A Tour of Time Series Analysis with R

James Balamuta, Stéphane Guerrier and Roberto Molinari

2016-08-20

Contents

Preface

This text is designed as an introduction to time series analysis and is used as a support document for the class STAT 429 (Time Series Analysis) given at the University of Illinois at Urbana-Champaign. It preferable to always access the text online rather than a printed to be sure you are using the latest version. The online version so affords additional features over the traditional PDF copy such as a scaling text, variety of font faces, and themed backgrounds. However, if you are in need of a local copy, a **pdf version** is also available.

This document is under active development and as a result is likely to contains many errors. As Montesquieu puts it:

> "*La nature semblait avoir sagement pourvu à ce que les sottises des hommes fussent passagères, et les livres les immortalisent.*"

If you notice any errors, we would be grateful if you would let us know. To let us know about the errors, there are two options available to you. The first and subsequently the fastest being if you are familiar with GitHub and know RMarkdown, then make a pull request and fix the issue yourself!. Note, in the online version, there is even an option to automatically start the pull request by clicking the edit button in the top-left corner of the text.

The second option, that will have a slightly slower resolution time is to send an email to `balamut2 AT illinois DOT edu` that includes: the error and a possible revision. Please put in the subject header: `[TTS]`.

Bibliographic Note

This text is heavily inspired by the following three execellent references:

1. "*Time Series Analysis and Its Applications*", Third Edition, Robert H. Shumway & David S. Stoffer.
2. "*Time Series for Macroeconomics and Finance*", John H. Cochrane.
3. "*Cours de Séries Temporelles: Théorie et Applications*", Volume 1, Arthur Charpentier.

Rendering Mathematical Formulae

Throughout the book, there will be mathematical symbols used to express the material. Depending on the version of the book, there are two different render engines.

- For the online version, the text uses MathJax to render mathematical notation for the web. In the event the formulae does not load for a specific chapter, first try to refresh the page. 9 times out of 10 the issue is related to the software library not loading quickly.
- For the pdf version, the text is built using the recommended AMS LaTeX symbolic packages. As a result, there should be no issue displaying equations.

An example of a mathematical rendering capabilities would be given as:

$$a^2 + b^2 = c^2$$

R Code Conventions

The code used throughout the book will predominately be `R` code. To obtain a copy of `R`, go to the Comprehensive R Archive Network (CRAN) and download the appropriate installer for your operating system.

When `R` code is displayed it will be typeset using a `monospace` font with syntax highlighting enabled to ensure the differentiation of functions, variables, and so on. For example, the following adds 1 to 1

```
a = 1L + 1L
a
```

Each code segment may contain actual output from `R`. Such output will appear in grey font prefixed by `##`. For example, the output of the above code segment would look like so:

```
## [1] 2
```

Alongside the PDF download of the book, you should find the R code used within each chapter.

License

Figure 1: This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Chapter 1

Introduction

Prévoir consiste à projeter dans l'avenir ce qu'on a perçu dans le passé. Henri Bergson

After reading this chapter you will be able to:

- Describe what a *time series* is.
- Perform explore data analysis on time series data.
- Evaluate different characteristics of a time series.
- Classify basic time series models by equation and plots.
- Manipulate a time series equation using *backsubstitution*.

1.1 Time Series

Generally speaking a *time series* (or stochastic process) corresponds to set of "repeated" observations of the same variable such as price of a financial asset or temperature in a given location. In terms of notation a time series is often written as

$$(X_1, X_2, ..., X_n) \quad \text{or} \quad (X_t)_{t=1,...,n}\,.$$

The time index t is contained within either the set of reals, $\mathbb{R}$, or integers, $\mathbb{Z}$. When $t \in \mathbb{R}$, the time series becomes a *continuous-time* stochastic process such a Brownian motion, a model used to represent the random movement of particles within a suspended liquid or gas, or an ElectroCardioGram (ECG) signal, which corresponds to the pulpitations of the heart. However, within this text, we will limit ourselves to the case the later case. That is, the focus will be on cases where $t \in \mathbb{Z}$ better known as *discrete-time* processes. *Discrete-time* processes are where a variable is measured sequentially at fixed and equally spaced intervals in time akin to 1.1. This implies that we will assume two tenents:

1. t is not random e.g. the time at which each observation is measured is known, and
2. the time between two consequtive observation is constant.

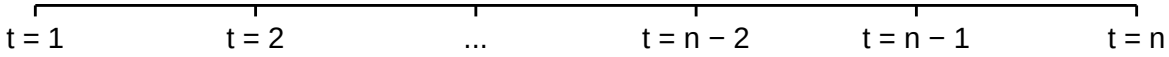


Figure 1.1: Discrete-time can be thought of as viewing a number line with equally spaced points.

Moreover, the term "time series" can also represent a probability model for set of observations. For example, one of the fundamental probability models used in time series analysis is called a *white noise* process and is defined as

$$W_t \overset{iid}{\sim} N(0, \sigma^2).$$

This statement simply means that (W_t) is normally distributed and independent over time. This model may appear to be dull but as we will see it is a crucial component to constructing intricate and riveting models. Unlike the white noise process, time series are typically *not* independent over time. Suppose that the temperature in Champaign is unusually low, then it is reasonable to assume that tomorrow's temperature will also be low. Indeed, such behavior would suggest the existence of a dependency over time. The time series methods we will discuss in this text consists of parametric models used to characterize (or at least approximate) the joint distribution of (X_t). Often, time series models can be decomposed into two components the first of which is what we call a *signal*, say (Y_t), and the second component is a *noise*, say (W_t), leading to the model

$$X_t = Y_t + W_t.$$

Typically, we have $E[Y_t] \neq 0$ while $E[W_t] = 0$ (although we may have $E[W_t|W_{t-1}, ..., W_1] \neq 0$). Such models impose some parametric structure which represents a convenient and flexible way of studying time series as well as a means to evalute *future* values of the series through forecasting. As we will see, predicting future values is one of the main aspects of time series analysis. However, making predictions is often a daunting task or as famously stated by Nils Bohr:

"*Prediction is very difficult, especially about the future.*"

There are plenty of examples predictions which were revealed to be completely erroneous. For example, Irving Fisher, Professor of Economics at Yale University, famously predicted three days before the 1929 crash:

"*Stock prices have reached what looks like a permanently high plateau*".

Another example is Thomas Watson, president of IBM, who said in 1943:

"*I think there is a world market for maybe five computers.*"

1.2 Exploratory Data Analysis for Time Series

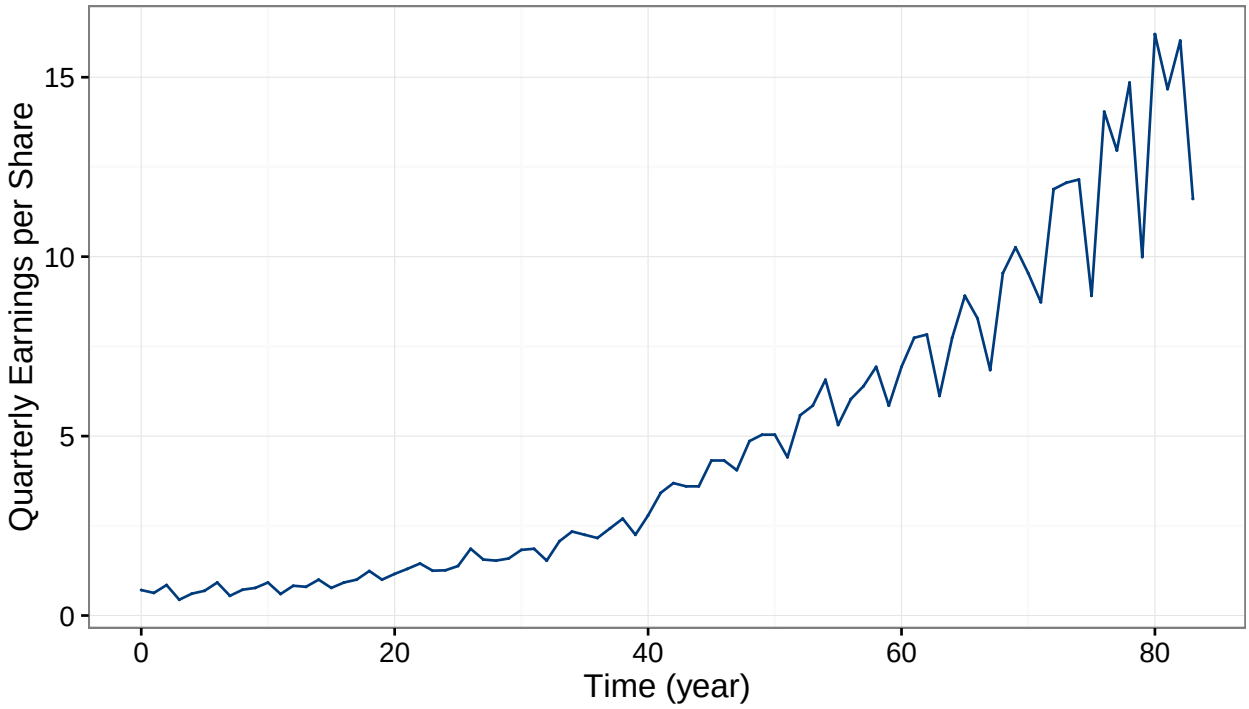
When dealing with relatively small time series (e.g. a few thousands), it is often useful to look at a graph of the original data. Such graphs can be informative to "detect" some features of a time series such as trends and the presence of outliers.

Indeed, a trend is typically deemed present in a time series when the data exhibit some form of long term increase or decrease or combination of increases or decreases. Such trends could be linear or non-linear and represent a important part of the "signal" of a model. Here are few examples of non-linear trends:

1. **Seasonal trends** (periodic): These are the cyclical patterns which repeat after a fixed/regular time period. This could be due to business cycles (e.g. bust/recession, recovery).
2. **Non-seasonal trends** (periodic): These patterns cannot be associated to seasonal variation and can for example to external variable. For example, impact of economic indicators on stock returns. Note that such trends are often hard to detect based on a graphical analysis of the data.

3. **"Other" trends**: These trends have typically no regular patterns and are over a segment of time, known as a "window", that change the statistical properties of a time series. A common example of such trends corresponds to vibrations observed before, during and after an earthquake.

Example: A traditional example of a time series is the quarterly earnings of the company Johnson and Johson. In the figure below, we present these earnings between 1960 and 1980:

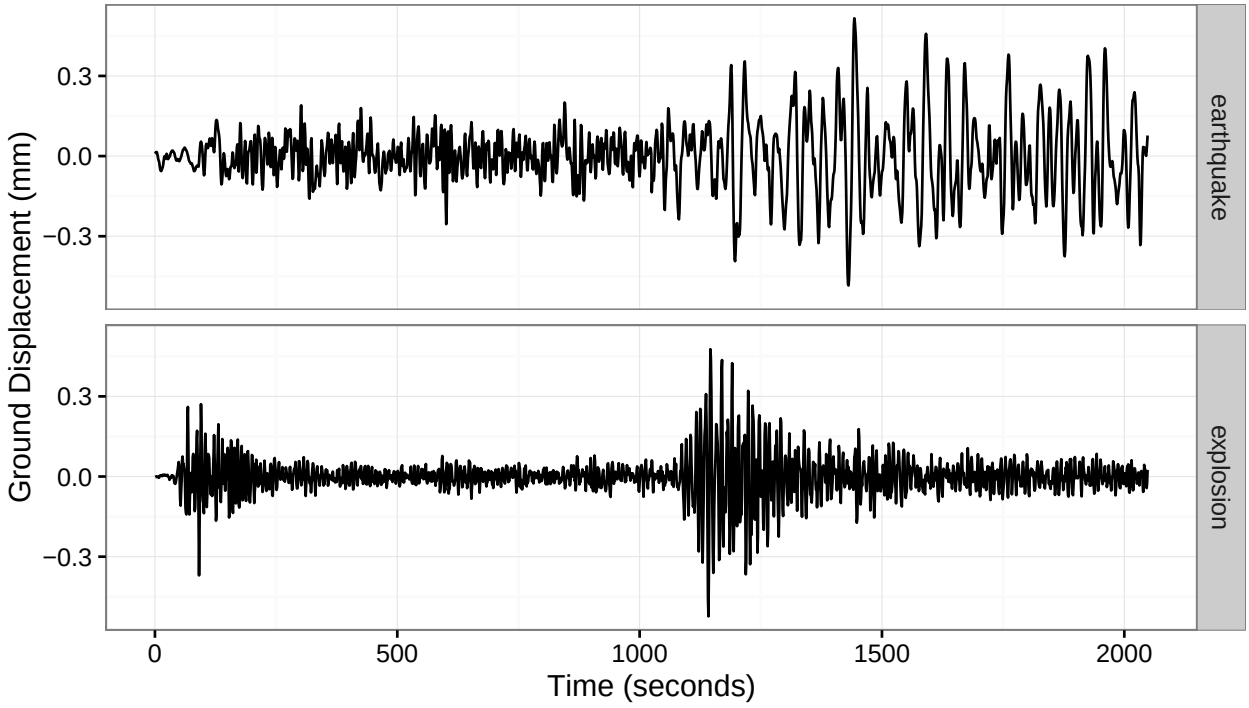


One trait that the graph makes evident is the data contains a non-linear increasing trend as well as a yearly seasonal component. In addition, one can note that the *variability* of the data seems to increase with time. Being able to make such observations provides actionable information to select a suitable models for the data.

Moreover, when observing "raw" time series data it is also interesting to evaluate if some the following phenomenon occur:

1. **Change in Means:** Does the mean of the process shift over time?
2. **Change in Variance:** Does the variance of the process evolves with time?
3. **Change in State:** Does the time series appear to change between "states" having distinct statistical properties?
4. **Outliers** Does the time series contain some "extreme" observations? Note that this is typically difficult to assess visually.

Example: In the figure below, we present an example of displacement recorded during an earthquake as well as explosion.

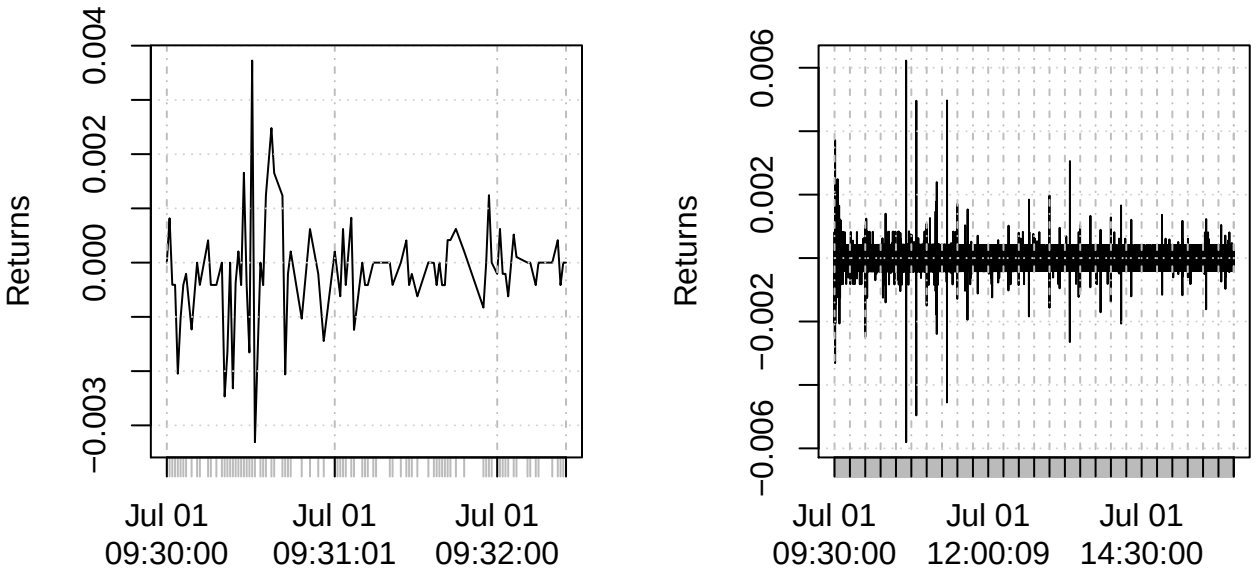


From the graph, it can be observed that the statistical properties of the time series appear to change over time. For instance, the variance of the time series shifts at around $t = 1150$ for both series. The shift in variance also opens "windows" where there appears to be distinct states. In the case of the explosion data this is particularly relevant around $t = 50, \cdots, 250$ and then again from $t = 1200, \cdots, 1500$. Even within these windows, there are "spikes" that could be considered as outliers most notably around $t = 1200$ for explosion series.

Next, we consider an example about high-frequency finance to illustate to limitation our current framework.

Example: The figure below present the returns (i.e. informally speaking the changes in price) for Starbuck's stock on the first of July 2011 during about 150 seconds (left panel) and about 400 minutes (right panel).

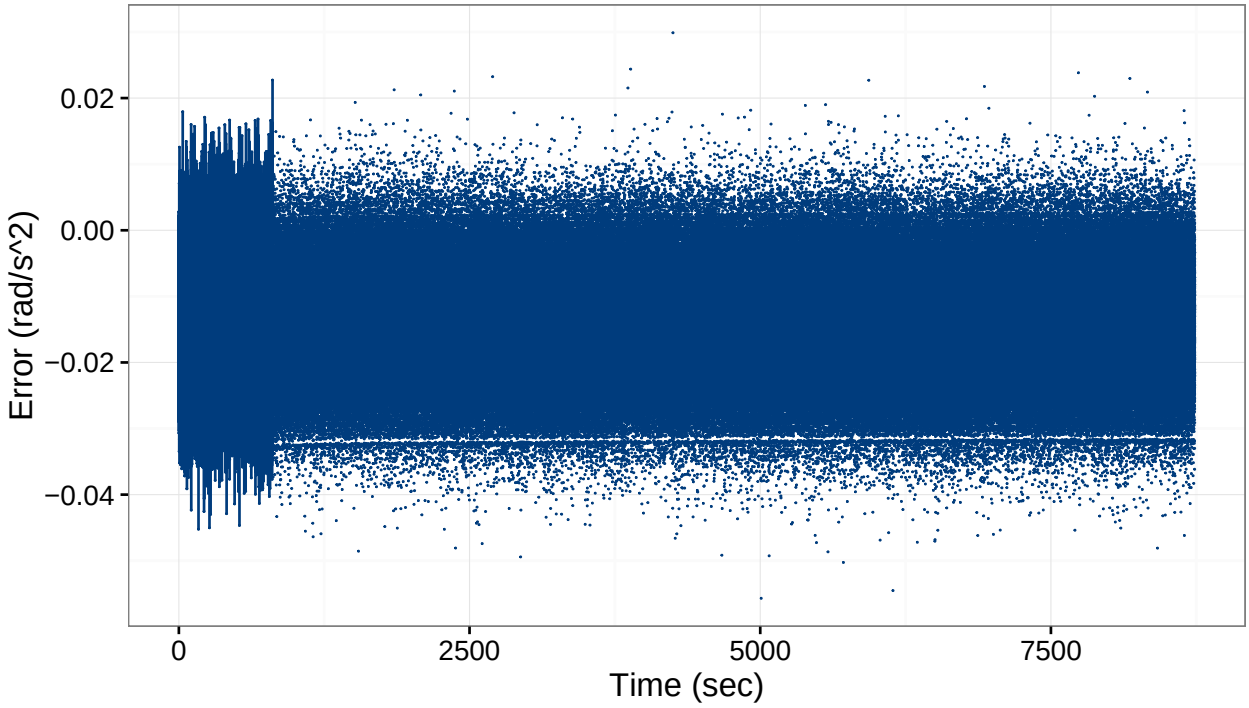
```
# Load packages
library(timeDate)

# Load "high-frequency" Starbucks returns for Jul 01 2011
data(sbux.xts, package = "highfrequency")

# Plot returns
par(mfrow = c(1,2))
plot(sbux.xts[1:89], main = " ", ylab = "Returns")
plot(sbux.xts, main = " ", ylab = "Returns")
```



It can be observed on the left panel that points are not equally spaced. Indeed, in high-frequency data interval between two points is typically not constant and is, even worse, a random variable. This implies that when a new observation will be available is in general unknown. On the right panel, one can observe that the variability of the data seems to change during the course of the trading day. Such phenomenon is well known in the finance community as a lot variation occurs at the start (and the end) of the day while the middle of the day is associated with small changes. Moreover, clear extreme observations can also be noted in this graph at around 09:30:34.

Finally, ...

```r
# Load packages
library(gmwm)

# Load IMU data
data(imu6, package = "imudata")
Xt = gts(imu6[,1], name = "Gyroscope data", unit = "sec", freq = 100)

# Plot gryoscope data
autoplot(Xt) + ylab("Error (rad/s^2)")
```



1.3 Basic Time Series Models

In this section, we introduce some simple time series models. Before doing so it is useful to define Ω_t as all the information avaiable up to time $t-1$, i.e.

$$\Omega_t = (X_{t-1}, X_{t-2}, ..., X_0)\,.$$

As we will see this compact notation is quite useful.

1.3.1 White noise processes

The building block for most time series models is the Gaussian white noise process, which can be defined as

$$W_t \overset{iid}{\sim} N\left(0, \sigma_w^2\right).$$

This definition implies that:

1. $E[W_t|\Omega_t] = 0$ for all t,
2. $\text{cov}\left(W_t, W_{t-h}\right) = \mathbf{1}_{h=0}\,\sigma^2$ for all t, h.

Therefore, this process present an absence of temporal (or serial) dependence and is homoskedastic (i.e it has a constant variance). This definition can be generalized in two sorts of processes, the *weak* and *strong* white noise. The process (W_t) is a weak white noise if

1. $E[W_t] = 0$ for all t,
2. $\text{var}\left(W_t\right) = \sigma^2$ for all t,
3. $\text{cov}\left(W_t, W_{t-h}\right) = 0$, for all t, and for all $h \neq 0$.

