Summer school: Machine Learning and Applied Statistics
Week 2

Almut E. D. Veraart
Room 551, Huxley Building
Department of Mathematics
Imperial College London
180 Queen's Gate, London, SW7 2AZ
E-Mail: a.veraart@imperial.ac.uk

July 2022

Contents

Part I

Monday 11 July 2022

Lecture 1

Lecture 1: Introduction to time series

1.1 Overview

In the second week of the summer school, we will be focussing on **time series analysis** with a particular focus on the following questions:

- What is a time series?
- Which statistical models can be used for describing time series data?
- How can such models be estimated and the goodness of the fit be assessed?
- How can we forecast time series data?
- How can we estimate risk measures such as *Value at Risk (VaR)* and *Expected shortfall (ES)* for (financial) time series?

Throughout the week, we will be looking at various case studies with a particular focus on financial time series.

These lecture notes will be self-contained, but for students who wish to read more, complementary reading material can be found in the following excellent textbooks: Brockwell & Davis (2016), Embrechts et al. (1997), Tsay (2010), McNeil et al. (2005), Hyndman & Athanasopoulos (2018).

Also, all the R code presented has been developed in RStudio using R-4.1.3 (2022-03-10).

1.2 What is a time series?

Definition 1. A *time series* is a collection of observations x_t, where t denotes the time point at which the observation is recorded.

In order to illustrate this concept, we study some examples in the following.

1.2.1 Example: Exchange rate data

We will start off with a time series of exchange rates.

I have downloaded the following data from the Bank of England website[1]: Daily Euro (EUR) versus Pound Sterling (GBP) exchange reference rate data published by the European Central Bank over the time period from 04 Jan 2000 to 31 May 2022 (weekends are excluded) resulting in 5664 observations. The data is stored in the file EUR-GBP-Rates.csv.

[1] https://www.bankofengland.co.uk/boeapps/database/index.asp?Travel=NIxAZxRSx&TD=21&TM=Jun&TY=2022&into=GBP&CurrMonth=5&startDD=22&startMM=5&startYYYY=2021&From=Rates&C=C8J&G0Xtop.x=1&G0Xtop.y=1

```r
#Read in the data:
MyData <- read.csv("EUR-GBP-Rates.csv", header = TRUE, sep = ",", dec
    = ".",fileEncoding="UTF-8-BOM")

#View the data
View(MyData)

#Print the first part of the data
head(MyData)

#Print the last part of the data
tail(MyData)

#Convert the dates into the correct date format
MyDates <- as.Date(MyData$Date, "%d/%m/%Y")

#Plot the data
plot(MyDates,MyData$GBPtoEUR,type="l", xlab="Time",ylab="EUR/GBP
    exchange rate")
plot(MyDates,MyData$EURtoGBP,type="l", xlab="Time",ylab="GBP/EUR
    exchange rate")

#You can also check the length of your dataset using
length(MyData$GBPtoEUR)
```

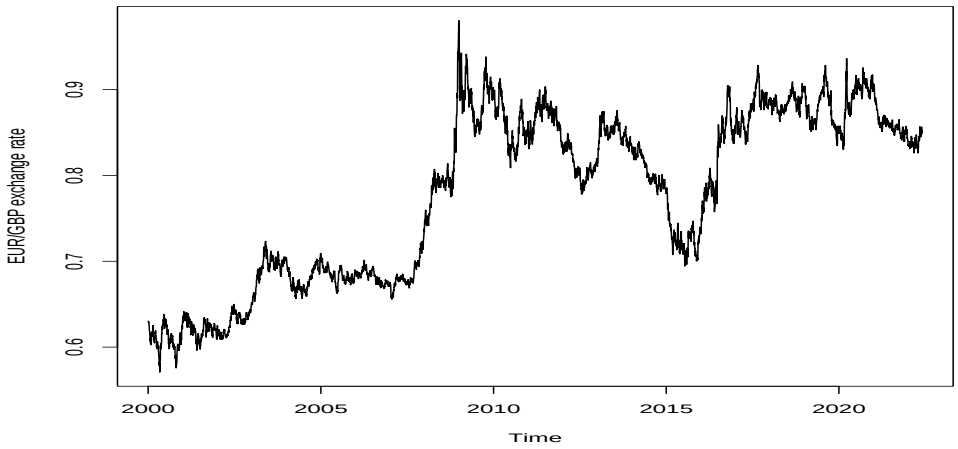


Figure 1.1: Time series of the daily Euro (EUR) versus Pound Sterling (GBP) exchange reference rate data published by the European Central Bank over the time period from 04 Jan 2000 to 31 May 2022 (without weekends). The exchange rate tells you how many pounds you need to buy/sell 1 Euro.

Exercise 2. *Describe the time series. Can you think of any events which might have influenced the exchange rate?*

1.2.2 Example: UK population data

Next we study the yearly UK population from 1971 to 2020 (i.e. 50 observations), which have been downloaded from the Office for National Statistics[2]. The data are the mid-year population estimates related to the population which usually resides in the UK – including long-term migrants, but excluding short-term migrants.

```
#Read in the data (ignoring the first seven rows which include the
    data description)
MyData <- read.csv("UK-Population.csv", header = TRUE, sep = ",", dec=
    ".", skip=7)

#View the data
View(MyData)

#Print the first part of the data
head(MyData)

#Print the last part of the data
tail(MyData)

#Briefly check the structure of the data
str(MyData) #outcome: a data.frame containing integers

#Convert the data into a time series using function ts
population <- ts(MyData[,2],start=1971, end=2020, frequency =1)

#Plot the data
plot(population, type = "p", pch=19)
```

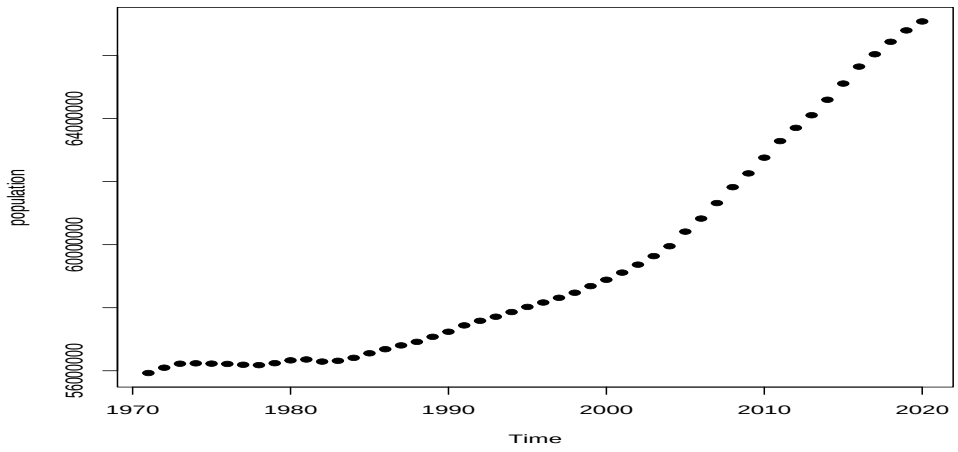


Figure 1.2: Yearly UK population from 1971 to 2020.

Exercise 3. *Describe the time series. Can you spot any* trend *in the data?*

[2] https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/timeseries/ukpop/pop

1.2.3 Example: French electricity consumption data

Next we study the weekly electricity consumption data (in MW) from 1996 to 2009 (731 observations), which are available in the R package `opera`. The data set was provided by EDF R&D. It contains weekly measurements of the total electricity consumption in France from 1996 to 2009, together with several covariates, including temperature, industrial production indices (source: INSEE) and calendar information.

```
#Install the package opera (this only needs to be done once!)
#install.packages("opera")
#Load the package opera
library(opera)

#Load the data set
attach(electric_load)

#Convert the data into a time series using the function ts
LoadTS <-ts(Load,start=1996, frequency =52)

#Check the length of the time series
length(LoadTS)

#Plot the data
plot(LoadTS,type ="l", ylab="Weekly load in MW")

detach(electric_load)
```

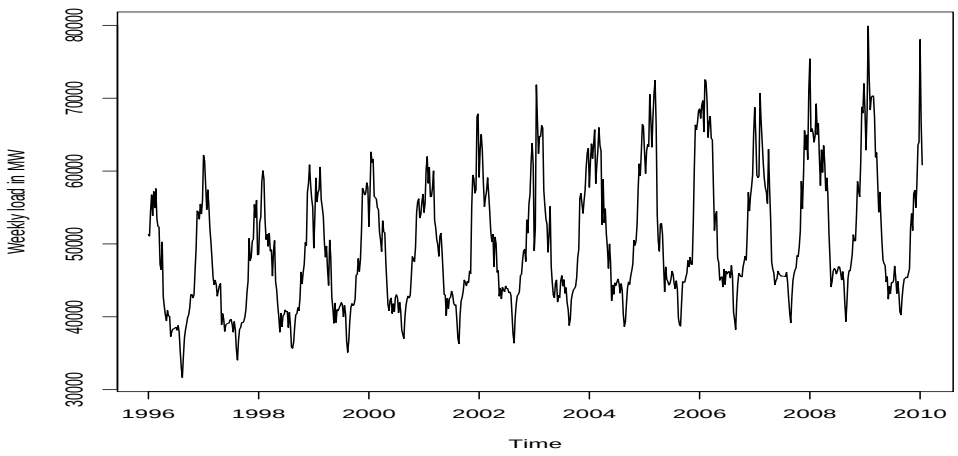


Figure 1.3: Weekly French electricity consumption data (in MW) from 1996 to 2009.

Exercise 4. *Describe the time series. Can you spot any* seasonality *in the data?*

1.2.4 Example: Daily bitcoin prices

Next we study the daily bitcoin closing prices from 17/09/2014 to 22/06/2022 (2836 observations). They can be downloaded from Yahoo finance[3].

[3] https://finance.yahoo.com/quote/BTC-USD/history?period1=1410912000&period2=1655856000&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true

```r
#Read in the data:
MyData <- read.csv("BTC-USD.csv", header = TRUE, sep = ",", dec = ".")

#View the data
View(MyData)

#Print the first part of the data
head(MyData)

#Print the last part of the data
tail(MyData)

#Convert the dates into the correct date format
MyDates <- as.Date(MyData$Date, "%d/%m/%Y")

#Plot the data
plot(MyDates,MyData$Close,type="l", xlab="Time",ylab="Bitcoin price is
    USD")
```

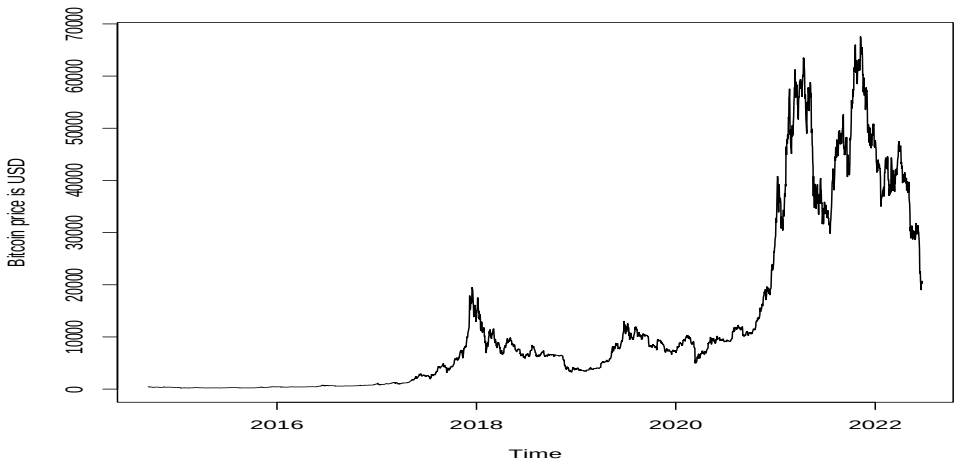


Figure 1.4: Daily bitcoin closing prices from 17/09/2014 to 22/06/2022.

Exercise 5. *Describe the time series. Do you think it is likely that such a time series can be described by a* stationary stochastic process*?*

Lecture 2

Lecture 2: Introduction to probability

Now we want to formalise the concept of a time series and explain that a time series can be viewed as a realisation of a stochastic process. For this we need to recall the concept of a *random variable* and define what we mean by a *stochastic process*. We want to define the ingredients of a probability model.

2.1 Sample space

We call Ω the *sample space*, i.e. the set of all possible outcomes of an experiment. The elements of Ω are typically denoted by ω and called *sample points*. Let's recap some concepts from set theory:

- Subsets of Ω are collections of elements of Ω and called *events*. Notation: A is a subset of Ω can be written as $A \subseteq \Omega$ meaning that every element of A is also an element of Ω.
- We write $\omega \in A$ if the element ω is a member of A and $\omega \notin A$ if the element ω is not a member of A.
- We denote the empty set by $\emptyset$. Note that the empty set contains no points, i.e. $\omega \notin \emptyset$ for all $\omega \in \Omega$.
- Every subset A of the sample space Ω satisfies $\emptyset \subseteq A \subseteq \Omega$.

Definition 6. $\mathcal{F}$ is a *set of events* which we are allowed to consider (technically: a σ-algebra),[1]

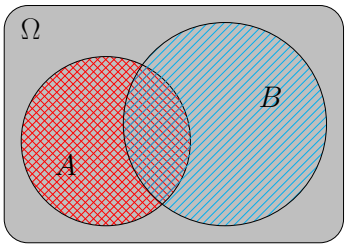
Suppose that $A, B \subset \Omega$ are events (i.e. $A, B \in \mathcal{F}$), then

- the union $A \cup B = \{\omega \in \Omega : \omega \in A \text{ or } \omega \in B\}$ is the event that at least one of A and B occurs (this is the *inclusive* "or"),
- the intersection $A \cap B = \{\omega \in \Omega : \omega \in A \text{ and } \omega \in B\}$ is the event that both A and B occur,
- the complement $A^c = \{\omega \in \Omega : \omega \notin A\}$ is the event that occurs if and only if A does not occur.

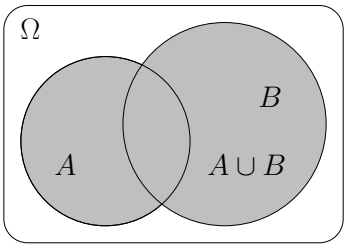
2.1.1 Venn diagrams

We typically use so-called *Venn diagrams* to illustrate concepts from set theory such as the union, intersection and complement of sets introduced above. Consider a sample space Ω with subsets $A, B \subseteq \Omega$.

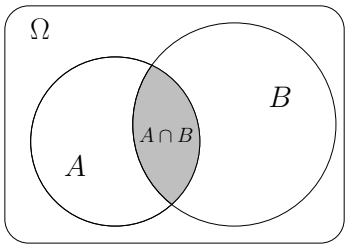
[1] A nonempty collection of subsets of Ω denoted by $\mathcal{F}$ is called an *σ-algebra* if $\mathcal{F}$ is closed under complements and countable unions. I.e. $E \in \mathcal{F} \Rightarrow E^c \in \mathcal{F}$ and $E_1, E_2, \cdots \in \mathcal{F} \Rightarrow \cup_{i=1}^{\infty} E_i \in \mathcal{F}$.



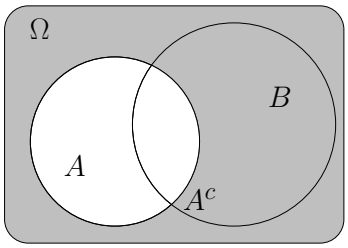
(a) $A, B \subseteq \Omega$



(b) $A \cup B$



(c) $A \cap B$



(d) A^c

Figure 2.1: We consider a sample space Ω with subsets $A, B \subseteq \Omega$ which are depicted in Figure 2.1a. The grey area in Figure 2.1b depicts the union $A \cup B$. The grey area in Figure 2.1c depicts the intersection $A \cup B$, and the grey area in Figure 2.1d depicts the complement A^c.

2.2 Definition of probabilities

Definition 7. The probability measure P is a function from $\mathcal{F}$ into the real numbers $\mathbb{R}$ which satisfies three conditions:

(i) $0 \leq \mathrm{P}(A) \leq 1$ for all events $A \in \mathcal{F}$,

(ii) $\mathrm{P}(\Omega) = 1$ and $\mathrm{P}(\emptyset) = 0$ (where $\emptyset$ denotes the empty set),

(iii) For any sequence of disjoint events $A_1, A_2, A_3, \cdots \in \mathcal{F}$ we have

$$\mathrm{P}(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathrm{P}(A_i).$$

[Note that by "disjoint events" we mean that $A_i \cap A_j = \emptyset$ for all $i \neq j$.]

2.2.1 Examples

Example 8. We start with the classical example of flipping a fair coin.

Figure 2.2: Flipping a fair coin.

We write H for heads and T for tail. The sample space is given by $\Omega = \{H, T\}$. The set of events can be taken as $\mathcal{F} = \{\emptyset, \{H\}, \{T\}, \Omega\}$ which is the collection of all subsets of Ω. Since we are considering a "fair" coin, we have that $\mathrm{P}(\{H\}) = \mathrm{P}(\{T\}) = \frac{1}{2}$, where we typically shorten the notation to:

$$\mathrm{P}(H) = \mathrm{P}(T) = \frac{1}{2}.$$

Moreover, we have $\mathrm{P}(\emptyset) = 0$ and $\mathrm{P}(\Omega) = 1$.

2.3 Random variables

Definition 9. Let Ω be a sample space. A *random variable* (r.v.) is a function from Ω into the real numbers $\mathbb{R}$, i.e. $X : \Omega \to \mathbb{R}$.

Note that

- Despite the name, a random variable is a *function* and not a variable.
- We typically use capital letters such as X, Y, Z to denote random variables.
- The value of the random variable X at the sample point ω is given by $X(\omega)$ and is called a *realisation* of X.
- The randomness stems from $\omega \in \Omega$ (we don't know which outcome ω appears in the random experiment), the mapping itself given by X is deterministic.

2.3.1 Example

Example 10. Let $\Omega = [0, 1]$ and define the random variable $X : \Omega \to \mathbb{R}$ by $X(\omega) = 3\omega$.

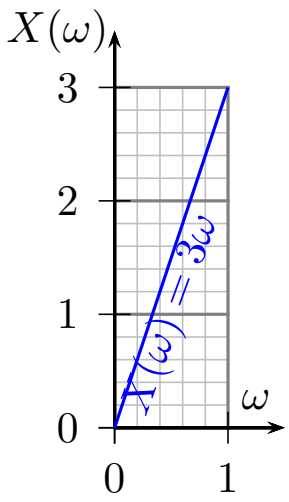


Figure 2.3: Plot of the function $X : [0,1] \to [0,3]$ given by $X(\omega) = 3\omega$.

Suppose that the probability measure P has the following property: For $0 \leq a \leq b \leq 1$, we have that $\mathrm{P}([a,b]) = b - a$, i.e. the probability of an event given by an interval is given by the *length of that interval*[2]. We want to find $\mathrm{P}(X \leq x)$ for $x \in \mathbb{R}$. Note that, for $x \in [0,3]$,

$$\begin{aligned}\{X \leq x\} &= \{\omega \in \Omega : X(\omega) \leq x\} = \{\omega \in \Omega : 3\omega \leq x\} \\ &= \{\omega \in \Omega : \omega \leq x/3\} = [0, x/3].\end{aligned}$$

Then

$$\mathrm{P}(X \leq x) = \begin{cases} 0, & \text{for } x < 0, \\ \frac{x}{3}, & \text{for } x \in [0,3], \\ 1, & \text{for } x > 3. \end{cases}$$

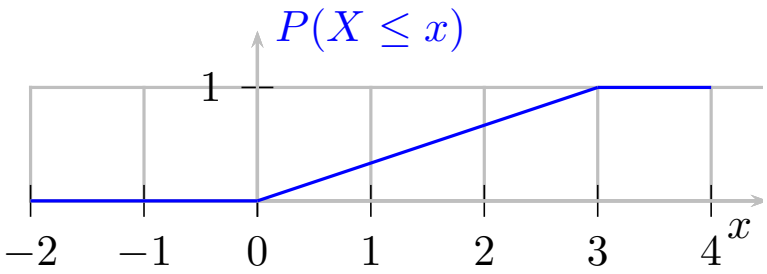


Figure 2.4: Plot of the function $\mathrm{P}(X \leq x)$. This is in fact the so-called cumulative distribution function of the random variable X, which we will study in the next chapter.

[2]Here the probability measure is in fact the so-called Lebesgue measure restricted to the interval $[0,1]$.

Lecture 3

Lecture 3: Probability distributions, expectation and variance

3.1 Probability distributions

Definition 11. Let X be a random variable on a sample space Ω. The *probability distribution* of X is the collection of probabilities $\mathrm{P}(X \in B)$ for sets B of real numbers.

