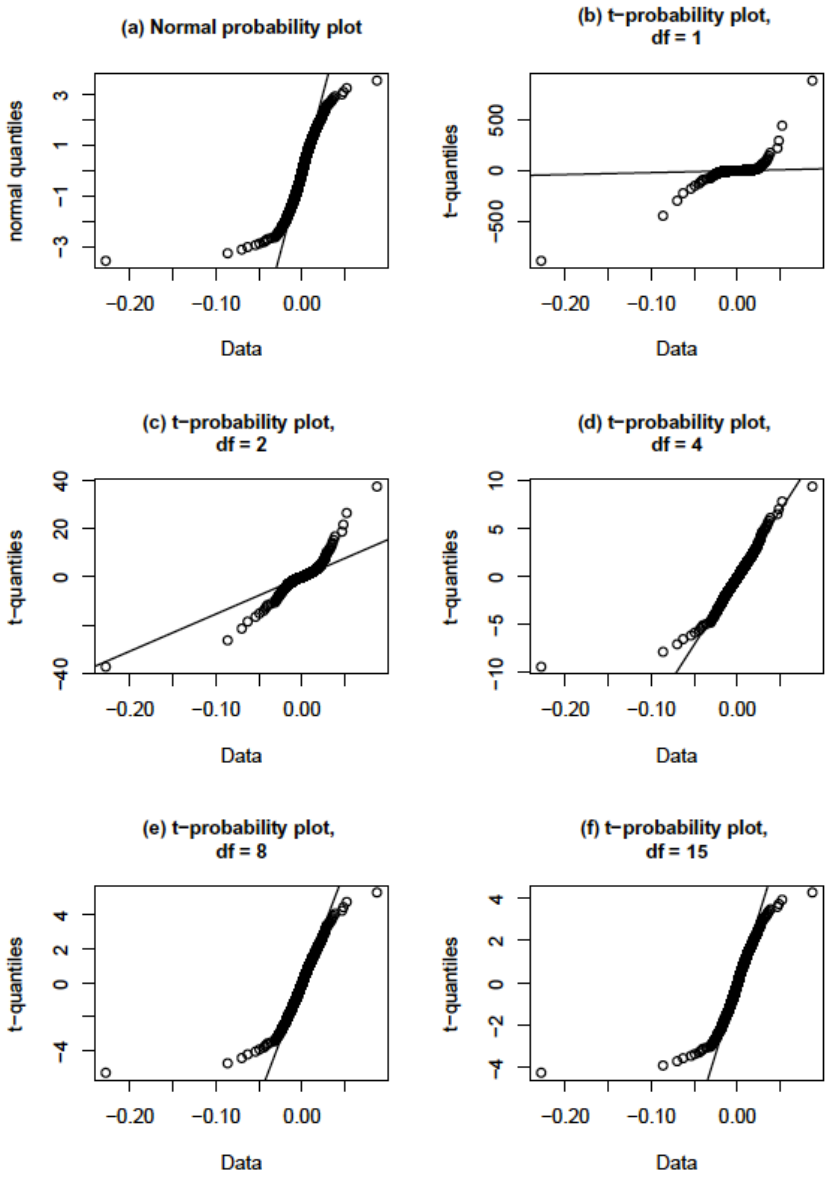


Figure 2.15: QQ plots of S&P500 log returns.

As mentioned before, QQ plots are not only useful for comparing a sample with a theoretical model, but also for comparing two samples with each other. If both samples have the same size, then we plot their order statistics against each other. Otherwise, one computes the same sets of sample quantiles for each sample and plots them.

The interpretation of a convex, concave, ... pattern is similar as before:

- Concave (resp. convex) implies that sample on x-axis is more right skewed (resp. left skewed) than sample on y-axis.
- Convex-concave (resp. concave-convex) implies that sample on x-axis is more (resp. less) heavy-tailed than sample on y-axis.

Remark 2.7.4. A QQ plot tells us nothing about dependencies between the samples!

Example 2.7.5. We look at sample QQ plots for all 3 pairs of the 3 time series (S&P 500 returns, changes in DM/dollar rate, changes in risk-free rate).