Note that this definition does not imply that W_t and W_{t-h} are independent (for $h \neq 0$) but simply uncorrelated. However, the notion of indepence is used to define a *strong* white noise as

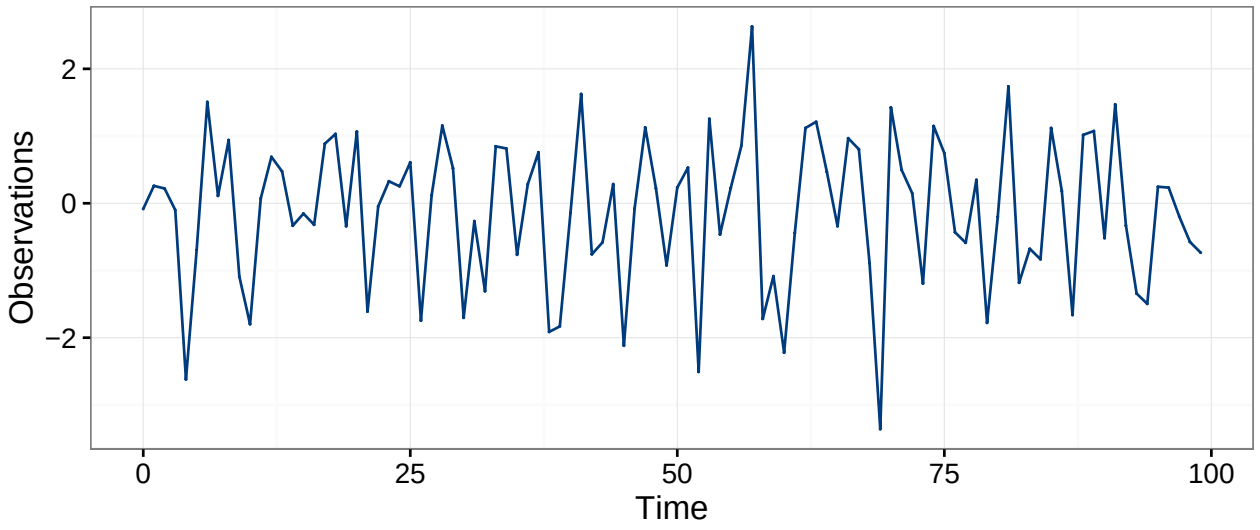
1. $E[W_t] = 0$ and $\text{var}(W_t) = \sigma^2 < \infty$, for all t,
2. $F(W_t) = F(W_{t-h})$, for all t, h (where $F(W_t)$ denotes the distribution of W_t),
3. W_t and W_{t-h} are independent for all t and for all $h \neq 0$.

It is clear from these definitions that if a process is a strong white noise it is also a weak white noise. However, the converse is not true a shown in the following example:

Example: Let $X_t \overset{iid}{\sim} F_t$, where F_t denote a Student distribution with t degrees of freedom. Such process is a weak but not a strong white noise.

The code below presents an example of how to simulate a Gaussian white noise process

```
# This code simulates a gaussian white noise process
n = 100                                  # process length
sigma2 = 1                               # process variance
Xt = gen.gts(WN(sigma2 = sigma2), N = n)
plot(Xt)
```

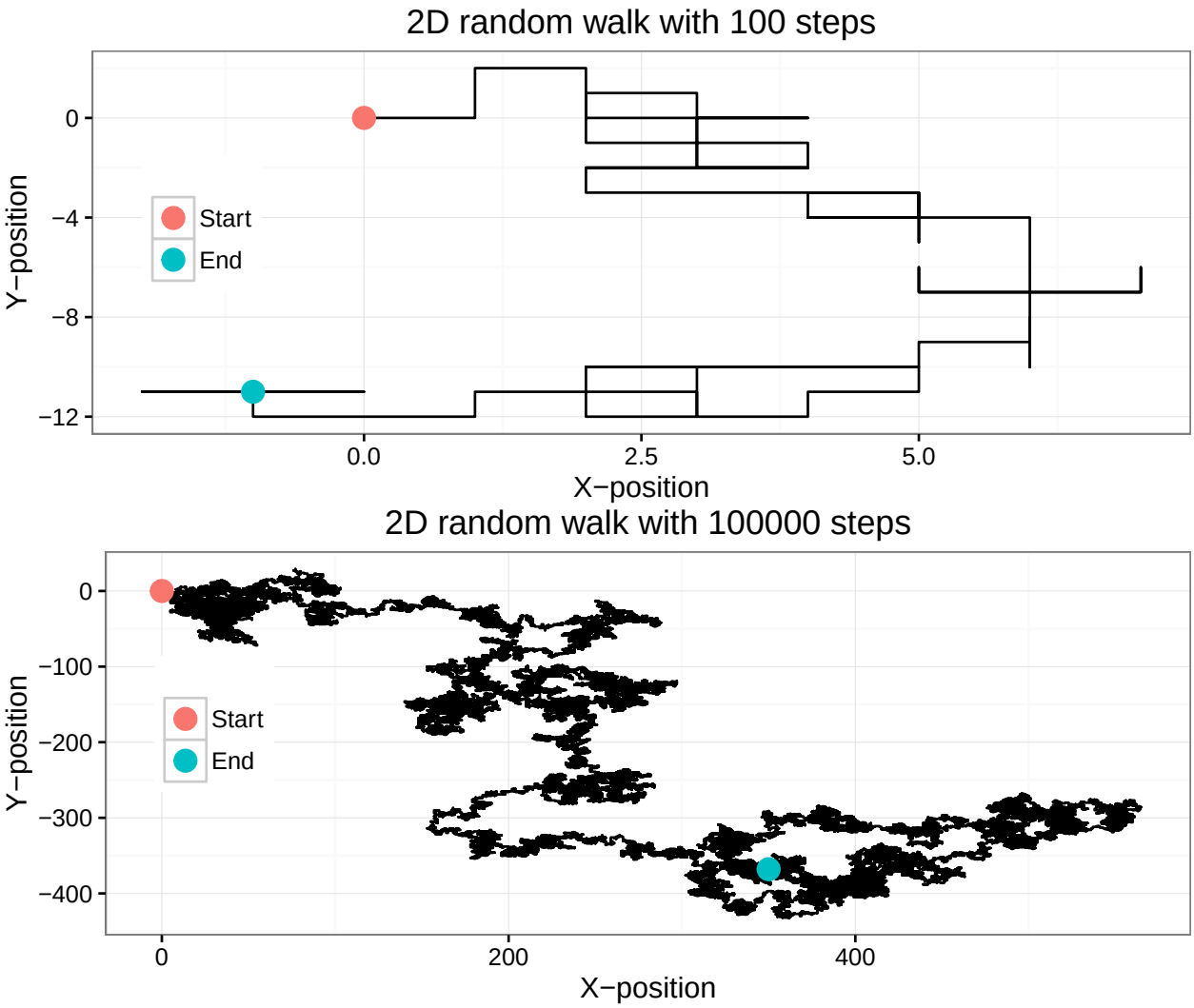


1.3.2 Random Walk Processes

The term *random walk* was first introduce by Karl Pearson in the early 19 hunders. As for the white noise, there exist a large range of random walk processes. For example, one of the simplest form of random walk are be explained as follows: suppose that you are walking on campus and your next step can either be on your left, your right, forward or backward (each with equal probability). Two realizations of such processes are represented below:

```
# Function computes direction random walk moves
RW2dimension = function(steps = 100){
  # Initial matrix
  step_direction = matrix(0, steps+1, 2)

  # Start random walk
  for (i in seq(2, steps+1)){
```

```
    # Draw a random number from U(0,1)
    rn = runif(1)

    # Go right if rn \in [0,0.25)
    if (rn < 0.25) {step_direction[i,1] = 1}

    # Go left if rn \in [0.25,0.5)
    if (rn >= 0.25 && rn < 0.5) {step_direction[i,1] = -1}

    # Go forward if rn \in [0.5,0.75)
    if (rn >= 0.5 && rn < 0.75) {step_direction[i,2] = 1}

    # Go backward if rn \in [0.75,1]
    if (rn >= 0.75) {step_direction[i,2] = -1}
  }

  # Cumulative steps
  position = data.frame(x = cumsum(step_direction[, 1]),
                        y = cumsum(step_direction[, 2]))

  # Mark start and stop locations
  start_stop = data.frame(x = c(0, position[steps+1, 1]),
                          y = c(0, position[steps+1, 2]),
                          type = factor(c("Start","End"),
                                        levels = c("Start","End")))

  # Plot results
  ggplot(mapping = aes(x = x, y = y)) +
    geom_path(data = position) +  # Mimics type = 'l'
    geom_point(data = start_stop, aes(color = type), size = 4) +
    theme_bw() +
    labs(
      x = "X-position",
      y = "Y-position",
      title = paste("2D random walk with", steps, "steps"),
      color = ""
    ) + theme(legend.position = c(0.1, 0.55))
}

# Plot 2D random walk with 10^2 and 10^5 steps
set.seed(2)

RW2dimension(steps = 10^2)
RW2dimension(steps = 10^5)
```



Such processes inspired Karl Pearson's famous quote that

> "*the most likely place to find a drunken walker is somewhere near his starting point.*"

Empirical evidence of this phenomenon is not too hard to find on a Friday night in Champaign. In this class, we only consider one very specific form of randon walk, namely the Gaussian random walk which can be defined as:

$$X_t = X_{t-1} + W_t,$$

where W_t is a Gaussian white noise and with initial condition $X_0 = c$ (typically $c = 0$). This process can be expressed differently by *backsubstitution* as follows:

$$\begin{aligned} X_t &= X_{t-1} + W_t \\ &= (X_{t-2} + W_{t-1}) + W_t \\ &= \vdots \\ X_t &= \sum_{i=1}^{t} W_i + X_0 = \sum_{i=1}^{t} W_i + c \end{aligned}$$

The code below presents an example of how to simulate a such process

1.3.3 Autoregressive Process of Order 1

An autoregressive process of order 1 or AR(1) is a generalization of both the white noise and random walk process which are both special case of an AR(1). A (Gaussian) AR(1) process can be defined as

$$X_t = \phi X_{t-1} + W_t,$$

where W_t is a Gaussian white noise. Clearly, an AR(1) with $\phi = 0$ is a Gaussian white noise and when $\phi = 1$ the process becomes a random walk.

Remark: We generally assume that an AR(1) (as well as other time series models) have zero mean. The reason for this assumption is only to simplfy the notation but it is easy to consider an AR(1) process around an arbitrary mean μ, i.e.

$$(X_t - \mu) = \phi\,(X_{t-1} - \mu) + W_t,$$

which is of course equivalent to

$$X_t = (1 - \phi)\,\mu + \phi X_{t-1} + W_t.$$

Thus, we will generally onyl work with zero mean processes since adding means is simple.

Remark: An AR(1) is in fact a linear combination of the past realisations of the white noise W_t. Indeed, we have

$$\begin{aligned} X_t &= \phi_t X_{t-1} + W_t = \phi\,(\phi X_{t-2} + W_{t-1}) + W_t \\ &= \phi^2 X_{t-2} + \phi W_{t-1} + W_t = \phi^t X_0 + \sum_{i=0}^{t-1} \phi^i W_{t-i}. \end{aligned}$$

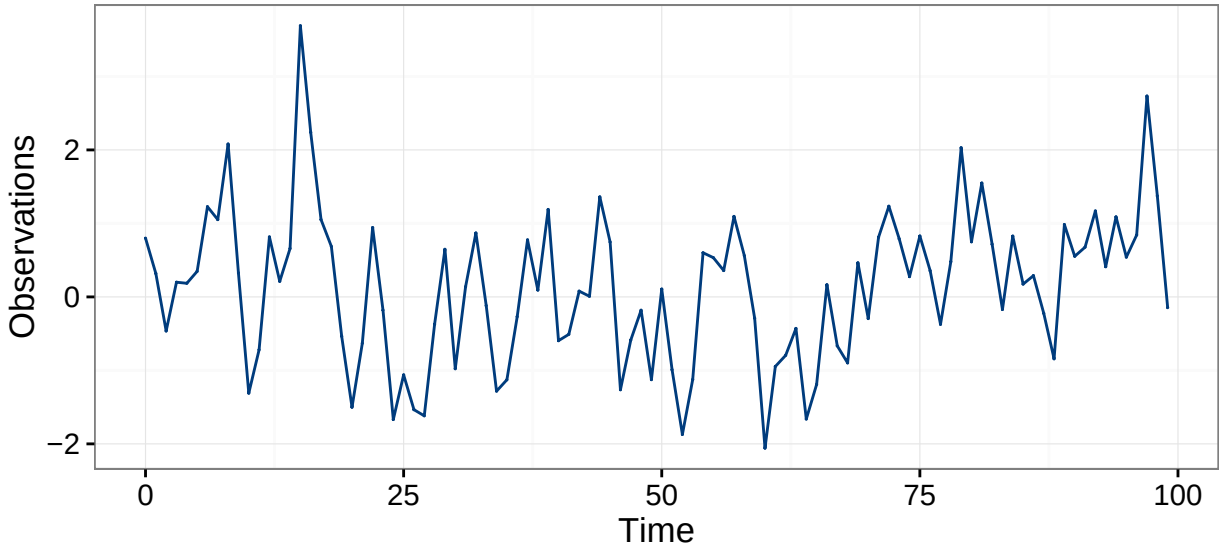
Under the assumption of infinite past (i.e. $t \in \mathbb{Z}$) and $|\phi| < 1$, we obtain

$$X_t = \sum_{i=0}^{\infty} \phi^i W_{t-i},$$

since $\lim_{i\to\infty}\ \phi^i X_{t-i} = 0$.

The code below presents an example of how an AR(1) can be simulated

```
# This code simulate a gaussian random walk process
n = 100                                  # process length
phi = 0.5                                # phi parameter
sigma2 = 1                               # innovation variance
Xt = gen.gts(AR1(phi = phi, sigma2 = sigma2), N = n)
plot(Xt)
```



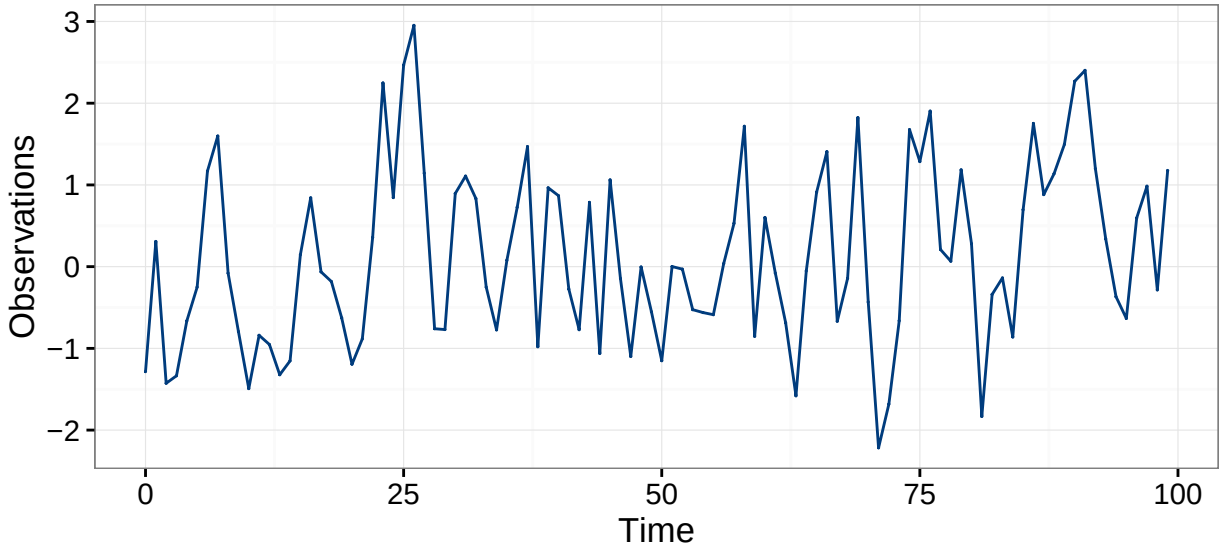
1.3.4 Moving Average Process of Order 1

As we have seen in the previous example, an AR(1) can be expressed as a linear combination of all past observation of (W_t), the next process, called a moving average process of order 1 or MA(1) is (in some sense) a "truncated" version of an AR(1). It is defined as

$$X_t = \theta W_{t-1} + W_t, \tag{1.1}$$

where (again) W_t denotes a Gaussian white noise process. An example on how generate an MA(1) is given below:

```
# This code simulates a gaussian white noise process
n = 100                                  # process length
sigma2 = 1                               # innovation variance
theta = 0.5                              # theta parameter
Xt = gen.gts(MA1(theta = theta, sigma2 = sigma2), N = n)
plot(Xt)
```



1.3.5 Linear Drift

A linear drift is a very simple detemrinistic time series model which can be expressed as

$$X_t = X_{t+1} + \omega,$$

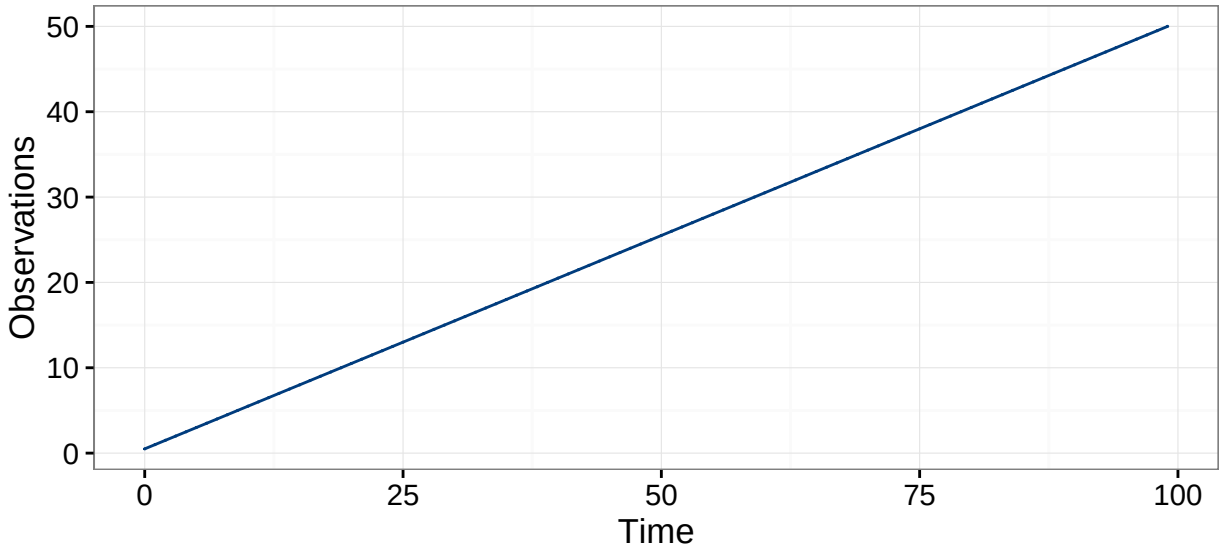
where ω is a constant and with the initial condition $X_0 = c$, an arbitrary constant (typically zero). This process can be expressed in a more familiar form as follows:

$$X_t = X_{t-1} + \omega = (X_{t-2} + \omega) + \omega = t\delta + c$$

Therefore, a (linear) drift corresponds to a simple linear model with slope ω and intercept c.

A drift can simply be generated used the code below:

```
# This code simulate a linear drift with 0 intercept
n = 100                                  # process length
omega = 0.5                              # slope parameter
Xt = gen.gts(DR(omega = omega), N = n)
plot(Xt)
```



1.3.6 Composite Stochastic Processes

A composite stochastic processes can be defined as the sum of underlying (or latent) stochastic processes. In this text, we will use the term *latent time series* as a synomym to composite stochastic processes. A simple example of such process is for example

$$Y_t = Y_{t-1} + W_t + \delta$$
$$X_t = Y_t + Z_t,$$

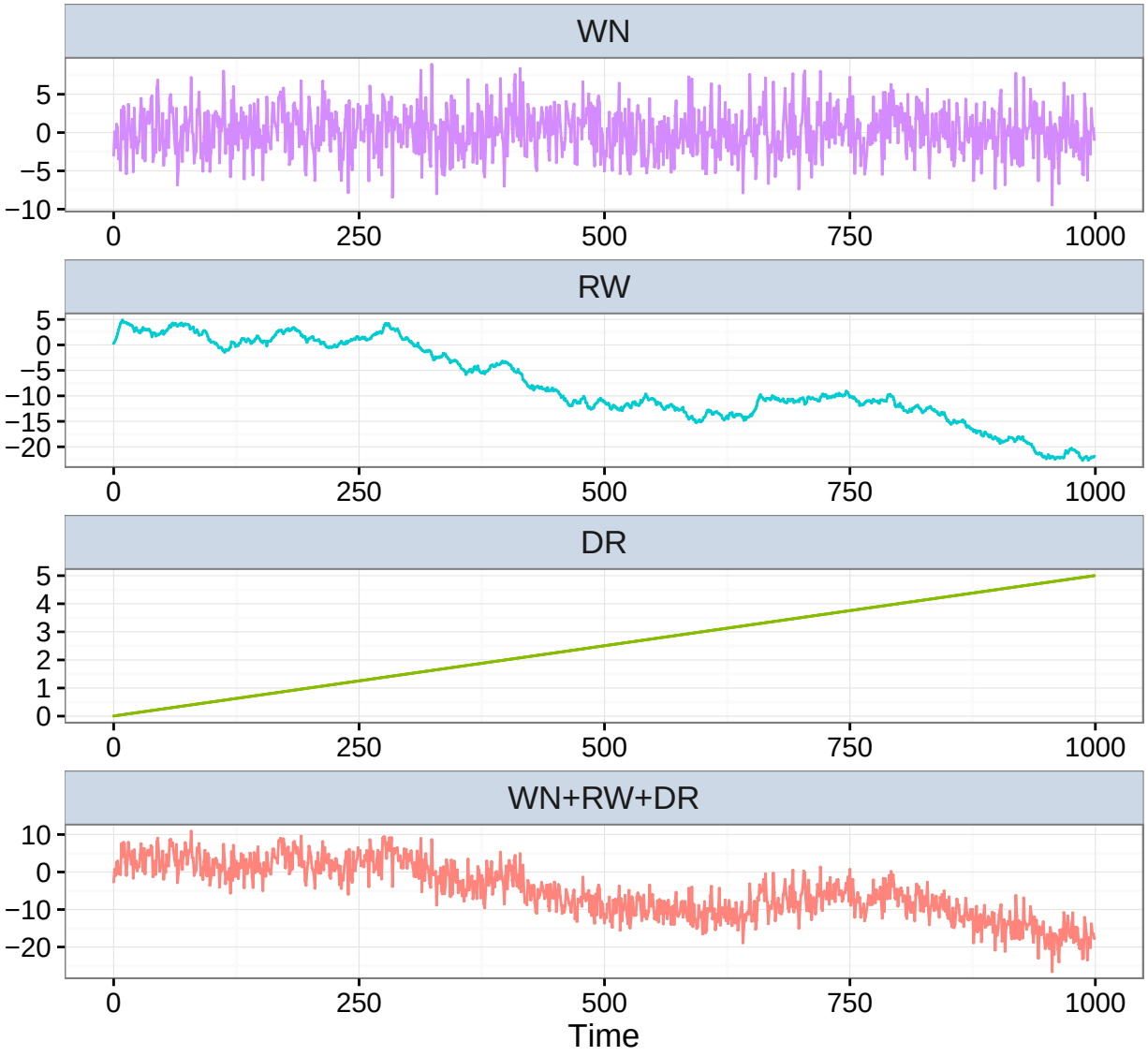
where W_t and Z_t are two independent Gaussian white noise processes. This model often used as first tool to approximate the number of individuals in the context ecological population dynamic. For example, suppose we want to study the popiulation of Chamois in the Swiss Alpes so let Y_t denote the "true" number of individual in this population at time t. It is reasonable that Y_t is (approximately) the population at the

previous time $t-1$ (e.g the previous year) plus a random variation and a drift. This random variation is due to the natural random in ecological population and reflects changes in number of predators, in aboundance of food or weather condition. On the other hand, the drift is often of particular interest for ecologist as it can used to determine the "long" term trends for the population (e.g. is the population increasing, stable or decreasing). Of course, Y_t (the number of individauls) is typically unknown and we observed a noisy version of it, denotes as X_t. This process corresponds to the true population plus a measruementr error as some Chamois may not be observed individuals and some counted several times. Interestingly, this process can clearly be expressed as a *latent time series model* (or composite stochastic process) as follows:

$$\begin{aligned} R_t &= R_{t-1} + W_t \\ S_t &= \delta t \\ X_t &= R_t + S_t + Z_t, \end{aligned}$$

where R_t, S_t and Z_t denote, respectively, a radnom walk, a drift and a white noise. The code below can be used to simulate such data:

```
n = 1000                                    # process length
delta = 0.005                               # delta parameter (drift)
sigma2 = 10                                 # variance parameter (white noise)
gamma2 = 0.1                                # innovation variance (random walk)
model = WN(sigma2 = sigma2) + RW(gamma2 = gamma2) + DR(omega = delta)
Xt = gen.lts(model, N = n)
plot(Xt)
```



In the above graph, the three latent (unobserved) processes are first depicted (i.e. white noise, random walk and drift) and then the sum of the three is present (i.e. (X_t)).