Definition 12. A random variable is said to be *discrete* if there exists a finite or countably infinite set $\{k_1, k_2, \dots\}$ of real numbers such that

$$\sum_i \mathrm{P}(X = k_i) = 1,$$

where the sum ranges over all points in $\{k_1, k_2, \dots\}$.

Definition 13. The probability mass function (p.m.f.) of a discrete random variable X is the function p_X defined by $p_X(k) = \mathrm{P}(X = k)$ for all possible values k of X (for which $\mathrm{P}(X = k) > 0$).

Often we drop the subscript X and write p for the p.m.f. of X.

Definition 14. Let X be a random variable. If there is a function f_X such that

$$\mathrm{P}(X \leq x) = \int_{-\infty}^{x} f_X(t)dt,$$

for all $x \in \mathbb{R}$, then f_X is called the *probability density function* (p.d.f.) of X and X is called a *continuous random variable*.

- Often we drop the subscript X and write f for the p.d.f. of X.
- Note that a probability density function must satisfy $f(t) \geq 0$ for all $t \in \mathbb{R}$ and $\int_{-\infty}^{\infty} f(t)dt = 1$.
- For a continuous random variable X,we always have that $P(X = x) = 0$ for all $x \in \mathbb{R}$ and $\mathrm{P}(a < X \leq b) = \mathrm{P}(a < X < b) = \int_a^b f(t)dt$.

We note that p.m.f.s are only defined for discrete random variables and p.d.f.s only for continuous random variables. However, the so-called cumulative distribution function can be used to describe the distribution of *any* random variable.

Definition 15. Let X be a random variable on the sample space Ω. Its *cumulative distribution function* (c.d.f.) is defined as

$$F_X(x) = \mathrm{P}(X \leq x), \quad \text{for all } x \in \mathbb{R}.$$

Often we drop the subscript X and write F for the c.d.f. of X. Note that

$$\mathrm{P}(a < X \leq b) = \mathrm{P}(X \leq b) - \mathrm{P}(X \leq a) = F_X(b) - F_X(a).$$

If X is a discrete random variable, then

$$F(x) = \mathrm{P}(X \leq x) = \sum_{k:k\leq x} \mathrm{P}(X = k).$$

If X is a continuous random variable, then

$$F(x) = \mathrm{P}(X \leq x) = \int_{-\infty}^{x} f(t)dt.$$

3.2 Expectation of random variables

In applications, we typically study the mean, variance and possibly higher moments of random variables. So let us briefly recall how such objects can be defined.

Definition 16. For a discrete random variable X, the *expectation* or *mean* is given by

$$\mathrm{E}(X) = \sum_{k} k\mathrm{P}(X = k),$$

where we sum over all possible values k of X. For a continuous random variable, the *expectation/mean* is given by

$$\mathrm{E}(X) = \int_{-\infty}^{\infty} x f_X(x)dx.$$

An important consequence of the above definition is the following fact.

Suppose that X is a r.v.s and let g be a real-valued function defined on the range of X. If X is discrete, then

$$\mathrm{E}(g(X)) = \sum_{k} g(k)\mathrm{P}(X = k),$$

whereas if X is continuous, then

$$\mathrm{E}(g(X)) = \int_{-\infty}^{\infty} g(x) f_X(x)dx.$$

Example 17. Choose $g(x) = x^n$ for $n \in \mathbb{N}$. Then, we obtain for discrete X:

$$\mathrm{E}(X^n) = \sum_{k} k^n\mathrm{P}(X = k),$$

and for continuous X:

$$\mathrm{E}(X^n) = \int_{-\infty}^{\infty} x^n f_X(x)dx,$$

which we call the nth moment of X (which might not always exist!).

Definition 18. The variance of a random variable X with mean $\mu = \mathrm{E}(X)$ is defined as

$$\mathrm{Var}(X) = \mathrm{E}[(X - \mu)^2].$$

Often we write $\sigma^2 = \mathrm{Var}(X)$.

We note that

$$\mathrm{Var}(X) = \mathrm{E}(X^2) - (\mathrm{E}(X))^2.$$

We have the following important properties: For a random variable X with finite mean and variance and real numbers a, b, we have

$$\mathrm{E}(aX + b) = a\mathrm{E}(X) + b, \qquad \mathrm{Var}(aX + b) = a^2\mathrm{Var}(X).$$

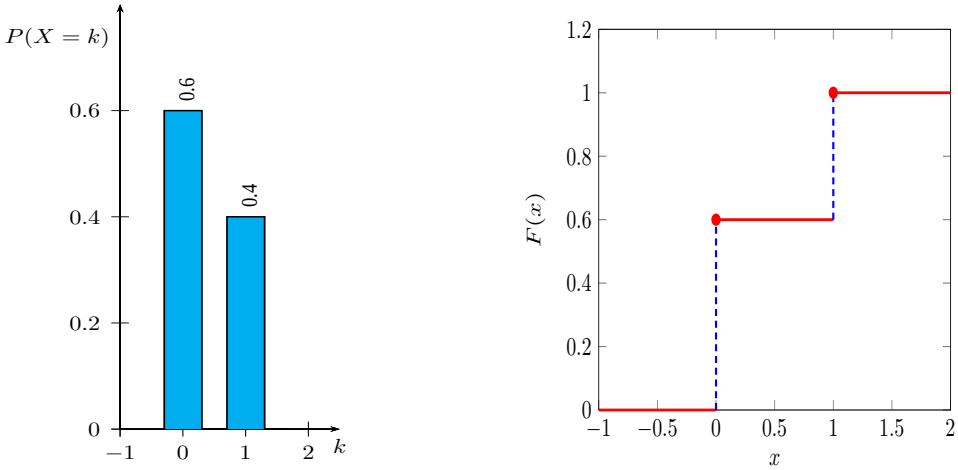
3.3 Example

Example 19. A so-called *Bernoulli* random variable records the outcome of a single experiment with two possible outcomes which are denoted by $\{0, 1\}$. We call $p \in [0, 1]$ the *success probability*. Then

$$\mathrm{P}(X = 1) = p, \qquad \mathrm{P}(X = 0) = 1 - p.$$

We typically write $X \sim \mathrm{Ber}(p)$ for a Bernoulli random variable with parameter p. We can now derive the c.d.f. of a Bernoulli random variable and obtain

$$F(x) = \mathrm{P}(X \leq x) = \sum_{k \leq x} \mathrm{P}(X = k) = \begin{cases} 0, & \text{for } x < 0, \\ 1 - p, & \text{for } x \in [0, 1), \\ 1, & \text{for } x \geq 1. \end{cases}$$



(a) P.m.f. of $X \sim \mathrm{Ber}(0.4)$

(b) C.d.f. of $X \sim \mathrm{Ber}(0.4)$

Figure 3.1: Consider a Bernoulli random variable X with parameter $p = 0.4$. Its probability mass function is depicted in Figure 3.1a and its cumulative distribution function in Figure 3.1b.

We can now compute the mean and the variance as follows:

$$\mathrm{E}(X) = \sum_k k\mathrm{P}(X = k) = 0\mathrm{P}(X = 0) + 1\mathrm{P}(X = 1) = \mathrm{P}(X = 1) = p.$$

The second moment is given by

$$\mathrm{E}(X^2) = \sum_k k^2\mathrm{P}(X = k) = 0^2\mathrm{P}(X = 0) + 1^2\mathrm{P}(X = 1) = \mathrm{P}(X = 1) = p.$$

Hence

$$\mathrm{Var}(X) = \mathrm{E}(X^2) - (\mathrm{E}(X))^2 = p - p^2 = p(1 - p).$$

Lecture 4

Problem class 1

4.1 Case study: Basic properties of exchange rate data

In this case study, we will be analysing some of the basic properties of the time series of the exchange rates of the EUR versus the GBP from 3 May 1999 to 30 April 2019. This time series contains the values for all weekdays in that time period and contains 5118 obersvations. For now, we will ignore the fact that weekends are excluded, and just view the data as a daily time series of weekday data.

During the problem class, please make sure that you are working through the assignment by yourself or in small groups. Please answer the multiple choice questions provided.

4.1.1 Visualisation [Expected time: 45min]

1. Open RStudio and start a new R script and save it. (”File – New File – R Script”)

2. Save the data file ”EUR-GBP-Rates.csv” in the same folder as your R Script and, in RStudio, click on ”Session – Set Working Directory – To Source File Location”. Now you are ready to go!

3. Read in the data using the code from the lecture notes:

```
#Read in the data:
MyData <- read.csv("EUR-GBP-Rates.csv", header = TRUE, sep = ",",
    dec = ".",fileEncoding="UTF-8-BOM")
```

4. Question: How can you get more information on the use of the function `read.csv`? a) By typing `?read.csv` into the R Console in RStudio. b) By typing `read.csv` into Google. c) All of the above

5. Question: Which of the following statements is correct. You can view the dataset by a) running the code `view(MyData)`, b) running the code `View(MyData)`, c) All of the above.

6. Question: How can you determine the number of observations in your dataset? By a) running the code `length(MyData)`, b) running the code `ncol(MyData)`, c) running the code `nrow(MyData)`, d) None of the above. [Please consult the manuals of these functions when answering the question.]

7. Next we want to get a better understanding of how we can plot the time series data. For this, let us briefly check how our data are saved in R: When you run the code

```
str(MyData)
```

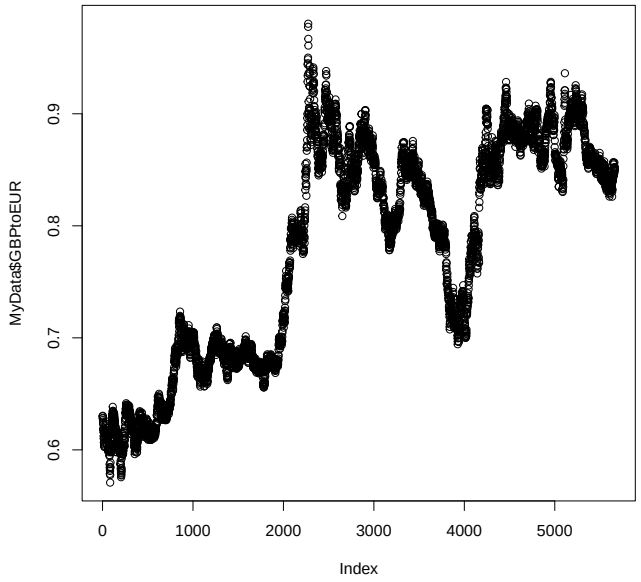
then the `str` function will display the internal structure of an R object. In our case, it will tell us that `MyData` is a "data.frame" consisting of 5664 obervations of three variables. The first variable is called "Date" and its values are of type "chr" (character), the second variable is called "EURtoGBP" and is of type "num" (numeric), and the third variable is called "GBPtoEUR" and is of type "num" (numeric).

8. Question: If you want to print all the values of the GBPtoEUR time series you can use a) the code `MyData[,3]`, b) the code `MyData$GBPtoEUR`, c) All of the above.

9. The code

```
plot(MyData$GBPtoEUR)
```

produces the following picture:



By adding more arguments to the plotting function, make the following changes to the plot:

- Change the line type from points to a solid line (Hint: `type`).
- Change the colour of the graph from black to blue (Hint: `col`).
- Add the title "EUR/GBP exchange rate" to the plot (Hint: `main`).
- Add the title "Time" to the x axis (Hint: `xlab`).
- Add the title "GBP per EUR" to the y axis (Hint: `ylab`).
- Increase the font size of the plot title from the default size 1 to size 1.25 (Hint: `cex.main`).
- Increase the font size of the labels of the axes from the default size 1 to size 1.5 (Hint: `cex.lab`).
- Increase the font size of the tick labels of the axes from the default size 1 to size 2 (Hint: `cex.axis`).
- Plot the values of the time series against their true dates
  (a) Hint 1: `MyData$Date`.

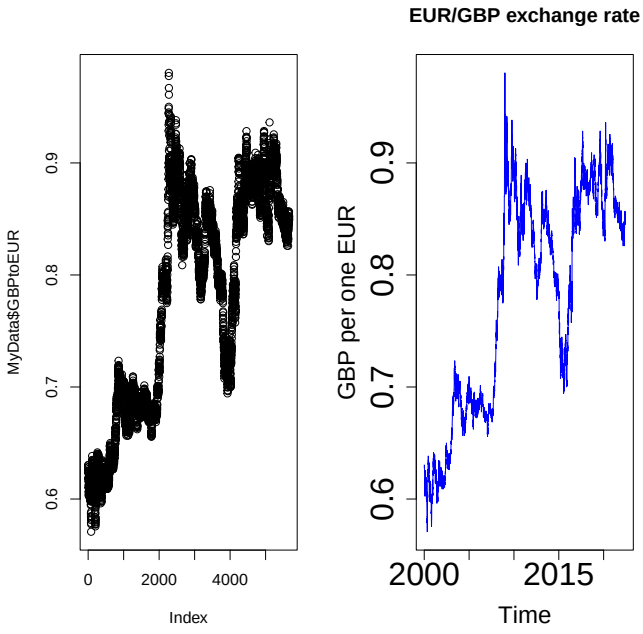
(b) Hint 2: If you obtain an error message, then recall that the dates are currently stored as "charactors", so you will first need to convert them to true dates using the function `as.Date`.

(c) Hint 3: Convert the dates into the correct date format:
`MyDates <- as.Date(MyData$Date, "%d/%m/%Y")`

- You can now compare your initial plot and your modified plot side by side using the function `par`: **[PLEASE ONLY READ THE FOLLOWING CODE AFTER COMPLETING THE ABOVE EXERCISES!]**

```
par(mfrow=c(1,2))
plot(MyData$GBPtoEUR)
plot(MyDates, MyData$GBPtoEUR, type="l",col="blue", main="EUR/GBP
    exchange rate", xlab="Time", ylab="GBP per one EUR", cex.main
    =1.25, cex.lab=1.5, cex.axis=2)
par(mfrow=c(1,1))
```

which produces the following picture:



Note that by running the code `par(mfrow=c(1,1))` you re-set the plotting environment to ensure that you only plot one picture at a time in your R panel rather than two pictures side by side.

4.1.2 Sample moments and empirical distribution [Expected time: 45min]

Next we want to study some summary statistics of the data, in particular their *empirical moments*, and have a first look at their *empirical distribution*. Note that our time series consists of $n = 5118$ observations denoted by $x_1, \ldots, x_n$.

1. **Sample mean:** First we want to compute the (sample) mean. Recall that we have defined the mean (or expectation) as follows: For a random variable X, the *expectation* or *mean* is given by

$$\mathrm{E}(X) = \begin{cases} \sum_k k\mathrm{P}(X = k) & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} x f_X(x) dx & \text{if } X \text{ is continuous.} \end{cases}$$

Note that the mean $\mathrm{E}(X)$ as defined above cannot be directly computed from the data, but rather needs to be *estimated* (or approximated). Typically we think of the expectation/mean as the average value and hence approximate it by the so-called *sample mean*. The sample mean is given by

$$\overline{x} = \frac{1}{n}\sum_{t=1}^{n} x_t.$$

Note that under certain conditions the sample mean will get closer and closer to the true mean $\mathrm{E}(X)$ as we get more and more observations. [We say that (under certain conditions) the sample mean converges to $\mathrm{E}(X)$ in probability as $n \to \infty$.] For our time series we have that $\overline{x} = 0.7594$. You can compute the sample mean directly using the following R code. Please try it out!

```
#You can use the mean function
mean(MyData$GBPtoEUR)

#Or you can compute it yourself using a for loop:
n <- length(MyData$GBPtoEUR) # determine the length of the time series
tmp <- 0 # initialise a temporary variable
for(i in 1:n){
  tmp <- tmp + MyData$GBPtoEUR[i] #Add all the elements of the vector containing the time series
}
mean <- tmp/n #Divide the sum of all entries of the data vector by the number of observations
mean # print the mean
```

2. **Sample variance:** Recall the definition of the variance: The variance of a random variable X with mean $\mu = \mathrm{E}(X)$ is defined as

$$\mathrm{Var}(X) = \mathrm{E}[(X-\mu)^2].$$

We define the *sample variance* of our time series as

$$\hat{\gamma}(0) := \frac{1}{n}\sum_{t=1}^{n}(x_t - \overline{x})^2,$$

which can be computed using
`acf(MyData$Value, type ="covariance", plot="FALSE")$acf[1]`. We find that the sample variance is equal to 0.0097. Note that we will revisit the use of the `acf` function in the next lecture.

Alternatively you can write your own code for computing the sample variance:

```
#Compute the mean first (as above)
n <- length(MyData$GBPtoEUR) # determine the length of the time series
tmp <- 0 # initialise a temporary variable
for(i in 1:n){
  tmp <- tmp + MyData$GBPtoEUR[i] #Add all the elements of the vector containing the time series
}
mean <- tmp/n #Divide the sum of all entries of the data vector by the number of observations
mean # print the mean
```

```
9
10 #Compute the sample variance
11 tmp <- 0 # Set the temporary variable back to 0
12 for(i in 1:n){
13   tmp <- tmp + (MyData$GBPtoEUR[i]-mean)^2 #Add all the elements
       of the vector of time series
14 }
15 variance <- tmp/n #Divide the sum of all entries of the data
     vector by the number of observations
16 variance
```

Note that the R function `var` implements the sample variance with a different scaling factor $\frac{1}{n-1}\sum_{t=1}^{n}(x_t-\overline{x})^2$. Hence if you want to use this function to compute the sample variance as we have defined it, then you need to use `var(MyData$GBPtoEUR)*(n-1)/n` .

3. **Empirical distribution:** Finally we want to study the (empirical) distribution of our observations. Recall that the cumulative distribution function (c.d.f.) of a random variable X is defined as $F(x) = \mathrm{P}(X \leq x)$. The distribution function can be approximated by its empirical counterpart called the *empirical cumulative distribution function* (e.c.d.f.). The e.c.d.f. of observations $x_1, \ldots, x_n$ evaluated at x is defined as the number of x_i which are smaller or equal to x divided by n, more formally

$$F_n(x) = \frac{1}{n}\#\{x_i : x_i \leq x\},$$

where for a set A, $\#A$ stands for "number of elements in set A". Plot the empirical distribution function for our data using `plot.ecdf(MyData$GBPtoEUR)`. When looking at the definition of the c.d.f. and the e.c.d.f which of the following properties are true/false

(a) The c.d.f. and e.c.d.f. can only take values in the interval $[0, 1]$. a) True b) False

(b) The c.d.f. and e.c.d.f. are functions which are nondecreasing in the sense that for $x_1 < x_2$ $F(x_1) \leq F(x_2)$ and $F_n(x_1) \leq F_n(x_2)$. a) True b) False

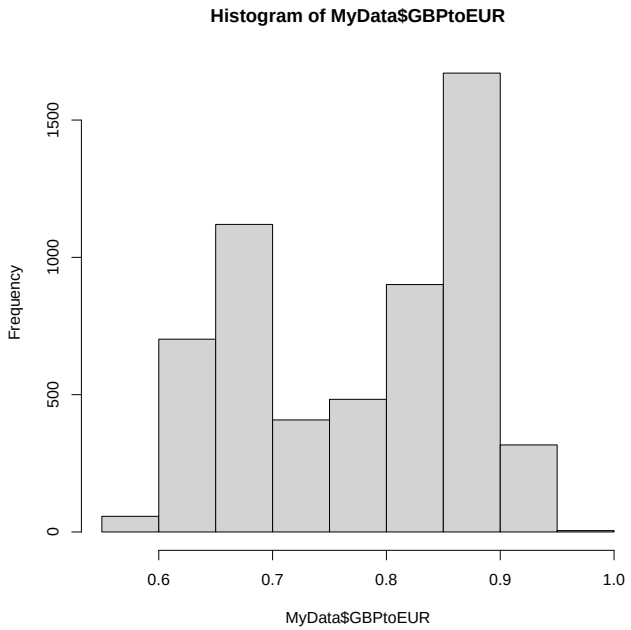
(c) The c.d.f and e.c.d.f. satisfy $\lim_{x\to-\infty} F(x) = 0, \lim_{x\to\infty} F(x) = 1, \lim_{x\to-\infty} F_n(x) = 0, \lim_{x\to\infty} F_n(x) = 1$ a) True b) False

4.1.3 Bonus material: Histograms and (empirical) quantiles

If you have already completed the material above, please continue here. This material will be revisited in a later lecture.

1. **Histograms and empirical density:** We want to get a better understanding of the empirical distribution of our time series. A widely used tool is the so-called *histogram*.

