Applied Financial Econometrics
Slides

Rolf Tschernig — Florian Brezina
University of Regensburg
April 2023[1]

Contents

General Information

Contact

Rolf Tschernig
Building RW(L), 5th floor, Room 515
Tel.: (+49) 941/943 2737
Fax: (+49) 941/943 4917
Email: rolf.tschernig@ur.de

Homepage
https://www.uni-regensburg.de/business-economics-and-management-information-systems/economics-tschernig/homepage/index.html

Schedule and locations

See homepage
https://www.uni-regensburg.de/business-economics-and-management-information-systems/economics-tschernig/teaching/master/applied-financial-econometrics/index.html

Prerequisites

The bachelor module Einführung in die Ökonometrie or even better the master module Methods of Econometrics or comparable courses.

Grading

The current grading rules are provided at GRIPS or the handbook of modules. The final exam will take place during the exam period after the lecture period.

Software

In this course the software R is used. The R programs used in this course are shown in Appendix A and available on GRIPS.

Literature

- Basic textbook(s):
  - Cochrane, J.H. (2005). *Asset Pricing*, rev. ed., Princeton University Press. (See the book page of John Cochrane for more material about the book, including lecture videos based on the book. This book is mostly relevant for chapter 7 in these slides.)
  - Hamilton, J.D. (1994). *Time Series Analysis*, Princeton University Press. (A classic and excellent text book for univariate time series analysis that provides very detailed explanations and derivations. Not strong on cointegration which however is not part of this course.)
  - Kirchgässner, G., Wolters, J. and Hassler, U. (2013). *Introduction to Modern Time Series Analysis*, Springer, Berlin. (In the campus network of OTH full text available.)
  - Lütkepohl, Helmut und Krätzig, Markus (2004, 2008). *Applied Time Series Econometrics*,

Cambridge University Press. (In the campus network full text available.)

- Introduction to software R:
  - Kleiber, C. and Zeileis, A. (2008). *Applied Econometrics with R*, Springer, New York. (In the campus network full text available.)
  - Ligges, U. (2008). *Programmieren mit R*, Springer, Berlin. (In the campus network full text available.)

Additional Reading:

- Introductory level:
  - Brooks, C. (2008, 2014, 2019). *Introductory Econometrics for Finance*, 2./3./4. ed., Cambridge University Press. (In the campus network full text of 2008 available.)
  - Diebold, F.X. (2007). *Elements of Forecasting*, 4. ed., Thomson/South-Western.
  - Wooldridge, J.M. (2009, 2016, 2020). *Introductory Econometrics. A Modern Approach*, Thomson South-Western.
- Graduate level:
  - Campbell, J.Y., Lo, A.W. and MacKinlay A.C. (1997). *The Econometrics of Financial Markets*, Princeton University Press.

- Enders, W. (2010, 2015). *Applied Econometric Time Series*, Wiley.
- Franke, J., Härdle, W., and Hafner, C. (2011, 2015, 2019). *Statistics of Financial Markets. An Introduction*, Springer. (Advanced, old edition in German available.)
- Tsay, R.S. (2010). *Analysis of Financial Time Series*, Wiley. (In the campus network full text available.)

- German Reading:
  - Deistler, M. and Scherrer, W. (2018). Modelle der Zeitreihenanalyse, Birkhäuser (Advanced, In the campus network full text available.)
  - Franke, J., Härdle, W., and Hafner, C. (2004). *Einführung in die Statistik der Finanzmärkte*, 2. ed., Springer. (Advanced, newer English edition available.)
  - Kreiß, J.-P. and Neuhaus, G. (2006). *Einführung in die Zeitreihenanalyse*, Springer. (Advanced, In the campus network full text available.)
  - Neusser, K. and Wagner, M. (2022). *Zeitreihenanalyse in den Wirtschaftswissenschaften*, 4. Auflage, Springer Gabler. (In the campus network full text available.)

1. Introduction

1.1. Themes

- How to measure returns and risks of financial assets?
- Are asset returns predictable? In the short run - in the long run?
  $\longrightarrow$ requires command of time series econometrics
- Does the risk of an asset vary with time? Is this important? How can one model time-varying risk?
- Is the equity premium (excess returns of stocks over bonds) really that high?
- How can one explain variations in stock returns across various stocks?

For **outline of the course** see Contents

This course provides an introduction to the basics of financial econometrics, mainly to analyzing financial time series. There are many more topics in financial econometrics that cannot be covered by this course but are treated in advanced textbooks such as Franke, Härdle, und Hafner (2019) or Tsay (2010).

A selection of advanced topics not treated here is:

- Statistics of extreme risks
- Credit risk management and probability of default
- Interest rate models and term structure models
- Analyzing high-frequency data and modeling market microstructure
- Analyzing and estimating models for options
- Multivariate time series models
- Technical methods such as state-space models and the Kalman filter, principal components and factor models, copulae, nonparametric methods, ...

1.2. Some basics

- **Return** R_t (or gross return)

$$R_t = \frac{P_t + D_t}{P_{t-1}}$$

- **Net return**

$$\frac{(P_t - P_{t-1}) + D_t}{P_{t-1}} = R_t - 1$$

- **Log returns** r_t or continuously compounded returns
  - Recall: $\ln(1) = 0$, $\frac{\partial \ln(x)}{\partial x} = \frac{1}{x}$. Taking a Taylor expansion of degree 1 at x_0 delivers

$$\ln x \approx \ln x_0 + \frac{\partial \ln(x)}{\partial x}_{|x_0}(x - x_0) = \ln x_0 + \frac{1}{x_0}(x - x_0)$$

    Thus, expanding at $x_0 = 1$, one has for x close to 1

$$\ln x \approx x - 1$$

  - Replacing x by R_t gives

$$r_t = \log(R_t) \approx R_t - 1$$

- **Real prices**

$$real\ price_t(t) = \frac{P_t}{CPI_t}$$

Note that if real prices should be given in prices of year s, then one has to compute

$$real\ price_t(s) = P_t \frac{CPI_s}{CPI_t}$$

- **Real return**

$$real\ return_t = \frac{P_t/CPI_t + D_t/CPI_t}{P_{t-1}/CPI_{t-1}} = \frac{P_t + D_t}{P_{t-1}} \frac{CPI_{t-1}}{CPI_t} = R_t \frac{CPI_{t-1}}{CPI_t}$$

- **Log real returns**

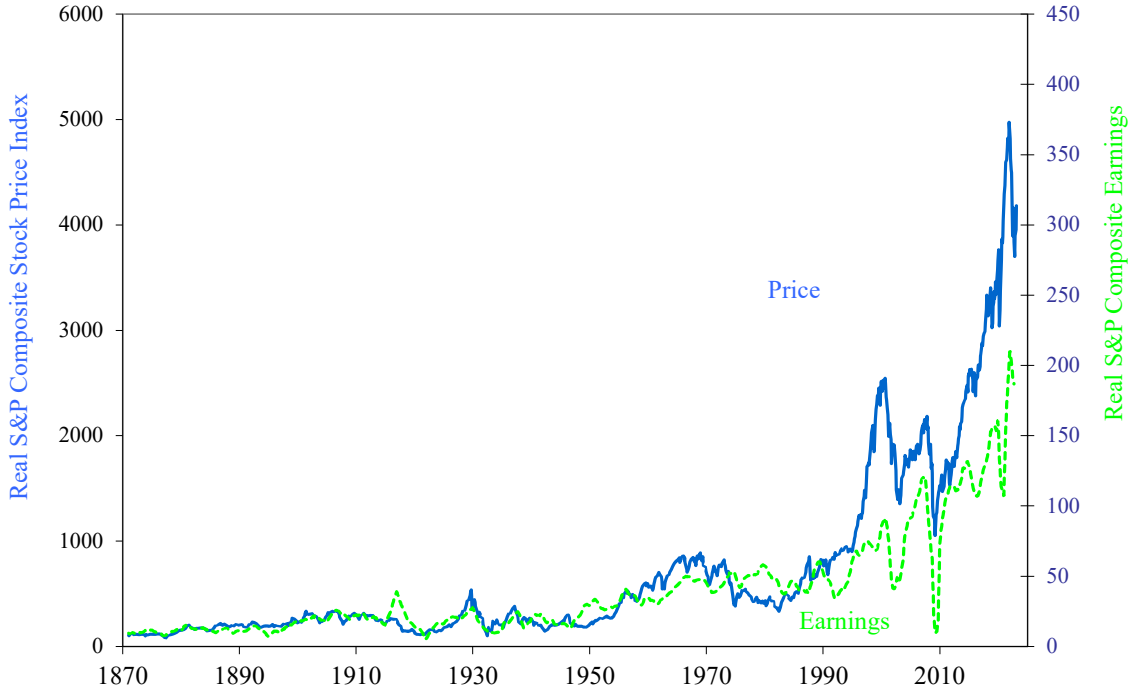
$$\begin{aligned} logged\ real\ return_t &= \log\left(R_t \frac{CPI_{t-1}}{CPI_t}\right) \\ &= \log(R_t) + \log CPI_{t-1} - \log CPI_t \\ &= r_t + \log CPI_{t-1} - \log CPI_t \end{aligned}$$

- **Excess log returns** of asset A over asset B

$$\begin{aligned} excess\ log\ return_t &= \log(R_t^A) - \log(R_t^B) \\ &= r_t^A - r_t^B \\ &= r_t^A + \log CPI_{t-1} - \log CPI_t - \left(r_t^B + \log CPI_{t-1} - \log CPI_t\right) \\ &= excess\ log\ real\ return_t \end{aligned}$$

A first look at data: S&P 500 Composite Index

Real prices of the S&P 500 Composite and real earnings, January 1871 – February 2023,
Source: homepage of Robert Shiller: www.econ.yale.edu/~shiller/



Are real prices RP_t predictable?

- Estimating a simple model

$$RP_t = \alpha_0 + \alpha_1 RP_{t-1} + u_t, \quad t = 1, \ldots, 1825 \tag{1.1}$$

What do you learn from that? Potential pitfalls?

Table 1.1.: Real Prices of the S&P 500, Jan 1971 - Feb 2023, OLS

```
Call:
lm(formula = RP_t ~ 1 + RP_tm1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.30955 -0.01915  0.00368  0.02340  0.41294

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.001497   0.006167   0.243    0.808
RP_tm1      1.000086   0.001008 992.211   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0409 on 1823 degrees of freedom
Multiple R-squared:  0.9982,Adjusted R-squared:  0.9982
F-statistic: 9.845e+05 on 1 and 1823 DF,  p-value: < 2.2e-16
```

See appendix A.1 for the R-program and the data source.

Potential issues when estimating a model like (1.1):

- lagged endogeneous regressors,
- nonstationarity of regressor,
- heteroskedastic errors,
- lag length.

Models where the regressors are exclusively lagged endogenous variables are called **autoregressive models**, see section 2.1.

1.3. Stochastic processes

First, some more terms:

Stochastic process or **random process**:

A stochastic process $\{y_t\}_{t\in\mathbb{T}}$ is a collection of random variables y_t's with their indices t ordered within the index set $\mathbb{T}$.

If the index t represents time, stochastic processes are also called time series processes.

In this lecture, only time discrete stochastic processes are considered with $\mathbb{T} = \mathbb{Z}$ or $\mathbb{T} = \mathbb{N}$.

An observed **realization** of a stochastic process is called **trajectory** or **sample path**.