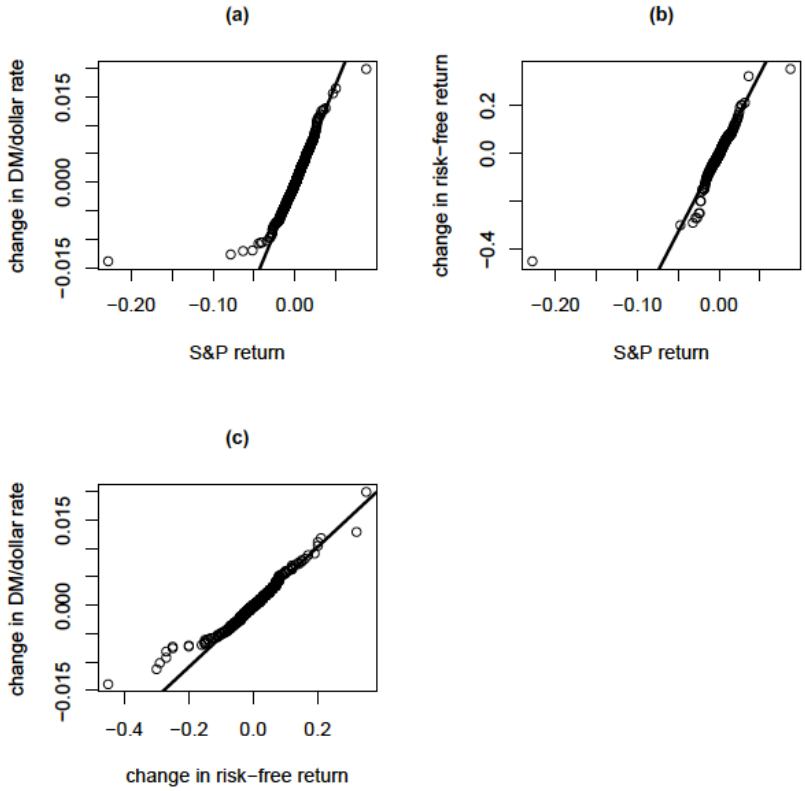


Figure 2.16: Sample QQ plots for 3 pairs of time series.

The S&P 500 has more extreme outliers than others. We see that changes in DM/dollar and risk-free rate have similar shapes, but changes in risk-free rate have slightly more extreme outliers in the left tail. However, this tells us nothing about dependencies between the series and moreover the 3 series were observed on different time intervals.

2.7.3 Tests of normality

We have seen 3 graphical ways to check for normality: boxplot, histogram and QQ plot. When viewing e.g. a normal plot, it is often difficult to judge whether any deviation from linearity is systematic or merely due to sampling variation. Therefore a really statistical test of normality is useful.

A formal test for normality is the **Shapiro-Wilk test** which has the following obvious null and alternative hypothesis:

$$H_0 : X \text{ is normally distributed.}$$
$$\text{versus}$$
$$H_1 : X \text{ is not normally distributed.}$$

The Shapiro-Wilk test corresponds with correlation r_Q of the points on a normal QQ plot and therefore lies between 0 and 1. If r_Q is close to 1, then there is a strong linear relationship in the QQ plot and the data points are probably generated from a normal distribution. In `R` the Shapiro-Wilk test can be executed by using the function `shapiro.test()`.

Alternatively, the **Anderson-Darling**, **Cramér-von Mises** and **Kolmogorov-Smirnov tests** compare the sample CDF to the normal CDF with mean equal to $\overline{Y}$ and variance equal to s_Y^2. The Kolmogorov-Smirnov-test is the maximum absolute difference between these two functions, while the other two tests are based on a weighted integral of the squared differences.

Another test for normality is the **Jarque-Bera test**, which compares the sample skewness and kurtosis to 0, the value under normality. The Jarque-Bera test statistic is given by

$$JB = n(\hat{\gamma}_3^2/6 + (\hat{\gamma}_4)^2/24)$$

which increases when $\hat{\gamma}_3$ and $\hat{\gamma}_4$ deviate from 0.

A large-sample approximation is used to calculate the p-value and under H_0, JB converges to χ_2^2. In `R` the Jarque-Bera test can be applied via the function `jarque.bera.test()`.

Remark 2.7.6. One should always check whether the deviation from normality is of practical importance, certainly when the sample size is large.

Yap and Sim (2011; Journal of Statistical Computation and Simulation) compared 8 tests of normality and it was found that Shapiro-Wilk test was as powerful as its competitors for both short- and long-tailed symmetric alternatives and was the most powerful for asymmetric alternatives.