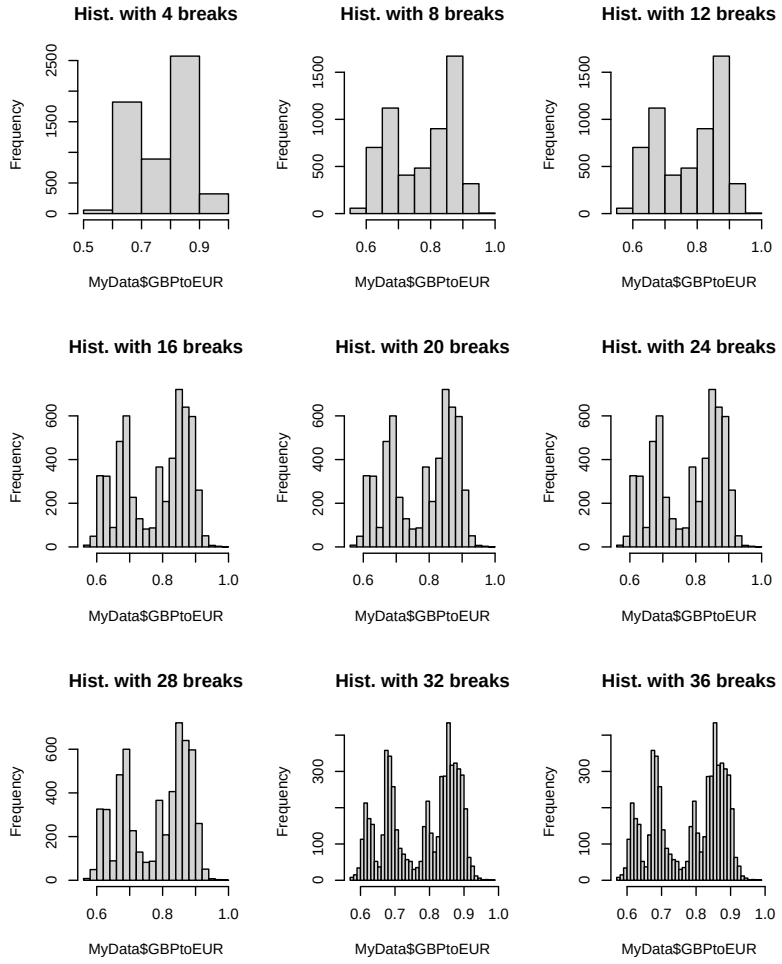
(a) Use the R function `hist` to plot a histogram.



(b) For constructing a histogram, you first need to divide the range of the data into suitable "bins" and then count how many data points fall into each point. Look at the manual page `?hist` to learn about some of the default settings of the R function `hist`.

(c) Now play around with some of the settings in the `hist` function, e.g. run the following code

```
hist(MyData$GBPtoEUR,breaks=4)
hist(MyData$GBPtoEUR,breaks=8)
hist(MyData$GBPtoEUR,breaks=12)
hist(MyData$GBPtoEUR,breaks=16)
hist(MyData$GBPtoEUR,breaks=20)
hist(MyData$GBPtoEUR,breaks=24)
hist(MyData$GBPtoEUR,breaks=28)
hist(MyData$GBPtoEUR,breaks=32)
hist(MyData$GBPtoEUR,breaks=36)
par(mfrow=c(1,1))
```



Does the shape of the histogram change with the number of break points? In particular, look at the number of *modes* (i.e. local maxima) in the histogram and study how this number changes with the number of break points.

2. **(Empirical) quantiles:** Run the code `summary(MyData$GBPtoEUR)` and study its output. It returns six numerical values: the estimates of the minimum, the first quartile, the median, the third quartile and the maximum of the observations. Do you know what these quantities mean? Let's study them in more detail.

   (a) The summary function reports both $\min\{x_1, \ldots, x_n\}$ and $\max\{x_1, \ldots, x_n\}$.

   (b) The median and quantiles in general are widely used for describing distributions:

   Definition 20. The *median* of a random variable X is any real value m that satisfies both

   $$\mathrm{P}(X \geq m) \geq \frac{1}{2} \text{ and } \mathrm{P}(X \leq m) \geq \frac{1}{2}.$$

   Since at least half the probability is on $\{X \geq m\}$ and $\{X \leq m\}$, you can view the median as the *midpoint* of the distribution. For our data, the median is estimated to be 0.7719 which is slightly smaller than the mean. Note that you can compute the median directly by using `median(MyData$GBPtoEUR)`.

   We can generalise the definition of the median using other fractions than 1/2:

Definition 21. Let $0 < p < 1$, then the *pth quantile* of a random variable X is defined as any real value x satisfying both

$$\mathrm{P}(X \geq x) \geq 1 - p \text{ and } \mathrm{P}(X \leq x) \geq p.$$

The summary statistics reports the 1st and 3rd empirical quartile, which correspond to the 0.25th and 0.75th empirical quantile respectively. You can run the code `?quantile` to read how the empirical/sample quantiles are computed in R.

Part II

Tuesday 12 July 2022

Lecture 5

Lecture 4: Stochastic models for time series

5.1 Formal definition of a stochastic process and a time series

Recall Definition 1, which states that a *time series* is a collection of observations x_t, where t denotes the time point at which the observation is recorded. Typically we consider an index set T_0 which collects all the time points when observations are available. For instance, we often have $T_0 = \{0, 1, 2, \ldots, n\}$ for $n \in \mathbb{N}$.

Definition 22. A stochastic process $X = (X_t)_{t \in T_0}$ is a collection of random variables X_t, where t denotes the time index and T_0 the index set.

For a fixed event $\omega \in \Omega$ we obtain the *realisation of the stochastic process* (sometimes also called a *sample path*) which is given by

$$x_t = X_t(\omega), \quad t \in T_0.$$

A time series can be viewed as a realisation of a stochastic process. Note that we typically use capital letters for random variables and small letters for their realisations.

We are interested in finding a suitable *time series model* for our time series $(x_t)_{t \in T_0}$. Since many objects of interest (such as asset prices, electricity demand, population growth) exhibit some random behaviour, we are looking for a *probabilistic description* of our time series. More precisely, we are looking for a stochastic process $(X_t)_{t \in T_0}$ for which our time series $(x_t)_{t \in T_0}$ is assumed to be a realisation. For instance, in the case when $T_0 = \{1, \ldots, n\}$, one possible approach would be to specify the *joint distribution* of the random vector $(X_1, \ldots, X_n)$. So far, we have only considered one-dimensional random variables. In time series analysis it is crucial to understand the *dependence* between subsequent observations. Hence, in the following, we will briefly extend the concepts of p.m.f.s, p.d.f.s and c.d.f.s to a multivariate setting. This will enable us to define what we mean by (in)dependence of random variables.

Remark 23. *Note that in the following we will use the notation X, $(X_t)_{t \in T_0}$, (X_t) interchangeably for a stochastic process.*

5.2 Random vectors

Definition 24. Let $n \in \mathbb{N}$. An n-dimensional random vector is a column vector

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} = (X_1, X_2, \ldots, X_n)^\top,$$

where X_i is a random variable for $i = 1, \ldots, n$.

Note that the sign $^\top$ stands for the *transpose* of a vector [or a matrix] and it means that a row (column) vector is changed into a column (row) vector where the elements remain in the same order.

5.2.1 Multivariate cumulative distribution function

Definition 25. The *joint/multivariate cumulative distribution function* of the random vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)^\top$ is defined as

$$F(x_1, \ldots, x_n) = \mathrm{P}(X_1 \leq x_1, \ldots, X_n \leq x_n),$$

for all real vectors $\mathbf{x} = (x_1, \ldots, x_n)^\top$.

Note that you can find the joint distribution of a subcollection of random variables by setting the corresponding x value to ∞. More precisely, we have

$$F_{X_1}(x_1) = \mathrm{P}(X_1 \leq x_1) = F(x_1, \infty, \ldots, \infty),$$
$$F_{(X_1,X_3)}(x_1, x_3) = \mathrm{P}(X_1 \leq x_1, X_3 \leq x_3) = F(x_1, \infty, x_3, \infty, \ldots, \infty).$$

5.2.2 Multivariate probability density and probability mass function

Exactly as in the univariate setting, we say that a random vector with c.d.f. F is *continuous* if F has a density function f which satisfies

$$F(x_1, \ldots, x_n) = \int_{-\infty}^{x_n} \cdots \int_{-\infty}^{x_1} f(t_1, \ldots, t_n) dt_1 \cdots dt_n.$$

In that case, the density can be derived by differentiating F with respect to all variables (i.e. by taking n *partial* derivatives)

$$f(x_1, \ldots, x_n) = \frac{\partial^n F(x_1, \ldots, x_n)}{\partial x_1 \cdots \partial x_n}.$$

Also, we say that a random vector is *discrete* if there exist real-valued vectors $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$ and a probability mass function $p(\mathbf{x}^{(i)}) = \mathrm{P}(\mathbf{X} = \mathbf{x}^{(i)}) = \mathrm{P}(X_1 = x_1^{(i)}, \ldots, X_n = x_n^{(i)})$ such that

$$\sum_{i=0}^{\infty} p(\mathbf{x}^{(i)}) = 1.$$

5.3 Independence of random variables

Definition 26. The random variables $X_1, \ldots, X_n$ are said to be *independent* if their joint cumulative distribution function factorises as follows:

$$\mathrm{P}(X_1 \leq x_1, \ldots, X_n \leq x_n) = \mathrm{P}(X_1 \leq x_1) \cdots \mathrm{P}(X_n \leq x_n),$$

i.e.,

$$F(x_1, \ldots, x_n) = F_{X_1}(x_1) \cdots F_{X_n}(x_n),$$

for all real numbers $x_1, \ldots, x_n$.

Note that in the discrete case, independence is equivalent to the factorisation of the joint probability mass function,

$$p(x_1, \ldots, x_n) = p_{X_1}(x_1) \cdots p_{X_n}(x_n),$$

and in the continuous case to the factorisation of the joint probability density function

$$f(x_1, \ldots, x_n) = f_{X_1}(x_1) \cdots f_{X_n}(x_n).$$

5.3.1 Expectation of functions of random vectors

Let $g : \mathbb{R}^n \to \mathbb{R}$ be a deterministic function, then we define the expectation of function g of a random vector $\mathbf{X}$ as

$$\begin{aligned}\mathrm{E}(g(\mathbf{X})) &= \int g(x_1,\ldots,x_n)dF(x_1,\ldots,x_n)\\ &= \begin{cases} \int\cdots\int g(x_1,\ldots,x_n)f(x_1,\ldots,x_n)dx_1\cdots dx_n, & \text{if } \mathbf{X} \text{ is continuous,}\\ \sum_{i_1}\cdots\sum_{i_n} g(x_{i_1},\ldots,x_{i_n})p(x_{i_1},\ldots,x_{i_n}), & \text{if } \mathbf{X} \text{ is discrete,}\end{cases}\end{aligned}$$

provided these integrals/sums are finite. In particular, for two random variables X, Y, we have (when choosing $g(x,y) = xy$):

$$\mathrm{E}(XY) = \begin{cases} \int\int xyf(x,y)dxdy, & \text{if } (X,Y)^\top \text{ is continuous,}\\ \sum_i\sum_j x_iy_jp(x_i,y_j), & \text{if } (X,Y)^\top \text{ is discrete,}\end{cases}$$

Exercise 27. *Consider two* **independent** *continuous random variables X and Y defined on the same sample space. Show that* $\mathrm{E}(XY) = \mathrm{E}(X)\mathrm{E}(Y)$.

Solution: Since X, Y are continuous, there is a joint density function $f = f_{(X,Y)}$. Since X and Y are independent, we have the factorisation $f_{(X,Y)}(x,y) = f_X(x)f_Y(y)$ for all real values x and y. Hence we have

$$\begin{aligned}\mathrm{E}(XY) &= \int\int xyf(x,y)dxdy = \int\int xyf_X(x)f_Y(y)dxdy\\ &= \int xf_X(x)dx\int yf_Y(y)dy = \mathrm{E}(X)\mathrm{E}(Y).\end{aligned}$$

Lecture 6

Lecture 5: The second-order approach to time series modelling

6.1 Covariance and correlation between random variables

Definition 28. Consider two (one-dimensional) random variables X and Y on the same sample space with expectations $\mu_X = \mathrm{E}(X)$ and $\mu_Y = \mathrm{E}(Y)$. The *covariance* of X and Y is defined as

$$\mathrm{Cov}(X, Y) = \mathrm{E}[(X - \mu_X)(Y - \mu_Y)]$$

if the expectation on the right hand side takes a finite value.

When we set $X = Y$, then the covariance simplifies to the variance:

$$\mathrm{Cov}(X, X) = \mathrm{E}[(X - \mu_X)^2] = \mathrm{Var}(X).$$

Exercise 29. *Use the linearity of the expectation to derive the alternative formula for the covariance:*

$$\mathrm{Cov}(X, Y) = \mathrm{E}(XY) - \mathrm{E}(X)\mathrm{E}(Y) = \mathrm{E}(XY) - \mu_X \mu_Y. \tag{6.1.1}$$

Solution:

$$\begin{aligned}\mathrm{Cov}(X, Y) &= \mathrm{E}[(X - \mu_X)(Y - \mu_Y)] = \mathrm{E}[XY - X\mu_Y - \mu_X Y + \mu_X \mu_Y] \\ &= \mathrm{E}(XY) - \mathrm{E}(X)\mu_Y - \mu_X \mathrm{E}(Y) + \mu_X \mu_Y \\ &= \mathrm{E}(XY) - \mu_X \mu_Y - \mu_X \mu_Y + \mu_X \mu_Y = \mathrm{E}(XY) - \mu_X \mu_Y.\end{aligned}$$

It is important to note that independent random variables always have zero covariance, but the converse does not hold in general!

Consider the case of independent continuous random variables X and Y. Then from Exercise 27 we know that $\mathrm{E}(XY) = \mathrm{E}(X)\mathrm{E}(Y) = \mu_X \mu_Y$. Hence

$$\mathrm{Cov}(X, Y) = \mathrm{E}(XY) - \mathrm{E}(X)\mathrm{E}(Y) = \mu_X \mu_Y - \mu_X \mu_Y = 0.$$

The proof of the discrete case works similarly.

Definition 30. Consider two (one-dimensional) random variables X and Y on the same sample space with expectations $\mu_X = \mathrm{E}(X)$ and $\mu_Y = \mathrm{E}(Y)$. The *correlation* of X and Y is defined as

$$\mathrm{Cor}(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}},$$

if $\mathrm{Cov}(X, Y), \mathrm{Var}(X), \mathrm{Var}(Y)$ take finite values.

When we set $X = Y$, then we have

$$\mathrm{Cor}(X, X) = \frac{\mathrm{Cov}(X, X)}{\sqrt{(\mathrm{Var}(X))^2}} = \frac{\mathrm{Var}(X)}{\sqrt{(\mathrm{Var}(X))^2}} = 1.$$

One can show that the correlation between two random variables X and Y can only take values in the interval $[-1, 1]$. We call X and Y

- *uncorrelated* if $\mathrm{Cor}(X, Y) = 0$,
- *positive correlated* if $\mathrm{Cor}(X, Y) > 0$,
- *negative correlated* if $\mathrm{Cor}(X, Y) < 0$.

6.2 Stationary time series models

6.2.1 Mean, autocovariance functions and second order stationarity

We think of a stochastic process $X = (X_t)_{t=0,\pm1,\ldots}$ as being stationary if its statistical properties are the same as the ones of the "time-shifted" process $(X_{t+h})_{t=0,\pm1,\ldots}$ for any $h \in \mathbb{Z}$. Rather than considering the joint distribution of the entire stochastic process, we rather use a second order perspective in time series modelling, which means that we focus on the statistical properties related to the first and second moment of the stochastic process X.

Definition 31. Consider a stochastic process $X = (X_t)$ with $\mathrm{E}(X_t^2) < \infty$ for all t.

- The *mean function* of X is defined as

$$\mu_X(t) = \mathrm{E}(X_t).$$

- The *covariance function* of X is defined as

$$\gamma_X(s, t) = \mathrm{Cov}(X_s, X_t) = \mathrm{E}[(X_s - \mu_X(s))(X_t - \mu_X(t))], \text{ for all } s, t.$$

Moreover, X is called *(weakly) stationary*[1] if (i) $\mu_X(t)$ is independent of t, and (ii) $\gamma_X(t + h, t)$ is independent of t for each h.

We can simplify the notation in the stationary case:

Definition 32. Consider a stationary stochastic process $X = (X_t)$.

- We define the *autocovariance function* (ACVF) of X at lag h as

$$\gamma_X(h) = \mathrm{Cov}(X_{t+h}, X_t),$$

- and the *autocorrelation function* (ACF) of X at lag h as

$$\rho_X(h) = \mathrm{Cor}(X_{t+h}, X_t) = \frac{\gamma_X(h)}{\gamma_X(0)}.$$

[1] There is a stronger concept of stationarity, sometimes referred to as *strict* stationarity, which we will not study in this course in more detail. However, it is important to note that the type of stationarity we have defined is sometimes called *weak/covariance/second order stationarity* to distinguish it more clearly from strict stationarity.

6.2.2 Sample mean and sample autocorrelation function

Let us now define the empirical counterparts of the mean and the autocorrelation function (which you have seen to some extent in yesterday's problem class!).

Definition 33. Suppose we have a time series with observations denoted by $x_1, \ldots, x_n$.

- The sample mean is given by

$$\overline{x} = \frac{1}{n}\sum_{t=1}^{n} x_t.$$

- We define the *sample variance* of our time series as

$$\hat{\gamma}(0) := \frac{1}{n}\sum_{t=1}^{n}(x_t - \overline{x})^2,$$

  and, more generally, the *sample autocovariance function* as

$$\hat{\gamma}(h) := \frac{1}{n}\sum_{t=1}^{n-|h|}(x_{t+|h|} - \overline{x})(x_t - \overline{x}), \text{ for } -n < h < n.$$

- The *sample autocorrelation function* is given by

$$\hat{\rho}(h) := \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \text{ for } -n < h < n.$$

Lecture 7

Lecture 6: A first example of a time series model

Let us study a first example of a time series model. You will see that this example is rather simple and in fact often too simple for applications. However, it does constitute an important building block for constructing more sophisticated time series models as we will see later in the course.

7.1 I.i.d. noise

Suppose we have random variables $X_1, X_2, \ldots$ which are all independent of each other and have identical distribution with zero mean. We call such a collection of random variables *i.i.d. noise*, where i.i.d. stands for independent and identically distributed. Let us briefly verify that i.i.d. noise constitutes a stationary process (provided its second moment exist). So let us assume that $\mathrm{E}(X_t^2) = \sigma^2 < \infty$ for all t. Then $\mathrm{E}(X_t) = 0$ does not depend on t. Also, $\gamma_X(t,t) = \mathrm{Var}(X_t) = \sigma^2$ and, for $h \neq 0$ we have $\gamma_X(t+h,t) = \mathrm{Cov}(X_{t+h}, X_t) = 0$. Hence $\gamma_X(t+h,t)$ does not depend on t for each h. We often write $(X_t) \sim IID(0, \sigma^2)$.

Let us consider the case of $n = 100$ i.i.d. random variables $X_1, \ldots, X_n$, where each random variable follows the standard normal distribution.

Definition 34. A random variable X has the *standard normal/standard Gaussian* distribution if it has density function $f(x) = \phi(x)$ with

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad \text{for } x \in \mathbb{R}.$$

Note that we typically write $X \sim N(0,1)$ since a standard normal random variable has mean zero and variance one. The c.d.f. is then denoted by $F(x) = \Phi(x)$ with

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt, \quad \text{for } x \in \mathbb{R}.$$

Unfortunately there is no explicit formula for the integral appearing in the c.d.f.!

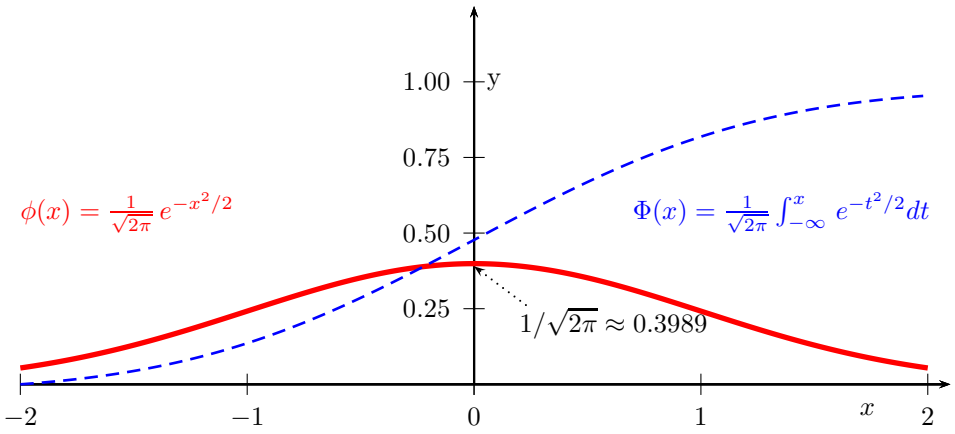


Figure 7.1: The red solid line depicts the standard Gaussian probability density function and the blue dashed line the corresponding cumulative distribution function.