Chapter 2

Autocorrelation and Stationarity

> "*One of the first things taught in introductory statistics textbooks is that correlation is not causation. It is also one of the first things forgotten.*", Thomas Sowell

In this chapter we will discuss and formalize a little how knowledge about X_{t-1} (or Ω_t) can provide us with some information about X_t. In particular, we will consisder the correlation (or covariance) of (X_t) at different times such as $\text{corr}(X_t, X_{t+h})$. This "form" of correlation (covariance) is called the *autocorrelation* (*autocovariance*) and is a very usefull tool in time series analysis. Without assuming that a time series present form of "stability", it would be rather difficult to estimate $\text{corr}(X_t, X_{t+h})$ as this quantity would dependent on both t and h leading to far parameters to estimate than observations. Therefore, the concept of *stationarity* is conveniant in this context as it allow (among other things) to assume that

$$\text{corr}(X_t, X_{t+h}) = \text{corr}(X_t + j, X_{t+h+j}),$$

implying that the autocorrelation (or autocovariance) is only function of the lag between observation. This two concepts will be discuss in this chapter. Before moving on, it is helpful to remind that correlation (or autocorrelation) is only approriate to measure a very spefic kind on dependence, i.e. linear dependence. There are many forms of dependency as illustrated in the bottom panels on the graph below, which all have a (true) zero correlation:

Note that several other metrics have been introduced in the litterature to assess the degree of "dependence" of two random variables but this goes beyond the material discussed in this text.

2.1 The Autocorrelation and Autocovariance Functions

2.1.1 Definitions

The *autocovariance function* of a series (X_t) is defined as

$$\gamma_x(t, t+h) = \text{cov}(x_t, x_{t+h}).$$

Since we generally consider stochastic processes with constant zero mean we often have

$$\gamma_x(t, t+h) = E[X_t X_{t+h}].$$

We normally drop the subscript referring to the time series if it is clear to the time series the autocovariance function is referencing. For example, we generally use $\gamma(t, t+h)$ instead of $\gamma_x(t, t+h)$. Moreover, the

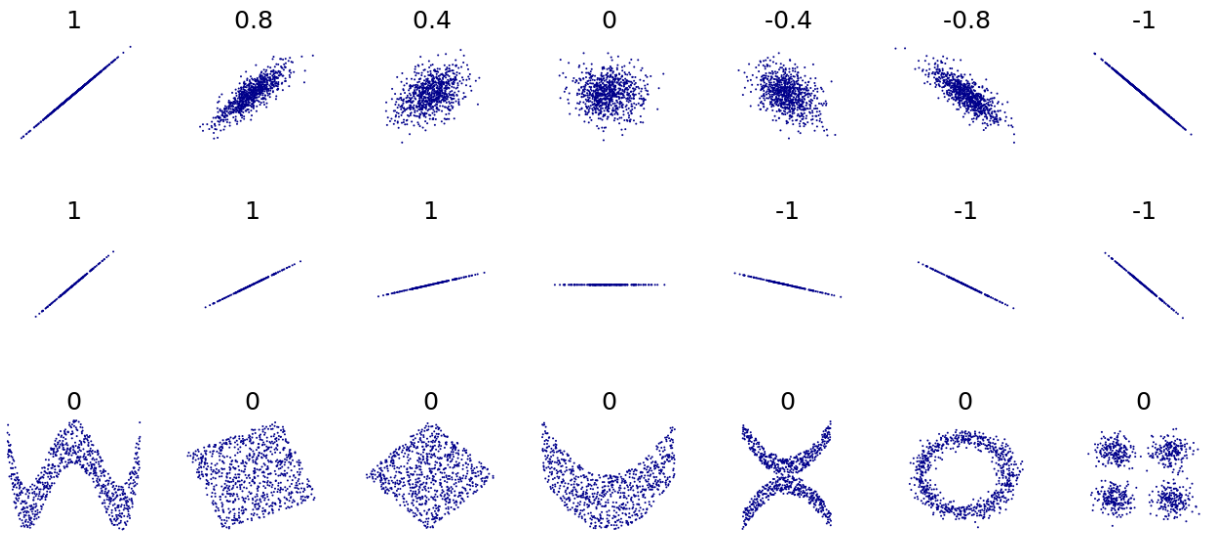


Figure 2.1: dependency

notation is even further simplify when the covariance of X_t and X_{t+h} is the same as that of X_{t+j} and X_{t+h+j} (for $j \in \mathbb{Z}$), i.e. that the covariance depends only on the time between observations and not the absolute date t. This is an important property call *stationarity*, which will be discuss in the next section. In this case, we simply use to following notation:

$$\gamma(h) = \operatorname{cov}(X_t, X_{t+h}).$$

A few other remarks:

1. The covariance function is **symmetric**. That is, $\gamma(h) = \gamma(-h)$ since $\operatorname{cov}(X_t, X_{t+h}) = \operatorname{cov}(X_{t+h}, X_t)$.
2. Note that $\operatorname{var}(X_t) = \gamma(0)$.
3. We have that $|\gamma(h)| \leq \gamma(0)$ for all h. The proof of this inequality follows from Cauchy-Schwarz inequality, i.e.
$$\begin{aligned}(|\gamma(h)|)^2 = \gamma(h)^2 &= (E[(X_t - E[X_t])(X_{t+h} - E[X_{t+h}])])^2 \\ &\leq E\left[(X_t - E[X_t])^2\right] E\left[(X_{t+h} - E[X_{t+h}])^2\right] = \gamma(0)^2.\end{aligned}$$
4. Just as any covariance, the $\gamma(h)$ is "scale dependent", $\gamma(h) \in \mathbb{R}$, or $-\infty \leq \gamma(h) \leq +\infty$, so of course we have:
   - if $|\gamma(h)|$ is "close" to zero, then X_t and X_{t+h} are "weakly" (linearly) dependent,
   - if $|\gamma(h)|$ is "far" from zero, then the two random variable present a "strong" (linear) dependence, but this is generally difficult to asses what "close" and "far" from zero means in this case.
5. $\gamma(h) = 0$ does not imply X_t and X_{t+h} are independent. This is only true in joint Gaussian case.

An important related statistic is the correlation of X_t with X_{t+h} or *autocorrelation* which is defined as

$$\rho(h) = \operatorname{corr}(X_t, X_{t+h}) = \frac{\gamma(h)}{\gamma(0)}.$$

Similarly to $\gamma(h)$, it is important to note that the above notation implies that the autocorrelation function is only a function of the lag h between observations. Thus, autocovariances and autocorrelations are one

possible way to describe the joint distribution of a time series. Indeed, the correlation of X_t with X_{t+1} is an obvious measure of how *persistent* a time series is.

Remeber that just as with any correlation:

1. $\rho(h)$ is scale free so it is much easier to interpret that than $\gamma(h)$.
2. $|\rho(h)| \leq 1$ since $|\gamma(h)| \leq \gamma(0)$.
3. Causation and correlation are two very different things!

2.1.2 A Fundamental Representation

Autocovariances and autocorrelation also turn out to be a very useful tool because they are one of the *fundamental representations* of time series. Indeed, if we consider a zero mean normally distrbuted process it is clear that its joint distribution is fully characterized by the autocariances $E[X_t X_{t+h}]$ (since the joint probability density only depends of these covariances). Once we know the autocovariances we know *everything* there is to know about the process and therefore: *if two processes have the same autocovariance function, then they are the same process.*

2.1.3 Admissible autocorrelation functions

Since the autocorrelation is related to a fundamental representation of time series it implies that one might be able to define a stochastic process by picking a set autocorrelation values. However, it turns out not every collection of numbers such as $\{\rho_1, \rho_2, ...\}$ is the autocorrelation of a process. Two conditions are required to ensure the validity of an autocorrelation sequence:

1. $\max_j \ |\rho_j| \leq 1$.
2. $\text{var}\left[\sum_{j=0}^{\infty} \alpha_j X_{t-j}\right] \geq 0$ for all $\{\alpha_0, \alpha_1, ...\}$.

The first condition is obvious and simply relects the fact that $|\rho(h)| \leq 1$ but the second is more difficult to verify. Let $\alpha_j = 0, \ j > 1$, then conditon 2 implies that

$$\text{var}\left[\alpha_0 X_t + \alpha_1 X_{t-1}\right] = \gamma_0 \begin{bmatrix} \alpha_0 & \alpha_1 \end{bmatrix} \begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \geq 0.$$

Thus, the matrix

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix}$$

must be positive semi-definite. Therefore,

$$\det(\boldsymbol{A}_1) = 1 - \rho_1^2$$

implying that $|\rho_1| < 1$. Next, let $\alpha_j = 0, \ j > 2$, then we must verify that:

$$\text{var}\left[\alpha_0 X_t + \alpha_1 X_{t-1} + \alpha_2 X_{t-2}\right] = \gamma_0 \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 \end{bmatrix} \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \geq 0.$$

Similarly, this implies that the matrix

$$\boldsymbol{A}_2 = \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}$$

must be positive semi-definite. It is easy to verify that

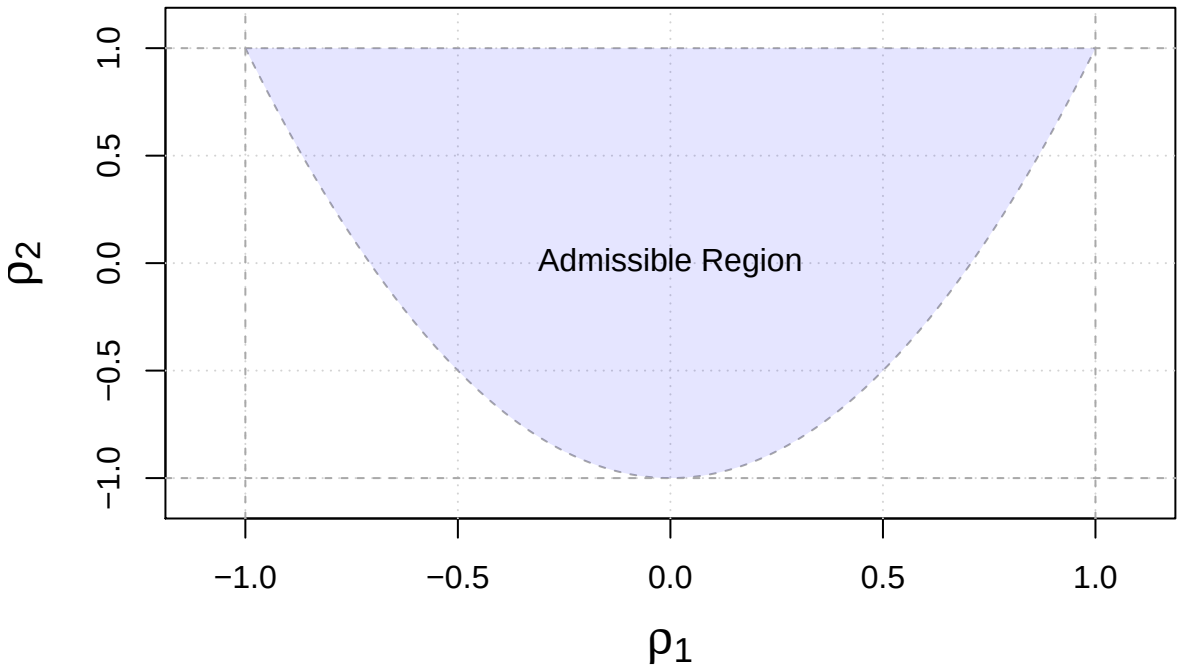
$$\det(\boldsymbol{A}_2) = (1-\rho_2)\left(-2\rho_1^2 + \rho_2 + 1\right).$$

It implies that $|\rho_2| < 1$ as well as

$$\begin{aligned} &-2\rho_1^2 + \rho_2 + 1 \geq 0 \Rightarrow 1 > \rho_2 \geq 2\rho_1^2 - 1 \\ &\Rightarrow 1 - \rho_1^2 > \rho_2 - \rho_1^2 \geq -(1-\rho_1^2) \\ &\Rightarrow 1 > \frac{\rho_2 - \rho_1^2}{1-\rho_1^2} \geq -1, \end{aligned}$$

imlying that ρ_1 and ρ_2 must lie in a parabolic shaped region defined by the above inequalities as illustrated in the figure below:

```
plot(NA, xlim = c(-1.1,1.1), ylim = c(-1.1,1.1), xlab = expression(rho[1]),
     ylab = expression(rho[2]), cex.lab = 1.5)
grid()

# Adding boundary of constraint |rho_1| < 1
abline(v = c(-1,1), lty = 2, col = "darkgrey")

# Adding boundary of constraint |rho_2| < 1
abline(h = c(-1,1), lty = 2, col = "darkgrey")

# Adding boundary of non-linear constraint
rho1 = seq(from = -1, to = 1, length.out = 10^3)
rho2 = (rho1^2 - 1) + rho1^2
lines(rho1, rho2, lty = 2, col = "darkgrey")

# Adding admissible region
polygon(c(rho1,rev(rho1)),c(rho2,rep(1,10^3)),
        border = NA, col= rgb(0,0,1, alpha = 0.1))

# Adding text
text(0,0, c("Admissible Region"))
```



Therefore, the restrictions on the autocorrelation are very complicated providing a motivation for other form of fundamental representation, which will explore later in this text. Before moving on the estimation of the autocorrelation and covariance function we first discuss the stationarity of (X_t), which will provide a conveninant framework in which $\gamma(h)$ and $\rho(h)$ can be used (rather that $\gamma(t, t+h)$ for example).

2.2 Stationarity

2.2.1 Definitions

There are two kinds of stationarity which are commonly used. They are defined below:

- A process (X_t) is *strongly stationary* or *strictly stationary* if the joint probability distribution of $\{X_{t-h}, ..., X_t, ..., X_{t+h}\}$ is independent of t for all h.
- A process (X_t) is *weakly stationary, covariance stationary* or *second order stationary* if $E[X_t]$, $E[X_t^2]$ are finite and $E[X_t X_{t-h}]$ depends only on h and not on t.

These types of stationarity are *not equivalent* and the presence of one kind of stationarity does not imply the other. That is, a time series can be strongly stationary but not weakly stationary and vice versa. In some cases, a time series can be both strong and weakly stationary, this is happends for example in the (joint) Gaussian case. Stationarity of (X_t) matters, because *it provides the framework in which averaging dependent data makes sense* allows to easily estimate quantities such as the autocorrelation function.

A few remarks:

- Strong stationarity *does not imply* weak stationarity. *Example*: an iid Cauchy process is strongly but not weakly stationary.
- Weak stationarity *does not imply* strong stationarity. *Example*: Consider the following weak white noise process: $X_{2t} = U_{2t}, X_{2t+1} = V_{2t+1}$, for $t = 1, ..., n$ where $U_t \overset{iid}{\sim} N(1, 1)$ and $V_t \overset{iid}{\sim} Exponential(1)$ is weakly stationary but *not* strongly stationary.
- Strong stationarity combined with bounded values of $E[X_t]$ and $E[X_t^2]$ *implies* weak stationarity
- Weak stationarity combined with normalityof the process *implies* strong stationarity.

2.2.2 Assessing Weak Stationarity of Time Series Models

It is important to understand how to verify if a postulated model is (waekly) stationary. In order to do so, we must ensure that the model satisfies three properties, i.e.

1. $E[X_t] = \mu_t = \mu < \infty$,
2. $\text{var}[X_t] = \sigma_t^2 = \sigma^2 < \infty$,
3. $\text{cov}(X_t, X_{t+h}) = \gamma(h)$.

In the following examples we evaluate the stationarity of the processes introduced in Section 1.3.

Example: Gaussian White Noise It is easy to verify that a this process is stationary. Indeed, we have:

1. $E[X_t] = 0$,
2. $\gamma(0) = \sigma^2 < \infty$,
3. $\gamma(h) = 0$ for $|h| > 0$.

Example: Random Walk To evaluate the stationarity of this process we first derive its properties:

1. $$E[X_t] = E[X_{t-1} + W_t] = E\left[\sum_{i=1}^{t} W_t + X_0\right]$$
$$= E\left[\sum_{i=1}^{t} W_t\right] + c = c$$

   Note that the mean here is constant since it depends only on the value of the first term in the sequence.

2. $$\text{var}(X_t) = \text{var}\left(\sum_{i=1}^{t} W_t + X_0\right) = \text{var}\left(\sum_{i=1}^{t} w_t\right) + \underbrace{\text{var}(X_0)}_{=0}$$
$$= \sum_{i=1}^{t} Var(w_t) = t\sigma_w^2.$$

   where $\sigma_w^2 = \text{var}(W_t)$. Therefore, the variance has a dependence on time contradicting our second property. Moreover, we have:
   $$\lim_{t\to\infty} \text{var}(X_t) = \infty.$$
   This process is therefore not weakly stationary.

3. Rgearding the autovariance of a random walk we have:
$$\gamma(h) = Cov(X_t, X_{t+h}) = Cov\left(\sum_{i=1}^{t} W_i, \sum_{j=1}^{t+h} W_j\right)$$
$$= Cov\left(\sum_{i=1}^{t} W_i, \sum_{j=1}^{t} W_j\right) = \min(t, t+h)\,\sigma_w^2$$
$$= (t + \min(0, h))\,\sigma_w^2,$$

which further illustrates that non-stationarity of this process.

Moreover, the autocorrelation of this process is given by

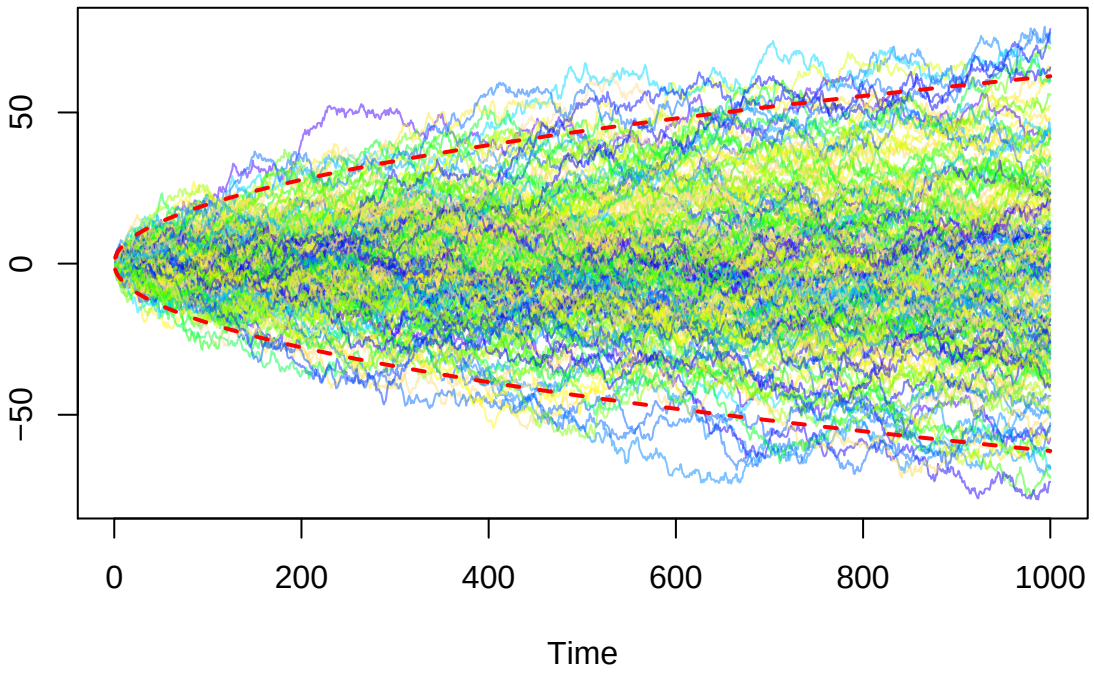
$$\rho(h) = \frac{t + \min(0, h)}{\sqrt{t}\sqrt{t+h}},$$

implying (for a fixed h) that

$$\lim_{t \to \infty} \rho(h) = 1.$$

In the following simulated example, we illustrate the non-stationary feature of such process:

```
# In this example, we simulate a large number of random walks
# Number of simulated processes
B = 200

# Length of random walks
n = 1000

# Output matrix
out = matrix(NA,B,n)

for (i in 1:B){
  # Simulate random walk
  Xt = cumsum(rnorm(n))

  # Store process
  out[i,] = Xt
}

# Plot random walks
plot(NA, xlim = c(1,n), ylim = range(out), xlab = "Time", ylab = " ")
color = sample(topo.colors(B, alpha = 0.5))
for (i in 1:B){
  lines(out[i,], col = color[i])
}

# Add 95% confidence region
lines(1:n, 1.96*sqrt(1:n), col = 2, lwd = 2, lty = 2)
lines(1:n, -1.96*sqrt(1:n), col = 2, lwd = 2, lty = 2)
```



The relationship between time and variance can clearly be observed in the above graph.