2.8 Transformations

Many statistical methods work best when data are normally distributed or at least symmetrically distributed and have constant variance. It is really important that one always checks if the assumptions that were made about the distribution of the data, are correct. To this purpose, graphical methods are often the most convenient way.

However real data often do not conform to assumptions of normality and homoscedasticity, i.e. constant variance. If this is the case, one needs to know how sensitive the estimator or test is to violations of the assumptions. A possible resource is to transform the data in order to normalize the data and/or to equalize the variance. To do so, one can perform the same mathematical operation on each piece of the original data.

Transformed data often exhibit less skewness and more constant variables compared to the original variables. For example, log stabilizes the variance of a variable whose conditional standard deviation is proportional to its conditional mean. Keep in mind that while transformations are important tools, they also alter the nature of the variables and they make the interpretation of the results more complex.

2.8.1 Geometry of transformations

Transformations can be beneficial because they stretch observations apart in some regions and push them together in other regions.

In case of right skewed data, a concave transformation will:

- stretch distances between observations at lower end of distribution.
- compress distances between observations at upper end of distribution.

The degree of stretching and compressing depends on the derivatives of the transformation function that is applied. For two values x and y close to each other, Taylor's theorem states that

$$|h(x) - h(y)| \approx h'(x)|x - y|.$$

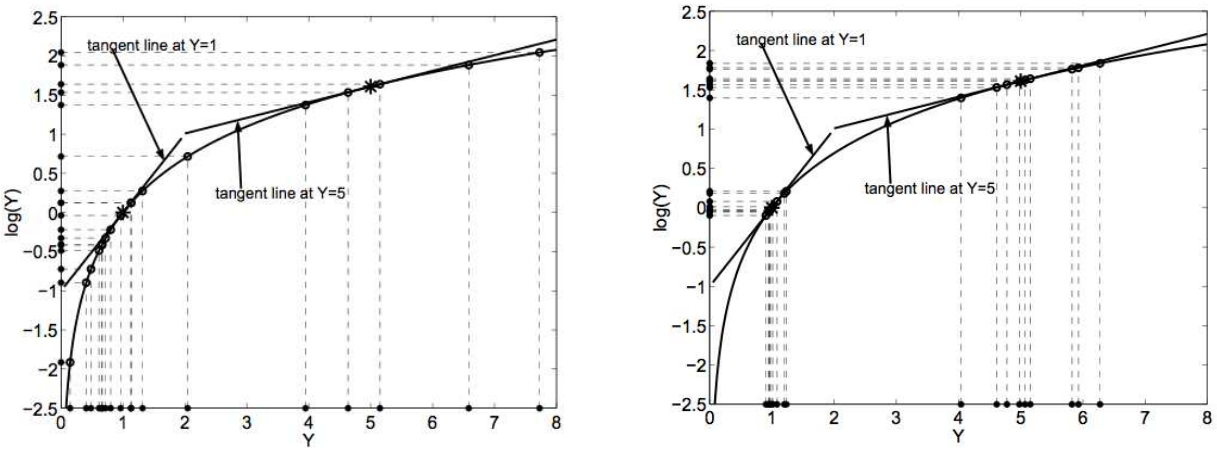
- $h(x)$ and $h(y)$ are pushed apart where $h'(x)$ is large.
- $h(x)$ and $h(y)$ are pushed together where $h'(x)$ is small.
- $h'(x)$ is a decreasing function of x if h is concave.

If the variance of data increases with its mean, a concave transformation will

- push more variable values closer together (for large values of data).
- push less variable values further apart (for small values of data).

We now look graphically at the impact of transformations in the following example.

Example 2.8.1. We perform a log transformation on two different sets of data.



(a) (b)

Figure 2.17: The impact of concave transformations.

Figure 2.17a shows skewed lognormal data on the horizontal axis which are transformed to symmetry by a log transformation. In Figure 2.17b there are 2 groups of responses, one with mean 1 and relatively small variance and one with mean 5 and relatively large variance. The variance of the second group is reduced because of the concavity of the transformation.

2.8.2 Box-Cox power transformation

There are as many potential types of data transformations as there are mathematical functions. Some of the more commonly discussed traditional transformations include: adding constants, square root, converting to logarithmic scales, inverting and reflecting, and applying trigonometric transformations such as sine wave transformations.