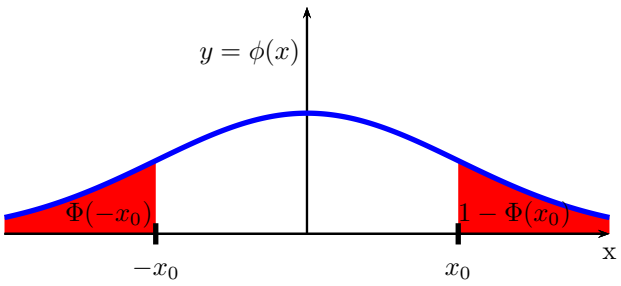


Figure 7.2: Note that the standard normal density is symmetric around 0, i.e. $\phi(x) = \phi(-x)$ for all x. This also implies that $\Phi(-x) = 1 - \Phi(x)$.

Definition 35. Let μ denote a real number and let $\sigma > 0$. A random variable X has the *normal/ Gaussian* distribution with mean μ and variance σ^2 if it has density function

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad \text{for } x \in \mathbb{R}.$$

Note that we typically write $X \sim N(\mu, \sigma^2)$.

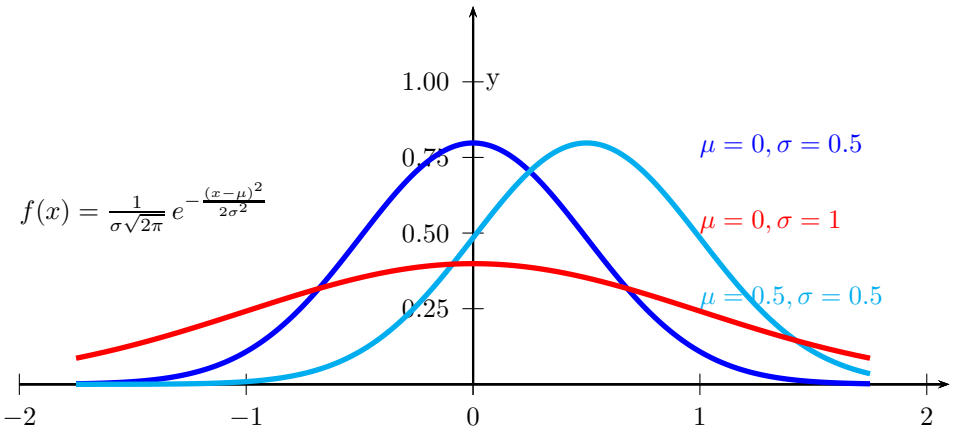


Figure 7.3: The red line depicts the standard Gaussian probability density function and the two blue lines show non-standard Gaussian probability density functions.

Example 36. Let us consider the case of $n = 100$ i.i.d. random variables $X_1, \ldots, X_n$, where each random variable follows the standard normal distribution.

We note that random variables with various distributions can be simulated in R. We will just use these tools without studying which algorithms are implemented to generated random variables.

```
#First we fix the seed of R's random number generator
#so that we can generate a simulation which can be reproduced
set.seed(1)
#Simulate 100 i.i.d N(0,4) variables
MyData <- rnorm(100,mean=0,sd=2)
```

You can compute some summary statistics as follows:

```
mean(MyData) #Compute the sample mean
var(MyData) #Compute the sample variance (Recall the different scaling
    factor!)
summary(MyData) #Compute the summary statistics of the data
```

7.1.1 Visualising the data

We will now continue with Example 36 and study how our data can be visualised. Using the code below, we generate six pictures, which we will explain in the following.

```
#Visualise the data:
par(mfrow=c(2,3))
#Plot the data
plot(MyData)
boxplot(MyData,main="Boxplot")
library(vioplot)
vioplot(MyData,col="gold", main="Violin plot")
#Plot the histogram which depicts the number of times an
#observation falls into each of the bins
hist(MyData, main="Histogram")
#The following function scales the classical histogram
#such that it approximates a probability density
```

```
13 library(MASS)
14 truehist(MyData,main="Scaled/true histogram and kernel density")
15 #Add the so-called empirical density to the plot
16 lines(density(MyData),col="red",lwd=3)
17 library(car)
18 qqPlot(MyData, main="Quantile-quantile plot")
```

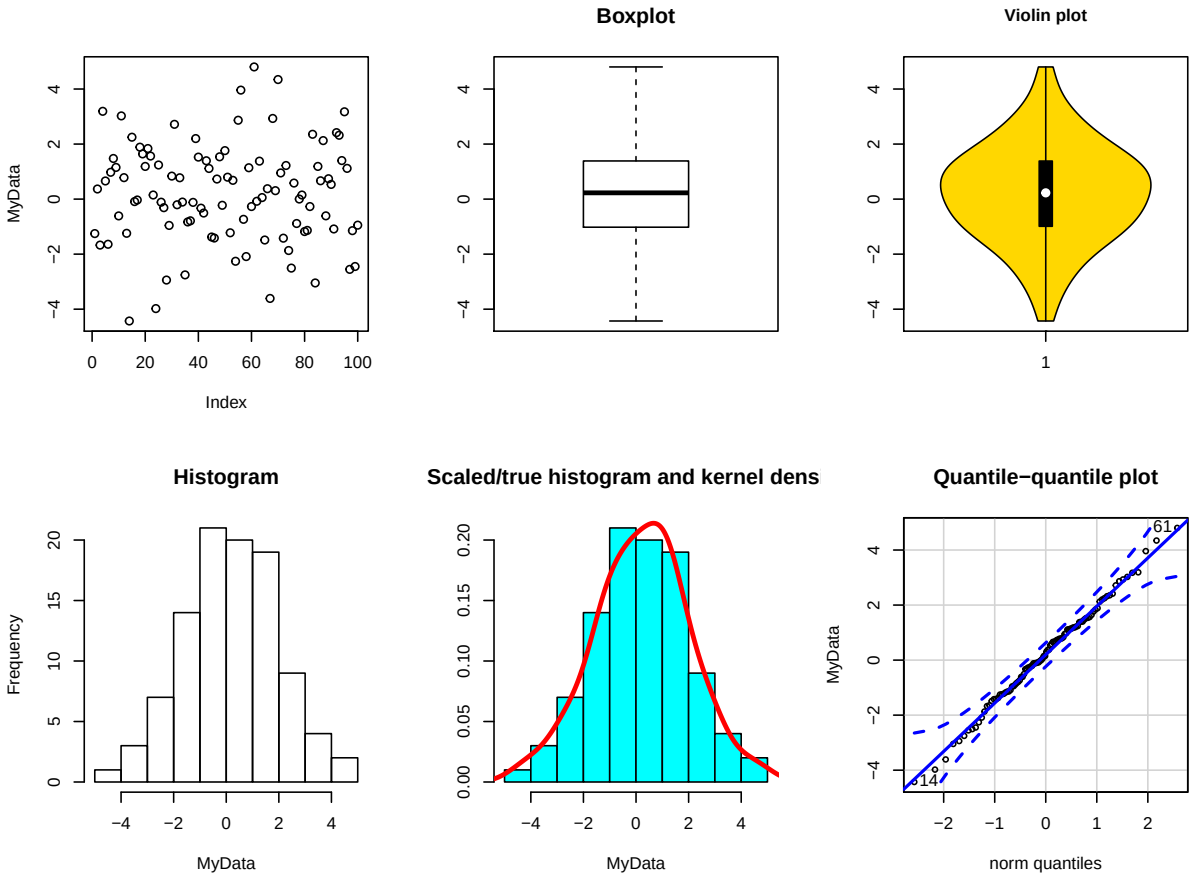


Figure 7.4: Visualisation of the data generated in Example 36.

As long as we are not dealing with too many observations, it is typically very useful to plot the data. We have already studied the basic `plot` function in R. In addition, there are various functions which can be used to give us some insights on the *distribution* of the empirical data. For instance, we often study

- box plots,
- violin plots,
- histograms and scaled histograms,
- quantile-quantile plots (short: QQ-plots).

In order to be able to interpret these plots, we need to know what *quantiles* of a distribution are.

Some of you might have read about quantiles in the bonus material from yesterday's case study. Quantiles provide you with cut points such that a certain percentage of your distribution/your observations fall below a certain value.

Definition 37. Let $0 < p < 1$, then the *pth quantile* of a random variable X is defined as any real value x satisfying both

$$\mathrm{P}(X \geq x) \geq 1 - p \text{ and } \mathrm{P}(X \leq x) \geq p.$$

Note that we call the 0.5 quantile the *median*, m say. Since at least half the probability is on $\{X \geq m\}$ and $\{X \leq m\}$, you can view the median as the *midpoint* of the distribution. Note that the theoretical quantiles are rarely available to us, so in practice we typically work with their *empirical counterparts*. Note that different computer packages typically use different methods for computing sample/empirical quantiles, so these estimates might vary depending on the software you use.

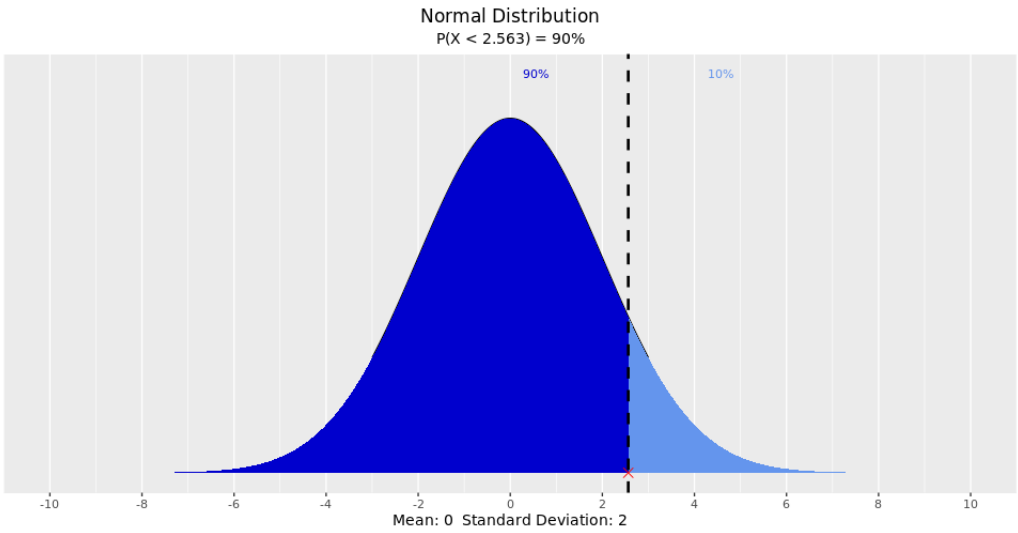


Figure 7.5: The pictures shows a plot of the probability density of the $N(0, 4)$ distribution and the red cross indicates the corresponding 90% quantile of the N(0,4) distribution.

A **box plot** graphically displays the following information of a vector of observations:

- The bounds of the "box" represent the first sample quartile (i.e. the 25% quantile, let's call it Q1) and the third sample quartile (i.e. the 75% quantile, let's call it Q3).
- The line inside the box represents the sample median (i.e. the 50% quantile).
- The interquartile range (IQR) is defined as $IQR = Q3 - Q1$.
- The "whiskers" are derived as follows: The lower whisker is given by $\max(\min(data), Q1 - 1.5IQR)$ and the upper whisker by $\min(\max(data), Q3 + 1.5IQR)$, respectively.

A **violin plot** combines a boxplot with a so-called kernel density plot which estimates the probability density of the observations.

A **histogram** depicts the number of times an observation falls into each of the bins. Sometimes we scale the histogram such that it approximates a probability density function.

A **quantile-quantile** (qq) plot plots the quantiles of two distributions against each other. Typically we plot the sample quantiles against an estimated distribution (such as the estimated normal distribution). If the two distributions are similar, we expect to see a straight line through $y = x$ in the qq plot.

We can compute and plot the sample autocorrelation function using the following code:

```
#Plot the sample autocorrelation of the data
par(mfrow=c(1,2))
acf(MyData)
#Remove the lag 0
plot(acf(MyData,plot=F)[1:20])
```

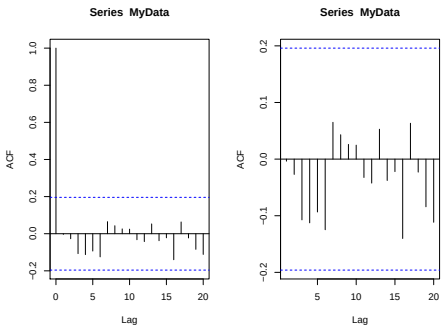


Figure 7.6: Sample autocorrelation function for our simulated data. In the second picture, we have removed the uninformative lag 0.

Not surprisingly, our sample from i.i.d. random variables does not exhibit any significant autocorrelation.

Part III

Wednesday 13 July 2022

Lecture 8

Lecture 7: General strategy for analysing time series

In this lecture we will study a general strategy to time series analysis, see (Brockwell & Davis 2016, p. 12), and will later run through the various steps in relevant case studies.

8.1 How to conduct a time series analysis - a step-by-step guide

Traditional time series analysis is typically performed using the following procedure:

Step 1: Plot the time series and analyse the main characteristics of the graph, in particular, you should look out for

- a trend,
- a seasonal component,
- any change points, where the behaviour of the time series changes abruptly,
- any extreme observations/outliers.

Step 2: Remove the trend and seasonal component and possibly apply a data transformation to produce a time series which appears to be *stationary*.

Step 3: Fit a (stationary) time series model to the (transformed) data.

Step 4: Assess the model fit and possibly compare models.

Step 5: If you want to do any forecasting, then you will need to forecast observations from your stationary time series model and then invert your procedures from Step 2 to obtain forecasts of your original time series.

Remark 38. *Note that alternative approaches to time series analysis exist, such as expressing the time series in terms of Fourier components, but we will not be able to cover them here.*

8.2 Step 1: Plotting the data

In the first lecture, we studied various examples of time series and discussed how to plot them in R.

8.2.1 Looking for breaks and outliers

When you plot the data and observe any apparent discontinuities or outliers, you might want to consider breaking the series up into homogeneous segments and analysing those segments separately. Recall out bitcoin example:

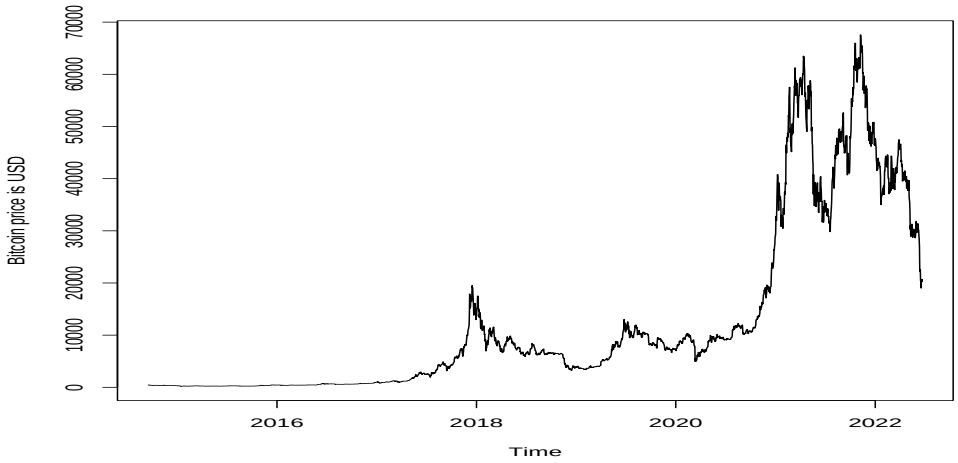


Figure 8.1: Daily bitcoin closing prices from 17/09/2014 to 22/06/2022.

You clearly observe that the behaviour of the time series changes substantially in mid 2017.

8.2.2 The classical decomposition model

When you study the graph of your time series, you might think that your data could be realisations of a stochastic process obtained from the *classical decomposition model*, which is defined as

$$X_t = m_t + s_t + Y_t,$$

where

- m_t is called the *trend component*, which is typically a slowly changing function,
- s_t is called the *seasonal component*, which is a periodic function with known period d, s(t+d)= s(t)
- Y_t is a *stochastic process* which is stationary (see Definition 31).

Remark 39. *You can use the R function* `stl` *to decompose a time series into these three components, see Figure 8.2.*

```
1 library(opera)
2 attach(electric_load)
3 LoadTS <-ts(Load,start=1996, frequency =52)
4 plot(stl(LoadTS, s.window="periodic"))
5 detach(electric_load)
```

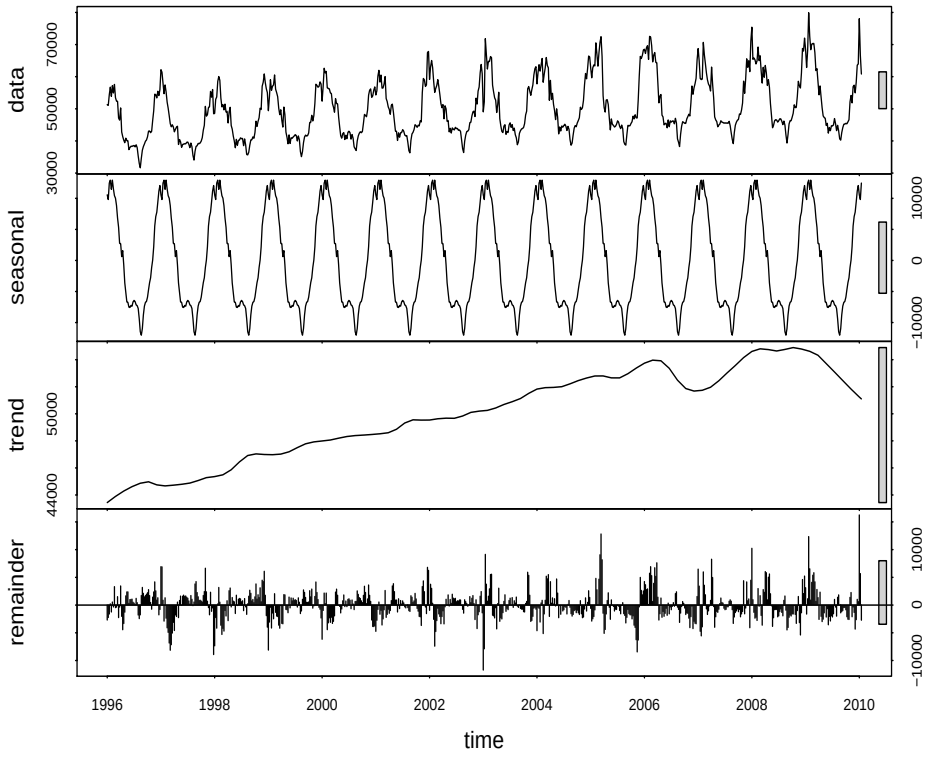


Figure 8.2: Decomposition of the electricity load time series (upper panel) into a seasonal component (second panel), a trend component (third panel) and a remainder (fourth panel).

Remark 40. *Note that there is also a multiplicative variant of the decomposition model (and also a mixed additive/multiplicative version). Also, if the fluctuations of the seasonal and noise component tend to increase with the corresponding level, then it might be worth considering a data transformation (e.g. using a logarithmic transform) first before applying a decomposition model.*

8.3 Step 2: Removing trend and seasonality

Two approaches are widely used for dealing with trend and seasonality: Either one can *difference* the data to generate a stationary time series, or one can fit a trend and/or seasonal model and remove the fitted model from the data to obtain a stationary time series. We will only study the former approach in this lecture, for the latter, see for instance (Brockwell & Davis 2016, Chapter 1.5) and the next problem class.

8.3.1 Differencing

Consider the stochastic process (X_t) and define the *lag-1 difference operator* ∇ by

$$\nabla X_t = X_t - X_{t-1} = (1 - B)X_t,$$

where B denotes the *back shift operator*

$$BX_t = X_{t-1}.$$

We can generalise the definitions of the difference and back shift operator to higher powers, i.e. $B^j(X_t) = X_{t-j}$ and $\nabla^j(X_t) = \nabla(\nabla^{j-1}(X_t))$, for $j \in \mathbb{N}$ with $\nabla^0(X_t) = X_t$. Consider a linear trend function:

$m_t = c_0 + c_1 t$. Then

$$\nabla m_t = m_t - m_{t-1} = c_0 + c_1 t - (c_0 + c_1(t-1)) = c_1,$$

which is a constant independent of time t. Similarly, one can remove any polynomial trend by applying the suitable (higher order) difference operator. This means that, given a time series (x_t), you can in principle difference the series repeatedly until you obtain a series which can be regarded as being generated by a stationary process. In practice, we typically only difference a few times (once or twice).