The simplest example of a stochastic process is:

White Noise (WN):
A collection of random variables $\{u_t\}_{t\in\mathbb{T}}$ is called white noise and denoted as

$$u_t \sim WN(0, \sigma^2), \quad t \in \mathbb{T},$$

if

- the unconditional mean is zero $E[u_t] = 0$ for all $t \in \mathbb{T}$, and
- the variance is identical for all t, i.e. $Var(u_t) = \sigma^2$ for all $t \in \mathbb{T}$, and
- the random variables are uncorrelated over time $Cov(u_t, u_s) = 0$ for all $t, s \in \mathbb{T}$ and $t \neq s$.

In case stock returns are white noise, what does this imply?

Notes:

- The properties of a white noise process do not imply that $E[u_t|u_{t-1}] = 0$:

$$\begin{aligned} Cov(u_t, u_{t-1}) &= E[u_t u_{t-1}] - E[u_t]E[u_{t-1}] = E[u_t u_{t-1}] \quad \text{(due to WN: } E[u_t] = 0) \\ &= E\left[u_{t-1}E[u_t|u_{t-1}]\right] = 0 \quad \text{(using LIE and property of WN.)} \end{aligned}$$

  Therefore, white noise does not imply that $g(u_{t-1}) := E[u_t|u_{t-1}] = 0$ since only the expectation of the product $E[u_{t-1}g(u_{t-1})]$ as well as $E[g(u_{t-1})]$ are zero by the properties of WN.

- White noise processes may exhibit dependence in the higher order moments (moments of order three or more). E.g., one has $E[u_t^2 u_s^2] \neq E[u_t^2]E[u_s^2]$).

Both issues are excluded if the u_t and u_s are **stochastically independent and identically (i.i.d.)** distributed, denoted as

$$u_t \sim IID(0, \sigma^2), \quad t \in \mathbb{T}.$$

The i.i.d. condition is automatically fulfilled if the random variables are normally distributed and uncorrelated. Then one has a stronger version of white noise: **Gaussian white noise**.

If one only wants to ensure that $E[u_t|u_{t-1}] = 0$ but allow for dependence in higher moments, then one has to assume that the process u_t is a martingale difference.

Martingale:
A stochastic process $\{y_t\}_{t\in\mathbb{T}}$ is called martingale if

$$E[y_t|y_{t-1}, y_{t-2}, \ldots] = y_{t-1} \tag{1.2}$$

holds.

Martingale Difference Sequence (MDS):
A process is called a martingale difference sequence$\{u_t\}_{t\in\mathbb{T}}$ if

$$E[u_t|u_{t-1}, u_{t-2}, \ldots] = 0 \tag{1.3}$$

holds (Hassler, 2019).

Therefore, if $\{y_t\}$ is a martingale, $u_t := y_t - y_{t-1}$, $t > 1$ and $u_1 = y_1$ is an MDS since (Davidson, 2000, Section 6.2.1)

$$E[u_t|u_{t-1}, \ldots] = E[y_t - y_{t-1}|y_{t-1}, \ldots, y_1] = y_{t-1} - y_{t-1} = 0.$$

2. The basics of time series modeling

2.1. Autoregressive processes

2.1.1. Autoregressive processes of order one (AR(1) processes)

A stochastic process $\{y_t\}_{t\in\mathbb{T}}$ that is generated by the following (stochastic) difference equation

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + u_t, \quad t \in \mathbb{T}$$

where the u_t's are white noise and $\mathbb{T} = \{0, 1, 2, \dots, \}$ or $\mathbb{T} = \mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots, \}$ is called **autoregressive process of order one (AR(1) process)**. If $\mathbb{T} = \mathbb{Z}$ we say that the process started a long time age.

The random variable u_t is called **error**, **disturbance**, **shock**, or **innovation**.

Simulating AR(1) processes

Using the `R` program given in appendix A.2 one can generate R different realizations of an AR(1) process. Since for simulations one always needs a presample value y_0, several options are possible to choose y_0. For generating the AR(1) series the `R` command `filter(x, filter, method, init)` is used:

- `x`: element t is given by $\alpha_0 + u_t$, $t = 1, \ldots, T$,
- `filter`: given by α_1,
- `method`: use `recursive` as the y_t's are computed recursively,
- `init`: contains the presample value y_0.

The `R` program also allows to illustrate non-ergodic AR(1) processes, see the handout of Methods of Econometrics for an explanation of ergodicity.

Statistical properties of an AR(1) process

- For $\alpha_1 = 0$ the stochastic process $y_t - \alpha_0$ is white noise.

- Expressing y_t as a weighted sum of past and present white noise terms (plus starting value)

$$\begin{aligned}
y_t &= \alpha_0 + \alpha_1 \left(\alpha_0 + \alpha_1 y_{t-2} + u_{t-1}\right) + u_t \\
&= \alpha_0 + \alpha_1\alpha_0 + \alpha_1^2 y_{t-2} + \alpha_1 u_{t-1} + u_t \\
&\vdots \\
&= \alpha_0 \left(1 + \alpha_1 + \alpha_1^2 + \cdots + \alpha_1^{j-1}\right) + \alpha_1^j y_{t-j} + \alpha_1^{j-1} u_{t-(j-1)} + \cdots + \alpha_1^2 u_{t-2} + \alpha_1 u_{t-1} + u_t \\
&= \alpha_0 \sum_{k=0}^{j-1} \alpha_1^k + \alpha_1^j y_{t-j} + \sum_{k=0}^{j-1} \alpha_1^k u_{t-k}.
\end{aligned}$$

  - For $|\alpha_1| < 1$ and $j \to \infty$ (process has run since ever) one has

$$\left(1 + \alpha_1 + \alpha_1^2 + \cdots + \alpha_1^j + \ldots\right) = 1/(1-\alpha_1)$$
$$\alpha_1^j y_0 \to 0$$

    and therefore

$$y_t = \alpha_0/(1-\alpha_1) + \sum_{j=0}^{\infty} \alpha_1^j u_{t-j}. \tag{2.1}$$

    The random variable y_t is a *weighted* infinite sum of past values and the present value of the white noise process $\{u_t\}$. The importance of shocks declines quickly. E.g. for $\alpha_1 = 0.9$ one has $0.9^{10} = 0.349$ and $0.9^{50} = 0.05$.

- For $\alpha_1 = 1$ one cannot let $j \to \infty$. Typically one chooses $j = t$ and obtains

$$y_t = t\alpha_0 + y_0 + u_t + u_{t-1} + \cdots + u_1, \quad t = 1, 2, \ldots.$$

  * If $\alpha_0 = 0$, the stochastic process $\{y_t\}_{t=1}^{\infty}$ is called **random walk** because y_t is an **unweighted** sum of random increments. Shocks keep their importance. One also says that y_t has long memory (even perfect memory).

  * For $\alpha_0 \neq 0$, it will be seen below that $E[y_t] = \alpha_0 t + E[y_0]$. Thus, one has a combination of a (pure) random walk with a deterministic trend or drift. In this case $\{y_t\}$ is called a **random walk with drift**.

- For $\alpha_1 > 1$ the influence of a shock increases with its distance to the present observation. One has an **explosive autoregressive model**.

- In order to obtain the **conditional expectation**

$$E[y_t|y_{t-1}] = \alpha_0 + \alpha_1 y_{t-1}$$

which is in general different from zero, the assumption $u_t \sim WN(0, \sigma^2)$ is not sufficient for the reason described in section 1.3.

One needs

- either the assumption that $\{u_t\}_{t\in\mathbb{T}}$ is a martingale differencee sequence (MDS) or

– the even stronger assumption $\{u_t\} \sim IID(0, \sigma^2)$

and in case of $\mathbb{T} = \mathbb{N}$ and a stochastic presample value y_0 that u_t and y_0 are stochastically independent for all $t > 0$. To see this, consider

$$\begin{aligned} E[y_t|y_{t-1}] &= \alpha_0 + \alpha_1 y_{t-1} + E[u_t|y_{t-1}] \\ &= \alpha_0 + \alpha_1 y_{t-1} + E\left[u_t \middle| \alpha_0 \sum_{k=0}^{t-2} \alpha_1^k + \alpha_1^{t-1} y_0 + \sum_{k=0}^{t-2} \alpha_1^k u_{t-1-k}\right] \end{aligned}$$

which reveals the possible dependence of u_t on past u_t's and y_0 that is not ruled out by the white noise assumption alone. By the independence of y_0 of all future u_t's one has $E[u_t|u_{t-1}, \ldots, u_1, y_0] = E[u_t|u_{t-1}, \ldots, u_1]$. And by the MDS assumption, the latter expression is zero.

- The **unconditional expectation** for period t is

$$\mu_t \equiv E[y_t] = \begin{cases} \alpha_0/(1-\alpha_1) & \text{if } |\alpha_1| < 1 \text{ and } t \in \mathbb{Z} \text{ — independent of } t, \\ \alpha_0(1 + \alpha_1 + \ldots + \alpha_1^{t-1}) + \alpha_1^t E[y_0] & \text{if } t = 1, 2, \ldots \text{ — dependent on } t, \\ t\alpha_0 + E[y_0] & \text{if } \alpha_1 = 1 \text{ and } t = 1, 2, \ldots \text{ — dependent on } t. \end{cases}$$

Knowing μ_t one can rewrite the autoregressive process as

$$y_t - \mu_t = \alpha_1(y_{t-1} - \mu_{t-1}) + u_t.$$

Check!

- **Unconditional variance**:

$$\begin{aligned} Var(y_t) &\equiv E[(y_t - \mu_t)^2] \\ &= E\left[(\alpha_1(y_{t-1} - \mu_{t-1}) + u_t)^2\right] \\ &= E\left[\alpha_1^2(y_{t-1} - \mu_{t-1})^2\right] + 2E[\alpha_1(y_{t-1} - \mu_{t-1})u_t] + E[u_t^2] \\ &= \alpha_1^2 Var(y_{t-1}) + 2 \cdot 0 + \sigma^2 \\ &= \alpha_1^2 Var(y_{t-1}) + \sigma^2. \end{aligned}$$

Again, further results depend on α_1:

Inserting iteratively delivers

$$Var(y_t) = \begin{cases} \sigma^2/(1-\alpha_1^2) & \text{if } |\alpha_1| < 1 \text{ and } t \in \mathbb{Z} \text{ — independent of } t, \\ \sigma^2 \sum_{j=0}^{t-1} \alpha_1^{2j} & \text{if } Var(y_0) = 0 \text{ and } t = 1, 2, \ldots \text{ — dependent on } t, \\ \sigma^2 t & \text{if } \alpha_1 = 1 \text{ and } Var(y_0) = 0 \text{ and } t = 1, 2, \ldots \text{ — dependent on } t. \end{cases}$$

- Definition of the **autocovariance function**:

$$Cov(y_t, y_s) \equiv E[(y_t - \mu_t)(y_s - \mu_s)]$$

Computation: Take for simplicity $s = t - 1$:

$$\begin{aligned} Cov(y_t, y_{t-1}) &= E\left[(\alpha_1(y_{t-1} - \mu_{t-1}) + u_t)\,(y_{t-1} - \mu_{t-1})\right] \\ &= E\left[\alpha_1(y_{t-1} - \mu_{t-1})^2 + u_t(y_{t-1} - \mu_{t-1})\right] \\ &= \alpha_1 E[(y_{t-1} - \mu_{t-1})^2] + E[u_t(y_{t-1} - \mu_{t-1})] \\ &= \alpha_1 Var(y_{t-1}). \end{aligned}$$