While many are familiar with selecting traditional transformations for improving normality, the Box-Cox transformation (Box and Cox, 1964) represents a family of power transformations that incorporates and extends the most popular options. However the Box-Cox power transformation is not a guarantee for normality, because it does not check for normality but for the smallest standard deviation. In that case the data has the highest likelihood to be normally distributed.

Definition 2.8.2. The **Box-Cox transformation** is a family of power transformations

$$y^{(\alpha)} = \begin{cases} \frac{y^\alpha - 1}{\alpha} & \alpha \neq 0 \\ \log(y) & \alpha = 0 \end{cases}$$

The subtraction of 1 from y^α and the division by α are not essential, but they make the transformation continuous in α at 0 since $\lim_{\alpha \to 0} \frac{y^\alpha - 1}{\alpha} = \log(y)$. In `R` a Box-Cox transformation can be done by using `boxcox()`.

This family of transformations incorporates many typical transformations:

α	transformation
1.00	no transformation (original data)
0.50	square root
0.33	cube root
0.25	fourth root
0.00	natural log
−0.50	reciprocal square root
−1.00	reciprocal (inverse)
2.00	square

Typically one looks for α ranging from -5 to +5 until the best value is found. If the response is right-skewed and the conditional response variance is an increasing function of the conditional response mean, then a concave transformation, i.e. Box-Cox with $\alpha < 1$, is used to remove the skewness and to stabilize the variance. If a Box-Cox transformation with $\alpha < 1$ is used, then the smaller α, the greater the effect of the transformation. Unfortunately, often the value of α that is the best for symmetrizing the data is not the same value of α that is the best for stabilizing the variance.

Example 2.8.3. We consider histograms of daily flows of natural gas in 3 different pipelines.

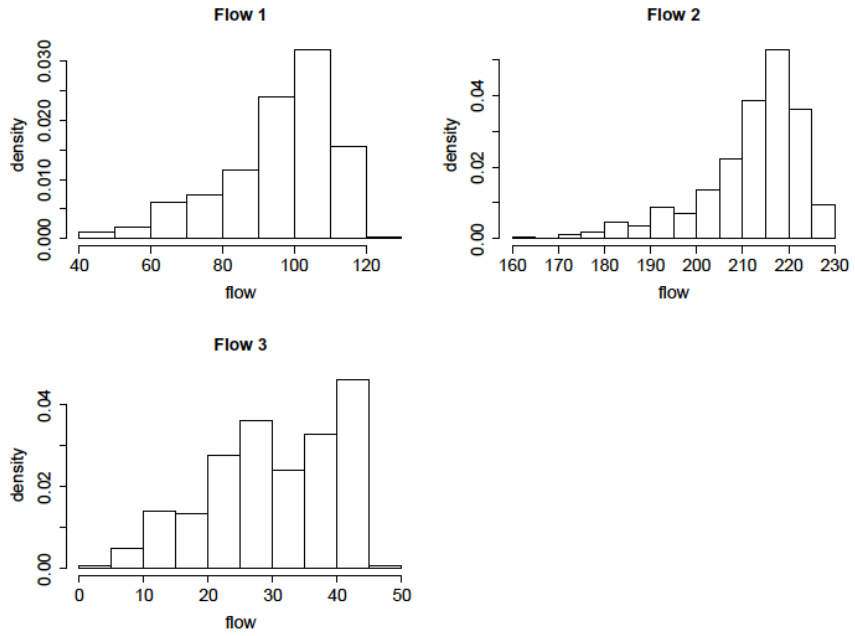


Figure 2.18: Histograms of daily flows of natural gas.

One can clearly observe that the distributions of the data are left-skewed, therefore we will use Box-Cox transformations with $\alpha > 1$. We now perform a Box-Cox transformation of the data of pipeline 1 and use $\alpha = 1, 2, 3, 4, 5$ and 6. The resulting kernel density estimates are shown in Figure 2.19.