Example: MA(1) Similarly to our previous examples, we attempt to verify the stationary properties for the MA(1) model defined in 1.1 **REF NOT WORKING**:

1. $$E[X_t] = E[\theta_1 W_{t-1} + W_t] = \theta_1 E[W_{t-1}] + E[W_t] = 0.$$

2. $$\text{var}(X_t) = \theta_1^2 \text{var}(W_{t-1}) + \text{var}(W_t) = \left(1+\theta^2\right)\sigma_w^2.$$

3. $$\begin{aligned} Cov(X_t, X_{t+h}) &= E[(X_t - E[X_t])(X_{t+h} - E[X_{t+h}])] = E[X_t X_{t+h}] \\ &= E[(\theta W_{t-1} + W_t)(\theta W_{t+h-1} + W_{t+h})] \\ &= E\left[\theta^2 W_{t-1}W_{t+h-1} + \theta W_t W_{t+h} + \theta W_{t-1}W_{t+h} + W_t W_{t+h}\right]. \end{aligned}$$

It is easy to see that $E[W_t W_{t+h}] = \mathbf{1}_{\{h=0\}}\sigma_w^2$ and therefore, we obtain

$$\text{cov}(X_t, X_{t+h}) = \left(\theta^2 \mathbf{1}_{\{h=0\}} + \theta \mathbf{1}_{\{h=1\}} + \theta \mathbf{1}_{\{h=-1\}} + \mathbf{1}_{\{h=0\}}\right)\sigma_w^2$$

implying the following autocovariance function:

$$\gamma(h) = \begin{cases} \left(\theta^2+1\right)\sigma_w^2 & h = 0 \\ \theta\sigma_w^2 & |h| = 1 \\ 0 & |h| > 1 \end{cases}.$$

Therefore, an MA(1) process is weakly stationary since both the mean and variance are constant over time and its covariance function is only a function of the lag h. Finally, we can easily obtain the autocorrelation for this process, which is given by

$$\Rightarrow \rho(h) = \begin{cases} 1 & h = 0 \\ \frac{\theta\sigma_w^2}{(\theta^2+1)\sigma_w^2} = \frac{\theta}{\theta^2+1} & |h| = 1 \\ 0 & |h| > 1 \end{cases}$$

Interestingly, we can note that $|\rho(1)| \leq 0.5$.

Example AR(1)

JAMES TO DO - USE MA(1) AS EXAMPLE, ADD DETAILS FROM HOMEWORK, CHANGE ϕ_1 in ϕ and add ref to chap 1. Thanks

Consider the AR(1) process given as:

$$y_t = \phi_1 y_{t-1} + w_t, \text{ where } w_t \overset{iid}{\sim} WN\left(0, \sigma_w^2\right)$$

This process was shown to simplify to:

$$y_t = \phi^t y_0 + \sum_{i=0}^{t-1} \phi_1^i w_{t-i}$$

In addition, we add the requirement that $|\phi_1| < 1$. This requirement allows for the process to be stationary. If $\phi_1 \geq 1$, the process would not converge. This way the process will be able to be written as a geometric series that converges:

$$\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}, \;\; |r| < 1$$

Next, we demonstrate how crucial this property is:

$$\begin{aligned}
\lim_{t\to\infty} E[y_t] &= \lim_{t\to\infty} E\left[\phi^t y_0 + \sum_{i=0}^{t-1} \phi_1^i w_{t-i}\right] \\
&= \lim_{t\to\infty} \underbrace{\phi^t y_0}_{|\phi|<1 \Rightarrow t\to\infty \;=\; 0} + \sum_{i=0}^{t-1} \phi_1^i \underbrace{E[w_{t-i}]}_{=0} \\
&= 0 \\
\lim_{t\to\infty} Var(y_t) &= \lim_{t\to\infty} Var\left(\phi^t y_0 + \sum_{i=0}^{t-1} \phi_1^i w_{t-i}\right) \\
&= \lim_{t\to\infty} \underbrace{Var\left(\phi^t y_0\right)}_{=0 \text{ since constant}} + Var\left(\sum_{i=0}^{t-1} \phi_1^i w_{t-i}\right) \\
&= \lim_{t\to\infty} \sum_{i=0}^{t-1} \phi_1^{2i} Var(w_{t-i}) \\
&= \lim_{t\to\infty} \sigma_w^2 \sum_{i=0}^{t-1} \phi_1^{2i} \\
&= \sigma_w^2 \cdot \underbrace{\frac{1}{1-\phi^2}}_{\text{Geometric Series}}
\end{aligned}$$

This leads us to being able to conclude the autocovariance function is:

$$\begin{aligned} Cov\left(y_t, y_{t+h}\right) &= Cov\left(y_t, \phi y_{t+h-1} + w_{t+h}\right) \\ &= Cov\left(y_t, \phi y_{t+h-1}\right) \\ &= Cov\left(y_t, \phi^{|h|} y_t\right) \\ &= \phi^{|h|} Cov\left(y_t, y_t\right) \\ &= \phi^{|h|} Var\left(y_t\right) \\ &= \phi^{|h|} \frac{\sigma_w^2}{1 - \phi_1^2} \end{aligned}$$

Both the mean and autocovariance function do not depend on time and, thus, the AR(1) process is stationary if $|\phi_1| < 1$.

If we assume that the AR(1) process is stationary, we can derive the mean and variance in another way. Without a loss of generality, we'll assume $y_0 = 0$.

Therefore:

$$\begin{aligned} y_t &= \phi_t y_{t-1} + w_t \\ &= \phi_1 \left(\phi_1 y_{t-2} + w_{t-1}\right) + w_t \\ &= \phi_1^2 y_{t-2} + \phi_1 w_{t-1} + w_t \\ &\vdots \\ &= \sum_{i=0}^{t-1} \phi_1^i w_{t-i} \end{aligned}$$

$$\begin{aligned} E\left[y_t\right] &= E\left[\sum_{i=0}^{t-1} \phi_1^i w_{t-i}\right] \\ &= \sum_{i=0}^{t-1} \phi_1^i \underbrace{E\left[w_{t-i}\right]}_{=0} \\ &= 0 \end{aligned}$$

$$\begin{aligned} Var\left(y_t\right) &= E\left[\left(y_t - E\left[y_t\right]\right)^2\right] \\ &= E\left[y_t^2\right] - \left(E\left[y_t\right]\right)^2 \\ &= E\left[y_t^2\right] \\ &= E\left[\left(\phi_1 y_{t-1} + w_t\right)^2\right] \\ &= E\left[\phi_1^2 y_{t-1}^2 + w_t^2 + 2\phi_1 y_t w_t\right] \\ &= \phi_1^2 E\left[y_{t-1}^2\right] + \underbrace{E\left[w_t^2\right]}_{=\sigma_w^2} + 2\phi_1 \underbrace{E\left[y_t\right]}_{=0} \underbrace{E\left[w_t\right]}_{=0} \\ &= \underbrace{\phi_1^2 Var\left(y_{t-1}\right) + \sigma_w^2 = \phi_1^2 Var\left(y_t\right) + \sigma_w^2}_{\text{Assume stationarity}} \end{aligned}$$

$$\begin{aligned} Var\left(y_t\right) &= \phi_1^2 Var\left(y_t\right) + \sigma_w^2 \\ Var\left(y_t\right) - \phi_1^2 Var\left(y_t\right) &= \sigma_w^2 \\ Var\left(y_t\right)\left(1 - \phi_1^2\right) &= \sigma_w^2 \\ Var\left(y_t\right) &= \frac{\sigma_w^2}{1 - \phi_1^2} \end{aligned}$$

2.3 Estimation of the Mean Function

If a time series is stationary, the mean function is constant and a possible estimator of this quantity is given by

$$\bar{X} = \frac{1}{n} \sum_{t=1}^{n} X_t.$$

This estimator is clearly unbiased and has the following variance:

$$
\begin{aligned}
\operatorname{var}\left(\bar{X}\right) &= \operatorname{var}\left(\frac{1}{n}\sum_{t=1}^{n} X_t\right) \\
&= \frac{1}{n^2}\operatorname{var}\left(\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times n}\begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}_{n\times 1}\right) \\
&= \frac{1}{n^2}\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times n}\begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & & \vdots \\ \vdots & & \ddots & \vdots \\ \gamma(n-1) & \cdots & \cdots & \gamma(0) \end{bmatrix}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}. \\
&= \frac{1}{n^2}\left(n\gamma(0) + 2(n-1)\gamma(1) + 2(n-2)\gamma(2) + \cdots + 2\gamma(n-1)\right) \\
&= \frac{1}{n}\sum_{h=-n}^{n}\left(1 - \frac{|h|}{n}\right)\gamma(h)
\end{aligned}
$$

In the white noise case, the above formula reduces to the usual $\operatorname{var}\left(\bar{X}\right) = \operatorname{var}(X_t)/n$.

Example: For an AR(1) we have $\gamma(h) = \phi^h\sigma_w^2\left(1-\phi^2\right)^2$, therefore, we obtain (after a bit of algebra):

$$\operatorname{var}\left(\bar{X}\right) = \frac{\sigma_w^2\left(n - 2\phi - n\phi^2 + 2\phi^{n+1}\right)}{n^2\left(1-\phi^2\right)\left(1-\phi\right)^2}.$$

Unfortunately, deriving such an exact formula is often difficult when considering more complexe models, however, asymptotic approximations are often employed simply the calculation. For example, in our case we have

$$\lim_{n\to\infty} n\operatorname{var}\left(\bar{X}\right) = \frac{\sigma_w^2}{\left(1-\phi\right)^2},$$

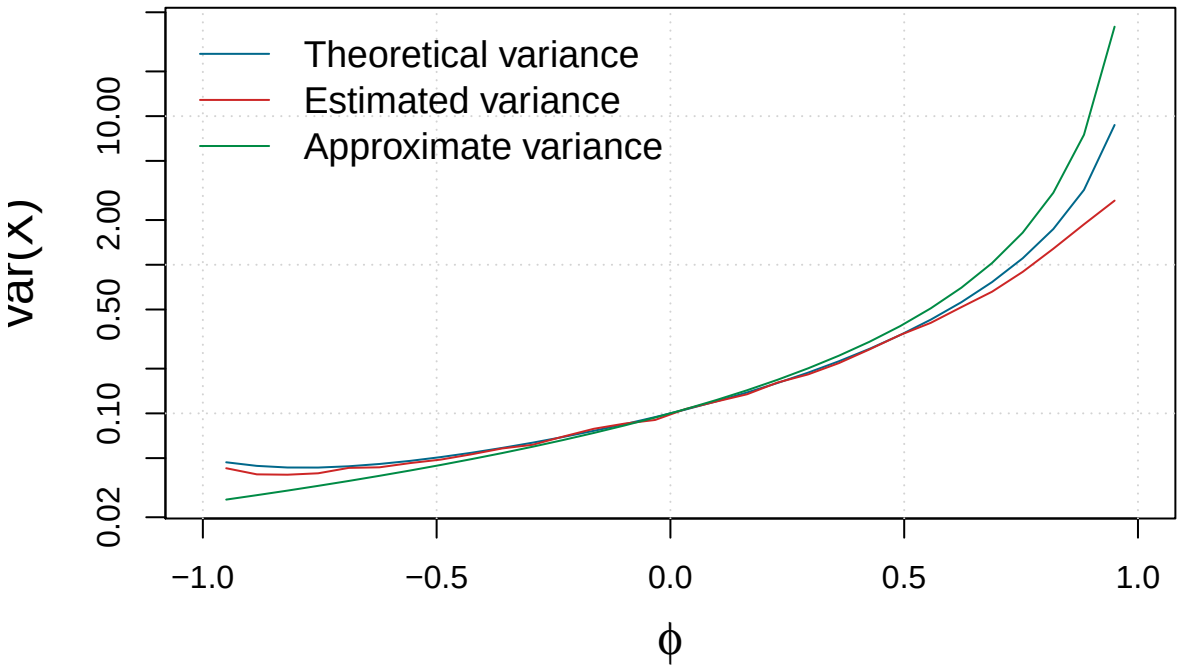
providing the following approximate formula:

$$\operatorname{var}\left(\bar{X}\right) \approx \frac{\sigma_w^2}{n\left(1-\phi\right)^2}.$$

Alternatively, simulation methods can also employed. The figure compares these three methods:

```
# Define sample size
n = 10

# Number of Monte-Carlo replications
B = 5000

# Define grid of values for phi
phi = seq(from = 0.95, to = -0.95, length.out = 30)

# Define result matrix
result = matrix(NA,B,length(phi))

# Start simulation
```

```
for (i in 1:length(phi)){
  # Define model
  model = AR1(phi = phi[i], sigma2 = 1)

  # Monte-Carlo
  for (j in 1:B){
    # Simulate AR(1)
    Xt = gen.gts(model, N = n)

    # Estimate Xbar
    result[j,i] = mean(Xt)
  }
}

# Estimate variance of Xbar
var.Xbar = apply(result,2,var)

# Compute theoretical variance
var.theo = (n - 2*phi - n*phi^2 + 2*phi^(n+1))/(n^2*(1-phi^2)*(1-phi)^2)

# Compute (approximate) vairance
var.approx = 1/(n*(1-phi)^2)

# Compare variance estimations
plot(NA, xlim = c(-1,1), ylim = range(var.approx), log = "y",
    ylab = expression(paste("var(", bar(X), ")")),
    xlab= expression(phi), cex.lab = 1.5)
grid()
lines(phi,var.theo, col = "deepskyblue4")
lines(phi, var.Xbar, col = "firebrick3")
lines(phi,var.approx, col = "springgreen4")
legend("topleft",c("Theoretical variance","Estimated variance","Approximate variance"),
       col = c("deepskyblue4","firebrick3","springgreen4"), lty = 1,
       bty = "n",bg = "white", box.col = "white", cex = 1.2)
```



2.4 Sample Autocovariance and Autocorrelation Functions

A natural estimator of the **autocovariance function** is given as:

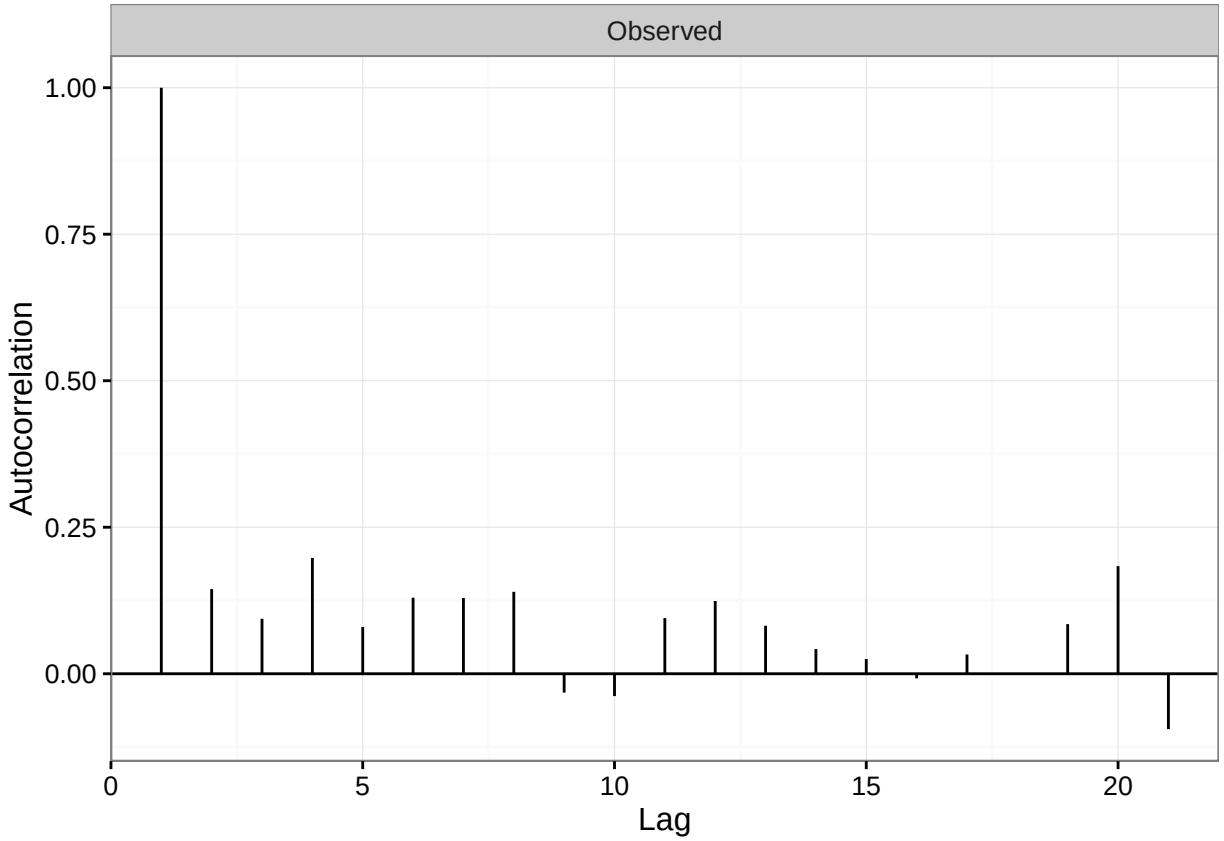
$$\hat{\gamma}(h) = \frac{1}{T}\sum_{t=1}^{T-h}\left(X_t - \bar{X}\right)\left(X_{t+h} - \bar{X}\right)$$

leading the following "plug-in" estimator of the **autocorrelation function**

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}.$$

A graphical representation of the autocorrelation function is often the first step of any time series analysis (assuming the process to be stationary). Consider the following simulated example:

```
# Load package
library(gmwm)

# Simulate 100 observation from a Gaussian white noise
Xt = gen.gts(WN(sigma2 = 1), N = 100)

# Compute autocorrelation
acf_Xt = ACF(Xt)

# Plot autocorrelation
plot(acf_Xt, show.ci = FALSE)
```

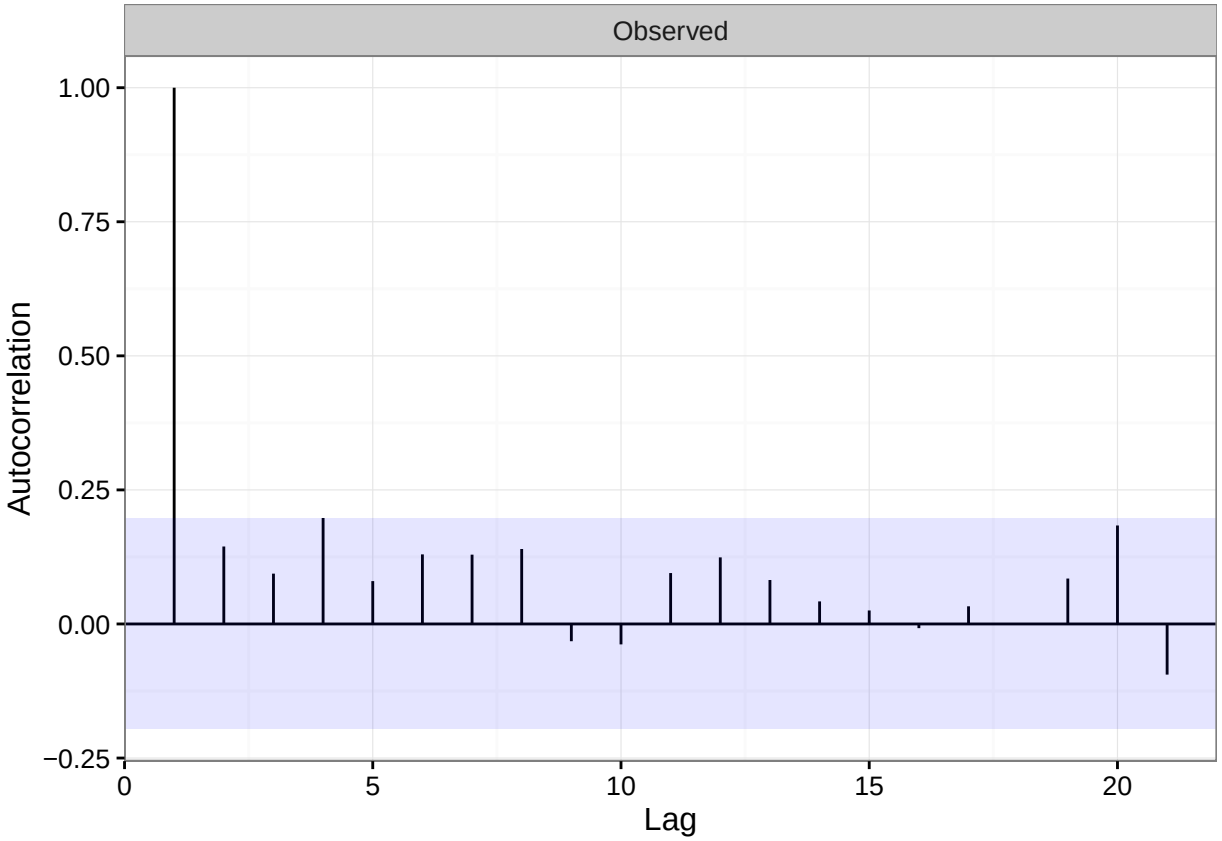


In this example, the true autocorrelation is equal to zero at any lag $h \neq 0$ but obviously the estimated autocorrelations are random variables and are not equal to their true values. It would therefore be usefull to have some knowledge about the variability of the sample autocorrelations (under some conditions) to assess whether the data comes from a completely random series or presents some significant correlation at some lags. The following result provide an asymptotic solution to this problem:

Theorem: If X_t is white noise with finite fourth moment, then $\hat{\rho}(h)$ is approximately normally distributed with mean 0 and variance n^{-1} for all fixed h.