Thus, with the results for the unconditional variance we obtain

$$Cov(y_t, y_{t-1}) = \alpha_1 Var(y_{t-1}) = \begin{cases} \alpha_1\sigma^2/(1-\alpha_1^2) & \text{if } |\alpha_1| < 1 \text{ and } t \in \mathbb{Z} \text{ — independent of } t, \\ \alpha_1\sigma^2 \sum_{j=0}^{t-2} \alpha_1^{2j} & \text{if } Var(y_0) = 0 \text{ and } t = 1, 2, \ldots \text{ — dependent on } t, \\ (t-1)\sigma^2 & \text{if } \alpha_1 = 1,\ Var(y_0) = 0,\ t = 1, 2, \ldots \text{ — dependent on } t. \end{cases}$$

Similarly one can show that

$$Cov(y_t, y_{t-k}) = \alpha_1^k Var(y_{t-k}) = \begin{cases} \alpha_1^k\sigma^2/(1-\alpha_1^2) & \text{if } |\alpha_1| < 1 \text{ and } t \in \mathbb{Z} \text{ — independent of } t, \\ \alpha_1\sigma^2 \sum_{j=0}^{t-1-k} \alpha_1^{2j} & \text{if } Var(y_0) = 0 \text{ and } t = 1, 2, \ldots \text{ — dependent on } t, \\ (t-k)\sigma^2 & \text{if } \alpha_1 = 1 \text{ and } Var(y_0) = 0,\ t = 1, 2, \ldots \text{ — dep. on } t. \end{cases}$$

- **Weak stationarity** or **Covariance stationarity**:
  A stochastic process $\{y_t\}$ is called weakly stationary if the first and second unconditional moment are independent of the time index t:
  - $E[y_t] = \mu_t = \mu$ for all $t \in \mathbb{T}$,
  - $Cov(y_t, y_{t-k}) = \gamma_k$ for all $t \in \mathbb{T}$.
- Thus, **if** $|\alpha_1| < 1$ and **it has run since ever**, **the AR(1) process is weakly stationary**.
  - Stochastic processes for which
    $$\lim_{t\to\infty} E(y_t) = \mu,$$
    $$\lim_{t\to\infty} Cov(y_t, y_{t-k}) = \gamma_k,$$
    are called **asymptotically stationary**. Of course any stationary process is also asymptotically stationary.
  - If a process is not asymptotically stationary, it is **nonstationary**.
  - There is another concept of stationarity, frequently called **strict stationarity**. A stochastic process is called strictly stationary if the joint distribution of $(y_{t_1}, y_{t_2}, \ldots, y_{t_m})$ is the same as for $(y_{t_1+h}, y_{t_2+h}, \ldots, y_{t_m+h})$ for any $t_1, \ldots, t_m$ and for all $m \in \mathbb{N}$, $h \in \mathbb{Z}$.

    Is a weakly stationary AR(1) process also strictly stationary? If not, which additional assump-

tions do you need?

- **Properties of a (weakly) stationary AR(1) process**:
  - The autocovariances are always different from zero albeit they may be very small.
  - The autocovariances converge to zero exponentially fast if $|\alpha_1| < 1$:

$$\gamma_k = \alpha^k \gamma_0.$$

Therefore, stationary AR(1) processes are called stochastic processes with **short memory**. The effect of a shock u_t in time t has a negligible effect on y_{t+h} for h large. The opposite holds for a random walk where the effect of u_t stays the same for any y_{t+h} in the future! Therefore, **random walks are said to have long memory**.

- Further remark: If $\{y_t\}$ is weakly stationary, it can be represented as a weighted sum of past and present white noise terms plus a constant. This result holds for **all** stationary stochastic processes and is known as the **Wold decomposition**, see (2.1).
- If one replaces in (2.1) α_1^j by ϕ_j, then one obtains a so called moving average process

$$y_t = \mu + \sum_{j=0}^{\infty} \phi_j u_{t-j}, \quad t \in \mathbb{Z},$$

see also section 2.2.

2.1.2. Autoregressive processes of higher order

- **Some more notation**:
  - **Backshift** or **lag operator**

$$Lu_t \equiv u_{t-1}.$$

    Thus, $L^2 u_t = L(Lu_t) = Lu_{t-1} = u_{t-2}$ and $L^k u_t = u_{t-k}$.

  - **Differencing operator**

$$\Delta x_t = (1 - L)x_t = x_t - Lx_t = x_t - x_{t-1}.$$

- A stochastic process $\{y_t\}$ is called **autoregressive process of order** p (**AR(p) process**) if

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + u_t, \quad t \in \mathbb{T}$$

  where u_t is white noise. The index set can be $\mathbb{T} = \mathbb{Z}$ or $\mathbb{T} = \{p, p+1, p+2, \ldots\}$.

  Alternative representations with lag operator

$$\begin{aligned} y_t - \alpha_1 y_{t-1} - \cdots - \alpha_p y_{t-p} &= \alpha_0 + u_t \\ 1 - \alpha_1 L y_t - \cdots - \alpha_p L^p y_t &= \alpha_0 + u_t \\ (1 - \alpha_1 L - \cdots - \alpha_p L^p) y_t &= \alpha_0 + u_t \\ \alpha(L) y_t &= \alpha_0 + u_t \end{aligned}$$

where $\alpha(L) \equiv (1 - \alpha_1 L - \cdots - \alpha_p L^p)$ is called an AR(p) lag polynomial.

- **Relationship of an AR(2) process with AR(1) processes**:
  Example: Let $\{w_t\}$ and $\{u_t\}$ be a weakly stationary AR(1) process and white noise, respectively. Then

$$w_t = \lambda_1 w_{t-1} + u_t, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots,$$
$$(1 - \lambda_1 L) w_t = u_t.$$

  Now consider the process $\{y_t\}$

$$y_t = \lambda_2 y_{t-1} + w_t, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots,$$
$$(1 - \lambda_2 L) y_t = w_t$$

  – Is the stochastic process $\{y_t\}$ weakly stationary?

  – Can the stochastic process $\{y_t\}$ be represented as an AR(p) process?

Answer:

$$(1-\lambda_2 L)y_t = \frac{u_t}{(1-\lambda_1 L)} \quad \text{does this work?}$$

$$(1-\lambda_2 L)(1-\lambda_1 L)y_t = u_t$$

$$((1\underbrace{-\lambda_1 L - \lambda_2 L}_{-\alpha_1 L} + \underbrace{\lambda_1\lambda_2 L^2}_{-\alpha_2 L^2})y_t = u_t$$

$$y_t - \alpha_1 y_{t-1} - \alpha_2 y_{t-2} = u_t$$

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + u_t$$

with $\alpha_1 = \lambda_1 + \lambda_2, \quad \alpha_2 = -\lambda_1\lambda_2$.

- In general it holds that one can factor an AR(p) process as

$$1 - \alpha_1 z - \cdots - \alpha_p z^p = (1-\lambda_1 z)\cdots(1-\lambda_p z)$$

where the values λ_i can be complex numbers.

- **Complex numbers** $\mathbb{C}$:
  - Define $i = \sqrt{-1}$. Thus $i^2 = -1$. i is called an **imaginary** number. Any number obtained by multiplying i with a scalar produces again an imaginary number, e.g. $5i$.
  - A **complex number** consists of a real part and imaginary part

$$z = a + bi$$

where a and b are real scalars.

– Examples: $\lambda_1 = a + bi$, $\lambda_2 = a - bi$:

$$-\alpha_1 = -(a+bi) - (a-bi) = -a - bi - a + bi = -2a \quad \text{real number}$$
$$-\alpha_2 = (a+bi)(a-bi) = a^2 - abi + abi - b^2 i^2 = a^2 - b^2(-1) = a^2 + b^2 \quad \text{real number}$$
$$(a+bi)(a+bi) = a^2 + abi + abi + b^2 i^2 = a^2 + b^2(-1) + 2abi = a^2 - b^2 + 2abi \quad \text{complex number}$$

– For more information on complex numbers see the article on Wikipedia or Neusser und Wagner (2022, Appendix A).

- The solutions z^* to the so-called characteristic equation

$$(1 - \lambda_1 z) \cdots (1 - \lambda_p z) = 0$$

are called its roots.

- Each factor $(1 - \lambda_i z)$ can be viewed as AR(1) polynomial which produces a weakly stationary process *if* its root is outside the unit circle, that is

$$|z_i| > 1 \quad \text{for all } i = 1, \ldots, k. \tag{2.2}$$

Equivalently

$$|\lambda_i| < 1 \quad \text{for all } i = 1, \ldots, k.$$

Thus, (2.2) is called **stability condition** of an AR(p) process. One also says: All roots of the AR(p) polynomial $\alpha(L)$ are outside the unit circle.

Frequently (2.2) is also called **stationarity condition**. This, however, is not entirely correct because the process is only asymyptotically stationary if it starts in $t = 0$ even if the stability condition holds. Why?

- Is for exactly one factor the root $z = 1$ (let's say for $\lambda_1 = 1$) and all others fulfil the stability condition, then one has

$$(1 - z)(1 - \lambda_2 z) \cdots (1 - \lambda_p z) = (1 - z)(1 - \alpha_1^* z - \cdots - \alpha_{p-1}^* z^{p-1})$$
$$\alpha(L) = (1 - L)\alpha^*(L)$$

  where $\alpha^*(L)$ fulfils the stability condition (2.2) and the AR(p) process contains a **random walk** component. One also says that it contains a **unit root** or is **integrated of order 1**,

$$y_t \sim I(1).$$

- In general: An AR(p) process $\{y_t\}$ is said to be **integrated of order** d, if

$$\alpha(L) = (1 - L)^d \alpha^*(L)$$

  and $\alpha^*(L)$ fulfils the stability condition (2.2).

  The integration parameter d may take real values $\rightarrow$ long memory models (Tschernig, 1994, Chapter 3).

- **Moments of a stationary AR(p) process**:
  - **Mean**: $E(y_t) = \mu, \quad$ for all t.

    It holds that

    $$\alpha_0 = \alpha(L)\mu = \alpha(1)\mu = \mu(1 - \alpha_1 - \alpha_2 - \cdots - \alpha_p)$$

    since from $\alpha(L)y_t = \alpha_0 + u_t$ one obtains $\alpha(L)(y_t - \mu) = -\alpha(L)\mu + \alpha_0 + u_t$ by subtracting $\alpha(L)\mu$ on both sides. If μ represents the mean, then $-\alpha(L)\mu + \alpha_0$ must be zero.
  - **Variance** and **autocovariance function**: cf. Hamilton (1994, p. 59, eq. (3.4.36))

    $$\gamma_k = \begin{cases} \alpha_1\gamma_1 + \alpha_2\gamma_2 + \cdots + \alpha_p\gamma_p + \sigma^2 & \text{for } k = 0 \\ \alpha_1\gamma_{k-1} + \alpha_2\gamma_{k-2} + \cdots + \alpha_p\gamma_{k-p} & \text{for } k = 1, 2, \ldots \end{cases} \tag{2.3}$$

    This system of equations is called **Yule-Walker equations**.
  - **Partial autocorrelation function** $Corr(y_t, y_{t-k}|y_{t-1}, \ldots, y_{t-k+1})$:
    In an AR(p) process it holds that

    $$a_k = Corr(y_t, y_{t-k}|y_{t-1}, \ldots, y_{t-k+1}).$$

    Thus, all partial autocorrelations for $k > p$ are zero since $a_k = \alpha_k = 0$ for $k > p$.

- It can be shown that the autocovariances of a stationary AR(p) process converge exponentially fast towards zero (Hamilton, 1994, p. 59), (Kirchgässner, Wolters, und Hassler, 2013, Example 2.4). For the AR(1) process this was shown in section 2.1.1.