Example 41. Let us revisit our example of the population data from the UK studied in the first lecture.

```
MyData <- read.csv("UK-Population.csv", header = TRUE, sep = ",", dec=
    ".", skip=7)
population <- ts(MyData[,2],start=1971, end=2017, frequency =1)
par(mfrow=c(1,3))
plot(population, type = "l", pch=19)
plot(diff(population),type = "l", pch=19)
plot(diff(population,differences=2),type = "l", pch=19)
```

Using the `diff` function in R, we can display the original data and the first and second order differences:

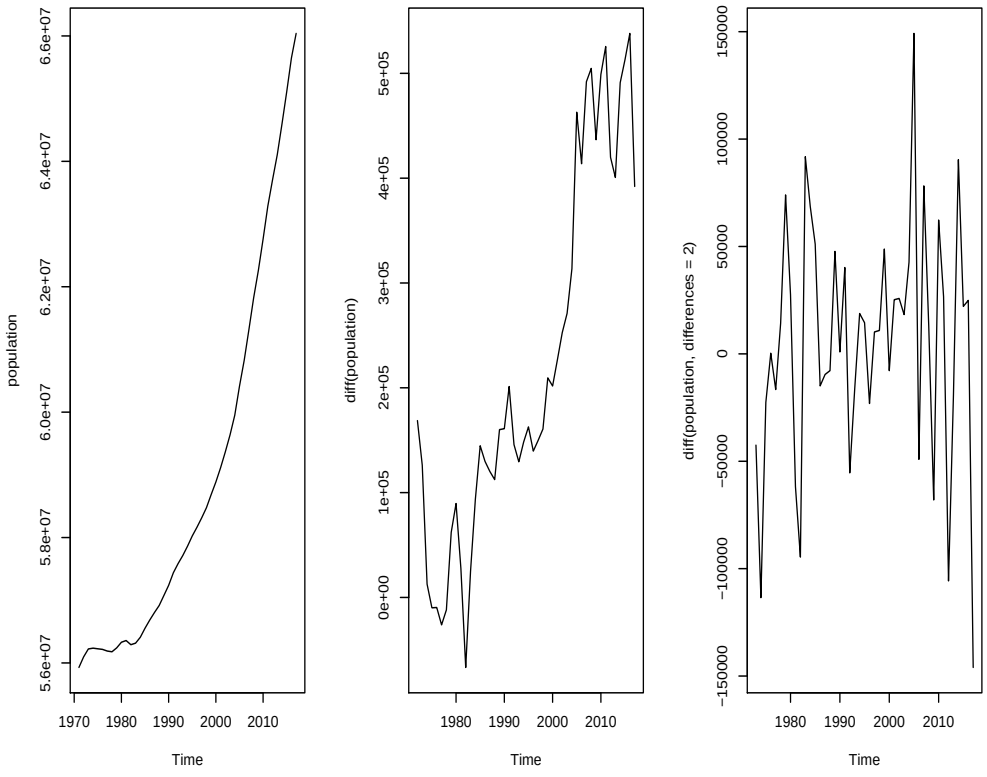


Figure 8.3: Yearly UK population from 1971 to 2017: Original data, first order differences and second order differences.

We observe that taking second order differences seems to remove the trend in the data and the new time series "looks" stationary. This conclusion still needs to be justified more precisely!

Exercise 42. *Write down the formula for* $\nabla^2 X_t$.

Solution: $\nabla^2 X_t = \nabla(X_t - X_{t-1}) = (X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) = X_t - 2X_{t-1} + X_{t-2}$.

Differencing can not only be used for removing trends, but also for removing seasonal features in a time series. For $d \in \mathbb{N}$, we define the *lag-d difference operator* ∇_d by

$$\nabla_d X_t = X_t - X_{t-d} = (1 - B^d) X_t$$

Consider the classical decomposition model, where s is assumed to have period d. Then

$$\nabla_d X_t = m_t - m_{t-d} + s_t - s_{t-d} + Y_t - Y_{t-d} = m_t - m_{t-d} + Y_t - Y_{t-d}.$$

Note that the trend component $m_t - m_{t-d}$ can be removed by applying a suitable power of the lag-1 difference operator.

Exercise 43. *Write down the formula for* $\nabla\nabla_d X_t$.

Solution: $\nabla\nabla_d X_t = \nabla(X_t - X_{t-d}) = (X_t - X_{t-d}) - (X_{t-1} - X_{t-d-1}) = X_t - X_{t-1} - X_{t-d} + X_{t-d-1}$.

Example 44. Let us revisit the example of the electricity load data from France studied in the first lecture.

```
library(opera)
attach(electric_load)
LoadTS <-ts(Load,start=1996, frequency =52)
par(mfrow=c(2,3))
plot(LoadTS,type ="l", ylab="Weekly load in MW") #Plot the original
    time series
plot(diff(LoadTS,lag=52))#Plot the seasonal differenced time series
plot(diff(diff(LoadTS,lag=52)))#Plot the series after applying the lag
    -1 and lag-52 difference operators
#Plot the corresponding empirical autocorrelation functions:
acf(LoadTS)
acf(diff(LoadTS,lag=52))
acf(diff(diff(LoadTS,lag=52)))
detach(electric_load)
```

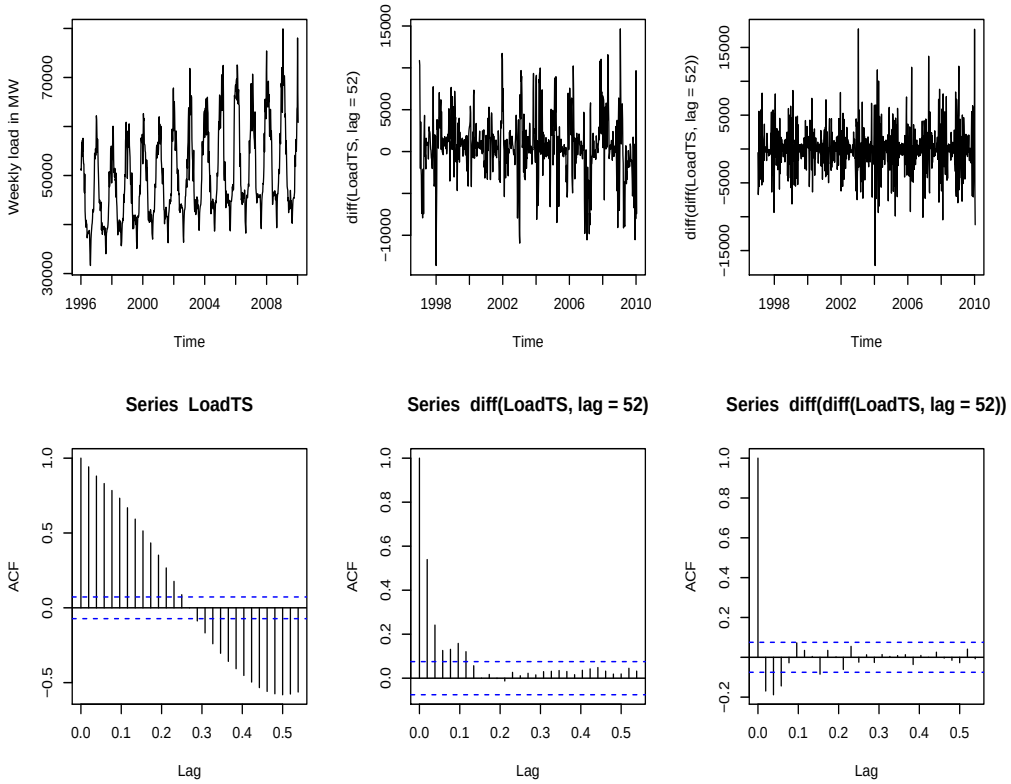


Figure 8.4: Weekly French electricity consumption data (in MW) from 1996 to 2009: The first row depicts the original time series (X_t), the seasonal differences $(\nabla_{52} X_t)$, and the differences $(\nabla\nabla_{52} X_t)$. The second row shows the corresponding empirical autocorrelation functions.

Lecture 9

Lecture 8: ARMA processes

In this chapter we want to study Step 3 of estimating a stationary time series model, possibly after trend and/or a seasonal component have been removed.

9.1 White noise and strict white noise processes

We will study the very popular class of *autoregressive moving average processes*. A so-called white noise process is the building block for many time series models. It is defined as follows:

Definition 45. The stochastic process $(X_t)_{t\in\mathbb{Z}}$ is a *white noise process* if it is stationary with autocorrelation function given by

$$\rho(h) = \begin{cases} 1, & \text{for } h = 0, \\ 0, & \text{for } h \neq 0. \end{cases}$$

If the white noise has zero mean and variance denoted by $\sigma^2 = \mathrm{Var}(X_t)$, then we typically write $WN(0, \sigma^2)$.

Remark 46. *Note that the definition of the white noise does not include any distributional assumption other than that its first and second moments exist. For example, white noise does not need to follow the Gaussian distribution.*

We have already introduced i.i.d. random variables yesterday. We will now introduce another name for them:

Definition 47. A stochastic process $(X_t)_{t\in\mathbb{Z}}$ is called a *strict white noise* (SWN), if it is a series of i.i.d. random variables with finite mean and finite variance. If the strict white noise has zero mean and variance denoted by $\sigma^2 = \mathrm{Var}(X_t)$, then we typically write $SWN(0, \sigma^2)$.

9.2 ARMA(p, q) processes

A very widely used class of linear time series models is the class of so-called ARMA(p, q) processes for $p, q \in \{0, 1, 2, \dots\}$.

Definition 48. Let $(\epsilon_t)_{t\in\mathbb{Z}}$ be WN(0,σ_ϵ^2). The stochastic process $(X_t)_{t\in\mathbb{Z}}$ is a *zero mean ARMA(p,q) process* if it is stationary and satisfies, for all $t \in \mathbb{Z}$,

help reduce the #parameters

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}, \tag{9.2.1}$$

where $\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q \in \mathbb{R}$ are the model parameters.[1]

[1] We also require that the (complex) polynomials $p_{AR}(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$ and $p_{MA}(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$ have no common factors.

want p, q small?

Remark 49. *Note that equation* (9.2.1) *is equivalent to*

autoregressive Moving average

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}, \quad \forall t \in \mathbb{Z}.$$

Note that we say that $(X_t)_{t\in\mathbb{Z}}$ is an ARMA(p,q) process with mean μ, if the centred process $(X_t - \mu)_{t\in\mathbb{Z}}$ is a zero-mean ARMA(p,q) process, i.e. it satisfies

$$(X_t - \mu) - \phi_1(X_{t-1} - \mu) - \cdots - \phi_p(X_{t-p} - \mu) = \epsilon_t + \theta_1\epsilon_{t-1} + \cdots + \theta_q\epsilon_{t-q}. \tag{9.2.2}$$

Exercise 50. *Verify that the mean of the process* (X_t) *satisfying equation* (9.2.2) *has indeed mean* μ.

Solution: We have

$$X_t = \mu + \phi_1(X_{t-1} - \mu) + \cdots + \phi_p(X_{t-p} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \cdots + \theta_q\epsilon_{t-q}.$$

Recall that we assume that (X_t) is stationary. So we can set $\mathrm{E}(X_t) = \mu^*$ for all t. We want to show that $\mu^* = \mu$. Now, taking expectations and using the fact that the expectation is linear, we deduce that

$$\begin{aligned}
\mathrm{E}(X_t) = \mu^* &= \mathrm{E}(\mu) + \mathrm{E}(\phi_1(X_{t-1} - \mu)) + \cdots + \mathrm{E}(\phi_p(X_{t-p} - \mu)) + \mathrm{E}(\epsilon_t) + \mathrm{E}(\theta_1\epsilon_{t-1}) + \cdots + \mathrm{E}(\theta_q\epsilon_{t-q}) \\
&= \mu + \phi_1\mathrm{E}((X_{t-1} - \mu)) + \cdots + \phi_p\mathrm{E}((X_{t-p} - \mu)) + \mathrm{E}(\epsilon_t) + \theta_1\mathrm{E}(\epsilon_{t-1}) + \cdots + \theta_q\mathrm{E}(\epsilon_{t-q}) \\
&= \mu + \phi_1(\mathrm{E}(X_{t-1}) - \mu) + \cdots + \phi_p(\mathrm{E}(X_{t-p}) - \mu) + \mathrm{E}(\epsilon_t) + \theta_1\mathrm{E}(\epsilon_{t-1}) + \cdots + \theta_q\mathrm{E}(\epsilon_{t-q}) \\
&= \mu + \phi_1(\mu^* - \mu) + \cdots + \phi_p(\mu^* - \mu) + \mathrm{E}(\epsilon_t) + \theta_1\mathrm{E}(\epsilon_{t-1}) + \cdots + \theta_q\mathrm{E}(\epsilon_{t-q}) \\
&= \mu + \phi_1(\mu^* - \mu) + \cdots + \phi_p(\mu^* - \mu).
\end{aligned}$$

So altogether we have

$$[1 - (\phi_1 + \cdots + \phi_p)](\mu^* - \mu) = 0,$$

which implies that $\mu^* = \mu$ since $1 - (\phi_1 + \cdots + \phi_p) \neq 0$. We assume that p_AR(z) != 0 , \forall |z| < 1

9.2.1 ARIMA(p,d,q) processes

I: Integrated ✓ Removing polynomial trend in data?

Note that the concept of differencing can be conveniently combined with the definition of an ARMA process, which leads to a so-called ARIMA process.

Definition 51. Let $p, q, d \in \{0, 1, 2, \ldots\}$. A stochastic process (X_t) is called an *ARIMA(p,d,q)* process if the differenced series (Y_t), where $Y_t = \nabla^d X_t$, is an ARMA(p,q) process.

Remark 52. *Note that an ARIMA(p,0,q) is identical to an ARMA(p,q) since no differencing is carried out.*

9.2.2 The first order moving average process or MA(1) process

If we set $p = 0$ and $q = 1$ in the definition of an ARMA process we obtain the MA(1) process. Also, in order to simplify the notation we will write θ instead of θ_1 for the parameter in the MA(1) process.

Let $(\epsilon_t)_{t\in\mathbb{Z}}$ be WN(0, σ_ϵ^2). The stochastic process $(X_t)_{t\in\mathbb{Z}}$ is a *zero mean MA(1) process* if it is stationary and satisfies

$$X_t = \epsilon_t + \theta\epsilon_{t-1}, \quad \text{for all } t \in \mathbb{Z}.$$

Exercise 53. *Compute the mean, variance and autocorrelation function of the MA(1) process.*[2]

[2] You may use without proof that for random variables X, Y, X_1, X_2, Y_1, Y_2 we have $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X, Y)$ and for deterministic constants $a, b, a_1, a_2, b_1, b_2 \in \mathbb{R}$ we have $\mathrm{Cov}(aX, bY) = ab\mathrm{Cov}(X, Y)$ and $\mathrm{Cov}(a_1X_1 + a_2X_2, b_1Y_1 + b_2Y_2) = a_1b_1\mathrm{Cov}(X_1, Y_1) + a_1b_2\mathrm{Cov}(X_1, Y_2) + a_2b_1\mathrm{Cov}(X_2, Y_1) + a_2b_2\mathrm{Cov}(X_2, Y_2)$.

Solution:

$$\mathrm{E}(X_t) = \mathrm{E}(\epsilon_t + \theta\epsilon_{t-1}) = \mathrm{E}(\epsilon_t) + \mathrm{E}(\theta\epsilon_{t-1}) = \mathrm{E}(\epsilon_t) + \theta\mathrm{E}(\epsilon_{t-1}) = 0.$$

$$\begin{aligned}\mathrm{Var}(X_t) &= \mathrm{Var}(\epsilon_t) + \mathrm{Var}(\theta\epsilon_{t-1}) + 2\mathrm{Cov}(\epsilon_t, \theta\epsilon_{t-1}) \\ &= \sigma_\epsilon^2 + \theta^2\sigma_\epsilon^2 + 2\theta\mathrm{Cov}(\epsilon_t, \epsilon_{t-1}) = (1+\theta^2)\sigma_\epsilon^2.\end{aligned}$$

For $h \in \mathbb{N}$, we have

$$\begin{aligned}\mathrm{Cov}(X_{t+h}, X_t) &= \mathrm{Cov}(\epsilon_{t+h} + \theta\epsilon_{t+h-1}, \epsilon_t + \theta\epsilon_{t-1}) \\ &= \mathrm{Cov}(\epsilon_{t+h}, \epsilon_t) + \theta\mathrm{Cov}(\epsilon_{t+h}, \epsilon_{t-1}) + \theta\mathrm{Cov}(\epsilon_{t+h-1}, \epsilon_t) + \theta^2\mathrm{Cov}(\epsilon_{t+h-1}, \epsilon_{t-1}) \\ &= \begin{cases} \theta\sigma_\epsilon^2, & \text{for } h = \pm 1, \\ 0, & \text{for } |h| > 1. \end{cases}\end{aligned}$$

Hence

$$\rho(h) = \mathrm{Cor}(X_{t+h}, X_t) = \begin{cases} 1, & \text{for } h = 0, \\ \theta/(1+\theta^2), & \text{for } h = \pm 1, \\ 0, & \text{for } |h| > 1. \end{cases}$$

We observe that the autocorrelation function of the MA(1) process cuts off at lag 1. More generally, one can show that the autocorrelation function of a MA(q) process cuts off at lag q.

9.2.3 The first order autoregressive process or AR(1) process

If we set $p = 1$ and $q = 0$ in the definition of an ARMA process we obtain the AR(1) process. Also, in order to simplify the notation we will write ϕ instead of ϕ_1 for the parameter in the AR(1) process.

Let $(\epsilon_t)_{t\in\mathbb{Z}}$ be WN(0,σ_ϵ^2). The stochastic process $(X_t)_{t\in\mathbb{Z}}$ is a *zero mean AR(1) process* if it is stationary and satisfies

$$X_t = \phi X_{t-1} + \epsilon_t, \quad \text{for all } t \in \mathbb{Z}.$$

Suppose that $|\phi| < 1$ (such AR(1) processes are called *causal*). Then

$$\begin{aligned}X_t &= \phi X_{t-1} + \epsilon_t \\ &= \phi(\phi X_{t-2} + \epsilon_{t-1}) + \epsilon_t = \phi^2 X_{t-2} + \phi\epsilon_{t-1} + \epsilon_t = \cdots \\ &= \phi^k X_{t-k} + \sum_{j=0}^{k-1} \phi^j \epsilon_{t-j} \longrightarrow \sum_{j=0}^{\infty} \phi^j \epsilon_{t-j}, \text{ as } k \to \infty.\end{aligned}$$

MA(inf) process

The above asymptotic results requires a proper proof and a statement in which sense we have convergence. Such a discussion is beyond the scope of this course.

Exercise 54. *Compute the mean, variance and autocorrelation function of the AR(1) process.*

Solution:

$$\mathrm{E}(X_t) = \mathrm{E}(\theta X_{t-1} + \epsilon_t) = \theta\mathrm{E}(X_{t-1}) + \mathrm{E}(\epsilon_t) = \theta\mathrm{E}(X_{t-1}).$$

This implies that $\mathrm{E}(X_t) = 0$. For the variance, we have

$$\begin{aligned}\gamma(0) = \mathrm{Var}(X_t) &= \mathrm{Var}(\phi X_{t-1} + \epsilon_t) \\ &= \phi^2\mathrm{Var}(X_{t-1}) + \mathrm{Var}(\epsilon_t) + 2\phi\mathrm{Cov}(X_{t-1}, \epsilon_t) = \phi^2\gamma(0) + \sigma_\epsilon^2 + 0.\end{aligned}$$

This implies that $\gamma(0) = \sigma_\epsilon^2/(1-\phi^2)$. For $h \in \mathbb{N}$, we have

$$\gamma(h) = \mathrm{Cov}(X_{t+h}, X_t) = \mathrm{Cov}(\phi X_{t+h-1} + \epsilon_{t+h}, X_t)$$

$$= \phi \text{Cov}(X_{t+h-1}, X_t) + \text{Cov}(\epsilon_{t+h}, X_t) = \phi\gamma(h-1) + 0 = \cdots$$
$$= \phi^h \gamma(0).$$

Hence

$$\rho(h) = \text{Cor}(X_{t+h}, X_t) = \frac{\gamma(h)}{\gamma(0)} = \phi^{|h|}, \text{ for } h \in \mathbb{Z}.$$

Lecture 10

Lecture 9: Estimating ARMA processes and assessing the model fit

10.1 Model estimation

There is a variety of methods which can be used for estimating the parameters in ARMA processes. Here we will only mention one method: the so-called maximum-likelihood approach.

10.1.1 Maximum likelihood estimation

Consider a general ARMA(p,q) process with mean μ and suppose you know the order of the process, i.e. p and q. We need to estimate the $p+q+1$ parameters $\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$. Let us shorten the notation and write $\boldsymbol{\zeta} = (\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)^\top$ for the $p+q+1$-dimensional parameter vector. In addition, you might want to estimate the variance σ_ϵ^2 of the white noise process which appears in the definition of the ARMA process. The maximum likelihood estimation technique finds the parameter values which maximise the probability of obtaining the data which we have observed.