Using on this result, we now have an approximate method to assess whether peaks in sample autocorrelation are significant by determining whether the observed peak lies outside the interval $\pm 2/\sqrt{T}$ (i.e. an approximate 95% confidence interval). Returning to our previous example:

```
# Plot autocorrelation with confidence bands
plot(acf_Xt)
```

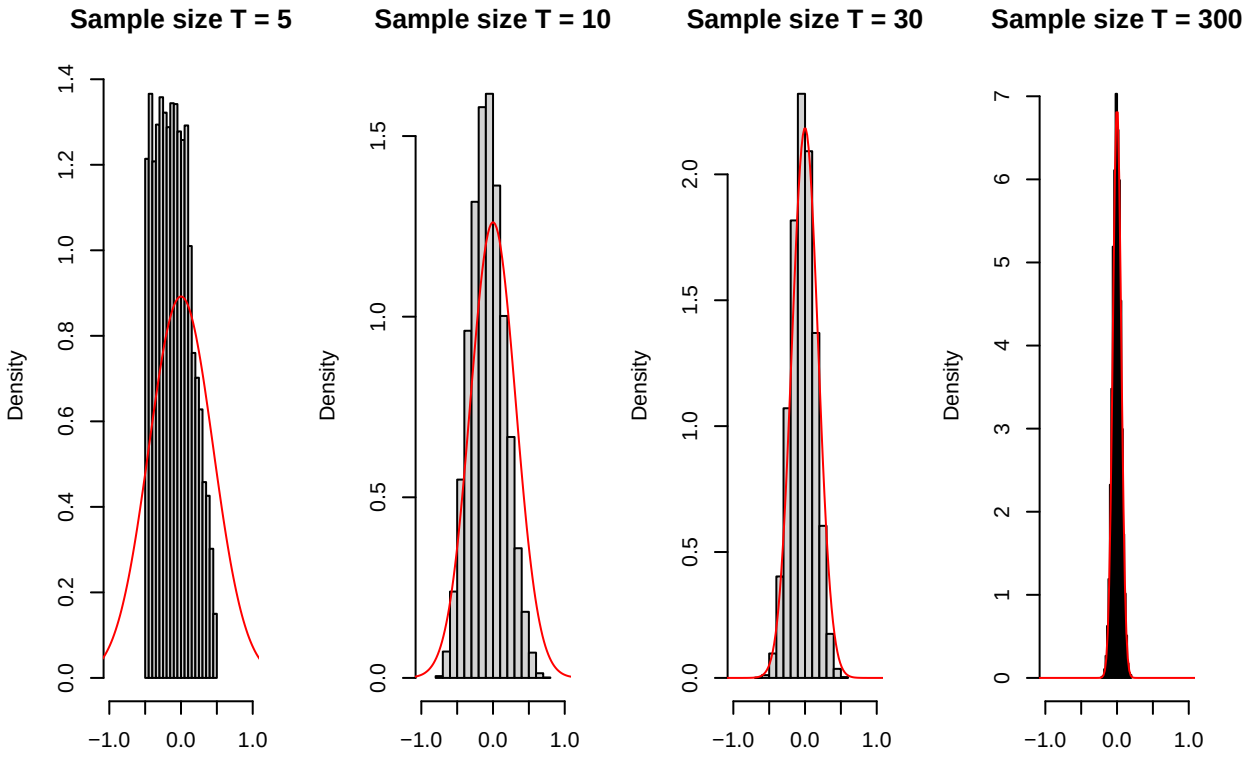


It can now be observed that most peaks lies within the interval $\pm 2/\sqrt{T}$ suggesting that the true data generating process is completely random (in the linear sense).

Unfortunately, this method is asymptotic (it relies on the central limit theorem) and there no "exact" tools that can be used in this case. In the simulation study below consider the "quality" of this result for $h = 3$ considering different sample sizes:

```
# Number of Monte Carlo replications
B = 10000

# Define considered lag
h = 3

# Sample size considered
T = c(5,10,30,300)

# Initialisation
result = matrix(NA,B,length(T))

# Set seed
set.seed(1)

# Start Monte Carlo
for (i in 1:B){
  for (j in 1:length(T)){
    # Simluate process
    Xt = rnorm(T[j])
```

```
    # Save autocorrelation at lag h
    result[i,j] = acf(Xt, plot = FALSE)$acf[h+1]
  }
}

# Plot results
par(mfrow = c(1,length(T)))
for (i in 1:length(T)){
  # Estimated empirical distribution
  hist(result[,i], col = "lightgrey", main = paste("Sample size T =",T[i]), probability = TRUE, xlim = c

  # Asymptotic distribution
  xx = seq(from = -10, to = 10, length.out = 10^3)
  yy = dnorm(xx,0,1/sqrt(T[i]))
  lines(xx,yy, col = "red")
}
```

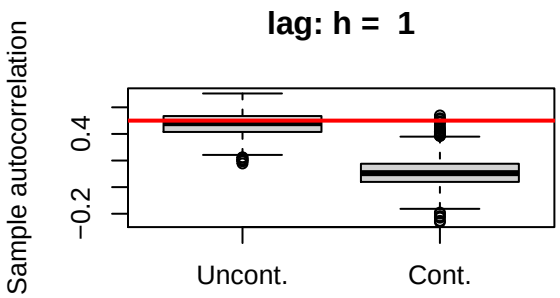


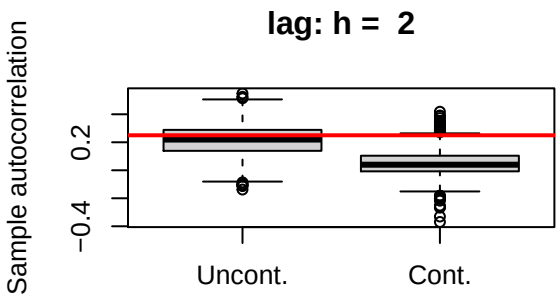
It can clearly be observed that asymptotic approximation is quite poor when $T = 5$ but as the sample size increases the approximation becomes more appropriate and is nearly perfect with $T = 300$.

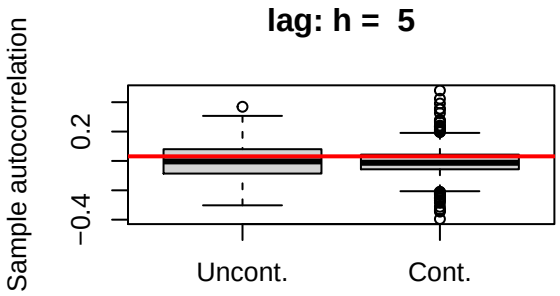
2.5 Robustness Issues

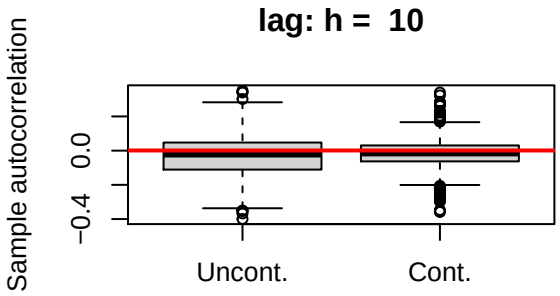
```
# Define sample size
n = 100
```

```
# Define proportion of "extreme" observation
alpha = 0.05

# Extreme observation are generated from N(0,sigma2.cont)
sigma2.cont = 10

# Number of Monte-Carlo replications
B = 1000

# Define model AR(1)
phi = 0.5
sigma2 = 1
model = AR1(phi = phi, sigma2 = sigma2)

# Initialization of result array
result = array(NA, c(B,2,20))

# Start Monte-Carlo
for (i in 1:B){
  # Simulate AR(1)
  Xt = gen.gts(model, N = n)

  # Create a copy of Xt
  Yt = Xt

  # Add a proportion alpha of extreme observations to Yt
  Yt[sample(1:n,round(alpha*n))] = rnorm(round(alpha*n), 0, sigma2.cont)

  # Compute ACF of Xt and Yt
  acf_Xt = ACF(Xt)
  acf_Yt = ACF(Yt)

  # Store ACFs
  result[i,1,] = acf_Xt[1:20]
  result[i,2,] = acf_Yt[1:20]
}


# Compare empirical distribution of ACF based on Xt and Yt

# Vector of lags considered (h <= 20)
lags = c(1,2,5,10) + 1

# Make graph
par(mfrow = c(2,2))

for (i in 1:4){
  boxplot(result[,1,lags[i]], result[,2,lags[i]], col = "lightgrey",
          names = c("Uncont.","Cont."), main = paste("lag: h = ", lags[i]-1),
          ylab = "Sample autocorrelation")
  abline(h = phi^(lags[i]-1), col = 2, lwd = 2)
}
```


lag: h = 1
Sample autocorrelation
−0.2
0.4
Uncont.
Cont.


lag: h = 2
Sample autocorrelation
−0.4
0.2
Uncont.
Cont.


lag: h = 5
Sample autocorrelation
−0.4
0.2
Uncont.
Cont.


lag: h = 10
Sample autocorrelation
−0.4
0.0
Uncont.
Cont.

Chapter 3

Basic Models

3.1 The Backshift Operator

Definition: **Backshift Operator**

The **Backshift Operator** is helpful when manipulating time series. When we backshift, we are changing the indices of the time series. e.g. $t \to t-1$. The operator is defined as:

$$Bx_t = x_{t-1}$$

If we were to repeatedly apply the backshift operator, we would receive:

$$\begin{aligned} B^2 x_t &= B\left(Bx_t\right) \\ &= B\left(x_{t-1}\right) \\ &= x_{t-2} \end{aligned}$$

We can generalize this behavior as:

$$B^k x_t = x_{t-k}$$

The backshift operator is helpful for later decompositions in addition to making differencing operations more straightforward.

3.2 White Noise

The process name of white noise has meaning in the notion of colors of noise. Specifically, the white noise is a process that mirrors white light's flat frequency spectrum. So, the process has equal frequencies in any interval of time.

Definition: **White Noise**

w_t or ε_t is a **white noise process** if w_t are uncorrelated identically distributed random variables with $E\left[w_t\right] = 0$ and $Var\left[w_t\right] = \sigma^2$, for all t. We can represent this algebraically as:

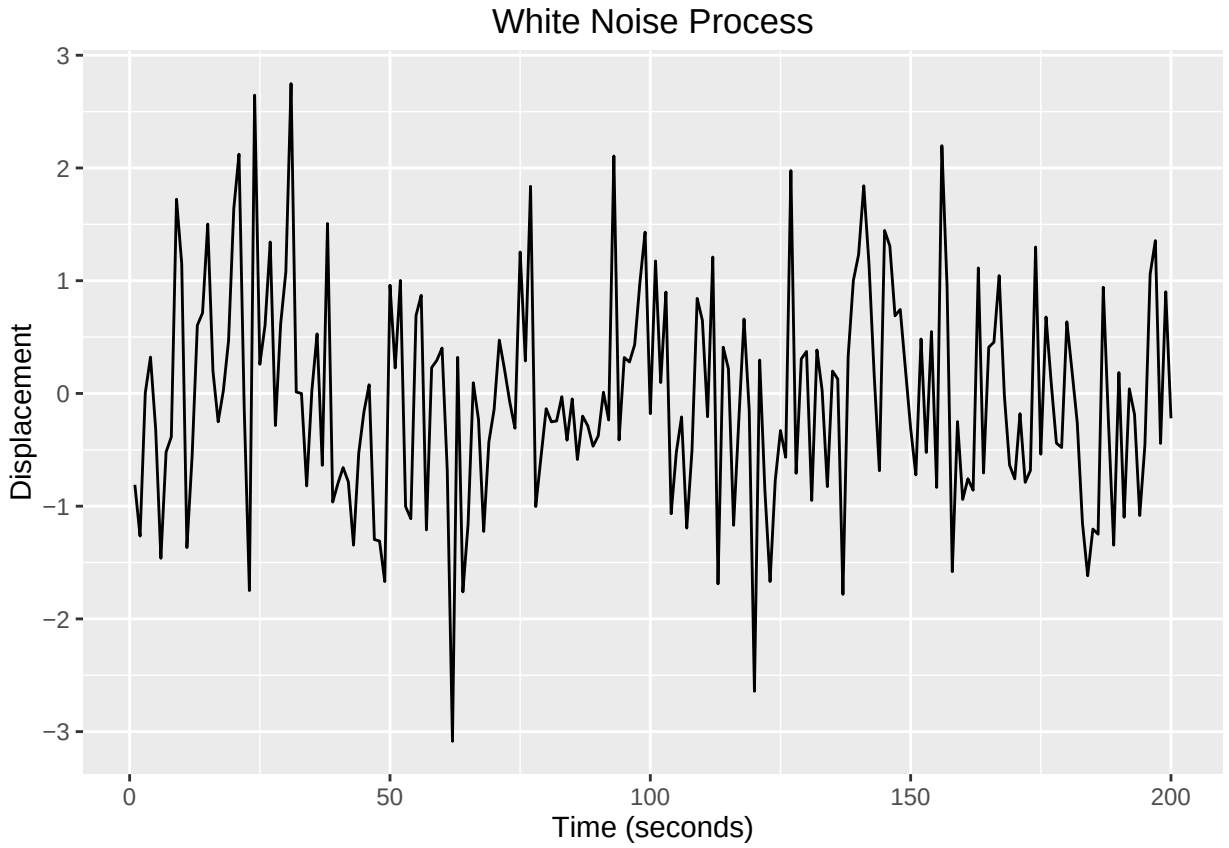
$$y_t = w_t,$$

where $w_t \overset{id}{\sim} WN\left(0, \sigma_w^2\right)$

Now, if the w_t are **Normally (Gaussian) distributed**, then the process is known as a **Gaussian White Noise** e.g. $w_t \overset{iid}{\sim} N\left(0, \sigma^2\right)$

To generate gaussian white noise use:

```
set.seed(1336)           # Set seed to reproduce the results
n  = 200                 # Number of observations to generate
wn = ts(rnorm(n,0,1))    # Generate Guassian white noise.

autoplot(wn) +
  ggtitle("White Noise Process") +
  ylab("Displacement") + xlab("Time (seconds)")
```



3.3 Moving Average Process of Order q = 1 a.k.a MA(1)

Definition: **Moving Average Process of Order (q = 1)**

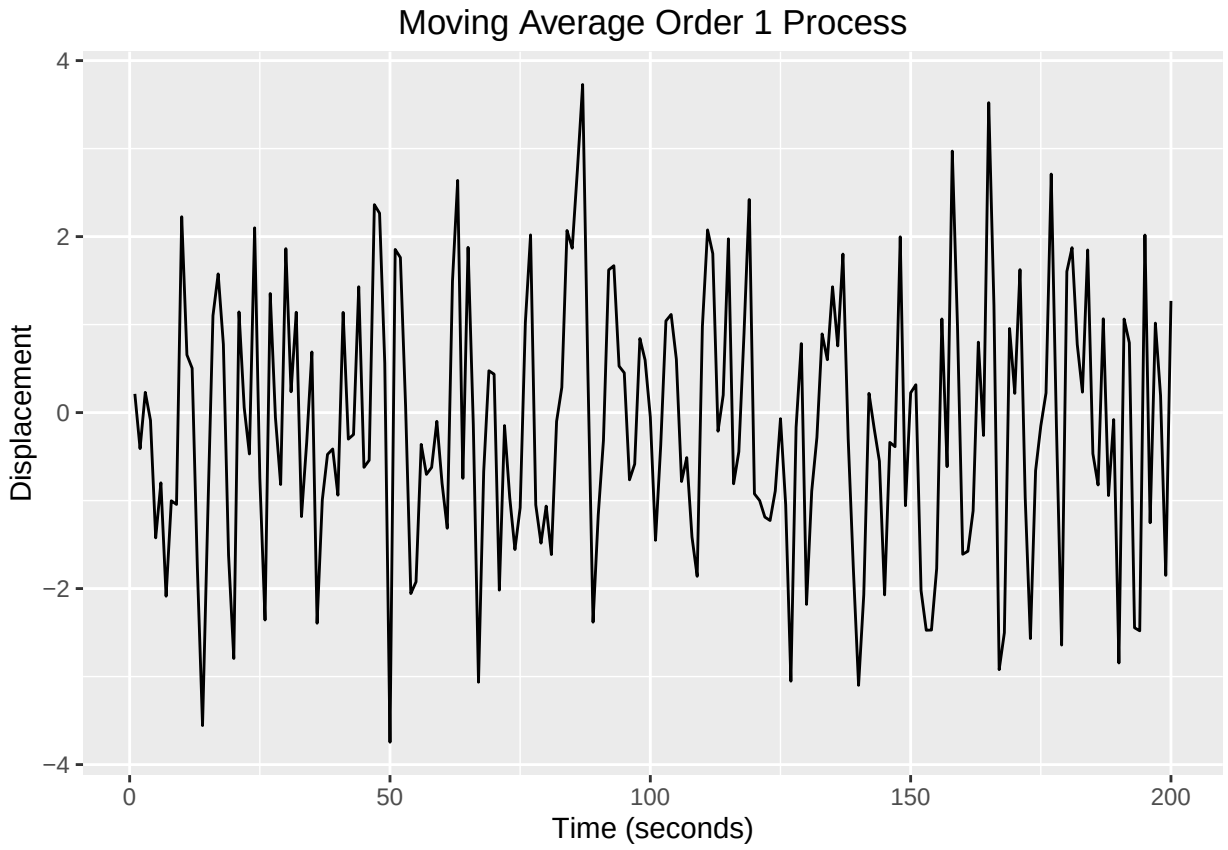
The concept of a **Moving Average Process of Order q** is a way to remove "noise" and emphasize the signal. The moving average achieves this by taking the local averages of the data to produce a new smoother time series series. The newly created time series is more descriptive, but it does influence the dependence within the time series.

This process is generally denoted as **MA(1)** and is defined as:

$$y_t = \theta_1 w_{t-1} + w_t,$$

where $w_t \overset{iid}{\sim} WN\left(0, \sigma_w^2\right)$

```r
set.seed(1345)  # Set seed to reproduce the results
n      = 200    # Number of observations to generate
sigma2 = 2      # Controls variance of Guassian white noise.
theta  = 0.3    # Handles the theta component of MA(1)


# Generate a white noise
wn = rnorm(n+1, sd = sqrt(sigma2))

# Simulate the MA(1) process
ma = rep(0, n+1)
for(i in 2:(n+1)) {
  ma[i] = theta*wn[i-1] + wn[i]
}

ma = ts(ma[2:(n+1)])      # Remove first item

autoplot(ma) +
  ggtitle("Moving Average Order 1 Process") +
  ylab("Displacement") + xlab("Time (seconds)")
```



3.4 Drift

Definition: **Drift**

A **drift process** has two components: time and a slope. As more points are accumlated over time, the drift will match the common slope form.

Specifically, the drift process has the following form:

$$y_t = y_{t-1} + \delta$$

with the initial condition $y_0 = c$.

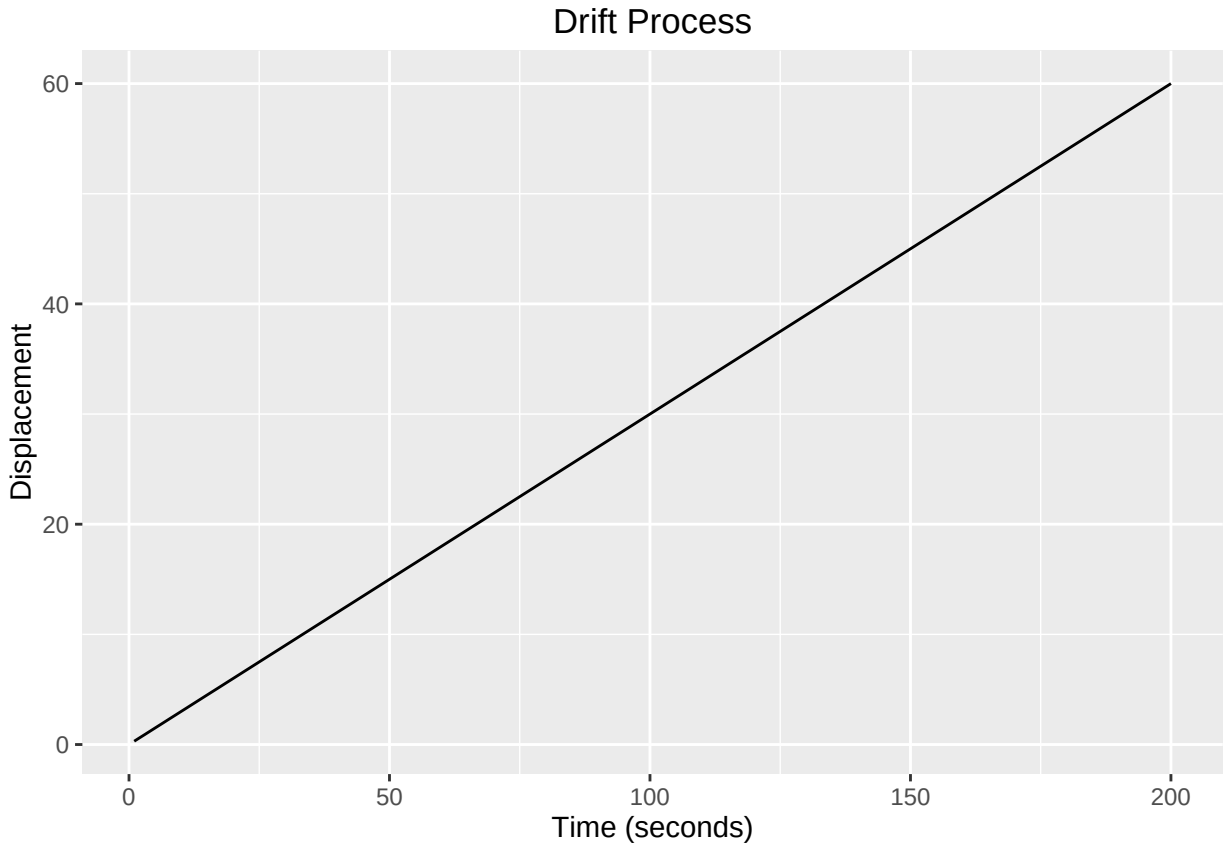
The process can be simplified using **backsubstitution** to being:

$$\begin{aligned} y_t &= y_{t-1} + \delta \\ &= (y_{t-2} + \delta) + \delta \\ &\vdots \\ &= \sum_{i=1}^{t} \delta + y_0 \\ y_t &= t\delta + c \end{aligned}$$

Again, note that a drift is similar to the slope-intercept form a linear line. e.g. $y = mx + b$.