- A stationary AR(p) process exhibits the following representation

$$y_t = \phi(L)u_t \tag{2.4}$$
$$y_t = u_t + \phi_1 u_{t-1} + \phi_2 u_{t-2} + \ldots + \phi_i u_{t-i} + \ldots$$

  where the lag polynomial $\phi(L)$ is determined by the following equations

$$\frac{1}{\alpha(L)} = \phi(L) \tag{2.5}$$
$$\alpha(L)\phi(L) = 1 \tag{2.6}$$

  The parameters of $\phi(L)$ can be determined by comparing coefficients (method of undetermined coefficients (Kirchgässner, Wolters, und Hassler, 2013, Section 2.1.2)):

$$\phi_j = \sum_{i=1}^{j} \phi_{j-i}\alpha_i, \quad j = 1, 2, \ldots, \quad \alpha_0 = 1, \alpha_i = 0 \text{ for } i > p$$

  The representation (2.4) is an example of an MA(∞) process, see section 2.2.3.

2.2. Moving average processes

2.2.1. MA(1) processes

- A stochastic process $\{y_t\}$ is called **moving average process of order 1 (MA(1) process)** if it fulfils the following equation

$$y_t = u_t + m_1 u_{t-1}, \quad t = \ldots, -2, -1, -, 1, 2, \ldots$$
$$y_t = (1 + m_1 L) u_t$$

  where $\{u_t\}$ is white noise.

- An MA(1) process can be simulated using the R program shown in section A.3.

- **Properties**:
  - Mean: $E(y_t) = 0$

– (Auto)covariance function:

$$\begin{aligned} Cov(y_t, y_{t-k}) &= E(y_t y_{t-k}) \\ &= E((u_t + m_1 u_{t-1})(u_{t-k} + m_1 u_{t-k-1})) \\ &= E(u_t u_{t-k}) + m_1 E(u_t u_{t-k-1}) \\ &+ m_1 E(u_{t-1} u_{t-k}) + m_1^2 E(u_{t-1} u_{t-k-1}) \end{aligned}$$

∗ Lag 0, $k = 0$:

$$\begin{aligned} Var(y_t) &= E(u_t^2) + m_1 E(u_t u_{t-1}) \\ &+ m_1 E(u_{t-1} u_t) + m_1^2 E(u_{t-1} u_{t-1}) \\ &= \sigma^2 + m_1^2 \sigma^2 = \gamma_0 \end{aligned}$$

∗ Lags -1,1, $k = 1$ oder $k = -1$:

$$\begin{aligned} Cov(y_t, y_{t-1}) &= E(u_t u_{t-1}) + m_1 E(u_t u_{t-2}) \\ &+ m_1 E(u_{t-1}^2) + m_1^2 E(u_{t-1} u_{t-2}) \\ &= m_1 \sigma^2 = \gamma_1 \end{aligned}$$

∗ For all lagged variables with $|k| \geq 2$ holds that

$$Cov(y_t, y_{t-k}) = 0 = \gamma_k$$

2.2.2. MA(q) processes

- A stochastic process $\{y_t\}$ is called **MA(q) process** if it has the following representation

$$y_t = u_t + m_1 u_{t-1} + \cdots + m_q u_{t-q}, \quad t = \ldots, -2, -1, -, 1, 2, \ldots$$

  where $\{u_t\}$ is white noise.

- Short notation using lag polynomials:

$$y_t = (m_0 + m_1 L + \cdots + m_q L^q) u_t, \quad m_0 = 1$$
$$y_t = m(L) u_t$$

- **Properties**:
  - Mean: $E(y_t) = 0$
  - (Auto)covariance function: Similarly to above we have
    - Lag 0, $k = 0$:

$$\begin{aligned} Var(y_t) = \gamma_0 &= \sigma^2 + m_1^2 \sigma^2 + \cdots m_q^2 \sigma^2 \\ &= \sigma^2 \left(1 + \sum_{i=1}^{q} m_i^2\right) = \sigma^2 \sum_{i=0}^{q} m_i^2 \end{aligned}$$

* Lag k, $-q \leq 0 < k \leq q$:

$$Cov(y_t, y_{t-k}) = \gamma_k = \sigma^2 \sum_{i=0}^{q-k} m_i m_{k+i}$$

* For all lags $|k| > q$:

$$Cov(y_t, y_{t-k}) = \gamma_k = 0$$

– Remarks:

* All **autocovariances for lags larger than q are 0**! Put differently: A shock that occurred q or more periods before does *not* influence the stochastic behavior of y_t !
* All **partial autocorrelations** are unequal zero for any lag, see later.

2.2.3. Moving Average of infinite order (MA(∞) processes)

- A stochastic process $\{y_t\}$ is called **MA(∞) process** if it has the following representation

$$\begin{aligned} y_t &= u_t + m_1 u_{t-1} + \cdots + m_q u_{t-q} + \cdots, \\ y_t &= (1 + m_1 L + \cdots + m_q L^q + \cdots) u_t \\ y_t &= m(L) u_t \end{aligned}$$

where $\{u_t\}$ is white noise.

- **Properties**:
  - Mean: $E(y_t) = 0$
  - (Auto)covariance function:
    - Variance

$$Var(y_t) = \sigma^2 \sum_{i=0}^{\infty} m_i^2$$

      Remark:
      The variance of an MA(∞) process only exists if the infinite sum $\sum_{i=0}^{\infty} m_i^2$ converges to a finite number. This a necessary condition for weak stationarity. Why?
      (Why is it not a necessary condition for strict stationarity?)
      In other words:
      Weak stationarity requires that the influence of a shock u_{t-k} decreases fast enough if the number of lags k increases.

    - Autocovariance function:

$$Cov(y_t, y_{t-k}) = \gamma_k = \sigma^2 \left(\sum_{i=1}^{\infty} m_i m_{k+i} \right)$$

- A MA(∞) process for which the MA polynomial $m(z)$ has all roots outside the unit circle is called **invertible** and has an AR(∞) or even AR(p) representation.

- **Remarks**:
  - In practice one cannot estimate an MA(∞) process since one cannot estimate the infinitely many parameters m_i. However, one can approximate an MA(∞) process by an MA(q) process with high order q. This only makes sense if the DGP that generated the observed time series does not have autocovariances for lags larger q.
  - If small but non-zero autocovariances occur for lags larger than q, then using a stationary AR(p) model can be more appropriate since any stationary AR(p) process can be represented as an MA(∞) process. See (2.1) for an AR(1) process and (2.4) for an AR(p) process.
  - Summary:
    * AR(p) process: $\gamma_k \neq 0$ for $k > p$, $a_k = 0$ for $k > p$
    * MA(q) process: $\gamma_k = 0$ for $k > q$, $a_k \neq 0$ for $k > q$.

    These properties can be used to select between AR and MA models in practice if one has reliable estimators for the autocorrelation and partial autocorrelation function. This is an integral part of the **Box-Jenkins** model specification strategy.
  - Sometimes, neither an AR(p) nor an MA(q) process is appropriate, then a combination of

both may do it: the ARMA(p, q) process.

2.3. ARMA processes

- A stochastic process $\{y_t\}$ is called **autoregressive moving average process (ARMA(p, q) process)** with AR order p and MA order q if it follows the equation

$$\alpha(L)y_t = m(L)u_t$$

  and $\{u_t\}$ is white noise and if $\{y_t\}$ is weakly stationary.

- An ARMA(1,1) process can be simulated using the `R` program shown in section A.3.

- **Properties**:
  - Mean: $E(y_t) = 0$ for all periods t
  - Autocovariance function:
    * For $|k| \geq \max(p, q)$ holds

$$\gamma_k = \alpha_1\gamma_{k-1} + \alpha_2\gamma_{k-2} + \cdots + \alpha_p\gamma_{k-p}.$$

      For large lags the ARMA process performs like an AR process.

  * For $|k| < \max(p, q+1)$ the computation is somewhat more complicated than in the AR(p) case, see e.g. Hamilton (1994, Section 3.6).

– How does the partial autocorrelation function behave?

– The stationarity condition for an ARMA(p, q) process and an AR(p) process are identical.

– Therefore, a stationary ARMA process exhibits an MA(∞) representation that can be obtained via comparing coefficients like (2.5).

– Is the MA polynomial invertible, then the ARMA(p, q) process can be written as an AR(∞) process. Invertibility requires:

$$1 + m_1 z + \cdots + m_q z^q \neq 0 \quad \text{for } |z| \leq 1$$

where z can be complex.

– **Attention**:
If the AR polynomial $\alpha(L)$ and the MA polynomial $m(L)$ have common roots, then some or all parameters of these polynomials are not identified, (see examples) and cannot be estimated. Therefore one needs a reliable determination of the AR and MA order. More on that later when model selection criteria are discussed.

Examples:

*

$$y_t = u_t$$
$$(1 - \alpha_1 L) y_t = (1 - \alpha_1 L) u_t$$

so that $m_1 = -\alpha_1$.

*

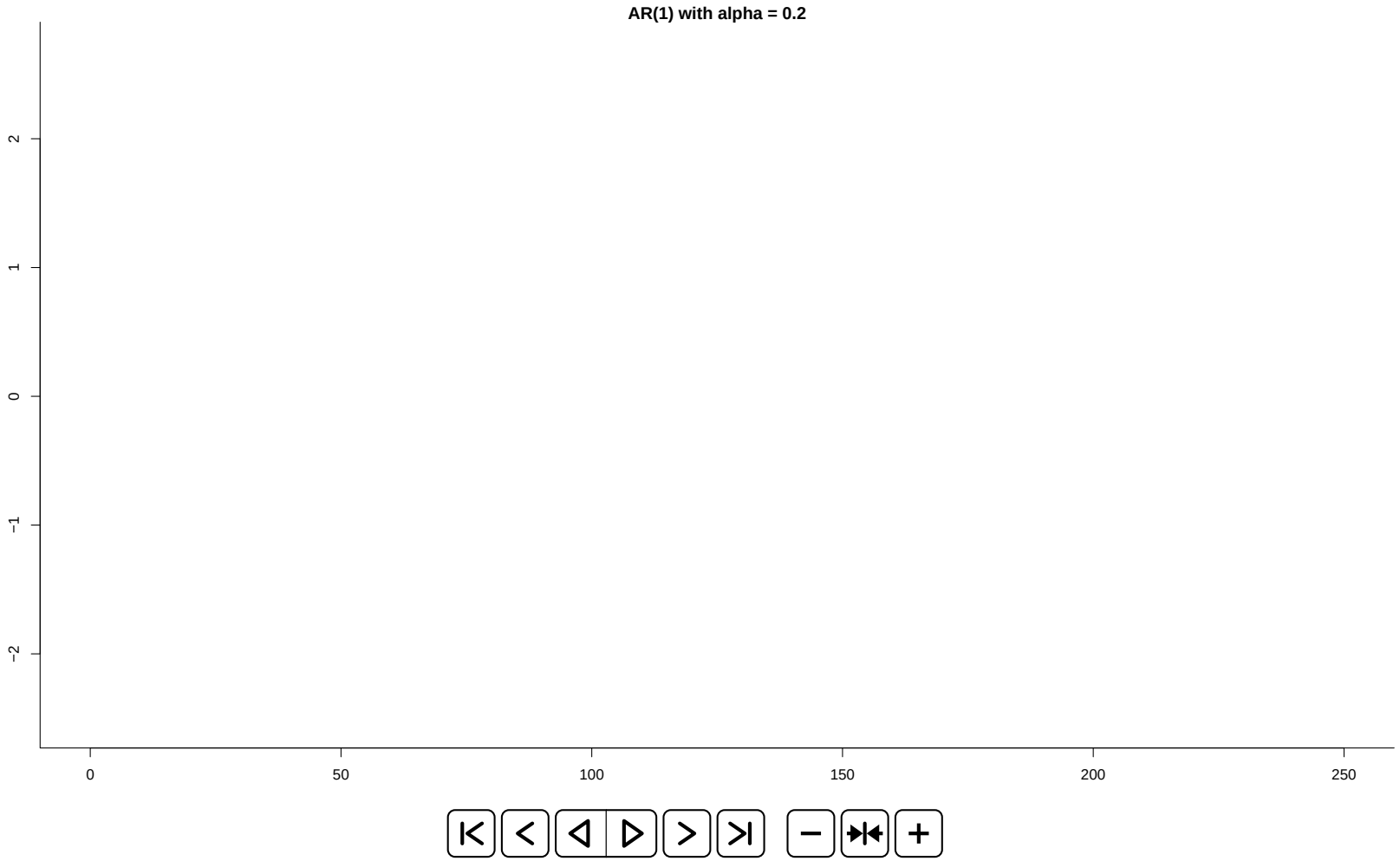
$$(1 - \lambda_1 L)(1 - \lambda_2 L) y_t = (1 - \lambda_1 L)(1 + \lambda_3 L) u_t$$
$$(1 - \underbrace{(\lambda_1 + \lambda_2)}_{\alpha_1} L + \underbrace{\lambda_1 \lambda_2}_{-\alpha_2} L^2) y_t = (1 + \underbrace{(-\lambda_1 + \lambda_3)}_{m_1} L + \underbrace{\lambda_1 \lambda_3}_{m_2} L^2) u_t$$
$$(1 - \underbrace{\lambda_2}_{\alpha_1'} L) y_t = (1 + \underbrace{\lambda_3}_{m_1'} L) u_t$$