More precisely, consider our time series $x_1, \ldots, x_n$ which we regard as realisations of the random vector $\mathbf{X} = (X_1, \ldots, X_n)^\top$. Suppose that the random vector has a joint density function denoted by f. Rather than writing $f(x_1, \ldots, x_n)$ we will write $f(x_1, \ldots, x_n|\boldsymbol{\zeta})$ to indicate that the joint density function depends on the parameter vector $\boldsymbol{\zeta}$. Then the likelihood function is given by $L(\boldsymbol{\zeta}) = f(x_1, \ldots, x_n|\boldsymbol{\zeta})$ and the maximum likelihood estimator $\widehat{\boldsymbol{\zeta}}$ is a vector, out of all possible choices of $\boldsymbol{\zeta}$, which maximises $L(\boldsymbol{\zeta})$.

Example 55. In the case when $\mathbf{x} = (x_1, \ldots, x_n)^\top$ is assumed to be a realisation of a Gaussian time series model $\mathbf{X} = (X_1, \ldots, X_n)^\top$ with zero mean, then the likelihood of $\mathbf{x} = (x_1, \ldots, x_n)^\top$ is given by

$$L(\boldsymbol{\zeta}) = (2\pi)^{-n/2} (\det(\Gamma_n))^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}^\top \Gamma_n^{-1} \mathbf{x}\right),$$

where $\Gamma_n = \mathrm{E}(\mathbf{X}^\top \mathbf{X})$ is the covariance matrix which is assumed to be nonsingular. Note that Γ_n depends on the model parameters $\boldsymbol{\zeta}$.

10.1.2 Order selection

We typically use an information criterion for estimating the order (i.e. p and q) in the ARMA(p,q) process. Information criteria give you a method for finding the right balance between having a model with high likelihood, but not too many parameters. So they typically include the likelihood, denoted by L, and a penalty term involving the number of parameters. Suppose our time series has n observations. In the following, we will denote by log the natural logarithm, i.e. the inverse function of the exponential function.

Three information criteria are widely used:

AIC Akaike's Information Criterion for ARMA(p,q) processes:

$$AIC = -2\log(L) + 2(p+q+b+1),$$

where $b = 1$ for non-zero mean $\mu \neq 0$ and $b = 0$ for zero mean $\mu = 0$.

AICC the Corrected AIC:

Better

$$AICC = AIC + \frac{2(p+q+b+1)(p+q+b+2)}{n-p-q-b-2},$$

BIC the Bayesian Information Criterion

$$BIC = AIC + (\log(n) - 2)(p+q+b+1).$$

We select the best model by minimising AIC, AICC or BIC. Note that the penalty terms in the AIC and AICC are asymptotically equivalent when $n \to \infty$, but AICC puts a higher penalty on large-order models, whereas the AIC is often prone to overfitting.

10.2 Goodness of fit

Suppose you have estimated an ARMA model. We will now turn to Step 4 in our step-by-step guide, where we will need to assess the model fit (and possibly compare models).

10.2.1 Computing the residuals of an ARMA process

Consider (for simplicity) a zero-mean ARMA(p,q) process as defined in equation (9.2.1):

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}.$$

We can rearrange this equation to obtain:

$$\epsilon_t = X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} - (\theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}).$$

Let $\hat{\phi}_1, \ldots \hat{\phi}_p, \hat{\theta}_1, \ldots, \hat{\theta}_q$ denote the parameter estimates obtained through (maximum likelihood) estimation. Given our time series $(x_1, \ldots, x_n)$ we can compute the residuals as the difference between the original data and the so-called fitted values:

$$\hat{\epsilon}_t = x_t - \hat{\phi}_1 x_{t-1} - \cdots - \hat{\phi}_p x_{t-p} - (\hat{\theta}_1 \hat{\epsilon}_{t-1} + \cdots + \hat{\theta}_q \hat{\epsilon}_{t-q}),$$

for $t = 1, \ldots, n$. Note that we need to choose initial values $x_0, \ldots, x_{1-p}, \hat{\epsilon}_0, \ldots, \hat{\epsilon}_{1-q}$. For instance, you could set $\hat{\epsilon}_0 = \cdots = \hat{\epsilon}_{1-q} = 0$ and $x_0 = \cdots = x_{1-p} = \bar{x}$. You can check the R help pages for more details on this (depending on which function/package you choose).

10.2.2 Are the residuals white noise?

For our ARMA(p,q) model to describe the data well, we expect that the residuals are (strict) white noise. Let us discuss how this requirement can be checked in practice. Recall the definition of the sample autocorrelation function, see Definition 33.

For a time series $(x_1, \ldots, x_n)^\top$ (being a realisation of $(X_1, \ldots, X_n)^\top$), we denote the sample mean by $\overline{x} = \frac{1}{n}\sum_{t=1}^n x_t$ and the *sample autocovariance function* by

$$\hat{\gamma}_x(h) := \frac{1}{n}\sum_{t=1}^{n-|h|} (x_{t+|h|} - \overline{x})(x_t - \overline{x}), \text{ for } -n < h < n.$$

Then the *sample autocorrelation function* is given by

$$\hat{\rho}_x(h) := \frac{\hat{\gamma}_x(h)}{\hat{\gamma}_x(0)}, \text{ for } -n < h < n.$$

We have already used the R function `acf` to display the *correlogram*, which is the plot

$$\{(h, \hat{\rho}_x(h)) : h = 0, 1, 2, \dots\}.$$

Did you notice the blue dotted lines in the correlogram produced in R??

Suppose briefly that $(X_1, \dots, X_n)$ is a strict white noise. In that case, one can show that, for large n, the sample autocorrelations $(\hat{\rho}_x(1), \dots, \hat{\rho}_x(h))^\top$ behave like observations from i.i.d. random variables with $N(0, 1/n)$ distribution. That means that we expect that 95% of our observations fall between the 0.025 and 0.975 quantiles of the $N(0, 1/n)$ distribution, which are given by $-1.96/\sqrt{n}$ and $1.96/\sqrt{n}$, respectively, and that 5% of our observations fall outside these bands[1]. Hence, we typically plot these bounds with the sample autocorrelations. If more than 5% of the estimated correlations are outside the bounds, we would conclude that our data are not realisations of a white noise process[2].

Remark 56. *When checking the goodness of fit of your estimated ARMA process, you would compute the sample acf of the residuals, i.e. $(\rho_{\hat{\epsilon}}(1), \dots, \rho_{\hat{\epsilon}}(h))^\top$, to check whether they can be regarded as realisations of a white noise process.*

Portmanteau tests

In addition to plotting sample autocorrelations with the corresponding confidence bounds, we typically also carry out so-called Portmanteau tests. Here we tests whether the the first H, say, sample autocorrelations are significantly bigger than a strict white noise process would suggest.

Box-Pierce test The Box-Pierce test statistics is given by

$$Q_{BP} = n \sum_{j=1}^{H} \hat{\rho}(j)^2.$$

Ljung-Box test The Ljung-Box test statistics is given by

$$Q_{LB} = n(n+2) \sum_{j=1}^{H} \frac{\hat{\rho}(j)^2}{n-j}.$$

In both cases, a large value of Q_{BP} or Q_{LB} suggests that the data do not come from a strict white noise series. In order to decide what we mean by a "large value", we use the fact, that in the case of a strict white noise, both statistics follow asymptotically a χ^2-distribution with $H - K$ degrees of freedom, where K denotes the number of model parameters (so in the ARMA(p,q)-case with zero mean, we have $K = p + q$ and if the tests are applied to raw data, then we set $K = 0$)[3]

Note that the Ljung-Box test is generally preferred to the Box-Pierce test.

Remark 57. *In the definitions of Q_{BP} and Q_{LB}, we used the notation for the sample acf $\hat{\rho}$ without any subscripts. If you are applying the test to your residuals from the ARMA(p,q) process, you will need to use the corresponding sample acfs $\hat{\rho}_{\hat{\epsilon}}$.*

Remark 58. *You will need to decide how many sample autocorrelations to include in the test statistics above. Various rules of thumb exist. For instance, Hyndman & Athanasopoulos (2018) suggest to choose $H = 10$ for non-seasonal data and $H = 2d$ for seasonal data, where d denotes the period of the seasonality. In addition, they stress that H should not be taken too large and should be at most $H = n/5$.*

[1] We say that a 95% (asymptotic) confidence interval is given by $(-1.96/\sqrt{n}, 1.96/\sqrt{n})$.

[2] Technically, we would *reject the null hypothesis* that the data are generated by a strict white noise process.

[3] More precisely, in practice we would reject the null hypothesis that the data are sampled from strict white noise at significance level $\alpha \in (0, 1)$ (typically, $\alpha \in \{0.01, 0.05\}$) if Q_{LB} (or Q_{BP}) is greater than the $1 - \alpha$-quantile of the χ^2-distribution with $H - K$ degrees of freedom.

Lecture 11

Problem class 2

This problem class is split into two parts. In the first half, you are expected to work through given R code to familiarise yourself with some key R functions used for time series analysis. Here, we will be working with simulated data, so that we actually know the so-called *data generating process*.

In the second half of the problem class, we will turn to a case study of financial data where your task will be to explore some of the so-called stylised facts of financial time series.

11.1 Case study: Simulation study of an ARMA(1,1) process with polynomial trend [45min]

11.1.1 Simulating an ARMA(1,1) process

Let us first simulate a time series from an ARMA(1,1) process using the `arima.sim` function from the `stats` package.

```
library(ggplot2)
library(forecast)
#Read the manual of the arima.sim function in the stats package
?arima.sim

#Set the length of the time series
n <- 500

#Simulate an ARMA(1,1) process with standard normal white noise and
    phi=0.5, theta=0.3
#Fix the seed for the random number generation
set.seed(1)
x <- arima.sim(model = list(ar = 0.5, ma = 0.3), n = n, sd=5)  # ARMA
    (1,1) process with N(0,5^2) strict white noise
plot(x, type="l")
acf(x)

#Alternatively you can use autoplot() from the ggplot2 package
autoplot(x)
```

11.1.2 Fitting various ARMA processes and assessing the model fit

Next we fit an AR(1), MA(1) and ARMA(1,1) model to the data and print the estimated parameters.

```
fit_ar1 <- Arima(x, order=c(1,0,0))
fit_ma1 <- Arima(x, order=c(0,0,1))
fit_arma11 <- Arima(x, order=c(1,0,1))

#Print the parameter estimates (and compare them to the true values!)
fit_ar1
fit_ma1
fit_arma11
```

You can use the `checkresiduals()` function from the `forecast` package to assess the model fit. It plots the residuals, their sample autocorrelation function and their empirical distribution with a fitted normal distribution added as a red line. Note that the results of the Ljung-Box test are also reported where $Q*$ denotes the Ljung-Box test statistics which we denoted by Q_{LB} in the lecture notes. If the test statistic is large and hence the p-value is small (e.g. below 0.05 or 0.01 say), we would reject the null hypothesis of strict white noise.

```
checkresiduals(fit_ar1)
checkresiduals(fit_ma1)
checkresiduals(fit_arma11)

#You can also use the auto.arima function to automatically select the
    order and fit an ARIMA process
fit <- auto.arima(x,ic = "aicc")
fit
checkresiduals(fit)
```

In your simulation, vary the length of the time series, e.g. choose n=100 and n=200 and check whether you would have selected the correct model!

11.1.3 Adding (and a removing!) a trend function to (from) the simulated data

Let us now add a trend function of the form

$$m_t = 5 + 0.03t + 0.005t^2$$

to the ARMA process.

```
#Define the trend function m(t)=a+bt+ct^2
m <-function(t,a,b,c){
  a+b*t+c*t^2
}
#Set a=5, b=0.03, c=0.005
trend <- m((1:n),5,0.03,0.005)
plot(trend,type="l")
```

Suppose that (X_t) denotes the ARMA(1,1) process, then be define

$$Y_t = m_t + X_t.$$

```
y <- trend + x
plot(y, type="l")
```

Next compare two ways of removing the trend: First, try differencing the data:

```
###Use differencing to get rid off trend
model1 <-auto.arima(y,ic="aicc")
model1
checkresiduals(model1)
```

Second, fit a parametric model to the trend function and subtract it from the observations.

```
###Fit a parametric model
timepoints <- (1:n)
#Use non-linear least squares to fit the trend function
fit_trend <- nls(y ~ m(timepoints,a,b,c), start = list(a=0,b=0,c=0))
summary(fit_trend)

ahat <- coef(fit_trend)[1]
bhat <- coef(fit_trend)[2]
chat <- coef(fit_trend)[3]

plot(y,type="l")
lines(timepoints, predict(fit_trend, newdata=timepoints), col=2)

#Remove trend
detrended_data <-y-predict(fit_trend, newdata=timepoints)
model2 <- auto.arima(detrended_data,ic="aicc")
model2
```

11.1.4 Bonus material: Forecasting

Suppose now that you want to make predictions from your time series models and you want to study which of your models is good at predicting future observations. In order to do that, we typically split the available data into a *training* set and a *test* set. We then fit various models to the data in the training set, then make predictions from the models for the time period covered in the test set and compare the predicted values with the true observations in the test set. Various performance measures can be used for assessing forecast performance. We start off with a visual assessment, but will compute quantitative performance measures as well:

```
#Split the sample into a training and a test sample
#Here we take 80% of the data for the training sample
length_training <- floor(80/100*n) # we round it down in case we don't
    have an integer-valued number
length_test <- n-length_training
#Use the window function to split the dataset:
y.train <- window(y,start=1, end=length_training)
y.test <- window(y, start=length_training+1, end=n)


###Use differencing to get rid off trend
model1 <-auto.arima(y.train,ic="aicc")
model1
checkresiduals(model1)

###Fit a parametric model
timepoints <- (1:length_training)
fit_trend <- nls(y.train ~ m(timepoints,a,b,c), start = list(a=0,b=0,c
   =0))
```

```
summary(fit_trend)
#The second parameter is not significant, so we refit the model
    excluding b
fit_trend <- nls(y.train ~ m(timepoints,a,0,c), start = list(a=0,c=0))
summary(fit_trend)

ahat <- coef(fit_trend)[1]
chat <- coef(fit_trend)[2]

plot(y.train,type="l")
lines(timepoints, predict(fit_trend, newdata=timepoints), col=2)

#Remove trend
detrended_data <-y.train-predict(fit_trend, newdata=timepoints)
model2 <- auto.arima(detrended_data,ic="aicc")
model2
checkresiduals(model2)
```

We can now make predictions from the two fitted models using the `predict` function:

```
model1.pred <- predict(model1, n.ahead = length_test)$pred
model2.pred <- predict(model2, n.ahead = length_test)$pred +m(((length
    _training+1):(length_training+length_test)),ahat,0,chat)
```

Let us compare the two predictions visually:

```
plot(model1.pred,col = "blue", lty = 2)
lines(model2.pred, col = "green", lty = 2)
```

The above picture is not really useful, so we better plot the data from the test dataset first and then add the predicted values from the two models:

```
plot.ts(y.test)
lines(model1.pred, col = "blue", lty = 2)
lines(model2.pred, col = "green", lty = 2)
```

Note that you can use the `forecast` function to make predictions and create so-called fan plots which display prediction intervals which are very useful for real applications, where you might not need a point forecast but a possible range in which future observations will most likely fall:

```
forecast(model1,h=10)$mean
plot(forecast(model1,h=length_test,fan=TRUE))#$mean
lines(y, col="black",lwd=2)


y_fulltrend <-c(predict(fit_trend, newdata=(1:n)),m(((length_training
    +1):(length_training+length_test)),ahat,0,chat))
y_detrended <- y - y_fulltrend
plot(forecast(model2,h=length_test,fan=TRUE))
lines(y_detrended, col="black",lwd=2)
```

Quantitative measures of forecast performance can be computed using the `accuracy` function from the `forecast` package.

```
accuracy(model1)
accuracy(model2)
```

The measures which will be calculated are the mean error, the root mean square error, mean absolute error, mean percentage error, mean absolute percentage error, mean absolute scaled error and the autocorrelation of errors at lag 1.

We will be studying some of the theoretical concepts of forecasting ARMA processes in tomorrow's lecture, but if you would like to do additional reading on the topic, please take a look at the excellent online textbook Hyndman & Athanasopoulos (2018).

11.2 Case study: Stylised facts of financial time series [45 min]

The so-called *stylised facts* of financial time series refer to statistical properties of financial time series which can be observed in almost all financial time series, see for instances Cont (2001, p.201). In this part of the problem class, you are expected to study some of the most prominent stylised facts and identify them in your dataset. We will be looking at daily closing prices (in USD) for the Google stock price recorded over a two year time period from 01 January 2020 to 22 June 2022. The dataset has been downloaded from the Yahoo Finance website https://uk.finance.yahoo.com/quote/GOOG/history?p=GOOG.

11.2.1 Computing the log returns

Suppose that the daily closing price is denoted by S_t. Given a time interval denoted by $\Delta > 0$ we define the log return at scale Δ as

$$r(t, \Delta) = \log(S_{t+\Delta}) - \log(S_t).$$

A common choice for the scale is $\Delta = 1$. In our case, such a choice would lead to *daily* log returns.

Exercise 59. *Read in the data and compute the daily log returns. Plot the prices and the log returns and compute some summary statistics of the data.*

```
#Read in the data
GOOG <- read.csv("GOOG.csv", header = TRUE, sep = ",", dec = ".")
#Use the daily closing prices and compute the log returns
GOOG_prices <-GOOG$Close #daily closing prices
GOOG_logreturns <-diff(log(GOOG_prices))

#Plot the prices and log returns
par(mfrow=c(1,2))
plot(GOOG_prices, type="l")
plot(GOOG_logreturns, type ="l")

#Compute summary statistics
library(fBasics)
basicStats(GOOG_prices)
basicStats(GOOG_logreturns)
```

11.2.2 Stylised fact: Absence of autocorrelation

Autocorrelations between asset log returns are typically insignificant and are typically only observed at high frequencies (from around 20 minute observations to even more frequent observations). Can you verify this claim in your dataset? Run the following checks and interpret your results.

```
library(stats)
library(forecast)
par(mfrow=c(1,1))
acf(GOOG_logreturns,lag=100)
```

```
Box.test(GOOG_logreturns,lag=10,type="Ljung-Box")
auto.arima(GOOG_logreturns)
```

11.2.3 Stylised fact: Heavy tails

The distribution of the log returns typically follows a distribution with heavier tails than the Gaussian distribution. Fit a Gaussian distribution to your log returns and look at the corresponding quantile-quantile plot. What do you observe?

```
library(car)
qqPlot(GOOG_logreturns, main="Quantile-quantile plot")
```

If the Gaussian distribution does not work well, which other distribution(s) could we use to describe log returns? It appears that the so-called *normal inverse Gaussian* (NIG) and other variants of the so-called *generalised hyperbolic* (GH) distribution typically work well in financial applications. The `ghyp` package will help us to explore these distributions in more detail. There are various parametrisations of the GH distribution and you can find many useful details on this in Luethi & Breymann (2013). The class of generalised hyperbolic distributions contains 11 distributions as special cases:

1. Asymmetric generalised hyperbolic (GH) distribution,
2. symmetric generalised hyperbolic (GH) distribution,
3. asymmetric hyperbolic (H) distribution,
4. symmetric hyperbolic (H) distribution,
5. asymmetric Student t (t) distribution,
6. symmetric Student t (t) distribution,
7. asymmetric variance gamma (VG) distribution,
8. symmetric variance gamma (VG) distribution,
9. asymmetric normal inverse Gaussian (NIG) distribution,
10. symmetric normal inverse Gaussian (NIG) distribution,
11. Gaussian distribution.

There is a convenient function `stepAIC.ghyp` which fits all 11 distributions and compares the fit according to the AIC. When you run the following code, which distribution is considered the best one according to the AIC?

```
library(ghyp)
stepAIC.ghyp(GOOG_logreturns,silent=TRUE)
```

We can also fit individual distributions such as the VG, NIG, Student t or Gaussian distribution:

```
#Fit a VG distribution:
GOOG.VGfit <- fit.VGuv(GOOG_logreturns, silent = TRUE)
#Fit an NIG distribution:
GOOG.NIGfit <- fit.NIGuv(GOOG_logreturns, silent = TRUE)
#Fit a Student t distribution:
GOOG.tfit <- fit.tuv(GOOG_logreturns, silent = TRUE)
#Fit a Gaussian distribution:
GOOG.gaussfit <- fit.gaussuv(GOOG_logreturns)
```

You can plot the histograms with the fitted densities (and the Gaussian density for comparison):

```
hist(GOOG.VGfit)
hist(GOOG.NIGfit)
hist(GOOG.tfit)
```

We clearly see that the VG, the NIG and the Student t fit is better than the Gaussian fit. Let us zoom into the tails of the fitted distribution by looking at the quantile-quantile plots:

```
#Look at qq-plots next using the function qqghyp
#Let us start with the variance gamma fit:
qqghyp(GOOG.VGfit, ghyp.col = "red", line = TRUE) #Compare Gaussian (
    black) and VG (red)
qqghyp(GOOG.VGfit,  gaussian=FALSE, ghyp.col = "red", line = TRUE) #
    Only VG

#Compare Gaussian, VG and student-t
qqghyp(GOOG.VGfit, ghyp.col = "red", line = TRUE)
qqghyp(GOOG.NIGfit,add = TRUE, ghyp.col = "green", line = TRUE)
qqghyp(GOOG.tfit,add = TRUE, ghyp.col = "blue", line = TRUE)

#Compare VG and NIG
qqghyp(GOOG.VGfit,  gaussian=FALSE, ghyp.col = "red", line = TRUE)
qqghyp(GOOG.NIGfit, gaussian=FALSE, add = TRUE, ghyp.col = "green",
    line = TRUE)
```

Do these pictures confirm your findings from running the `stepAIG.ghyp` function?