To generate a drift use:

```
n     = 200             # Number of observations to generate
drift = .3              # Drift Control
dr    = ts(drift*(1:n)) # Generate drift sequence (e.g. y = drift*x + 0)

autoplot(dr) +
  ggtitle("Drift Process") +
  ylab("Displacement") + xlab("Time (seconds)")
```



3.5 Random Walk

In 1906, Karl Pearson coined the term 'random walk' and demonstrated that "the most likely place to find a drunken walker is somewhere near his starting point." Empirical evidence of this phenomenon is not too hard to find on a Friday night in Champaign.

Definition: **Random Walk**

A **random walk** is defined as a process where the current value of a variable is composed of the past value plus an error term that is a white noise. In algebraic form,

$$y_t = y_{t-1} + w_t$$

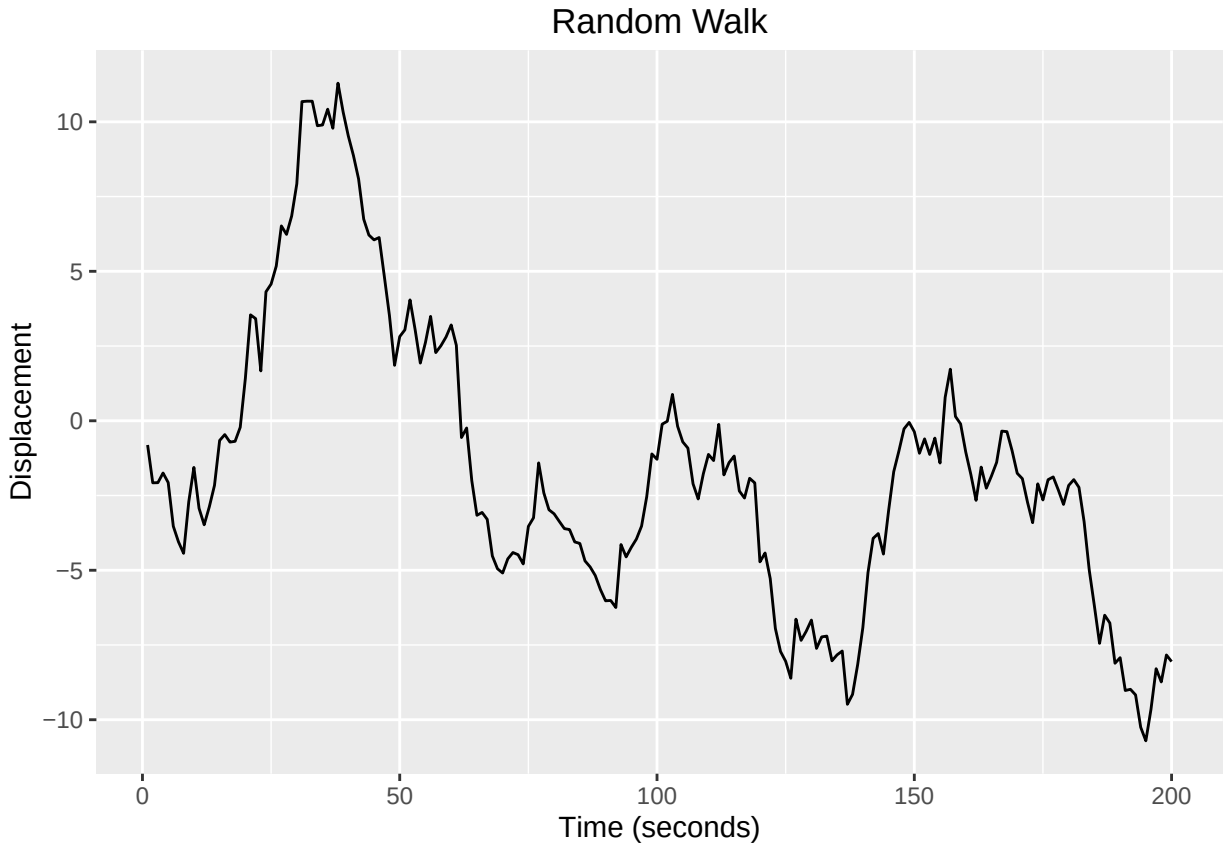
with the initial condition $y_0 = c$.

The process can be simplified using **backsubstitution** to being:

$$\begin{aligned} y_t &= y_{t-1} + w_t \\ &= (y_{t-2} + w_{t-1}) + w_t \\ &\vdots \\ y_t &= \sum_{i=1}^{t} w_i + y_0 = \sum_{i=1}^{t} w_i + c \end{aligned}$$

To generate a random walk, we use:

```r
set.seed(1336)        # Set seed to reproduce the results
n  = 200              # Number of observations to generate
w  = rnorm(n,0,1)     # Generate Guassian white noise.
rw = ts(cumsum(w))    # Cumulative sum

# Create a data.frame to graph in ggplot2
autoplot(rw) +
  ggtitle("Random Walk") +
  ylab("Displacement") + xlab("Time (seconds)")
```



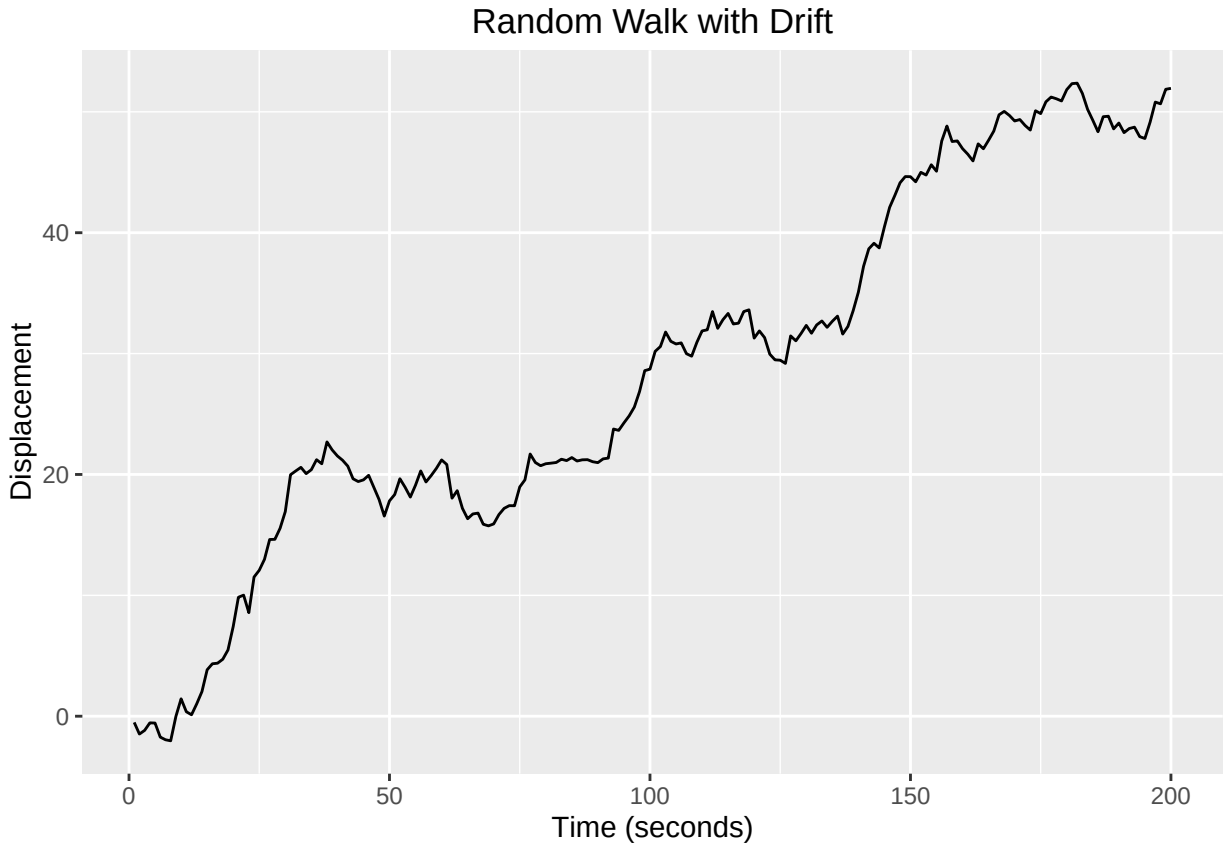
3.6 Random Walk with Drift

In the previous case of a random walk, we assumed that drift, δ, was equal to 0. What happens to the random walk if the drift is not equal to zero? That is, what happens with the initial condition $y_0 = c$?

$$\begin{aligned} y_t &= y_{t-1} + w_t + \delta \\ &= (y_{t-2} + w_{t-1} + \delta) + w_t + \delta \\ &\vdots \\ y_t &= \sum_{i=1}^{t} (w_i + \delta) + y_0 = \sum_{i=1}^{t} w_i + t\delta + c \end{aligned}$$

To generate a random walk with drift we use:

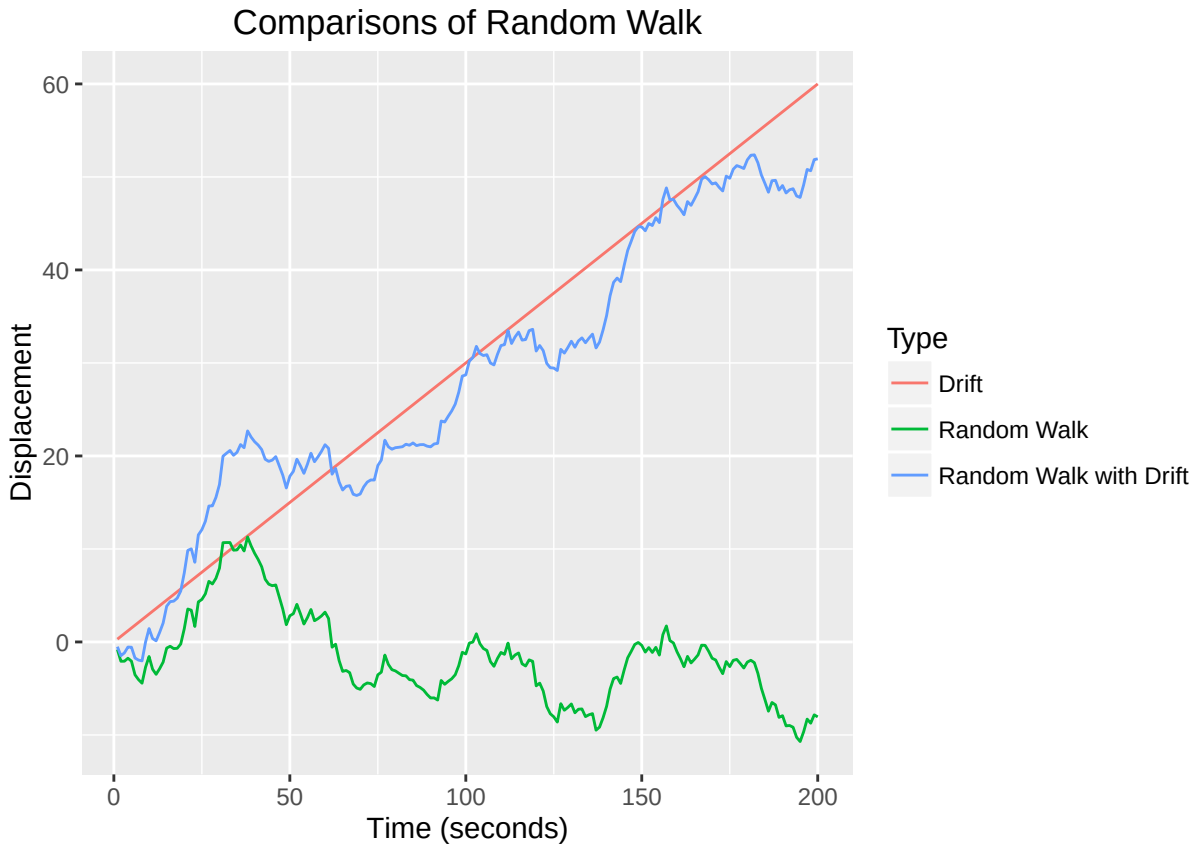
```
set.seed(1336)        # Set seed to reproduce the results
n     = 200           # Number of observations to generate
drift = .3            # Drift Control

w = rnorm(n,0,1)      # Generate Guassian white noise.
wd = w + drift        # Add a drift
rwd = ts(cumsum(wd)) # Cumulative sum

# Create a data.frame to graph in ggplot2
autoplot(rwd) +
  ggtitle("Random Walk with Drift") +
  ylab("Displacement") + xlab("Time (seconds)")
```



Notice the difference the drift makes upon the random walk:

```
# Add identifiers
drift.df = data.frame(Index = 1:n, Data = drift*(1:n), Type = "Drift")

rw.df = data.frame(Index = 1:n, Data = rw, Type = "Random Walk")

rwd.df = data.frame(Index = 1:n, Data = rwd, Type = "Random Walk with Drift")

combined.df = rbind(drift.df, rw.df, rwd.df)

ggplot(data = combined.df, aes(x = Index, y = Data, colour = Type)) +
  geom_line() +
```

```
  ggtitle("Comparisons of Random Walk") +
  ylab("Displacement") + xlab("Time (seconds)")
```



3.7 Autoregressive Process of Order p = 1 a.k.a AR(1)

Definition: **Autoregressive Process of Order p = 1**

This process is generally denoted as **AR(1)** and is defined as: $y_t = \phi_1 y_{t-1} + w_t$,

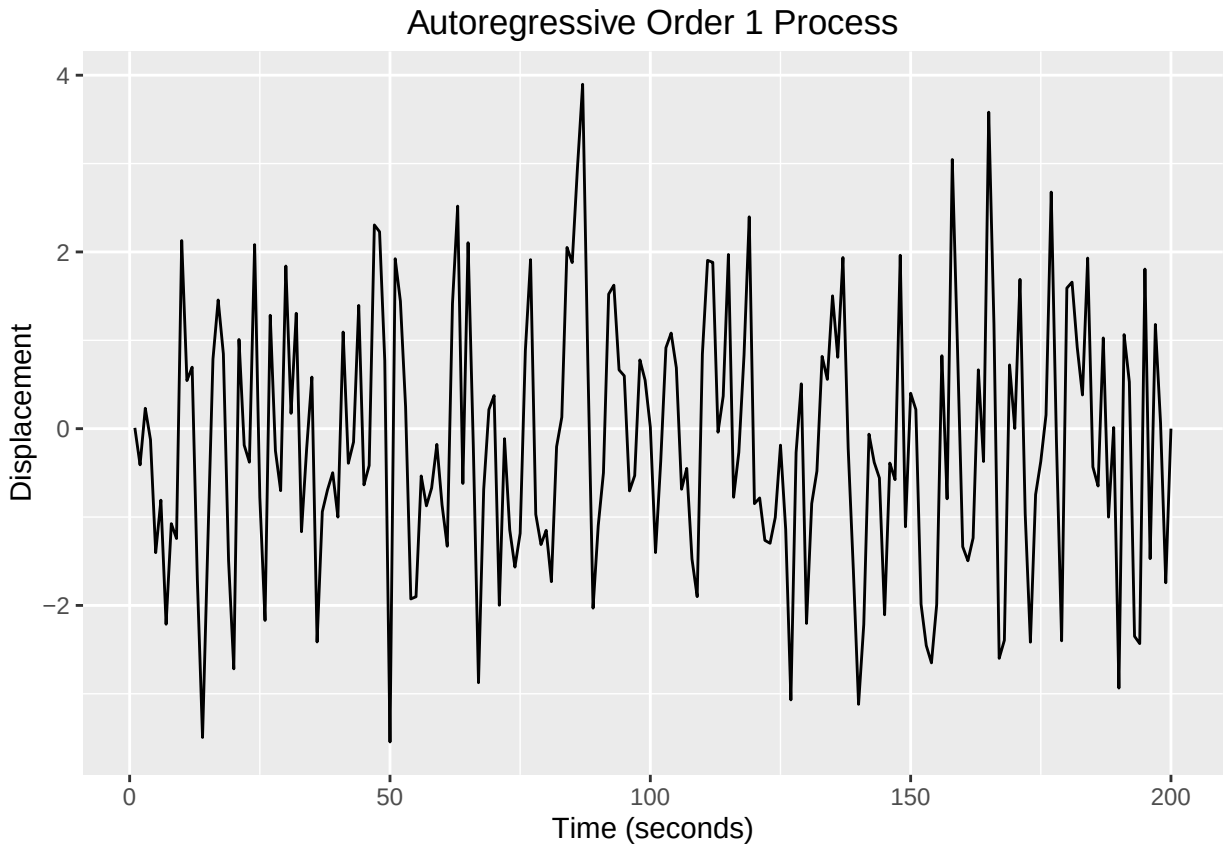
where $w_t \overset{iid}{\sim} WN\left(0, \sigma_w^2\right)$

If $\phi_1 = 1$, then the process is equivalent to a random walk.

The process can be simplified using **backsubstitution** to being:

$$\begin{aligned} y_t &= \phi_t y_{t-1} + w_t \\ &= \phi_1 \left(\phi_1 y_{t-2} + w_{t-1}\right) + w_t \\ &= \phi_1^2 y_{t-2} + \phi_1 w_{t-1} + w_t \\ &\vdots \\ &= \phi^t y_0 + \sum_{i=0}^{t-1} \phi_1^i w_{t-i} \end{aligned}$$

```r
set.seed(1345)  # Set seed to reproduce the results
n      = 200    # Number of observations to generate
sigma2 = 2      # Controls variance of Guassian white noise.
phi    = 0.3    # Handles the phi component of AR(1)

wn = rnorm(n+1, sd = sqrt(sigma2))

# Simulate the MA(1) process
ar = rep(0,n+1)
for(i in 2:n) {
  ar[i] = phi*ar[i-1] + wn[i]
}

ar = ts(ar[2:(n+1)])

autoplot(ar) +
  ggtitle("Autoregressive Order 1 Process") +
  ylab("Displacement") + xlab("Time (seconds)")
```



Chapter 4

ARMA

4.1 Definition

4.2 MA / AR Operators

4.3 Redundancy

4.4 Causal + Invertible

4.5 Estimation of Parameters

Consider a time series given by $x_t \sim ARMA(p, q)$. This gives us with a paramter space Ω that looks like so:

$$\vec{\varphi} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_p \\ \theta_1 \\ \vdots \\ \theta_q \\ \sigma^2 \end{bmatrix}$$

In order to estimate this parameter space, we must assume the following three conditions:

1. The process is casual
2. The process is invertible
3. The process has Gaussian innovations.

Innovations are a time series equivalent to residuals. That is, an innovation is given by $x_t - \hat{x}_t^{t-1}$, where $\hat{x}_t^{t-1}$ is the prediction at time t given $t-1$ observations and x_t is the true value observed at time t.

There are two main ways of performing such an estimation of the parameter space.

1. Maximum Likelihood / Least Squares Estimation [MLE / LSE]

2. Method of Moments (MoM)

To begin, we'll explore using the MLE to perform the estimation.

4.5.1 Maximum Likelihood Estimation

Definition Consider $X_n = (X_1, X_2, \ldots, X_n)$ with the joint density $f(X_1, X_2, \ldots, X_n; \theta)$ where $\theta \in \Theta$. Given $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$ is observed, we have the likelihood function of θ as

$$L(\theta) = L(\theta | x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n | \theta)$$

If the X_i are iid, then the likelihood simplifies to:

$$L(\theta) = \prod_{i=1}^{n} f\left(x_i | \theta\right)$$

However, that's a bit painful to maximize with calculus. So, we opt to use the log of the function since derivatives are easier and the logarithmic function is always increasing. Thus, we traditionally use:

$$l\left(\theta\right) = \log\left(L\left(\theta\right)\right) = \sum_{i=1}^{n} \log\left(f\left(x_i | \theta\right)\right)$$

From maximizes the likelihood function $L(\theta)$, we get the **maximum likelihood estimate (MLE)** of θ. So, we end up with a value that makes the observed data the "most probable."

Note: The likelihood function is **not** a probability density function.

4.5.1.1 $AR(1)$ with mean μ

Consider an $AR(1)$ process given as $y_t = \phi y_{t-1} + w_t$, $w_t \overset{iid}{\sim} N\left(0, \sigma^2\right)$, with $E\left[y_t\right] = 0$, $|\phi| < 1$.

Let $x_t = y_t + \mu$, so that $E\left[x_t\right] = \mu$.

Then, $x_t - \mu = y_t$. Substituting in for y_t, we get:

$$\begin{aligned} y_t &= \phi y_{t-1} + w_t \\ \underbrace{(x_t - \mu)}_{=y_t} &= \phi \underbrace{(x_{t-1} - \mu)}_{=y_t} + w_t \\ x_t &= \mu + \phi\left(x_{t-1} - \mu\right) + w_t \end{aligned}$$

In this case, x_t is an $AR(1)$ process with mean μ.

This means that we have:

1. $E\left[x_t\right] = \mu$

2.

$$\begin{aligned} Var(x_t) &= Var(x_t - \mu) \\ &= Var(y_t) \\ &= Var\left(\sum_{j=0}^{\infty} \phi^j w_{t-j}\right) \\ &= \sum_{j=0}^{\infty} \phi^{2j} Var(w_{t-j}) \\ &= \sigma^2 \sum_{j=0}^{\infty} \phi^{2j} \\ &= \frac{\sigma^2}{1-\phi^2}, \text{ since } |\phi| < 1 \text{ and } \sum_{k=0}^{n} ar^k = \frac{a}{1-r} \end{aligned}$$

So, $x_t \sim N\left(\mu, \frac{\sigma^2}{1-\phi^2}\right)$.

Note that the distribution of x_t is normal and, thus, the density function of x_t is given by:

$$\begin{aligned} f(x_t) &= \sqrt{\frac{1-\phi^2}{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \cdot \frac{1-\phi^2}{\sigma^2} \cdot (x_t - \mu)^2\right) \\ &= (2\pi)^{-\frac{1}{2}} \left(\sigma^2\right)^{-\frac{1}{2}} \left(1-\phi^2\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2} \cdot \frac{1-\phi^2}{\sigma^2} \cdot (x_t - \mu)^2\right) \quad [1] \end{aligned}$$

We'll call the last equation [1].

4.5.1.2 Conditioning time $x_t|x_{t-1}$

Now, consider $x_t|x_{t-1}$ for $t > 1$.