- Useful commands in `R` for computing the MA(∞) representation as well as the autocorrelation and partial autocorrelation function are given in the `R` program in section A.4.

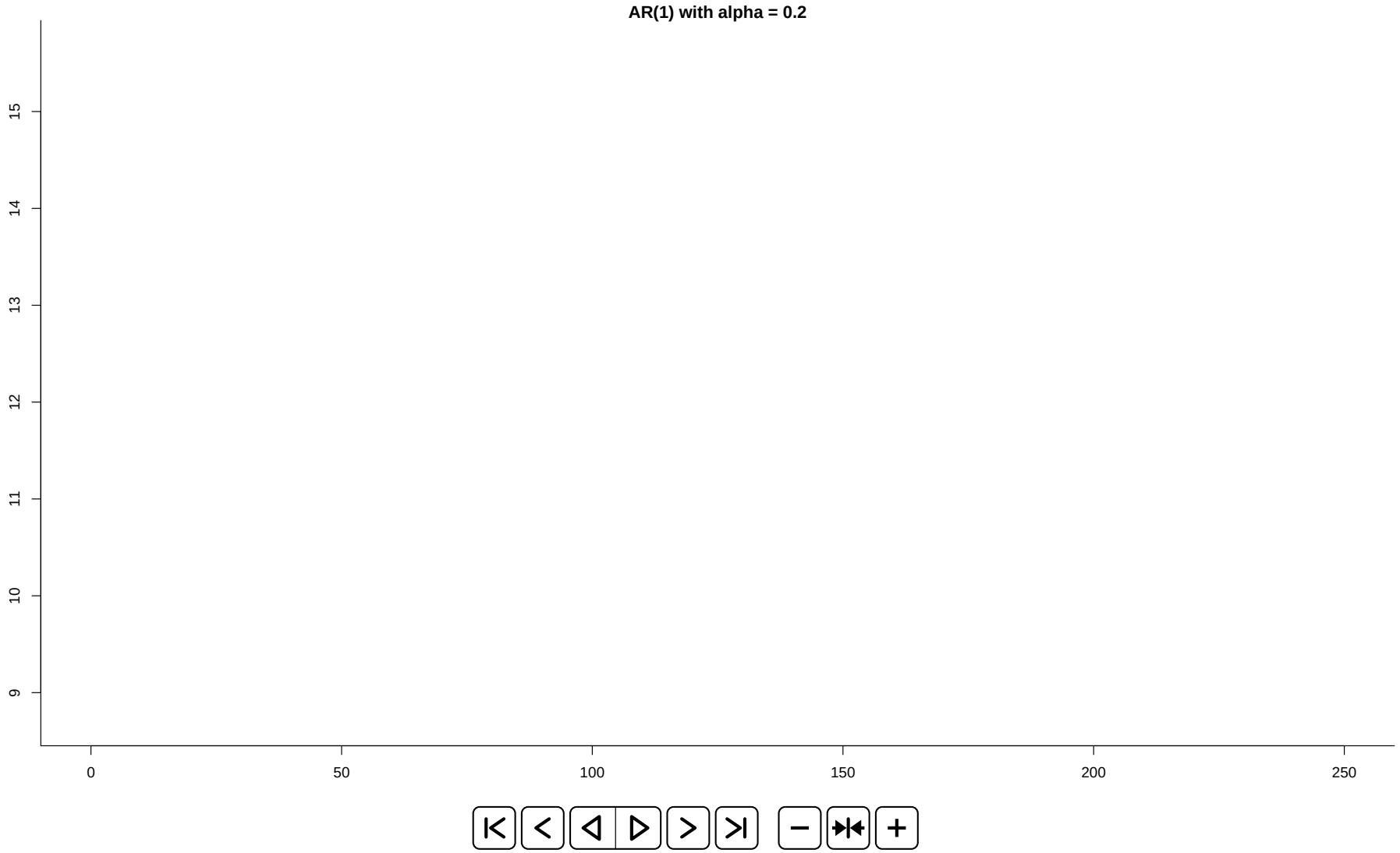
2.4. Trajectory examples

Note: The animations work when using Adobe Acrobat Reader or Adobe Acrobat. They do not work in Vorschau of the Apple operating system.

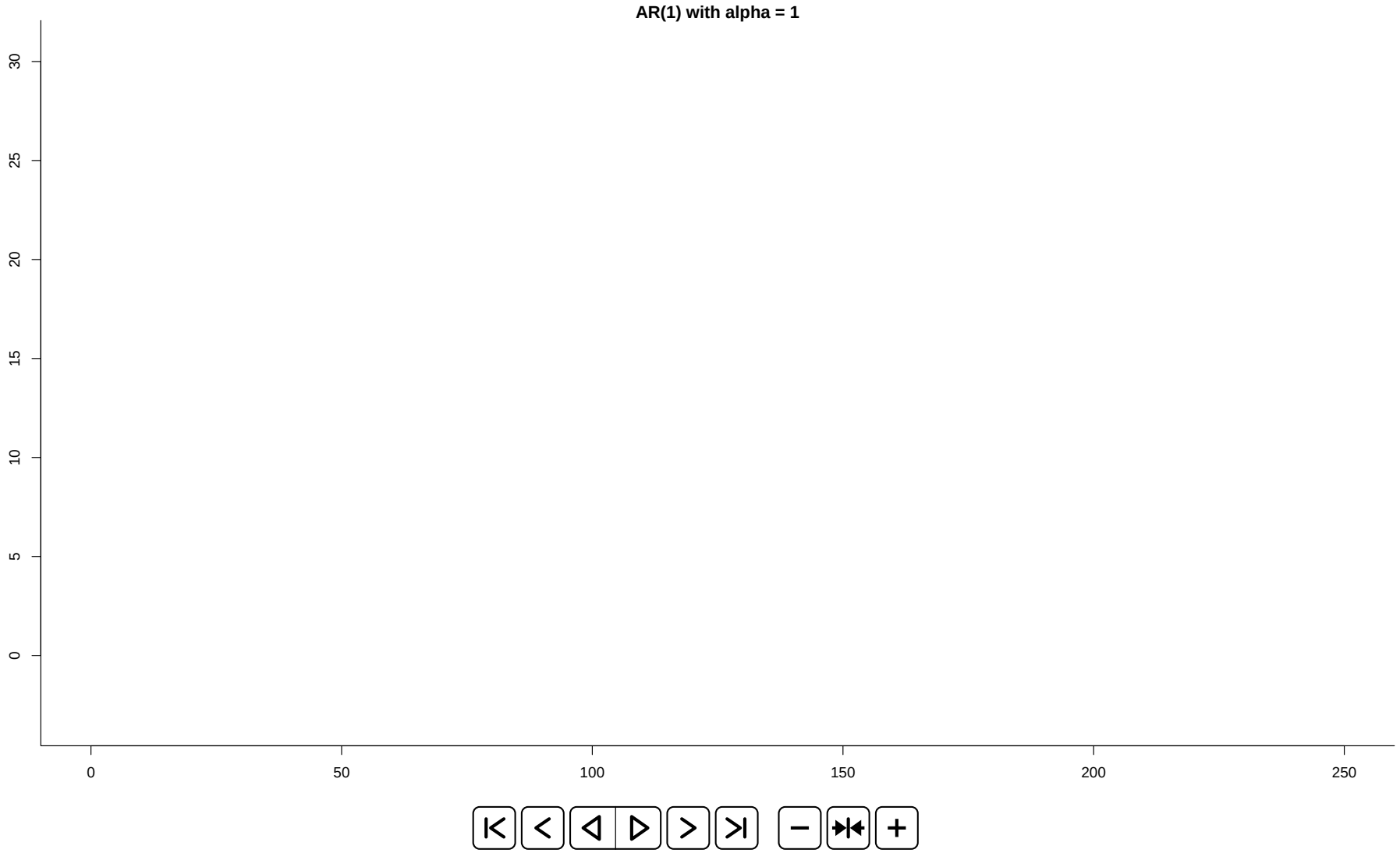
$y_t = 0.2y_{t-1} + u_t$, where u_t $i.i.d. N(0,1)$ and $y_0 = 0$.


AR(1) with alpha = 0.2
2
1
0
−1
−2
0
50
100
150
200
250

$y_t = 10 + 0.2y_{t-1} + u_t$, where u_t $i.i.d. N(0,1)$ and $y_0 = 0$.



$y_t = y_{t-1} + u_t$, where u_t $i.i.d. N(0,1)$ and $y_0 = 0$.



2.5. Estimation

2.5.1. OLS Estimation of AR(p) Models

AR(p) process

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + u_t$$

with $\{u_t\}$ being white noise

- **Some more notation and convention**:
  - One has p presample values $y_{-p+1}, \ldots, y_0$ and a sample with T observations $y_1, \ldots, y_T$. Hence, for estimating the parameters of an AR(p) process we need $T + p$ observations.

– Define the vectors and matrices

$$\alpha = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{pmatrix} \quad Y_{t-1} = \begin{pmatrix} 1 \\ y_{t-1} \\ y_{t-2} \\ \vdots \\ y_{t-p} \end{pmatrix}$$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{pmatrix}, \quad X = \begin{pmatrix} 1 & y_0 & y_{-1} & \cdots & y_{1-p} \\ 1 & y_1 & y_0 & \cdots & y_{2-p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & y_{T-1} & y_{T-2} & \cdots & y_{T-p} \end{pmatrix}, u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_T \end{pmatrix}$$

Thus, one can write

$$y = X\alpha + u$$

- **OLS estimator**

$$\hat{\alpha} = (X'X)^{-1}X'y = \left(\sum_{t=1}^{T} Y_{t-1}Y_{t-1}' \right)^{-1} \sum_{t=1}^{T} Y_{t-1}y_t$$

Example: If $p = 1$, one has $y_t = \alpha y_{t-1} + u_t$ and

$$\hat{\alpha} = \frac{\sum_{t=1}^{T} y_{t-1} y_t}{\sum_{t=1}^{T} y_{t-1}^2} \underbrace{=}_{\text{conditions?}} \alpha + \frac{\sum_{t=1}^{T} y_{t-1} u_t}{\sum_{t=1}^{T} y_{t-1}^2}$$

or

$$\hat{\alpha} - \alpha = \frac{\sum_{t=1}^{T} y_{t-1} u_t}{\sum_{t=1}^{T} y_{t-1}^2}$$

All the important properties of the OLS estimator can be illustrated for a simple AR(1) model.