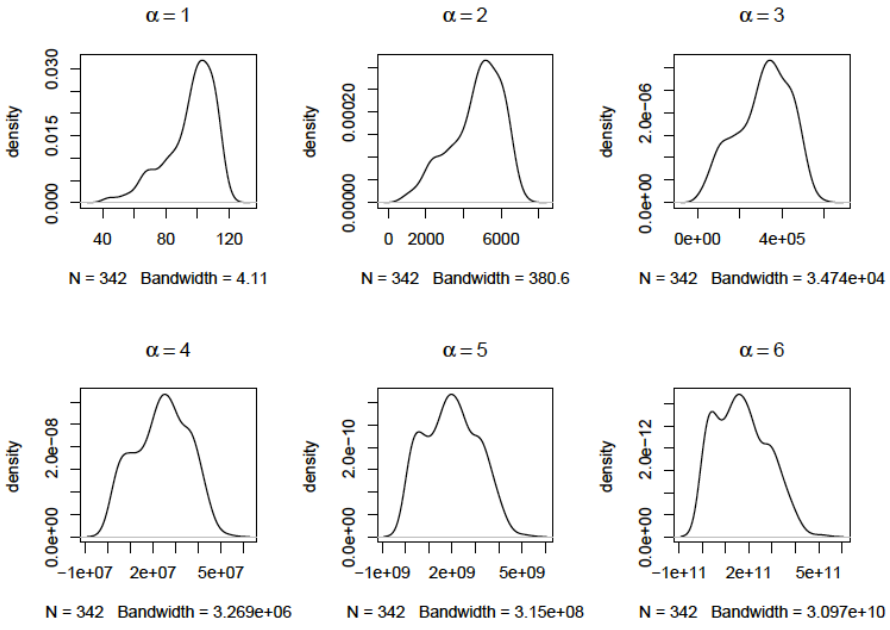
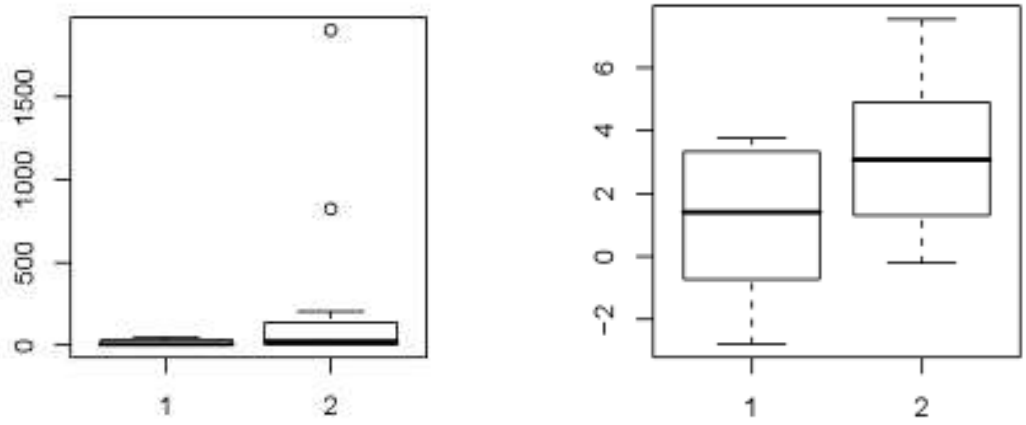


Figure 2.19: Kernel density estimates for the data of pipeline 1 after performing a Box-Cox transformation.

If we take $\alpha \in [3, 4]$, most of the left-skewness is removed. However, we need to be careful because overtransformation is also possible: for $\alpha \geq 5$ right-skewness is induced.

2.8.3 t-test and transformation

We show here the use of the t-test and the importance of transformations. Let us first generate 15 observations from the lognormal (1,4) distribution and from the lognormal (3,4) distribution. Boxplots for each population are shown in Figure 2.20a.



(a) Lognormal populations.

(b) Normal populations.

Figure 2.20: Boxplots of the two populations.

We test the null-hypothesis that the two populations have the same mean by using a t-test. The two-sided independent samples t-test has a p-value of 0.105. From this we could conclude that the means are not significantly different. However we ignored an important assumption of the t-test: the population has to be normally distributed!

We now log-transform the data, the corresponding boxplots are shown in Figure 2.20b. The transformed data now satisfy the assumptions of the t-test that the two populations are normally distributed with the same variance. Therefore, we can now test the null-hypothesis that the two populations have the same mean by using a t-test. The two-sided independent samples t-test has a p-value of 0.00467, which is an indication that the difference is highly significant.

All statistical estimators and tests make certain assumptions. One should always check if these assumptions are satisfied. Graphical methods are often the most convenient way! If assumptions are not met, one should check the sensitivity of the estimator or test to violations. If the estimator or test is non-robust, then one should look for an estimator or test that is suitable for data or one has to transform the data.