The mean is given by:

$$\begin{aligned} E[x_t|x_{t-1}] &= E[\mu + \phi(x_{t-1} - \mu) + w_t|x_{t-1}] \\ &= \mu + \phi(x_{t-1} - \mu) \end{aligned}$$

This is the case since $E[x_{t-1}|x_{t-1}] = x_{t-1}$ and $E[w_t|x_{t-1}] = 0$

Now, the variance is:

$$\begin{aligned} Var(x_t|x_{t-1}) &= Var(\mu + \phi(x_{t-1} - \mu) + w_t|x_{t-1}) \\ &= \underbrace{Var(\mu + \phi(x_{t-1} - \mu)|x_{t-1})}_{=0} + Var(w_t|x_{t-1}) \\ &= Var(w_t) \\ &= \sigma^2 \end{aligned}$$

Thus, we have: $x_t \sim N\left(\mu + \phi(x_{t-1} - \mu), \sigma^2\right)$.

Again, note that the distribution of x_t is normal and, thus, the density function of x_t is given by:

$$\begin{aligned} f(x_t) &= \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \cdot [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2\right) \\ &= (2\pi)^{-\frac{1}{2}} \left(\sigma^2\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2} \cdot [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2\right) \quad [2] \end{aligned}$$

And for this equation we'll call it [2].

4.5.2 MLE for σ^2 on $AR(1)$ with mean μ

Whew, with all of the above said, we're now ready to obtain an MLE estimate on an $AR(1)$.

Let $\vec{\theta} = \begin{bmatrix} \mu \\ \phi \\ \sigma^2 \end{bmatrix}$, then the likelihood of $\vec{\theta}$ is given by $x_1, \ldots, x_T$ is:

$$\begin{aligned} L\left(\vec{\theta}|x_1, \ldots, x_T\right) &= f\left(x_1, \ldots, x_T|\vec{\theta}\right) \\ &= f\left(x_1\right) \cdot \prod_{t=2}^{T} f\left(x_t|x_{t-1}\right) \end{aligned}$$

The last equality is the result of us using a lag 1 of "memory." Also, note that $x_t|x_{t-1}$ must have $t > 1 \in \mathbb{N}$. Furthermore, we have dropped the parameters in the densities, e.g. $\vec{\theta}$ in $f(\cdot)$, to ease notation.

Using equations [1] and [2], we have:

$$L\left(\vec{\theta}|x_1, \ldots, x_T\right) = (2\pi)^{-\frac{T}{2}} \left(\sigma^2\right)^{-\frac{T}{2}} \left(1-\phi^2\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\left[\left(1-\phi^2\right)\left(x_t-\mu\right)^2 + \sum_{t=2}^{T}\left[\left(x_t-\mu\right) - \phi\left(x_{t-1}-\mu\right)\right]^2\right]\right)$$

For convenience, we'll define:

$$S\left(\mu, \phi\right) = \left(1-\phi^2\right)\left(x_t-\mu\right)^2 + \sum_{t=2}^{T}\left[\left(x_t-\mu\right) - \phi\left(x_{t-1}-\mu\right)\right]^2$$

Fun fact, this is called the "**unconditional** sum of squares."

Thus, we will operate on:

$$L\left(\vec{\theta}|x_1, \ldots, x_T\right) = (2\pi)^{-\frac{T}{2}} \left(\sigma^2\right)^{-\frac{T}{2}} \left(1-\phi^2\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2} S\left(\mu, \phi\right)\right)$$

Taking the log of this yields:

$$\begin{aligned} l\left(\vec{\theta}|x_1, \ldots, x_T\right) &= \log\left(L\left(\vec{\theta}|x_1, \ldots, x_T\right)\right) \\ &= -\frac{T}{2}\log\left(2\pi\right) - \frac{T}{2}\log\left(\sigma^2\right) + \frac{1}{2}\left(1-\phi^2\right) - \frac{1}{2\sigma^2} S\left(\mu, \phi\right) \end{aligned}$$

Now, taking the derivative and solving for the maximized point gives:

$$\begin{aligned} \frac{\partial}{\partial\sigma^2} l\left(\vec{\theta}|x_1, \ldots, x_T\right) &= -\frac{T}{2\sigma^2} + \frac{1}{2\sigma^4} S\left(\mu, \phi\right) \\ 0 &= -\frac{T}{2\sigma^2} + \frac{1}{2\sigma^4} S\left(\mu, \phi\right) \\ \frac{T}{2\sigma^2} &= \frac{1}{2\sigma^4} S\left(\mu, \phi\right) \\ \sigma^2 &= \frac{1}{T} S\left(\mu, \phi\right) \end{aligned}$$

Thus, the MLE for $\hat{\sigma}^2 = \frac{1}{T}S\left(\hat{\mu}, \hat{\phi}\right)$, where $\hat{\mu}$ and $\hat{\phi}$ are the MLEs for μ, ϕ that are obtained numerically via either *Newton Raphson* or a *Scoring Algorithm.* (More details in a numerical recipe book.)

4.5.2.1 Conditional MLE on $AR(1)$ with mean μ

A common strategy to reduce the dependency on numerical recipes is to simplify $l\left(\vec{\theta}|x_1, \ldots, x_T\right)$ by using $l^*\left(\vec{\theta}|x_1, \ldots, x_T\right)$:

$$\begin{aligned} l^*\left(\vec{\theta}|x_1, \ldots, x_T\right) &= \prod_{t=2}^{T} \log\left(f\left(x_t|x_{t-1}\right)\right) \\ &= \prod_{t=2}^{T} \log\left(\left(2\pi\right)^{-\frac{1}{2}}\left(\sigma^2\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2} \cdot \left[\left(x_t - \mu\right) - \phi\left(x_{t-1} - \mu\right)\right]^2\right)\right) \\ &= -\frac{\left(T-1\right)}{2}\log\left(2\pi\right) - \frac{\left(T-1\right)}{2}\log\left(\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{t=2}^{T}\left[\left(x_t - \mu\right) - \phi\left(x_{t-1} - \mu\right)\right]^2 \end{aligned}$$

Again, for convenience, we'll define:

$$S_c\left(\mu, \phi\right) = \sum_{t=2}^{T}\left[\left(x_t - \mu\right) - \phi\left(x_{t-1} - \mu\right)\right]^2$$

Fun fact, this is called the "**conditional** sum of squares."

So, we will use:

$$l^*\left(\vec{\theta}|x_1, \ldots, x_T\right) = -\frac{\left(T-1\right)}{2}\log\left(2\pi\right) - \frac{\left(T-1\right)}{2}\log\left(\sigma^2\right) - \frac{1}{2\sigma^2}S_c\left(\mu, \phi\right)$$

Taking the derivative with respect to μ gives:

$$\begin{aligned} \frac{\partial}{\partial\mu}l^*\left(\vec{\theta}|x_1, \ldots, x_T\right) &= -\frac{1}{2\sigma^2}\sum_{t=2}^{T} 2\left[\left(x_t - \mu\right) - \phi\left(x_{t-1} - \mu\right)\right]\left(\phi - 1\right) \\ &= \frac{1-\phi}{\sigma^2}\sum_{t=2}^{T}\left[\left(x_t - \mu\right) - \phi\left(x_{t-1} - \mu\right)\right] \\ &= \frac{1-\phi}{\sigma^2}\sum_{t=2}^{T}\left(x_t - \phi x_{t-1} - \mu\left(1-\phi\right)\right) \\ &= -\frac{\left(1-\phi\right)^2}{\sigma^2}\mu\left(T-1\right) + \frac{\left(1-\phi\right)}{\sigma^2}\sum_{t=2}^{T}\left(x_t - \phi x_{t-1}\right) \end{aligned}$$

Solving for μ^* gives:

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \mu} l^* \left(\vec{\theta} | x_1, \ldots, x_t \right) \\
0 &= -\frac{(1-\phi)^2}{\sigma^2} \mu^* (T-1) + \frac{(1-\phi^*)}{\sigma_*^2} \sum_{t=2}^{T} (x_t - \phi^* x_{t-1}) \\
\frac{(1-\phi^*)^2}{\sigma_*^2} \mu^* (T-1) &= \frac{(1-\phi^*)}{\sigma_*^2} \sum_{t=2}^{T} (x_t - \phi^* x_{t-1}) \\
\mu^* (1-\phi^*) (T-1) &= \sum_{t=2}^{T} (x_t - \phi^* x_{t-1}) \\
\mu^* &= \frac{1}{(1-\phi^*)(T-1)} \sum_{t=2}^{T} (x_t - \phi^* x_{t-1}) \\
\mu^* &= \frac{1}{1-\phi^*} \left[\underbrace{\frac{1}{T-1} \sum_{t=2}^{T} x_t}_{=\bar{x}_{(2)}} - \underbrace{\frac{\phi^*}{T-1} \sum_{t=2}^{T} x_{t-1}}_{=\bar{x}_{(1)}} \right] \\
\hat{\mu}^* &= \frac{1}{1-\phi^*} \left(\bar{x}_{(2)} - \phi \bar{x}_{(1)} \right)
\end{aligned}
$$

When T is large, we have the following:

$$
\bar{x}_{(1)} \approx \bar{x}, \bar{x}_{(2)} \approx \bar{x}
$$

$$
\begin{aligned}
\hat{\mu}^* &= \frac{1}{1-\phi^*} (\bar{x} - \phi^* \bar{x}) \\
&= \frac{\bar{x}}{1-\phi^*} (1-\phi^*) \\
&= \bar{x}
\end{aligned}
$$

Taking the derivative with respect to σ^2 and solving for σ^2 gives:

$$
\begin{aligned}
\frac{\partial}{\partial \sigma^2} l^* \left(\vec{\theta} | x_1, \ldots, x_T \right) &= -\frac{(T-1)}{2\sigma_*^2} + \frac{1}{2\sigma_*^4} S_c (\mu, \phi) \\
0 &= -\frac{(T-1)}{2\sigma_*^2} + \frac{1}{2\sigma_*^4} S_c (\mu, \phi) \\
\frac{(T-1)}{2\sigma_*^2} &= \frac{1}{2\sigma_*^4} S_c (\mu, \phi) \\
\hat{\sigma}_*^2 &= \frac{1}{T-1} S_c \left(\hat{\mu}^*, \hat{\phi}^* \right)
\end{aligned}
$$

Taking the derivative with respect to ϕ gives:

$$\begin{aligned}
\frac{\partial}{\partial\phi}l^*\left(\vec{\theta}|x_1,\ldots,x_T\right) &= -\frac{1}{2\sigma^2}\sum_{t=2}^{T}-2\left[\left(x_t-\mu\right)-\phi\left(x_{t-1}-\mu\right)\right]\left(x_{t-1}-\mu\right)\\
&= \frac{1}{\sigma^2}\sum_{t=2}^{T}\left[x_t-\phi x_{t-1}-\mu\left(1-\phi\right)\right]\left(x_{t-1}-\mu\right)\\
&= \frac{1}{\sigma^2}\sum_{t=2}^{T}\left[x_tx_{t-1}-\phi x_{t-1}^2-\mu\left(1-\phi\right)x_{t-1}-\mu x_t+\mu\phi x_{t-1}+\mu^2\left(1-\phi\right)\right]\\
&= \frac{1}{\sigma^2}\left[\begin{array}{l}\sum_{t=2}^{T}x_tx_{t-1}-\phi\sum_{t=2}^{T}x_{t-1}^2-\mu\left(1-\phi\right)\left(T-1\right)\bar{x}_{(1)}\\-\mu\left(T-1\right)\bar{x}_{(2)}+\phi\mu\left(T-1\right)\bar{x}_{(1)}+\mu^2\left(1-\phi\right)\left(T-1\right)\end{array}\right]
\end{aligned}$$

Solving for ϕ gives:

$$\begin{aligned}
0 &= \frac{\partial}{\partial\phi}l^*\left(\vec{\theta}|x_1,\ldots,x_T\right)\\
0 &= \sum_{t=2}^{T}x_tx_{t-1}-\hat{\phi}^*\sum_{t=2}^{T}x_{t-1}^2-\left(\bar{x}_{(2)}-\hat{\phi}^*\bar{x}_{(1)}\right)\left(T-1\right)\bar{x}_{(1)}-\frac{\bar{x}_{(2)}-\hat{\phi}^*\bar{x}_{(1)}}{1-\hat{\phi}^*}\left(T-1\right)\bar{x}_{(2)}\\
&\quad+\hat{\phi}^*\frac{\bar{x}_{(2)}-\hat{\phi}^*\bar{x}_{(1)}}{1-\hat{\phi}^*}\left(T-1\right)\bar{x}_{(1)}+\left(\frac{\bar{x}_{(2)}-\hat{\phi}^*\bar{x}_{(1)}}{1-\hat{\phi}^*}\right)^2\left(1-\hat{\phi}^*\right)\left(T-1\right)\\
&\vdots\\
&\text{Magic}\\
&\vdots\\
\hat{\phi}^* &= \frac{\sum_{t=2}^{T}\left(x_t-\bar{x}_{(2)}\right)\left(x_{t-1}-\bar{x}_{(1)}\right)}{\sum_{t=2}^{T}\left(x_{t-1}-\bar{x}_{(1)}\right)^2}
\end{aligned}$$

When T is large, we have:

$$\begin{aligned}
\sum_{t=2}^{T}\left(x_t-\bar{x}_{(2)}\right)\left(x_t-\bar{x}_{(1)}\right) &\approx \sum_{t=2}^{T}\left(x_t-\bar{x}\right)\left(x_{t-1}-\bar{x}\right)\\
\sum_{t=2}^{T}\left(x_{t-1}-\bar{x}_{(1)}\right)^2 &\approx \sum_{t=1}^{T}\left(x_t-\bar{x}\right)^2
\end{aligned}$$

$$\hat{\phi}^* = \frac{\sum_{t=2}^{T}\left(x_t-\bar{x}_{(2)}\right)\left(x_t-\bar{x}_{(1)}\right)}{\sum_{t=2}^{T}\left(x_{t-1}-\bar{x}_{(1)}\right)^2} \approx \frac{\sum_{t=2}^{T}\left(x_t-\bar{x}\right)\left(x_{t-1}-\bar{x}\right)}{\sum_{t=1}^{T}\left(x_t-\bar{x}\right)^2} = \hat{\rho}\left(1\right)$$

4.6 Method of Moments

The goal behind the estimation with Method of Moments is to match the theoretical moment (e.g. $E\left[x_t^k\right]$) with the sample moment (e.g $\frac{1}{n}\sum_{i=1}^{n} x_i^k$), where k denotes the moment.

This method often leads to suboptimal estimates for general ARMA models. However, it is quite optimal for $AR(p)$.

4.6.1 Method of Moments - AR(p)

Consider an $AR(p)$ process represented by:

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t$$

where $w_t \sim N(0, \sigma^2)$

To begin, we find the Covariance of the process when $h > 0$:

$$\begin{aligned} Cov\left(x_{t+h}, x_t\right) &\overset{(h>0)}{=} Cov\left(\phi_1 x_{t+h-1} + \cdots + \phi_p x_{t+h-p} + w_{t+h}, x_t\right) \\ &= \phi_1 Cov\left(x_{t+h-1}, x_t\right) + \cdots + \phi_p Cov\left(x_{t+h-p}, x_t\right) + Cov\left(w_{t+h}, x_t\right) \\ &= \phi_1 \gamma\left(h-1\right) + \cdots + \phi_p \gamma\left(h-p\right) \end{aligned}$$

Now, we turn our attention to the variance of the process:

$$\begin{aligned} Var\left(w_t\right) &= Cov\left(w_t, w_t\right) \\ &= Cov\left(w_t, w_t\right) + \underbrace{Cov\left(\phi_1 x_{t-1}, w_t\right)}_{=0} + \cdots + \underbrace{Cov\left(\phi_p x_{t-p}, w_t\right)}_{=0} \\ &= Cov\left(\underbrace{\phi_1 x_{t-1} + \cdots + \phi_p x_p + w_t}_{=x_t}, w_t\right) \\ &= Cov\left(x_t, w_t\right) \\ &= Cov\left(x_t, x_t - \phi_1 x_{t-1} - \cdots - \phi_p x_p\right) \\ &= Cov\left(x_t, x_t\right) - \phi_1 Cov\left(x_t, x_{t-1}\right) - \cdots - \phi_p Cov\left(x_t, x_{t-p}\right) \\ &= \gamma\left(0\right) - \phi_1 \gamma\left(1\right) - \cdots - \phi_p \gamma\left(p\right) \end{aligned}$$

Together, these equations are known as the **Yule-Walker** equations.

4.6.2 Yule-Walker

Definition

Equation form:

$$\begin{aligned} \gamma\left(h\right) &= \phi_1 \gamma\left(h-1\right) - \cdots - \phi_p \gamma\left(h-p\right) \\ \sigma^2 &= \gamma\left(0\right) - \phi_1 \gamma\left(1\right) - \cdots - \phi_p \gamma\left(p\right) \\ h &= 1, \ldots, p \end{aligned}$$

Matrix form:

$$\Gamma\vec{\phi} = \vec{\gamma}$$
$$\sigma^2 = \gamma(0) - \vec{\phi}^T\vec{\gamma}$$

$$\vec{\phi} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_p \end{bmatrix}_{p\times 1}, \vec{\gamma} = \begin{bmatrix} \gamma(1) \\ \vdots \\ \gamma(p) \end{bmatrix}_{p\times 1}, \Gamma = \{\gamma(k-j)\}_{j,k=1}^{p}$$

More aptly, the structure of Γ looks like the following:

$$\Gamma = \begin{bmatrix} \gamma(0) & \gamma(-1) & \gamma(-2) & \cdots & \gamma(1-p) \\ \gamma(1) & \gamma(0) & \gamma(-1) & \cdots & \gamma(2-p) \\ \gamma(2) & \gamma(1) & \gamma(0) & \cdots & \gamma(3-p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \gamma(p-3) & \cdots & \gamma(0) \end{bmatrix}_{p\times p}$$

Note, that we are able to use the above equations to effectively estimate $\vec{\phi}$ and σ^2.

$$\begin{bmatrix} \hat{\vec{\phi}} = \hat{\Gamma}^{-1}\hat{\vec{\gamma}} \\ \hat{\sigma}^2 = \hat{\gamma}(0) - \hat{\vec{\gamma}}^T\hat{\Gamma}^{-1}\hat{\vec{\gamma}} \end{bmatrix} \rightarrow \text{Yule - Walker Estimates}$$

For the second equation, we are effectively substituting in the first equation for $\hat{\vec{\phi}}$, hence the quadratic form $\hat{\vec{\gamma}}^T\hat{\Gamma}^{-1}\hat{\vec{\gamma}}$.

With this being said, there are a few nice asymptotic properties that we obtain for an $AR(p)$.

1. $\sqrt{T}\left(\hat{\vec{\phi}} - \vec{\phi}\right) \underset{t\to\infty}{\xrightarrow{L}} N\left(\vec{0}, \sigma^2\Gamma^{-1}\right)$
2. $\hat{\sigma}^2 \xrightarrow{p} \sigma^2$

Yule-Walker estimates are optimal in the sense that they have the smallest asymptotic variance i.e.

$$Var\left(\sqrt{T}\hat{\vec{\phi}}\right) = \sigma^2\Gamma^{-1}$$

However, they are not necessarily optimal with small sample sizes.

Conceptually, the reason for this optimality result is a consequence from the linear dependence between moments and variables.

This is not true for MA or ARMA, which are both nonlinear and suboptimal.

4.6.3 Estimates

Consider x_t as an $MA(1)$ process: $x_t = \theta w_{t-1} + w_t, w_t \overset{i.i.d}{\sim} N\left(0, \sigma^2\right)$

Finding the covariance when $h = 1$ gives:

$$\begin{aligned} Cov(x_t, x_{t-1}) &= Cov(\theta w_{t-1} + w_t, \theta w_{t-2} + w_{t-1}) \\ &= Cov(\theta w_{t-1}, w_{t-1}) \\ &= \theta\sigma^2 \end{aligned}$$

Finding the variance (e.g. $h = 0$) gives:

$$\begin{aligned} Cov(x_t, x_t) &= Cov(\theta w_{t-1} + w_t, \theta w_{t-1} + w_t) \\ &= \theta^2 Cov(w_{t-1}, w_{t-1}) + \underbrace{2\theta Cov(w_{t-1}, w_t)}_{=0} + Cov(w_t, w_t) \\ &= \theta^2\sigma^2 + \sigma^2 \\ &= \sigma^2\left(1 + \theta^2\right) \end{aligned}$$

This gives us the MA(1) ACF of:

$$\rho(h) = \begin{cases} 1 & h = 0 \\ \frac{\theta}{\theta^2+1} & h = \pm 1 \end{cases}$$

With this in mind, let's solve for possible θ values:

$$\begin{aligned} \rho(1) &= \frac{\theta}{\theta^2 + 1} \\ \Rightarrow \theta &= \left(\theta^2 + 1\right)\rho(1) \\ \theta &= \rho(1)\theta^2 + \rho(1) \\ 0 &= \rho(1)\theta^2 - \theta + \rho(1) \end{aligned}$$

Yuck, that looks nasty. Let's dig out an ol' friend from middle school known as the quadratic formula:

$$\theta = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Applying the quadratic formula leads to:

$$\begin{aligned} a = \rho(h), b = -1, c &= \rho(h) \\ \theta &= \frac{1 \pm \sqrt{1^2 - 4\rho(h)\rho(h)}}{2\rho(h)} \\ \theta &= \frac{1 \pm \sqrt{1 - 4[\rho(h)]^2}}{2\rho(h)} \end{aligned}$$

Thus, we have two possibilities:

$$\begin{aligned} \theta_1 &= \frac{1 + \sqrt{1 - 4[\rho(h)]^2}}{2\rho(h)} \\ \theta_2 &= \frac{1 - \sqrt{1 - 4[\rho(h)]^2}}{2\rho(h)} \end{aligned}$$

To ensure invertibility, we mandate that $|\rho(1)| < \frac{1}{2}$. Thus, we opt for θ_2.

So, our estimator is:

$$\hat{\theta} = \frac{1 - \sqrt{1 - 4[\hat{\rho}(1)]^2}}{2\hat{\rho}(1)}$$

Furthermore, it can be shown that:

$$\sqrt{T}\left(\hat{\theta} - \theta\right) \underset{T\to\infty}{\overset{L}{\to}} N\left(0, \frac{1 + \theta^2 + 4\theta^4 + \theta^6 + \theta^8}{\left(1 - \theta^2\right)^2}\right)$$

So, this is not a really optimal estimator…

4.7 Prediction (Forecast)

Chapter 5

Linear Regression with Autocorrelated Errors

In this chapter we discuss how the classical linear regression setting can be extended to accomodate for autocorrelated error. Before considering this more general setting, we start by discussing the usual linear regression model with Gaussian errors, i.e.