- **Statistical properties in finite samples**:
  - Consider

$$E[\hat{\alpha}] = \alpha + E\left[(X'X)^{-1}X'u\right]$$

$$= \alpha + E\left[\underbrace{E\left[(X'X)^{-1}X'u|X\right]}_{\neq 0}\right] = \alpha + E\left[(X'X)^{-1}X'\underbrace{E[u|X]}_{\neq 0}\right]$$

    The reason why the conditional expectation is not zero is that X is not exogenous in the present case because it consists of lagged y_t's that are weighted sums of u_t's. Thus, the u_t's are already contained in the condition. This is different from the simple linear model for cross-section data.

Therefore, the OLS estimator $\hat{\alpha}$ is **biased** in finite samples.

$$E[\hat{\alpha}] \neq \alpha$$

– The probability distribution of $\hat{\alpha}$ is not exactly known even if the u_t's are normally distributed. Note that in this case also the y_t's follow a normal distribution. Why? Consider the OLS estimator of a simple AR(1) model

$$\hat{\alpha} = \frac{\sum_{t=1}^{T} y_{t-1} y_t}{\sum_{t=1}^{T} y_{t-1}^2}$$

As one can see, the same normal random variables appear in the numerator and denominator. Thus, the OLS estimator is a highly nonlinear function of normal random variables for which the distribution is generally unknown.

– Example: Distribution of $\hat{\alpha}$, when the process is

$$y_t = 0.8 y_{t-1} + u_t$$

and $u_t \sim IN(0, 1)$: The finite sample distribution in figure 2.1 is *not* centered around the true value 0.8 but around 0.75 so there exists an **estimation bias**. Moreover, the finite-sample distribution deviates considerably from a normal distribution.

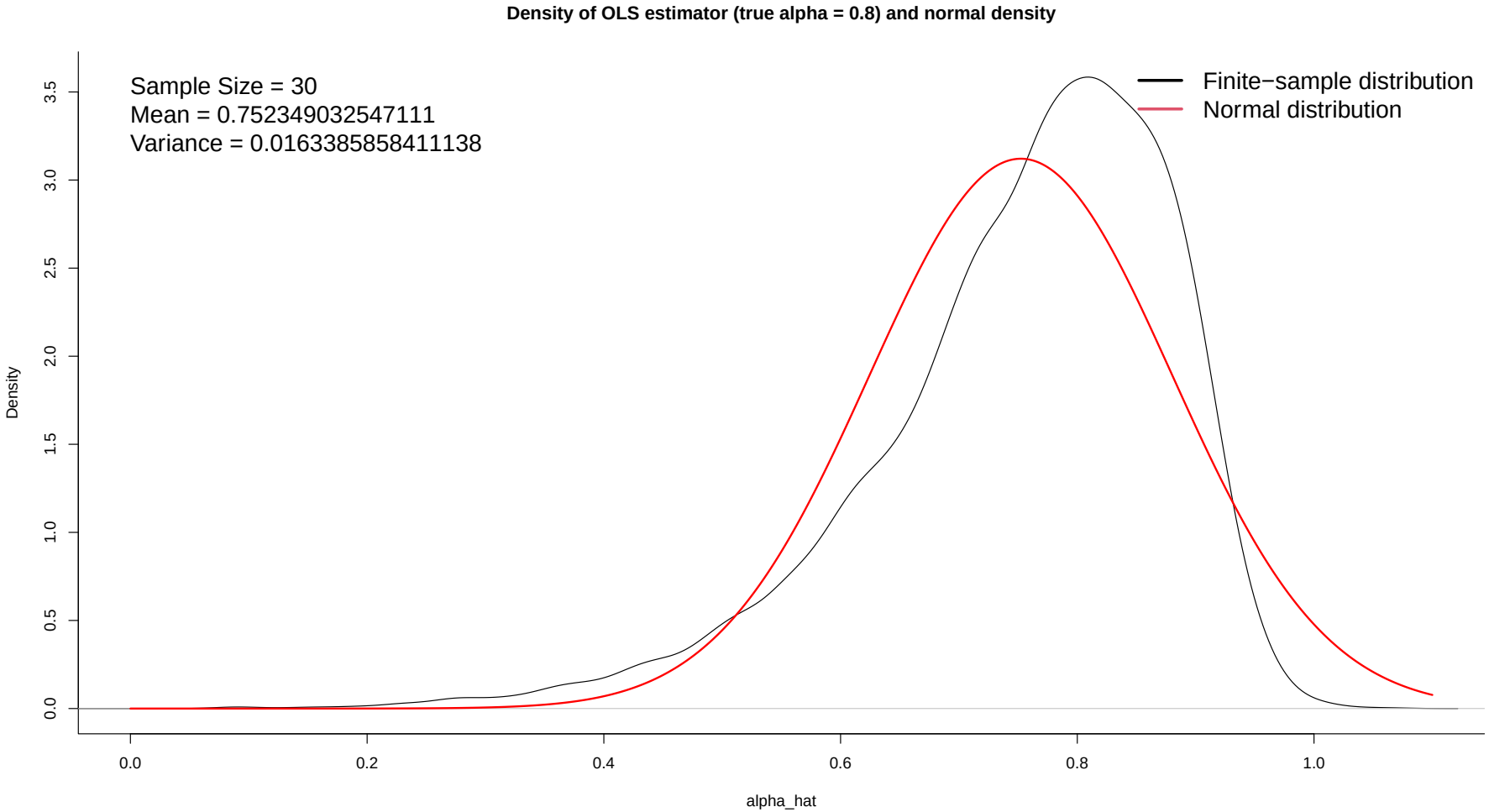


Figure 2.1.: Monte-Carlo simulation of finite-sample distribution of OLS for $n = 30$ observations using 10.000 replications. See appendix A.5 for the `R` program.

- **Statistical properties in very large samples — asymptotic properties**:
  - Consider the OLS estimator of the AR(1) model first:

$$\hat{\alpha} = \frac{\sum_{t=1}^{T} y_{t-1}y_t}{\sum_{t=1}^{T} y_{t-1}^2} = \frac{\frac{1}{T}\sum_{t=1}^{T} y_{t-1}y_t}{\frac{1}{T}\sum_{t=1}^{T} y_{t-1}^2}$$

Note that if the AR(1) process is stationary it holds that (see above)

$$Cov(y_t, y_{t-1}) = \alpha Var(y_t)$$

and thus

$$\alpha = \frac{Cov(y_t, y_{t-1})}{Var(y_t)}.$$

Note also that

$$\frac{1}{T}\sum_{t=1}^{T} y_{t-1}y_t$$

is an estimator of $Cov(y_t, y_{t-1})$ and

$$\frac{1}{T}\sum_{t=1}^{T} y_{t-1}^2$$

is an estimator of $Var(y_{t-1}) = Var(y_t)$.

– In order to check the properties of the OLS estimator for large samples, one may check the behavior of the covariance and the variance estimator. If for an arbitrary $\epsilon > 0$

$$\lim_{T\to\infty} P\left(\left|\frac{1}{T}\sum_{t=1}^{T} y_{t-1}^2 - Var(y_t)\right| < \epsilon\right) = 1 \qquad (+)$$

then the estimation error disappears. One says that the estimator $\frac{1}{T}\sum_{t=1}^{T} y_{t-1}^2$ converges in probability to the true value, which is $Var(y_t)$ in the present case. A short hand notation is

$$\operatorname{plim}_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} y_{t-1}^2 = Var(y_t).$$

An estimator for which the estimation error diminishes with increasing sample size is also called **consistent**.

It can be shown that the variance and the covariance estimator are both consistent if the autoregressive process is stationary. Moreover, for two arbitrary sequences of random numbers z_T and w_T, for which

$$\operatorname{plim} z_T = z \quad \operatorname{plim} w_T = w$$

it holds that

$$\operatorname{plim} z_T w_T = \operatorname{plim} z_T \operatorname{plim} w_T = zw.$$

Note that this property does not hold for expected values!

Thus, if the model is correctly specified and thus $Cov(y_t, y_{t-1}) = \alpha Var(y_t)$ holds, one has

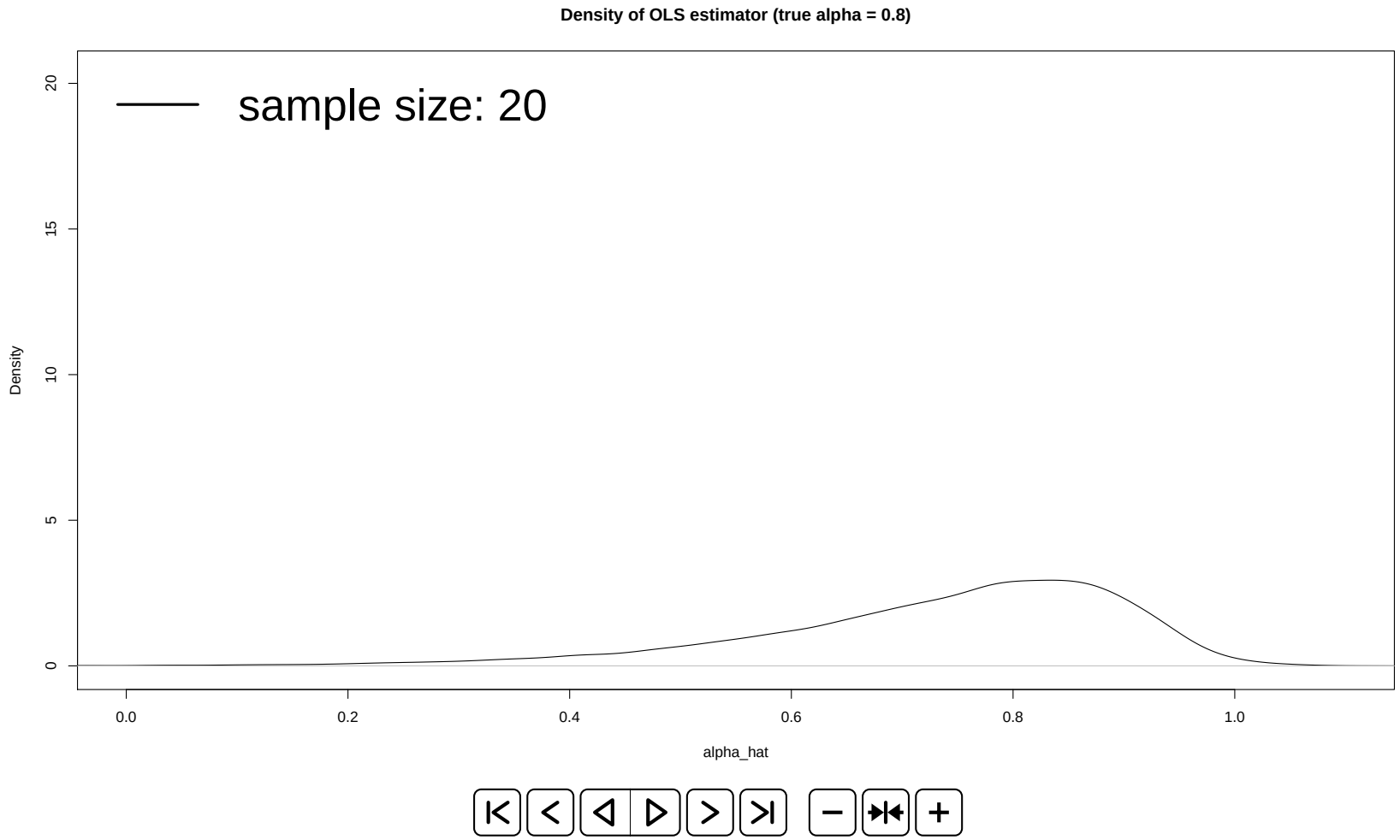
$$\operatorname{plim} \hat{\alpha} = \frac{\operatorname{plim} \frac{1}{T} \sum_{t=1}^{T} y_{t-1} y_t}{\operatorname{plim} \frac{1}{T} \sum_{t=1}^{T} y_{t-1}^2} = \alpha Var(y_t)/Var(y_t) = \alpha.$$

The OLS estimator is consistent.