11.2.4 Stylised fact: Aggregational Gaussianity

If you increase the time scale Δ over which you compute your log returns, then the corresponding empirical distribution resembles a Gaussian distribution more closely:

```
#Convert the data into a time series with 226 days per year (since
    weekends are excluded)
returnts <-ts(GOOG_logreturns, frequency =226)
#Use the aggregate function to compute various aggregates
twodaily <- aggregate(returnts, nfrequency =226/2, sum)
weekly <- aggregate(returnts, nfrequency =226/5, sum)
monthly <- aggregate(returnts, nfrequency =226/19, sum)

#Display the time series plots for four different choices of the time
    scale
par(mfrow=c(2,2))
plot(returnts, type="l")
plot(twodaily, type="l")
plot(weekly, type="l")
plot(monthly, type="l")

#Display the true histograms for four different choices of the time
    scale
library(MASS)
truehist(returnts)
truehist(twodaily)
truehist(weekly)
truehist(monthly)
```

```
21
22 #Display the quantile-quantile plots for four different choices of the
       time scale
23 qqPlot(returnts, main="Quantile-quantile plot")
24 qqPlot(twodaily, main="Quantile-quantile plot")
25 qqPlot(weekly, main="Quantile-quantile plot")
26 qqPlot(monthly, main="Quantile-quantile plot")
```

11.2.5 Stylised fact: Slow decay of the autocorrelation in absolute log returns

While there is typically (almost) no autocorrelation in the log returns, this changes when we consider the absolute value of log returns instead. Is this true for our dataset?

```
1 acf(abs(GOOG_logreturns))
2 Box.test(abs(GOOG_logreturns),lag=10,type="Ljung-Box")
```

Part IV

Thursday 14 July 2022

Lecture 12

Lecture 10: Predicting ARMA(p,q) processes

In this lecture we are going to describe how we can predict ARMA(p,q) processes. We will not be able to give a rigorous mathematical derivations since they rely on computations involving conditional expectations which are beyond the scope of this course, but we rather sketch the intuition.

12.1 Forecasting an ARMA(1,1) process

Consider an ARMA(1,1) process with zero mean given by

$$X_t = \phi X_{t-1} + \epsilon_t + \theta \epsilon_{t-1}.$$

Suppose we have realisations $x_1, \ldots, x_n$ from the ARMA(1,1) process and we would like to make a one (or even h) step ahead prediction, i.e. we would like to find the value $\hat{x}_{n+1}$ (or $\hat{x}_{n+h}$, for $h \in \mathbb{N}$). How do we proceed?
Step 1: Write down the ARMA(1,1) equation for time $n+1$

$$X_{n+1} = \phi X_n + \epsilon_{n+1} + \theta \epsilon_n.$$

Step 2: Replace all unknown quantities by suitable forecasts based on the information available to you up to time n. We will use the notation $\hat{x}_{n+1}$ for the predicted value of the time series at time $n+1$ given the information up to time n.

- Note that we assume that we have estimated the parameters ϕ, θ and hence replace them with their estimates $\hat{\phi}, \hat{\theta}$.
- We know the realisation x_n of X_n. Hence we use $\hat{x}_n = x_n$.
- What about $\hat{\epsilon}_{n+1}$? Given that we work under the white noise assumption, we can set $\hat{\epsilon}_{n+1} = 0$ (and even $\hat{\epsilon}_{n+h} = 0$).
- Finally we need to find $\hat{\epsilon}_n$. However, this residual is known to us after we have estimated the ARMA(1,1) process. We can write:

$$\epsilon_t = X_t - \phi X_{t-1} - \theta \epsilon_{t-1}.$$

- Hence (given starting values X_0, ϵ_0)

$$\epsilon_1 = X_1 - \phi X_0 - \theta \epsilon_0,$$
$$\epsilon_2 = X_2 - \phi X_1 - \theta \epsilon_1,$$

$$\vdots$$
$$\epsilon_n = X_n - \phi X_{n-1} - \theta\epsilon_{n-1}.$$

Hence (given starting values $x_0, \hat{\epsilon}_0$), we can recursively compute

$$\hat{\epsilon}_1 = x_1 - \hat{\phi}x_0 - \hat{\theta}\hat{\epsilon}_0,$$
$$\hat{\epsilon}_2 = x_2 - \hat{\phi}x_1 - \hat{\theta}\hat{\epsilon}_1,$$
$$\vdots$$
$$\hat{\epsilon}_n = x_n - \hat{\phi}x_{n-1} - \hat{\theta}\hat{\epsilon}_{n-1}.$$

We then get the following formula for the one-step ahead prediction:

$$\hat{x}_{n+1} = \hat{\phi}x_n + \hat{\theta}\hat{\epsilon}_n.$$

For the two-step ahead forecast we argue similarly:

Step 1: We write

$$X_{n+2} = \phi X_{n+1} + \epsilon_{n+2} + \theta\epsilon_{n+1}.$$

Step 2: We note that both white noise terms will vanish as mentioned above: $\hat{\epsilon}_{n+h} = 0$ for all $h \in \mathbb{N}$. Hence we have

$$\hat{x}_{n+2} = \hat{\phi}\hat{x}_{n+1} = \hat{\phi}(\hat{\phi}x_n + \hat{\theta}\hat{\epsilon}_n) = \hat{\phi}^2 x_n + \hat{\phi}\hat{\theta}\hat{\epsilon}_n.$$

This procedure can be iterated: Since, for $h \in \mathbb{N}, h \geq 2$, we have

$$X_{n+h} = \phi X_{n+h-1} + \epsilon_{n+h} + \theta\epsilon_{n+h-1},$$

the h-step ahead forecast is given by

$$\hat{x}_{n+h} = \hat{\phi}\hat{x}_{n+h-1} = \hat{\phi}^h x_n + \hat{\phi}^{h-1}\hat{\theta}\hat{\epsilon}_n.$$

Remark 60. *If you include a non-zero mean μ in the ARMA(1,1) process, then analogous arguments lead to the following formula for the h-step ahead prediction:*

$$\hat{x}_{n+h} = \hat{\mu} + \hat{\phi}^h(x_n - \hat{\mu}) + \hat{\phi}^{h-1}\hat{\theta}\hat{\epsilon}_n.$$

In the case when $|\hat{\phi}| < 1$, the long term forecast will converge to $\hat{\mu}$.

Remark 61. *The above arguments can be extended to derive suitable prediction formulas for general ARMA(p,q) processes.*

12.2 Case study

First, we simulate an ARMA(1,1) process of the form

$$X_t = 0.5X_{t-1} + \epsilon_t + 0.3\epsilon_{t-1}, \qquad (\epsilon_t) \sim \text{i.i.d.} N(0, 5^2),$$

and we plot the corresponding sample path and the sample autocorrelation:

```
1 library(forecast)
2 #Set the length of the time series
3 n <- 400
```

```
#Simulate an ARMA(1,1) process with standard normal white noise and
    phi=0.5, theta=0.3
set.seed(100)
# Simulate an ARMA(1,1) process with N(0,5^2) strict white noise
x <- arima.sim(model = list(ar = 0.5, ma = 0.3), n = n, sd=5)
plot(x, type="l")
acf(x)
```

In a next step, we split the sample into a *training* and a *test* set.

```
#Split the sample into a training and a test sample
length_test <- 30
length_training <- n-length_test
x.train <- window(x,start=1, end=length_training)
x.test <- window(x, start=length_training+1, end=n)
```

Next we fit an AR(1), MA(1) and ARMA(1,1) process and assess the model fit, by checking the corresponding residuals:

```
###Fit an AR(1), MA(1) and ARMA(1,1) process and check the residual
fit_ar1 <- Arima(x.train, order=c(1,0,0))
fit_ma1 <- Arima(x.train, order=c(0,0,1))
fit_arma11 <- Arima(x.train, order=c(1,0,1))
checkresiduals(fit_ar1)
checkresiduals(fit_ma1)
checkresiduals(fit_arma11)
#Note that you can plot the acf of the residuals also using the
    following command:
par(mfrow=c(1,1))
plot(acf(fit_arma11$residuals,lag=100,plot=F)[1:100])
```

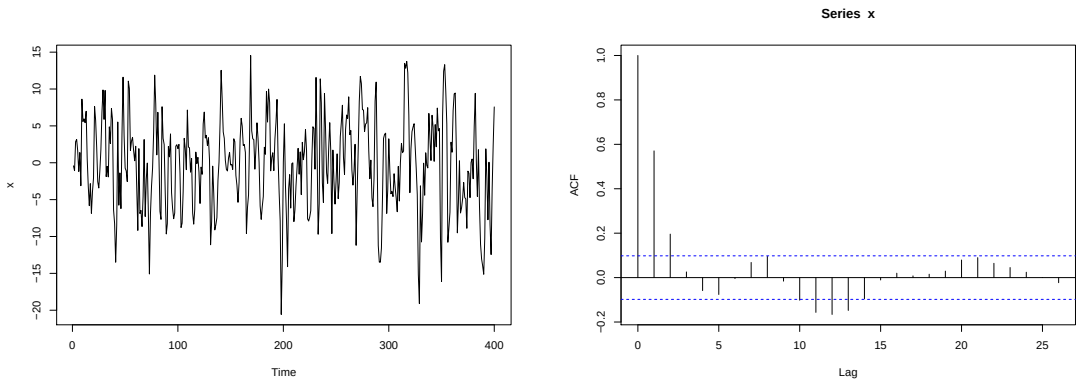
Using the `predict` function, we make predictions for the test sample and plot the predictions:

```
###Predict from the three models
modelar1.pred <- predict(fit_ar1, n.ahead = length_test)$pred
modelma1.pred <- predict(fit_ma1, n.ahead = length_test)$pred
modelarma11.pred <- predict(fit_arma11, n.ahead = length_test)$pred

plot(modelar1.pred,col = "blue", lty = 2,lwd=2, ylab="Predictions")
lines(modelma1.pred, col = "black", lty = 3,lwd=2)
lines(modelarma11.pred, col = "red", lty = 1,lwd=2)
legend(x="bottomright", legend=c("AR(1)", "MA(1)", "ARMA(1,1)"),
       col=c("blue", "black", "red"), lty=c(2,3,1), lwd=2, cex=1.1)

#Plot the forecast from the ARMA(1,1) process including the prediction
    intervals (up to 99\% level)
plot(forecast(fit_arma11,h=length_test,fan=TRUE))
lines(x, col="black",lwd=1)

#Alternatively, you can specify the level of the prediction interval
plot(forecast(fit_arma11,h=length_test,level=0.95))
lines(x, col="black",lwd=1)
```



(a) Realisation of an ARMA(1,1) process

(b) Sample acf of the ARMA(1,1) realisations

Figure 12.1: Simulated sample path of length $n = 400$ of an ARMA(1,1) process with $N(0, 5^2)$ white noise and parameters $\phi = 0.5$, $\theta = 0.3$.

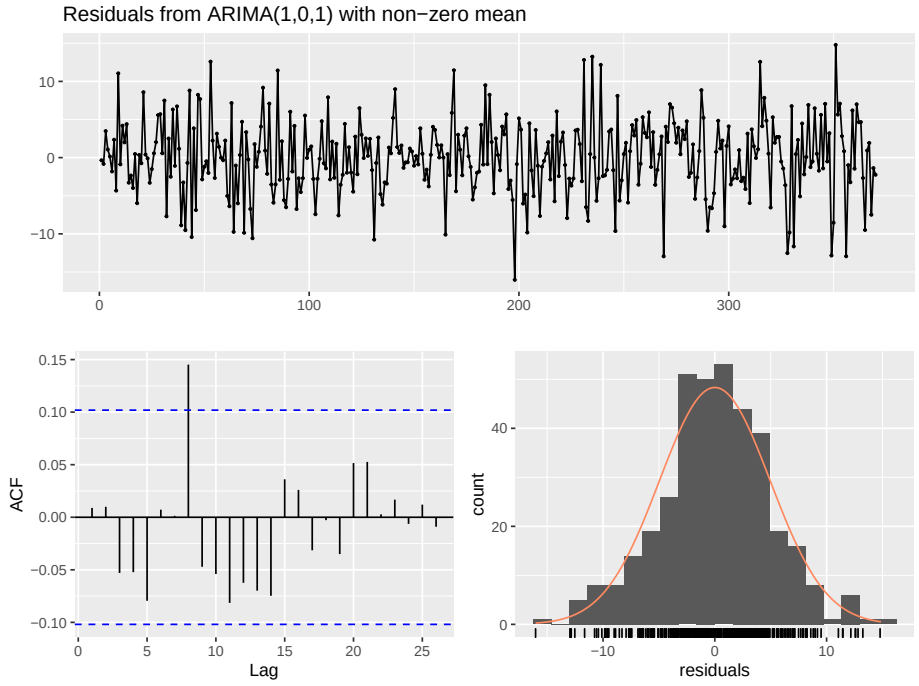


Figure 12.2: Residual diagnostics of a fitted ARMA(1,1) model based on the first 370 observations from our sample.

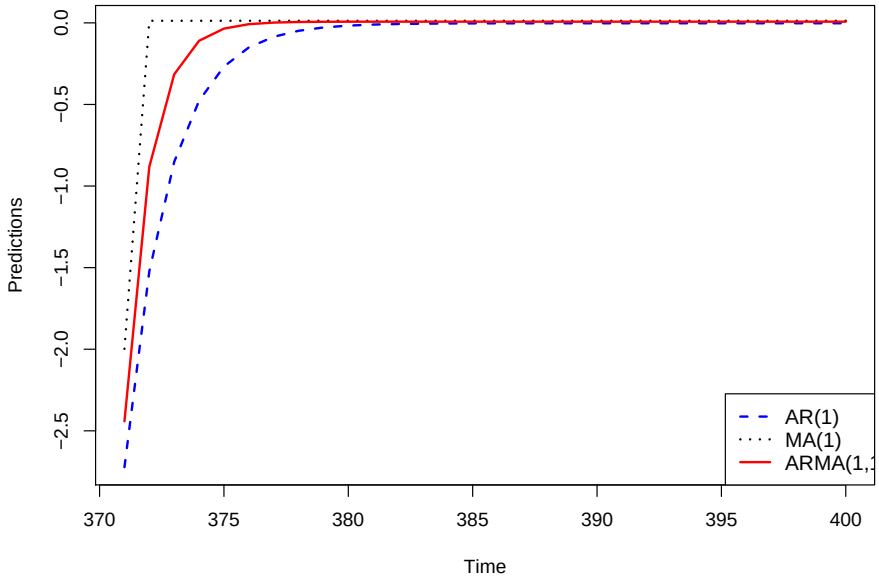


Figure 12.3: Comparing the 1-30 step ahead predictions of an AR(1), MA(1) and ARMA(1,1) model.

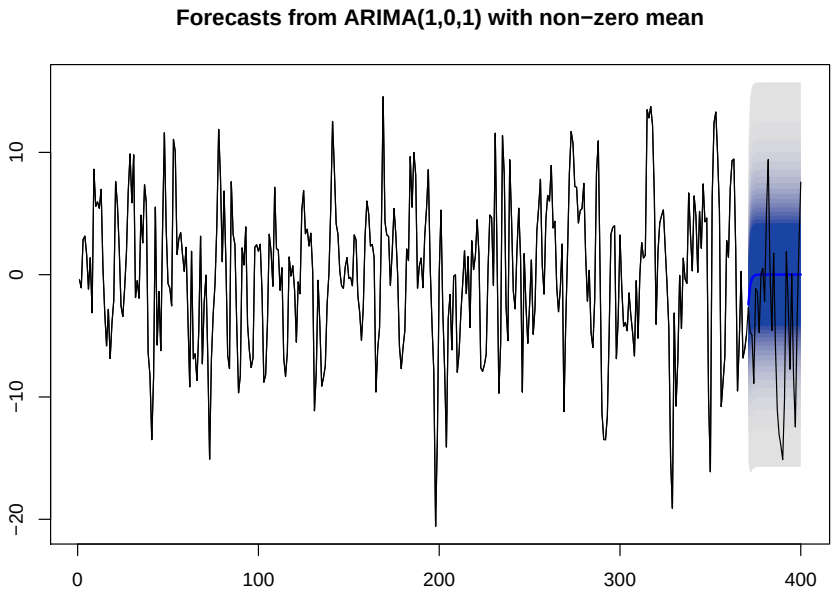


Figure 12.4: Graph of the 1-30 step ahead predictions from the fitted ARMA(1,1) process. A fan is plotted to indicate the 99% prediction intervals. The true observations are added in black.

12.3 Remarks

We have only touched upon the surface of forecasting time series. There are many other methods which can be used and you can find an excellent introduction in the online textbook Hyndman & Athanasopoulos (2018). Note also that topics like *forecast evaluation* and *probabilistic forecasting* are of utmost importance in applications.

Lecture 13

Lecture 11-12: A risk perspective: Value at risk and expected shortfall

13.1 Value at risk

Value at risk (VaR) (despite various shortcomings) is still a very widely used risk measure in financial risk management. We will discuss how VaR can be estimated and we also introduce a related risk measure called expected shortfall (ES). Suppose we have a financial portfolio consisting of one or more risky financial assets. In addition, let us fix a time horizon $\Delta > 0$ say. We associate a loss L with the portfolio whose cumulative distribution function is given by $F_L(l) = \mathrm{P}(L \leq l)$. The intuition behind the definition of the value at risk is to choose a maximum loss which is very unlikely to be exceeded. Let's formalise this idea.

Definition 62. Let $\alpha \in (0,1)$ denote a confidence level. The value at risk of the portfolio at confidence level α is the α-quantile of the loss distribution, denoted by VaR_α.

Typically, we choose $\alpha = 0.95$ or $\alpha = 0.99$.

Adapting the definition of a quantile introduced earlier (Definition 37): Let $0 < \alpha < 1$, then the VaR_α of a random variable L is defined as any real value satisfying both

$$\mathrm{P}(L \geq VaR_\alpha) \geq 1 - \alpha \text{ and } \mathrm{P}(L \leq VaR_\alpha) \geq \alpha.$$

Note that in the case when F_L is continuous and strictly increasing, then the VaR can be expressed in terms of the inverse of the cumulative distribution function: $VaR_\alpha = F_L^{-1}(\alpha)$.

Example 63. Suppose that $L \sim N(\mu, \sigma^2)$. For a confidence level $\alpha \in (0,1)$, we have

$$VaR_\alpha = \mu + \sigma\Phi^{-1}(\alpha).$$

Here $\Phi^{-1}(\alpha)$ denotes the αth quantile of the standard normal distribution. We can now verify that the VaR_α satisfies our definition: For this note that $L \sim N(\mu, \sigma^2)$ implies that $\frac{L-\mu}{\sigma} \sim N(0,1)$. Then

$$\begin{aligned}\mathrm{P}(L \geq VaR_\alpha) = \mathrm{P}\left(L \geq \mu + \sigma\Phi^{-1}(\alpha)\right) &= \mathrm{P}\left(\frac{L-\mu}{\sigma} \geq \Phi^{-1}(\alpha)\right) \\ &= 1 - \mathrm{P}\left(\frac{L-\mu}{\sigma} \leq \Phi^{-1}(\alpha)\right) = 1 - \Phi(\Phi^{-1}(\alpha)) = 1 - \alpha,\end{aligned}$$

and

$$\mathrm{P}(L \leq VaR_\alpha) = \mathrm{P}(L \leq \mu + \sigma\Phi^{-1}(\alpha)) = \mathrm{P}\left(\frac{L-\mu}{\sigma} \leq \Phi^{-1}(\alpha)\right) = \Phi(\Phi^{-1}(\alpha)) = \alpha.$$

In financial applications, we often find that the empirical loss distributions have heavier tails than suggested by the normal distribution. Hence let us study an example of a Student t loss distribution next:

Example 64. Suppose that $(L-\mu)/\sigma$ follows a standard Student t distribution with $\nu > 0$ degrees of freedom, i.e. the corresponding probability density[1] is given by

$$f_{(L-\mu)/\sigma}(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \text{ for } x \in \mathbb{R}.$$

For a confidence level $\alpha \in (0,1)$, we have

$$VaR_\alpha = \mu + \sigma t_\nu^{-1}(\alpha),$$

where $t_\nu^{-1}(\alpha)$ denotes the αth quantile of the standard Student t distribution.