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\varepsilon}^2 \mathbf{I}\right),$$

where $\boldsymbol{X}$ is a known $n \times p$ design matrix of rank p and $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown parameters. Under this setting, the MLE and LSE are equivalent (due to normality of $\boldsymbol{\varepsilon}$) and corresponds to the ordinary LS parameter estimates of $\boldsymbol{\beta}$, i.e.

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}, \tag{5.1}$$

leading to the (linear) prediction

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{S}\boldsymbol{y}$$

where $\boldsymbol{S} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$ denotes the "*hat*" matrix. The unbiased and maximum likelihood estimates of σ_{ε}^2 are, respectively, given by

$$\tilde{\sigma}_{\varepsilon}^2 = \frac{||\boldsymbol{y} - \hat{\boldsymbol{y}}||_2^2}{n-p} \quad \text{and} \quad \hat{\sigma}_{\varepsilon}^2 = \frac{||\boldsymbol{y} - \hat{\boldsymbol{y}}||_2^2}{n}, \tag{5.2}$$

where $||\cdot||_2$ denotes the L_2 norm. Throughout this chapter we assume that $0 < \sigma_{\varepsilon}^2 < \infty$. Under this setting (i.e. Gaussian iid errors) $\tilde{\sigma}_{\varepsilon}^2$ is distributed proportiinally to χ^2 random vcaraible with $n-p$ degrees of freedom independent of $\hat{\boldsymbol{\beta}}$ (a proof of this result can for example be found in ?????). Consequently, it follow that

$$\frac{\hat{\beta}_i - \beta_i}{(\boldsymbol{C})_i} \sim t_{n-p}, \tag{5.3}$$

where $(\boldsymbol{C})_i$ denotes the i-th diagonal element of the following matrix

$$\boldsymbol{C} = \operatorname{cov}\left(\hat{\boldsymbol{\beta}}\right) = \sigma_{\varepsilon}^2\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}, \tag{5.4}$$

and where $\hat{\beta}_i$ denotes the i-th element of $\hat{\boldsymbol{\beta}}$. Thus, this allows for a natural approach for testing coefficients and selecting models. Moreover, a common quantity used ton evaluate the "quality" of a model is the R^2, which corresponds to the proportion of variation explained by the model, i.e.

$$R^2 = \frac{\sum_{i=1}^n (y_i - \hat{\boldsymbol{y}}_i)^2 - \sum_{i=1}^n (y_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

where y_i and $\hat{y}_i$ denote, respectively, the i-th element of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, and $\bar{y}$ represent the mean value of the vector $\boldsymbol{y}$. This goodness-of-fit is widely used in practice but its limits are often misunderstood as illustrated in the example below.

Example: Suppose that we have two *nested* models, say $\mathcal{M}_1$ and $\mathcal{M}_2$, i.e.

$$\begin{aligned} \mathcal{M}_1 : &\quad \boldsymbol{y} = \boldsymbol{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}, \\ \mathcal{M}_2 : &\quad \boldsymbol{y} = \boldsymbol{X}_1\boldsymbol{\beta}_1 + \boldsymbol{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}, \end{aligned}$$

and assume that $\boldsymbol{\beta}_2 = \boldsymbol{0}$. In this case, it is interesting to compare the R^2 of both models, say R_1^2 and R_2^2. Using $\hat{\boldsymbol{y}}_i$ to denote the predictions made from model $\mathcal{M}_i$, we have that

$$||\boldsymbol{y} - \hat{\boldsymbol{y}}_1||_2^2 \geq ||\boldsymbol{y} - \hat{\boldsymbol{y}}_2||_2^2.$$

By letting $||\boldsymbol{y} - \hat{\boldsymbol{y}}_1||_2^2 = ||\boldsymbol{y} - \hat{\boldsymbol{y}}_2||_2^2 + c$ where c is a non-negartive constant we obtain:

$$R_1^2 = 1 - \frac{||\boldsymbol{y} - \hat{\boldsymbol{y}}_1||_2^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{||\boldsymbol{y} - \hat{\boldsymbol{y}}_2||_2^2 + c}{\sum_{i=1}^n (y_i - \bar{y})^2} = R_2^2 + \frac{c}{\sum_{i=1}^n (y_i - \bar{y})^2}.$$

This implies that $R_1^2 \leq R_2^2$, regardelss of the value of $\boldsymbol{\beta}_2$ and therefore the R^2 is essentially useless in terms of model selection. This results is well known and is further discuss in ??????REF *REGRESSION and TIME SERIES MODEL SELECTION< TSAI< CHAP 2*.

A more approriate measure of the goodness-of-fit of a particular model is for example Mallow's C_p introduced in **REF see STEF PHD**. This metric balances the error of fit against its complexity and can be defined as

$$C_p = ||\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2 + 2\hat{\sigma}_*^2 p, \tag{5.5}$$

where $\hat{\sigma}_*^2$ is an unbiased estimates of σ_ε^2, generally $\tilde{\sigma}_\varepsilon^2$ computed on a "low-bias" model (i.e. a sufficiently "large" model).

To understand how this result is derived, we let $\boldsymbol{y}_0$ denote an independent "copy" of $\boldsymbol{y}$ issued from the same data-generating process and let $E_0[\cdot]$ denotes the expectation under the distribution of $\boldsymbol{y}_0$ (conditionally on $\boldsymbol{X}$). Then, it can be argued that the following quantity is approriate at measuring the adequacy of model as it compares how $\boldsymbol{y}$ can be used to predict $\boldsymbol{y}_0$,

$$E\left[E_0\left[||\boldsymbol{y}_0 - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right]\right].$$

As we will see Mallow's C_p is an unbiased estimator of this quantity. There are several ways of showing it, one of them is presented here using the following "*optimism*" theorem. Note that this result is based on Theorem 2.1 of **REF MISSING, TWO HERE PHD STEF** and on the Optimism Theorem of ** REF MISSING EFRON COVARIACNE PAPER 2004 JASA**.

Theorem: Let $\boldsymbol{y}_0$ denote an independent "copy" of $\boldsymbol{y}$ issued from the same data-generating process and let $E_0[\cdot]$ denotes the expectation under the distribution of $\boldsymbol{y}_0$ (conditionally on $\boldsymbol{X}$). Then we have that,

$$E\left[E_0\left[||\boldsymbol{y}_0 - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right]\right] = E\left[||\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right] + 2\,\mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{y}, \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right).$$

Proof: We first expend $||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2$ as follows:

$$||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 = \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - 2\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right)^T\boldsymbol{X}\boldsymbol{\beta}.$$

Then, we define C and C* and used the above expension

$$\begin{aligned}
\mathrm{C} &= E\left[E_0\left[||\boldsymbol{y}_0 - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right]\right] = E_0\left[\boldsymbol{y}_0^T\boldsymbol{y}_0\right] - E\left[\hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}\right] - 2E\left[\left(E_0\left[\boldsymbol{y}_0\right] - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\boldsymbol{X}\hat{\boldsymbol{\beta}}\right], \\
\mathrm{C}^* &= E\left[||\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right] = E\left[\boldsymbol{y}^T\boldsymbol{y}\right] - E\left[\hat{\boldsymbol{\beta}}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\beta}}\right] - 2E\left[\left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\boldsymbol{X}\hat{\boldsymbol{\beta}}\right].
\end{aligned}$$

Next, we consider the difference between C and C*, i.e.

$$\begin{aligned}
\mathrm{C} - \mathrm{C}^* &= 2E\left[\left(\boldsymbol{y} - E_0\left[\boldsymbol{y}_0\right]\right)^T\boldsymbol{X}\hat{\boldsymbol{\beta}}\right] = 2\,\mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{y} - E_0[\boldsymbol{y}_0], \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right) + 2\,\mathrm{tr}\left(E\left[\boldsymbol{y} - E_0[\boldsymbol{y}_0]\right]E^T[\boldsymbol{X}\hat{\boldsymbol{\beta}}]\right) \\
&= 2\,\mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{y} - E_0[\boldsymbol{y}_0], \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right) = 2\,\mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{y}, \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right),
\end{aligned}$$

which concludes our proof. Note that in the above equation we used the following equality, which is based on two vector valued random variation of approriate dimensions:

$$E\left[\boldsymbol{X}^T\boldsymbol{Z}\right] = E\left[\mathrm{tr}\left(\boldsymbol{X}^T\boldsymbol{Z}\right)\right] = E\left[\mathrm{tr}\left(\boldsymbol{Z}\boldsymbol{X}^T\right)\right] = \mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{X}, \boldsymbol{Z}\right)\right) + \mathrm{tr}\left(E[\boldsymbol{X}]E^T[\boldsymbol{Z}]\right).$$

In the linear regression case with iid Gaussian errors we have:

$$\mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{y}, \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right) = \mathrm{tr}\left(\mathrm{cov}\left(\boldsymbol{y}, \boldsymbol{S}\boldsymbol{y}\right)\right) = \sigma_\varepsilon^2\,\mathrm{tr}\left(\boldsymbol{S}\right) = \sigma_\varepsilon^2 p.$$

Therefore,

$$\mathrm{C} = E\left[E_0\left[||\boldsymbol{y}_0 - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right]\right] = E\left[||\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2\right] + 2\sigma_\varepsilon^2 p,$$

yielding to the unbiased estimate

$$\widehat{\mathrm{C}} = C_p = ||\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2 + 2\hat{\sigma}_*^2 p.$$

An alternative famous goodness-of-fit criterion was proposed by Akaike (1969, 1973, 1974) **REF MISSING** and is given by

$$\mathrm{AIC} = \log\left(\hat{\sigma}_\varepsilon^2\right) + \frac{n + 2p}{n}. \tag{5.6}$$

where $\hat{\sigma}_\varepsilon^2$ denotes the MLE for σ_ε^2 defined in 5.2.

The AIC is based on a *divergence* (i.e. a generalization of the notion of distance) that informally speaking measures "how far" is the density of the estimated model compared to the "true" density. This divergence is called the Kullback-Leibler information which in this context can be defined for two densities of the same family as

$$\mathrm{KL} = \frac{1}{n} E\left[E_0\left[\log\left(\frac{f(\boldsymbol{y}_0|\boldsymbol{\theta}_0)}{f(\boldsymbol{y}_0|\hat{\boldsymbol{\theta}})}\right)\right]\right],$$

where we assume $\boldsymbol{\theta}_0$ and $\hat{\boldsymbol{\theta}}$ to denote, respectively, the true parameter vector of interest and an estimator $\boldsymbol{\theta}_0$ based on a postulated model. Similarly to the setting used to derive Mallow's C_p, the expectations $E[\cdot]$ and $E_0[\cdot]$ denote the expectation with respect to the densities of $\boldsymbol{y}$ and $\boldsymbol{y}_0$ (conditionally on $\boldsymbol{X}$). Note that $\hat{\boldsymbol{\theta}}$ dependences on $\boldsymbol{y}$ and not $\boldsymbol{y}_0$. Informally speaking this divergence measure how far is $f(\boldsymbol{y}_0|\boldsymbol{\theta}_0)$ from $f(\boldsymbol{y}_0|\hat{\boldsymbol{\theta}})$, where in the latter $\hat{\boldsymbol{\theta}}$ is estimated on $\boldsymbol{y}$, a sample independent from $\boldsymbol{y}_0$.

To derive the AIC we start by considering a generic a linear model $\mathcal{M}$ with parameter vector $\boldsymbol{\theta} = [\boldsymbol{\beta}^T \quad \sigma_\varepsilon^2]$. Indeed, we have that its density is given by

$$\begin{aligned} f(\boldsymbol{y}|\boldsymbol{\theta}) &= (2\pi)^{-n/2}\left|\sigma_\varepsilon^2\mathbf{I}\right|^{-1/2}\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T\left(\sigma_\varepsilon^2\mathbf{I}\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\beta_i)\right) \\ &= (2\pi)^{-n/2}\left(\sigma_\varepsilon^2\right)^{-n/2}\exp\left(-\frac{1}{2\sigma_\varepsilon^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right). \end{aligned}$$

Using this result and letting

$$\boldsymbol{\theta}_0 = [\boldsymbol{\beta}_0^T \quad \sigma_0^2] \qquad \text{and} \qquad \hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\beta}}^T \quad \hat{\sigma}^2],$$

where $\hat{\boldsymbol{\theta}}$ denotes the MLE for $\hat{\boldsymbol{\theta}}$, we obtain

$$\begin{aligned} \frac{1}{n}E\left[E_0\left[\log\left(\frac{f(\boldsymbol{y}_0|\boldsymbol{\theta}_0)}{f\left(\boldsymbol{y}_0|\hat{\boldsymbol{\theta}}\right)}\right)\right]\right] &= \frac{1}{n}E\left[E_0\left[\log\left(\frac{\left(\sigma_0^2\right)^{-n/2}}{\left(\hat{\sigma}^2\right)^{-n/2}}\right)+\log\left(\frac{\exp\left(-\frac{1}{2\sigma_0^2}(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)^T(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)\right)}{\exp\left(-\frac{1}{2\hat{\sigma}^2}\left(\boldsymbol{y}_0-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y}_0-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right)}\right)\right]\right] \\ &= -\frac{1}{2}E\left[\log\left(\frac{\sigma_0^2}{\hat{\sigma}^2}\right)\right] - \frac{1}{2n\sigma_0^2}E_0\left[(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)^T(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)\right] \\ &\quad + \frac{1}{2n}E\left[\frac{1}{\hat{\sigma}^2}E_0\left[\left(\boldsymbol{y}_0-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y}_0-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right]\right]. \end{aligned}$$

Next, we consider each term of the above equation. For the first term, we have

$$-\frac{1}{2}E\left[\log\left(\frac{\sigma_0^2}{\hat{\sigma}^2}\right)\right] = \frac{1}{2}\left(E\left[\log\left(\hat{\sigma}^2\right)\right] - \log\left(\sigma_0^2\right)\right).$$

For the second term, we obtain

$$-\frac{1}{2n\sigma_0^2}E_0\left[(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)^T(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)\right] = -\frac{1}{2}.$$

Finally, we have for the last term

$$\begin{aligned} \frac{1}{2n}E\left[\frac{1}{\hat{\sigma}^2}E_0\left[\left(\boldsymbol{y}_0-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T\left(\boldsymbol{y}_0-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right]\right] &= \frac{1}{2n}E\left[\frac{1}{\hat{\sigma}^2}E_0\left[\left(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0-\boldsymbol{X}\left(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\right)\right)^T\left(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0-\boldsymbol{X}\left(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\right)\right)\right]\right] \\ &= \frac{1}{2n}E\left[\frac{1}{\hat{\sigma}^2}\left[E_0\left[(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)^T(\boldsymbol{y}_0-\boldsymbol{X}\boldsymbol{\beta}_0)\right]\right]\right] \\ &\quad + \frac{1}{2n}E\left[\frac{1}{\hat{\sigma}^2}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}\right)^T\boldsymbol{X}^T\boldsymbol{X}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}\right)\right] \\ &= \frac{1}{2}E\left[\frac{\sigma_0^2}{\hat{\sigma}^2}\right] + \frac{1}{2n}E\left[\frac{\sigma_0^2}{\hat{\sigma}^2}\frac{\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}\right)^T\boldsymbol{X}^T\boldsymbol{X}\left(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}\right)}{\sigma_0^2}\right]. \end{aligned}$$

To simplify further this result it is usefull to remeber that

$$U_1 = \frac{n\hat{\sigma}^2}{\sigma_0^2} \sim \chi^2_{n-p}, \qquad U_2 = \frac{\left(\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}\right)^T \boldsymbol{X}^T \boldsymbol{X} \left(\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}\right)}{\sigma_0^2} \sim \chi^2_p,$$

and that U_1 and U_2 are independent. Moreover, we have that if $U \sim \chi^2_k$ then $E[1/U] = 1/(k-2)$. Thus, we obtain

$$\frac{1}{2n} E\left[\frac{1}{\hat{\sigma}^2} E_0\left[\left(\boldsymbol{y}_0 - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^T \left(\boldsymbol{y}_0 - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\right]\right] = \frac{n+p}{2(n-p-2)}.$$

Combining, the above result we have

$$\mathrm{KL} = \frac{1}{2}\left[E\left[\log\left(\hat{\sigma}^2\right)\right] + \frac{n+p}{(n-p-2)} + c\right],$$

where $c = -\log\left(\sigma_0^2\right) - 1$. Since the constant c is *common* to all models it can neglated for the purpose of model selection. Therefore, neglecting the constant we obtain that

$$\mathrm{KL} \propto E\left[\log\left(\hat{\sigma}^2\right)\right] + \frac{n+p}{(n-p-2)}.$$

Thus, an unbiased estimator of KL is given by

$$\mathrm{AICc} = \log\left(\hat{\sigma}^2\right) + \frac{n+p}{(n-p-2)},$$

since an unbiased estimator of $\log\left(\hat{\sigma}^2\right)$ is simply $\log\left(\hat{\sigma}^2\right)$. However, it can be observed that the result we derived is not equal to the AIC defined in (5.6). Indeed, this result is known as the bias-corrected AIC or AICc. To understand the relationship between the AIC and AICc it is instructif to consider their difference and letting n diverge to infinity, i.e.

$$\lim_{n\to\infty} \mathrm{AIC} - \mathrm{AICc} = \frac{2\left(p^2+2p+n\right)}{n\left(p-n-2\right)} = 0.$$

Therefore, the AIC is an asymptotically unbiased estimator of KL. In practice, the AIC and AICc provides very similar result expect when the sample size is rather small.

TO DO Talk about BIC

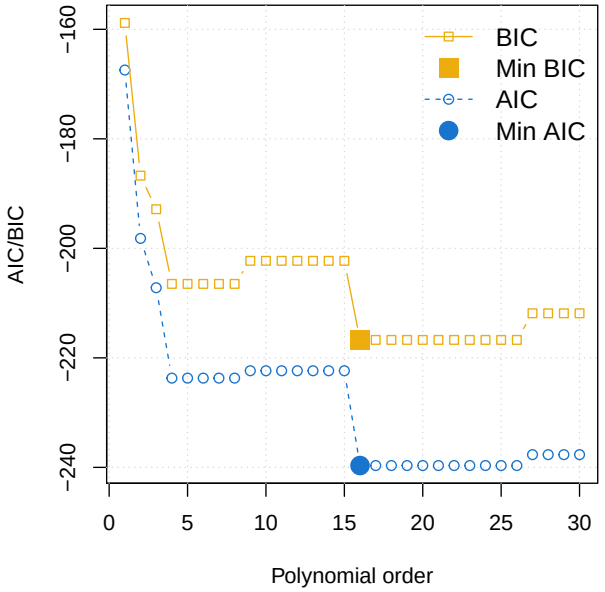
Illustration for model selection with linear model:

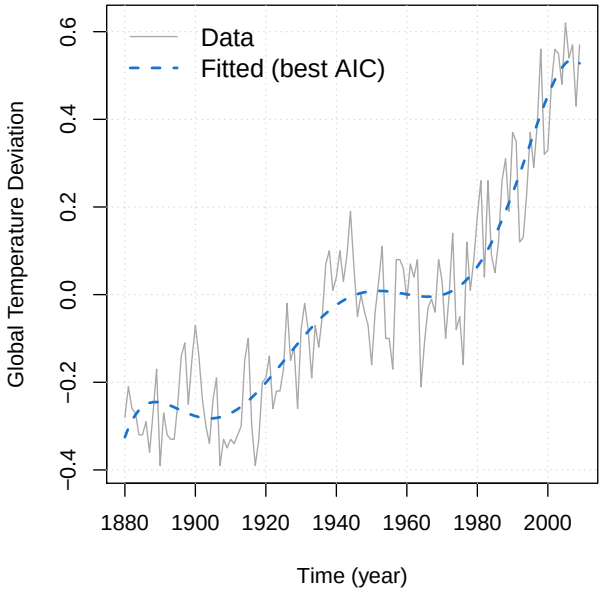
TO DO add comments

```
# Load libraries
library(astsa)

# Load data
data(gtemp)

# Degree of polynomial regression
deg_max = 30
```

```
# Construct design matrix (no intercept)
year = time(gtemp)
X = cbind(year)
for (i in 2:deg_max){X = cbind(X,year^i)}

# Define response vector
y = gtemp

# Initialisation
model.AIC = rep(NA,deg_max)
model.BIC = rep(NA,deg_max)
model.pred = matrix(NA,deg_max,length(y))

# Fit models
for (i in 1:deg_max){
  # Fit model
  model = lm(y~X[,1:i])

  # Compute AIC, BIC and \hat{y}
  model.AIC[i] = AIC(model)
  model.BIC[i] = BIC(model)
  model.pred[i,] = fitted(model)
}

# Compute best AIC and BIC
aic.best = which.min(model.AIC)
bic.best = which.min(model.BIC)

# Plot results
par(mfrow = c(1,2))
plot(NA, xlim = c(1,deg_max),
     ylim = range(cbind(model.AIC, model.BIC)),
     xlab = "Polynomial order", ylab = "AIC/BIC")
grid()
lines(model.AIC, type = "b", col = "dodgerblue3", lty = 2)
lines(model.BIC, type = "b", col = "darkgoldenrod2", pch = 22)
points(aic.best,model.AIC[aic.best], col = "dodgerblue3",
       pch = 16, cex = 2)
points(bic.best,model.BIC[bic.best], col = "darkgoldenrod2",
       pch = 15, cex = 2)

legend("topright", c("BIC","Min BIC","AIC","Min AIC"),
pch = c(22,15,21,16), pt.cex = rep(c(1,2),2),
col = rep(c("darkgoldenrod2","dodgerblue3"), each = 2),
lty = c(1,NA,2,NA), lwd = c(1,NA,1,NA),
bty = "n", bg = "white", box.col = "white", cex = 1.2)

plot(NA, xlim = range(year), ylim = range(y),
     xlab = "Time (year)",
     ylab = "Global Temperature Deviation")
grid()
lines(gtemp, col = "darkgrey")
lines(cbind(year,model.pred[aic.best,])[,2],
```

```
      col = "dodgerblue3", lty = 2, lwd = 2)

legend("topleft", c("Data","Fitted (best AIC)"),
       col = c("darkgrey","dodgerblue3"),
       lty = c(1,2), lwd = c(1,2),
       bty = "n", bg = "white", box.col = "white", cex = 1.2)
```





Chapter 6

State-Space Models

Chapter 7

Time Series Models of Heteroskedasticity

Appendix A

Appendix A

A.1 Subject

Appendix B

Appendix B

Bibliography