The following figure (animated) shows the distribution of the OLS estimator of the AR(1) model when the true data generating process is

$$y_t = 0.8 y_{t-1} + u_t$$

where $u_t \sim IN(0, 1)$. Sample sizes vary from 20 to 1000.



Note: variance diminishes with larger sample size

- Using a central limit theorem one can also show that

$$\sqrt{T}(\hat{\alpha} - \alpha) \xrightarrow{d} N(0, q)$$

  with some asymptotic variance q.

In order to state the asymptotic properties of the OLS estimator for AR(p) models we state the conditions that the unknown but true AR(p) process has to fulfil:

- **Assumption A**:

  1. The AR(p) process is stationary and correctly specified. This implies that the order p is correctly chosen and the errors are not serially correlated.
  2. The errors are homoskedastic $Var(u_t) = \sigma^2$ for all periods $t = 1, \ldots, T$.
  3. $E[u_t^4] < \infty$ for all periods $t = 1, \ldots, T$.

Under Assumption A the following holds for the OLS estimator $\hat{\alpha}$:

* It is **consistent**, i.e.

$$\lim_{T\to\infty} P\left(|\hat{\alpha}-\alpha|<\epsilon\right)\to 1$$

for arbitrary $\epsilon>0$ or in short hand notation

$$\text{plim}_{T\to\infty}\hat{\alpha}=\alpha.$$

* The OLS estimator is **asymptotically normally distributed**:

$$\sqrt{T}\left(\hat{\alpha}-\alpha\right)\xrightarrow{d}N\left(0,Q\right)$$

where Q denotes the **asymptotic covariance matrix**

$$Q=\sigma^2\text{plim}_{T\to\infty}\left(T^{-1}X'X\right)^{-1}=\sigma^2\text{plim}_{T\to\infty}\left(T^{-1}\sum_{t=1}^{T}Y_{t-1}Y_{t-1}'\right)^{-1}.$$

A less precise but more intuitive way to write this result for reasonably large T is

$$\hat{\alpha}\approx N\left(\alpha,\sigma^2\left(X'X\right)^{-1}\right)\quad\text{or}\quad\hat{\alpha}\approx N\left(\alpha,\sigma^2\left(\sum_{t=1}^{T}Y_{t-1}Y_{t-1}'\right)^{-1}\right).$$

– **Remarks**

* Cf. Lütkepohl und Kraetzig (2008, Section 2.4.1), for proofs see Brockwell und Davis (1991, Sections 8.7 and 8.8) who require $u_t \sim IID(0, \sigma^2)$ instead of the weaker white noise assumptions.

* Weaker conditions that only require stability of the AR polynomial and some regularity of the error process are possible, see e.g. Davidson (2000, Chapter 6).

* OLS estimation is used in `EViews` if the lags of a variable `Y` are specified by `Y(-1)`, `Y(-2)`, etc. and is automatically used in the software `JMulTi`. In `R`, function `ar()` uses a different estimation technique by default (using the Yule-Walker equations). You need to add the option `method = "ols"` to get the least squares estimator.

* The OLS estimator is even asymptotically normally distributed **if** $\{y_t\}$ contains one unit root **and** the AR order is at least 2. However, in this case the asymptotic covariance matrix Q is singular (which, for example, implies that F tests do not work). This is due to the faster convergence rate of the unit root compared to the stationary roots.

- **Asymptotic distribution of the OLS estimator in case of a simple random walk**
  - Note that a simple random walk process

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

    with $\{u_t\}$ being white noise does not fulfil Assumption A. What are then the asymptotic properties? To see that things change, consider again $\frac{1}{T}\sum_{t=1}^{T} y_{t-1}^2$. Does it converge with increasing sample size T to a finite quantity? Remember that in case of a random walk $Var(y_t) = t\sigma^2$.

  - The **asymptotic distribution** of the OLS estimator for the autoregression parameter α in case of a random walk as DGP is

$$T\left(\hat{\alpha} - 1\right) = \frac{\frac{1}{T}\sum_{t=1}^{T} y_{t-1}u_t}{\frac{1}{T^2}\sum_{t=1}^{T} y_{t-1}^2} \xrightarrow{d} \frac{\frac{1}{2}\left(X - 1\right)}{Z}$$

    where $X = [W(1)]^2 \sim \chi^2(1)$ and $Z \sim \int_0^1 [W(r)]^2 dr$ and X and Z are **not** independent. For specialists: The $W(r)$ denotes Brownian motion which is the continuous time version of a simple discrete time random walk. This asymptotic distribution is often called **Dickey-Fuller distribution**.

    Thus: In case of $p = 1$ and $\alpha_1 = 1$ one obtains a

    * **completely different** asymptotic distribution (graphic shows distribution of $T\left(\hat{\alpha} - 1\right)$:

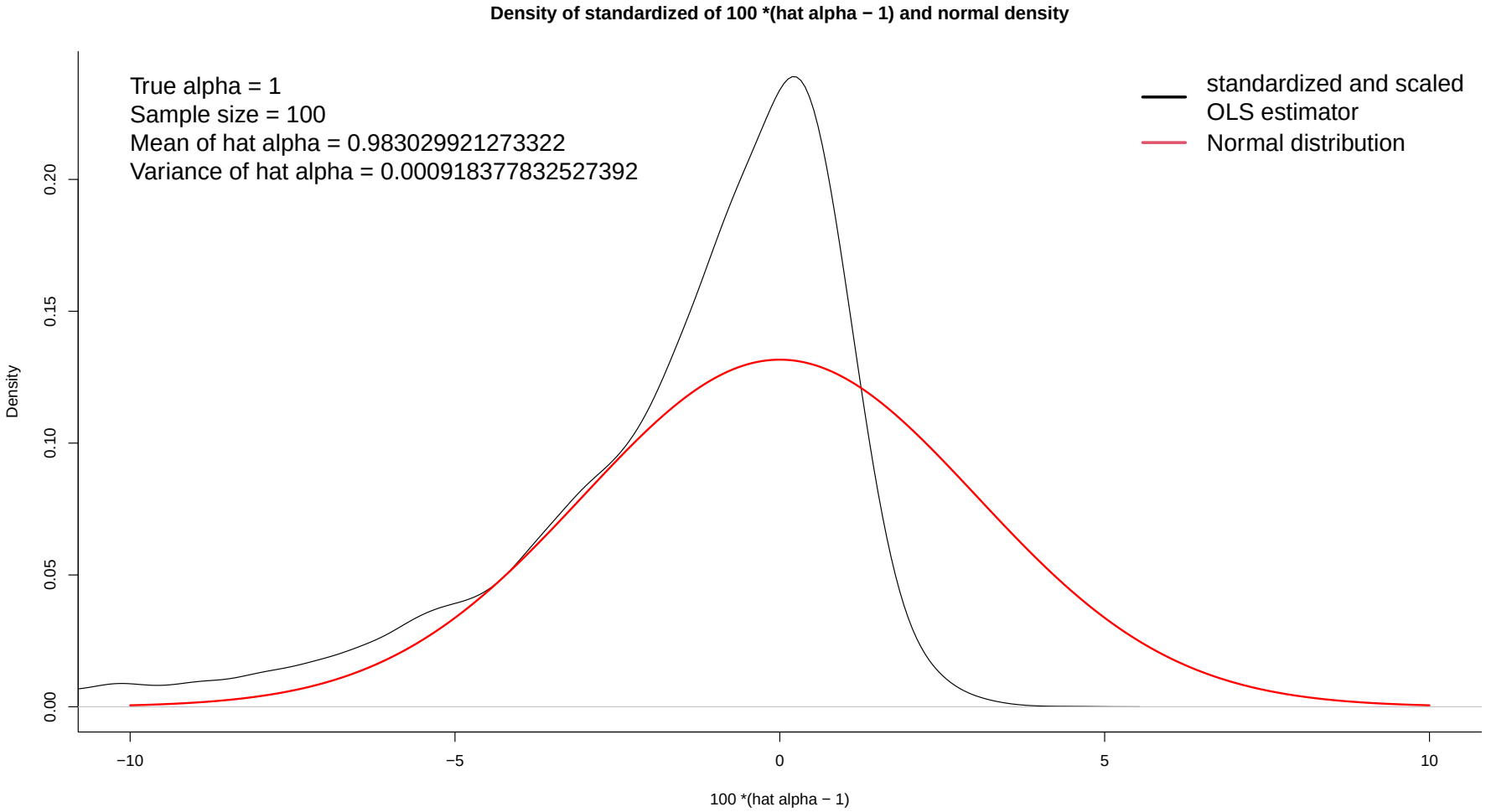
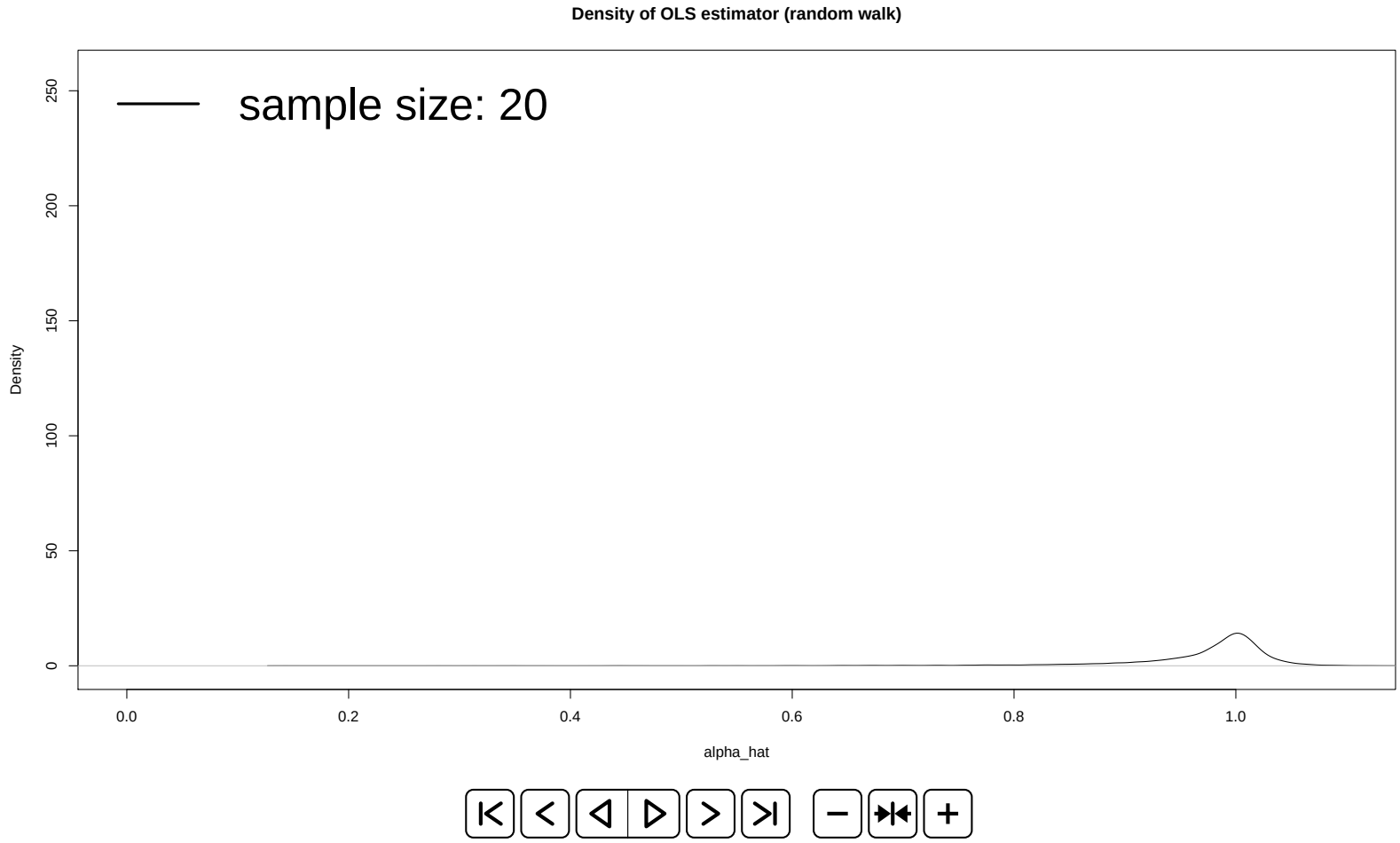