While the VaR is widely used in practice, there are important shortcomings you should be aware of when computing the VaR:

- Since the VaR_α is just the quantile of the loss distribution, it gives you no information on the *severity* of the loss which happens with a probability smaller than $1-\alpha$.
- The VaR is not subadditive, which means that if you have two portfolios with losses L_1 and L_2 the VaR of the aggregated portfolio loss $L = L_1 + L_2$ is not necessarily bounded by the individual VaRs, i.e. we do not necessarily have that

  $$VaR_\alpha(L) \leq VaR_\alpha(L_1) + VaR_\alpha(L_2).$$

  This is counter-intuitive to the fact that we expect the risk of a portfolio to decrease when we diversify the assets.
- When computing the VaR (and other risk measures), there is significant model risk. What if the normal/Student t assumption is wrong?
- You also need to be careful about which confidence level to choose in the VaR computations. Do you have enough observations to be able to reliably estimate extreme quantiles? You might need to apply tools from extreme value theory in some applications.

13.2 Expected shortfall

We mentioned above that the value at risk does not tell you anything about the severity of a highly unlikely loss. This shortcoming is addressed in the definition of another risk measure called expected shortfall (ES).

Definition 65. Consider a loss L with $\mathrm{E}(|L|) < \infty$. Then the expected shortfall (ES) at confidence level $\alpha \in (0,1)$ is defined as

$$ES_\alpha = \frac{1}{1-\alpha}\int_\alpha^1 VaR_u(L)du.$$

We observe that the ES computes the average over all VaR with confidence levels $u \geq \alpha$. Hence it does contain information on the severity of potential losses. Also, one can show that $ES_\alpha \geq VaR_\alpha$.

Remark 66. *For those of you who are familiar with the concept of a* conditional expectation *note that one can show that the ES is the expected loss given that the VaR was exceeded, i.e.* $ES_\alpha = \mathrm{E}(L|L \geq VaR_\alpha)$.

Example 67. Suppose that $L \sim N(\mu, \sigma^2)$. Then, for $\alpha \in (0,1)$, we get

$$ES_\alpha = \frac{1}{1-\alpha}\int_\alpha^1 VaR_u(L)du = \frac{1}{1-\alpha}\int_\alpha^1 (\mu + \sigma\Phi^{-1}(u))du = \cdots = \mu + \sigma\frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}.$$

[1] Note that the Gamma function is defined as $\Gamma(x) = \int_0^\infty u^{x-1}e^{-x}dx$ for $x > 0$.

A similar result can be derived in the case of a Student t distributed loss function.

Exercise 68. *Fill in the missing steps in the computations of Example 67.*

Solution: Since $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2})$, we have $\phi'(z) = -z\phi(z)$. Now, a change of variable argument with $u = \Phi(z), du = \phi(z)dz$ and an application of the reverse chain rule leads to

$$\int_\alpha^1 \Phi^{-1}(u)du = \int_{\Phi^{-1}(\alpha)}^\infty z\phi(z)dz = -\phi(z)|_{\Phi^{-1}(\alpha)}^\infty = \phi(\Phi^{-1}(\alpha)).$$

13.3 Case study

For an empirical example, we revisit the time series of the Google stock price we studied in yesterday's problem class: Recall that the data consists of daily closing prices (in USD) for the Google stock price recorded over a two year time period from 01 January 2020 to 22 June 2022. The dataset has been downloaded from the Yahoo Finance website https://uk.finance.yahoo.com/quote/GOOG/history?p=GOOG.

Suppose that the daily closing price is denoted by S_t. We compute the daily log returns as follows:

$$r_t = r(t,1) = \log(S_{t+1}) - \log(S_t).$$

We then define the process of the losses as $L_t = -r_t$. Recall that in the problem class we found that the NIG distribution describes the log returns best. Hence we compute the VaR and ES based on the estimated NIG distribution and for comparison also for the estimated Gaussian distribution: First we read in the data and compute the losses and plot them:

```
library(ghyp)
GOOG <- read.csv("GOOG.csv", header = TRUE, sep = ",", dec = ".")
GOOG_prices <-GOOG$Close #daily closing prices
```

As a side remark, we can check what kind of ARMA model could describe the prices. We note that the `auto.arima` function suggests a random walk model:

```
auto.arima(GOOG_prices)
```

Next, let us compute the log returns and then the daily ($\Delta = 1$) losses.

```
GOOG_logreturns <-diff(log(GOOG_prices))
#Compute the losses
GOOG_losses <- -GOOG_logreturns
plot(GOOG_losses, ylab="Google losses",type="l")
```

Next, we compare the model fit for various models from the GH class and then save the parameters for both the NIG and the Gaussian model.

```
a<-stepAIC.ghyp(GOOG_losses,silent=TRUE)
a

#Fit a VG distribution:
GOOG_VGfit <- fit.VGuv(GOOG_losses, silent = TRUE)
#Fit an NIG distribution:
GOOG_NIGfit <- fit.NIGuv(GOOG_losses, silent = TRUE)
#Fit a Gaussian distribution:
GOOG_gaussfit <- fit.gaussuv(GOOG_losses)

#Alternatively, you can get the fitted model parameters from the auto.
    arima function
```

```
#For the asymmetric VG, use:
a$all.models[[4]]
#For the asymmetric NIG, use:
a$all.models[[3]]
#For the Gaussian model, use:
a$all.models[[11]]
```

Next, we compute the VaR for various confidence levels for both the Gaussian and the NIG model and compare them:

```
###Compute Value at Risk
VaRlevels <- 1-c(0.1, 0.05, 0.01, 0.001)
#Compute the NIG and Gaussian quantiles:
GOOG_VG_VaR <- qghyp(VaRlevels,GOOG_VGfit)
GOOG_NIG_VaR <-qghyp(VaRlevels,GOOG_NIGfit)
GOOG_gauss_VaR <-qnorm(VaRlevels, mean=coef(GOOG_gaussfit)$mu, sd=coef
    (GOOG_gaussfit)$sigma)
#Plot the Gaussian, NIG  and VG quantiles:
barplot(rbind(GOOG_gauss_VaR, GOOG_NIG_VaR, GOOG_VG_VaR), beside =
    TRUE,
names.arg = paste(100 * VaRlevels, "percent"), density=c(10,20,30) ,
    angle=c(0,45,90), col = c("red", "blue", "green"),
ylab = "VaR (loss distribution)", xlab = "Level")
legend("topleft", c("Gaussian", "NIG", "VG"), density=c(10,20,30) ,
    angle=c(0,45,90),fill=c("red","blue", "green"))
```

We do the same analysis for the corresponding expected shortfalls:

```
## Confidence levels for expected shortfalls
ESlevels <- VaRlevels
GOOG_VG_ES <- ESghyp(ESlevels, GOOG_VGfit, distr="loss")
GOOG_NIG_ES <- ESghyp(ESlevels, GOOG_NIGfit, distr="loss")
GOOG_gauss_ES <- ESghyp(ESlevels, GOOG_gaussfit, distr="loss")

barplot(rbind(GOOG_gauss_ES, GOOG_NIG_ES, GOOG_VG_ES), beside = TRUE,
names.arg = paste(100 * VaRlevels, "percent"), density=c(10,20,30) ,
    angle=c(0,45,90), col = c("red", "blue", "green"),
ylab = "ES (loss distribution)", xlab = "Level")
legend("topleft", c("Gaussian", "NIG", "VG"), density=c(10,20,30) ,
    angle=c(0,45,90), fill=c("red","blue", "green"))
```

Finally, we explore various graphics to display and compare the VaR and ES for our two models:

```
#You can graphically compare the levels of the VaR and ES for the
    three different fitted distributions using:
par(mfrow=c(2,1))
barplot(rbind(GOOG_gauss_VaR, GOOG_NIG_VaR, GOOG_VG_VaR), beside =
    TRUE,
names.arg = paste(100 * VaRlevels, "percent"), density=c(10,20,30) ,
    angle=c(0,45,90), col = c("red", "blue", "green"),
ylab = "VaR (loss distribution)", xlab = "Level")
legend("topleft", c("Gaussian", "NIG", "VG"),density=c(10,20,30) ,
    angle=c(0,45,90), fill=c("red","blue", "green"))
barplot(rbind(GOOG_gauss_ES, GOOG_NIG_ES, GOOG_VG_ES), beside = TRUE,
names.arg = paste(100 * VaRlevels, "percent"), density=c(10,20,30) ,
    angle=c(0,45,90), col = c("red", "blue", "green"),
```

```
ylab = "ES (loss distribution)", xlab = "Level")
legend("topleft", c("Gaussian", "NIG", "VG"),density=c(10,20,30) ,
    angle=c(0,45,90), fill=c("red","blue", "green"))


#Next we plot the VaR and ES for the VG and Gaussian distributions in
    one picture:
par(mfrow=c(1,1))
barplot(rbind(GOOG_gauss_VaR, GOOG_gauss_ES, GOOG_VG_VaR, GOOG_VG_ES),
     beside = TRUE,
names.arg = paste(100 * VaRlevels, "percent"), density=c(10,20, 30,
    40) , angle=c(0,45,90,135), col = c("red", "blue", "green", "grey")
    ,
ylab = "VaR and ES (loss distribution)", xlab = "Level")
legend("topleft", c("Gaussian VaR", "Gaussian ES", "VG VaR", "VG ES"),
    density=c(10,20, 30, 40) , angle=c(0,45,90,135), fill=c("red","blue
    ", "green", "grey"))

#Finally we indicate the VaR and ES levels w.r.t the empirical density
    :
library(MASS)
truehist(GOOG_losses, main="Empirical density and true histogram of
    the losses", xlab="")
lines(density(GOOG_losses))
abline(v=GOOG_gauss_VaR[3],col="red",lwd=2)
text(GOOG_gauss_VaR[3], 30, "Gaussian VaR 99%", col = "red")
abline(v=GOOG_NIG_VaR[3],col="green",lwd=2)
text(GOOG_NIG_VaR[3], 20, "NIG VaR 99%", col = "green")
abline(v=GOOG_VG_VaR[3],col="grey",lwd=3)
text(GOOG_VG_VaR[3], 20, "VG VaR 99%", col = "grey")
```

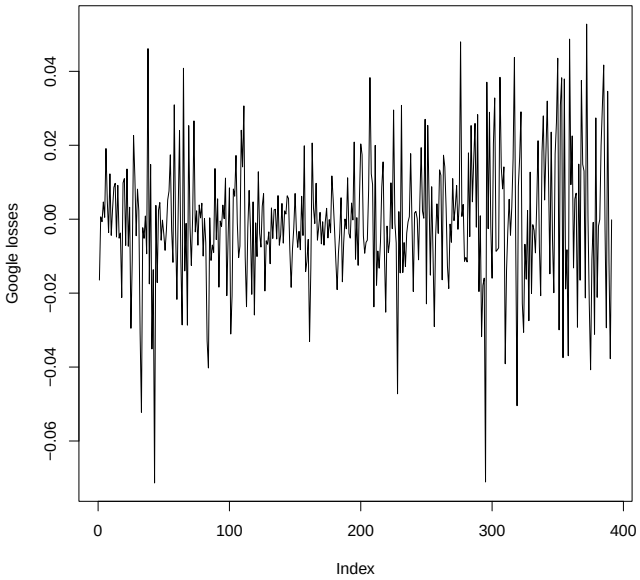
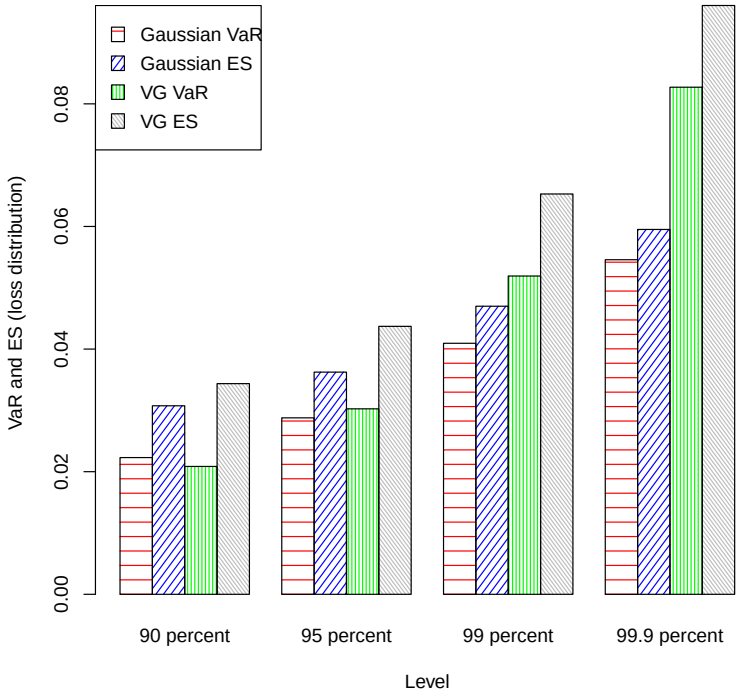
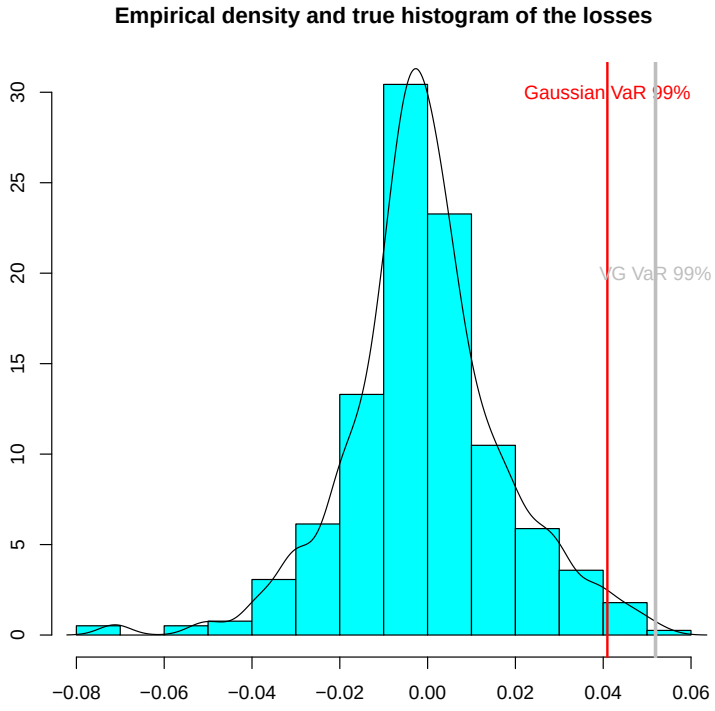


Figure 13.1: Time series of the losses computed based on the times series of the daily Google stock closing prices from 01 January 2020 to 22 June 2022.



(a)



(b)

Figure 13.2: Here we compare the value at risks and expected shortfalls based on the estimated Gaussian and VG distributions. Figure 13.2a depicts the various VaR and ES levels for four possible choices of confidence levels. In Figure 13.2b we display the true histogram of the loss distribution (with the estimated empirical density superimposed) and we highlight the VaR values for the fitted Gaussian and VG distributions.

13.4 Remarks

We have studied a first approach to computing risk measures such as Value at Risk and Expected Shortfall in financial applications. However, it is important to keep in mind that in practical applications we should typically apply a more sophisticated methodology than the one implemented above.

In the above computations, we treated the process of losses $(L_t)_{t\in\mathbb{Z}}$ as being i.i.d. random variables. A more flexible approach is to consider a (stationary) model of the form

$$L_t = \mu_t + \sigma_t Z_t,$$

where (Z_t) is a strict white noise process with zero mean and variance equal to 1. Moreover, we typically assume that μ_t and σ_t are known given the information up to time $t-1$. One can then show that the VaR and ES given the information until time t are given by

$$VaR_\alpha^t = \mu_{t+1} + \sigma_{t+1} VaR_\alpha(Z),$$
$$ES_\alpha^t = \mu_{t+1} + \sigma_{t+1} ES_\alpha(Z).$$

In applications, we could consider an ARMA process for modelling μ_t. In financial applications, we often observe that the volatility is time-varying which suggest a stochastic process model for σ_t as well. There is a very rich class of stochastic volatility models in the literature and if you would like to read more about this you could explore the class of GARCH processes, which are essentially ARMA processes for the squares. Given an ARMA-GARCH model one can then make 1-step (or multistep) ahead predictions and compute the VaR and ES for various different time horizons.

You can find explicit R examples on VaR and ES computations for such more sophisticated models in the QRM Tutorial `https://www.qrmtutorial.org/` associated with the textbook McNeil et al. (2005).

Mock exam questions

The exam will exist of multiple choice questions where you can choose amongst five possible answers. I will list some mock questions below so that you can get an idea of what to expect. Please remember, however, that the mock questions only cover a small part of the material. For the final exam, **all the material covered during the lectures and problem classes is examinable**.

Quiz 1 (Probability quiz)**.** *Consider the experiment where we roll a standard six-sided fair die.*

Figure 13.3: Rolling a fair die.

1. *The sample space associated with this experiment is given by*
   - *a)* $\Omega = \{1, 2, 3, 4, 5, 6\}$.
   - *b)* $\Omega = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$.
   - *c)* $\Omega = \{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$.
   - *d)* $\Omega = \{1, 6\}$.
   - *e) None of the above.*
2. *We set* $A = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$,$B = \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 6\}\}$, $C = \{\{2, 3\}, \{2, 4\}, \{2, 5\}, \{2, 6\}\}$, $D = \{\{3, 4\}, \{3, 5\}, \{3, 6\}, \{4, 5\}, \{4, 6\}, \{5, 6\}\}$. *The set of all subsets of the sample space* Ω *is given by*
   - *a)* $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
   - *b)* $\mathcal{F} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$
   - *c)* $\mathcal{F} = A \cup B \cup C \cup D$.
   - *d)* $\mathcal{F} = \{\emptyset, \Omega\}$.
   - *e) None of the above.*
3. *The probability that the outcome of the experiment is odd is given by*
   - *a)* $\mathrm{P}(1, 3, 5) = 1/6$.
   - *b)* $\mathrm{P}(1, 3, 5) = 1/3$.
   - *c)* $\mathrm{P}(1, 3, 5) = 1/2$.

d) $\mathrm{P}(1,3,5) = 1/4$.

e) *None of the above.*

Quiz 2 (Time series quiz)**. Question 1:**

Suppose you want to plot the correlogram of a time series. Which R function is best suited for the task?

a) *The* `auto.arima` *function from the* `forecast` *package.*

b) *The* `stl` *function from the* `stats` *package.*

c) *The* `acf` *function from the* `stats` *package.*

d) *The* `median` *function from the* `stats` *package.*

e) *The* `window` *function from the* `stats` *package.*

Question 2:

Let $(\epsilon_t)_{t\in\mathbb{Z}}$ *denote a white noise process with zero mean. Which of the following stochastic processes* $(X_t)_{t\in\mathbb{Z}}$ *describes an MA(1) process?*

a) $X_t = \epsilon_t + 0.4\epsilon_{t-1}$.

b) $X_t = 0.3X_{t-1} + \epsilon_t + 0.4\epsilon_{t-1}$.

c) $X_t = \epsilon_t + 0.4\epsilon_{t-1} + 0.2\epsilon_{t-2}$.

d) $X_t = 0.3X_{t-1} + \epsilon_t$.

e) *None of the above.*

13.5 Answers

Quiz 1: 1a, 2e, 3c
Quiz 2: 1c, 2a

Bibliography

Brockwell, P. J. & Davis, R. A. (2016), *Introduction to time series and forecasting*, Springer Texts in Statistics, third edn, Springer, [Cham].

Cont, R. (2001), 'Empirical properties of asset returns: stylized facts and statistical issues', *Quantitative Finance* **1**, 223–236.

Embrechts, P., Klüppelberg, C. & Mikosch, T. (1997), *Modelling extremal events (for insurance and finance)*, Vol. 33 of *Applications of Mathematics (New York)*, Springer-Verlag, Berlin. Corrected Fourth Printing 2003.

Hyndman, R. & Athanasopoulos, G. (2018), *Forecasting: principles and practice, 2nd edition*, OTexts, Melbourne, Australia. OTexts.com/fpp2. Accessed on 11/06/2019.

Luethi, D. & Breymann, W. (2013), *ghyp: A Package on Generalized Hyperbolic Distribution and Its Special Cases*. Vignettes for R package version 1.5.7.
URL: *https://CRAN.R-project.org/package=ghyp*

McNeil, A. J., Frey, R. & Embrechts, P. (2005), *Quantitative risk management (Concepts, techniques and tools)*, Princeton Series in Finance, Princeton University Press, Princeton, NJ.

Tsay, R. S. (2010), *Analysis of financial time series*, Wiley Series in Probability and Statistics, third edn, John Wiley & Sons Inc., Hoboken, NJ.