Figure 2.2.: Monte-Carlo simulation of Dickey-Fuller distribution for $n = 1000$ observations. See appendix A.6 for the `R` program.

- the OLS estimator converges much faster to the true value $\alpha = 1$. It is called **super consistent**. The variance of the OLS estimator converges much faster towards zero than in the stationary case. Here, the rate is T^{-1} instead of $T^{-1/2}$ in the stationary case. The reason is that with an increasing number of observations the variation of the regressor and the regressand increase — in contrast to the stationary case:



Note: even for $T = 20$ the variance of $\hat{\alpha}$ is much smaller than in the stationary case

* For conducting valid inference one has to check which distribution applies!! This can be done with an appropriate **unit root test**, see section 4.1.

* If the model contains **deterministic terms (constant, trend, seasonal dummies,**

structural breaks, etc.), the asymptotic distribution changes, see section 4.1.

* The result holds for u_t being white noise or Gaussian white noise.
* If the $\{u_t\}$ process is not white noise but **correlated**, **then the asymptotic distribution changes** $\longrightarrow$ **Phillips-Perron-Test**
* In order to avoid correlated u_t, one can also estimate a more general model, e.g. an AR(p) model. However, for testing purposes it has to be rewritten in a specific way $\longrightarrow$ **Augmented Dickey-Fuller-Test**, see section 4.1.

- **Estimation of the error variance** using the OLS residuals $\hat{u}_t = y_t - Y_{t-1}'\hat{\alpha}$
  - OLS estimator (for the model including a constant term)

$$\hat{\sigma}^2 = \frac{1}{T-p-1}\sum_{t=1}^{T}\hat{u}_t^2$$

  - Maximum Likelihood estimator

$$\tilde{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}\hat{u}_t^2$$

  - Asymptotic properties: under suitable conditions both estimators are consistent

$$\text{plim}_{T\to\infty}\hat{\sigma}_T^2 = \sigma^2$$
$$\text{plim}_{T\to\infty}\tilde{\sigma}_T^2 = \sigma^2$$

- A complete empirical analysis can be conducted using the `R` in section A.9. The data can be downloaded from Yahoo finance.

2.5.2. Alternative Estimators for AR(p) models

- **Nonlinear least squares**:
  A stationary AR(1) process can alternatively be written as (similarly for an AR(p) process)

$$y_t = \mu + v_t \tag{2.7a}$$
$$v_t = \alpha_1 v_{t-1} + u_t. \tag{2.7b}$$

  Inserting (2.7a) into (2.7b) delivers

$$y_t = \mu(1 - \alpha_1) + \alpha_1 y_{t-1} + u_t. \tag{2.8}$$

  This equation is nonlinear in the parameters μ and α_1 and can be estimated by nonlinear least squares. This is done in EViews if lagged endogenous variables are specified by `AR(1)`, `AR(2)`, etc.

  Be aware that in this case in the EViews output `C` refers to the estimated mean μ, not the estimated α_0.

- Both, the OLS and nonlinear LS estimator require that the first p observations of a sample are needed as starting values, thus reducing the number of observations by p. This can be avoided if one can assume $u_t \sim NID(0, \sigma^2)$. Then, one can apply **maximum likelihood (ML) estimation**. See section 5.2 for a brief introduction to maximum likelihood estimation of stationary AR(p) models and e.g. Hamilton (1994, Sections 5.2 and 5.3) for details.

2.5.3. Estimation of MA(q) and ARMA(p, q) models

The estimation of MA(q) and ARMA(p, q) models cannot be done by OLS directly. Mainly three procedures are used:

- 2-step procedure using OLS techniques:

  1. step: if the MA or ARMA process is invertible, it has an infinite AR representation. This can be approximated by an AR(p^*) model with p^* rather large. So fit an AR(p^*) model with $p^* >> \max(p, q)$ and calculate residuals $\hat{u}_t(p^*) = y_t - Y_{t-1}^{*\prime}\hat{\alpha}^*$, $t = 1, \ldots, T$

  2. step: Estimate ARMA(p, q) model with residuals $\hat{u}_t(p^*)$

  $$y_t = \nu + \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + u_t + m_1 \hat{u}_{t-1}(p^*) + \cdots + m_q \hat{u}_{t-q}(p^*)$$

  using the OLS estimator.

  Under suitable conditions asymptotically normally distributed.

- Maximum likelihood (ML) estimation. See section 5.2 for an introduction to ML estimation or Kapitel 5 in Fortgeschrittene Ökonometrie. This estimator is highly nonlinear in the ARMA parameters, requires nonlinear optimization techniques and starting values. The latter can be obtained via the first method. Under suitable conditions both estimators are asymptotically normally distributed.

A detailed treatment can be found in Hamilton (1994, Chapter 5). Technical details for algorithms and proofs can be found in Brockwell und Davis (1991, Chapter 8).

- Nonlinear least squares with backcasting techniques (EViews 5, User Guide, Chapter 17; EViews 6, User guide, Chapter 26)

3. Forecasting (financial) time series

3.1. Some general remarks and definitions

- To begin with, one may call **prediction** a general statement about the dependent variable given some conditioning information, e.g. predicting the wage of an individual knowing his/her age, experience, education etc.
- **Forecasting** is a particular type of prediction of a *future* value of the dependent variable or a value of it that is outside the current sample.
- A forecasting rule is any systematic operational procedure for making statements about future events.
- Before we continue, we need to define the conditioning **information set** for which we introduce the general notation $\mathcal{I}_t$. Intuitively, $\mathcal{I}_t$ contains all sets for which a probability can be assigned. In

case of lagged endogenous variables, for example in case of AR(p) processes, these sets contain all combinations of intervals in which past values could have been observed. In practice, one often writes for the given example $\mathcal{I}_{t-1} = \{y_{t-1}, y_{t-2}, \ldots\}$. Then in case of an AR(1) process

$$y_t = \alpha y_{t-1} + u_t, \quad u_t \sim i.i.d.(0, \sigma^2)$$

one has

$$E[y_t|\mathcal{I}_{t-1}] = \alpha y_{t-1} \neq E[y_t] = 0.$$

Thus, knowing the condition helps to predict y_t. Predictability is not only limited to the (conditional) mean. One may state this even more general w.r.t. to densities.

- A random variable is **unpredictable** with respect to a given information set $\mathcal{I}_{t-1}$ if the conditional probability distribution $F(y_t|\mathcal{I}_{t-1})$ is identical to the unconditional/marginal probability distribution $F(y_t)$

$$F(y_t|\mathcal{I}_{t-1}) = F(y_t).$$

- Any prediction is very likely to be false. Let $y_{T+h|T}$ denote a predictor for y_{T+h} based on some information set up to time T. Then the **prediction error** is

$$e_{T+h|T} = y_{T+h} - y_{T+h|T}.$$

To evaluate the prediction error one may e.g. miminize the

  - **mean squared error of prediction (MSEP)** $MSEP(y_{T+h|T}) \equiv E[(y_{T+h} - y_{T+h|T})^2]$

– **mean absolute error of prediction(MAEP)** $MAEP(y_{T+h|T}) \equiv E[|y_{T+h} - y_{T+h|T}|]$

The function of which the mean is taken is called **loss function**, in case of the $MSEP$ it is the **squared prediction error**. Note that it does not make much sense to minimize the squared error of prediction $\left(y_{T+h} - y_{T+h|T}\right)^2$ alone since the optimal predictor may vary from sample to sample even if the information set stays the same. One therefore takes the mean of all potential samples.

- By the law of iterated expectations, the MSEP (equivalently for the MAEP) can be written as

$$MSEP(y_{T+h|T}) = E\left[E[(y_{T+h} - y_{T+h|T})^2 \big| \mathcal{I}_T]\right], \tag{3.1}$$

where $\mathcal{I}_T$ denotes the information set used to compute $y_{T+h|T}$. Note that $E[(y_{T+h} - y_{T+h|T})^2 \big| \mathcal{I}_T]$ measures the mean of squared prediction errors for a given path up to time T due to the conditioning on $\mathcal{I}_T$ so that $y_{T+h|T}$ is the same while y_{T+h} varies randomly. Some authors call (more precisely) $E[(y_{T+h} - y_{T+h|T})^2 \big| \mathcal{I}_T]$ the mean squared error of prediction and denote the expression $E\left[E[(y_{T+h} - y_{T+h|T})^2 \big| \mathcal{I}_T]\right]$ as mean integrated squared error of prediction since it 'integrates' out the impact of history up to time T. Both are the same in case of ARMA processes so that we follow the standard textbook usage and call (3.1) MSEP (Lütkepohl, 2005, Section 2.2.2), (Hamilton, 1994, Chapter 4). These concepts in general differ in case of nonlinear time series processes.

- The **minimal mean squared prediction error** is obtained by using the conditional expectation

$E[y_{T+h}|\mathcal{I}_T]$ (does not minimize the MAEP), i.e. use $y_{T+h|T} = E[y_{T+h}|\mathcal{I}_T]$. Thus, forecasting by using conditional expectations implicitly means that one aims at minimizing the mean squared prediction error.

- Possible approaches for predicting time series data:
  - using exclusively past/current observations (univariate/multivariate time series models — example: AR(p)/VAR(p) model)
  - using regression models including conditioning/explanatory variables

    $$y_t = \beta_1 + \beta_2 x_{t2} + \cdots + \beta_k x_{t,k} + u_t$$

    where u_t is white noise and the x_{tj}'s are observable at time t. Having found a regression model explaining y_t is not sufficient for predictions since the x_{tj}'s are unknown in future periods. One then needs an auxiliary model for predicting all explanatory variables.
  - a combination of the pure time series and the pure regression approach: **dynamic econometric models**

    Example:

    $$y_t = \beta_0 + \alpha_1 y_{t-1} + \beta_1 x_{t1} + \beta_2 x_{t-1,1} + \beta_3 x_{t2} + u_t$$

- **Important issues**:
  - How to choose the best suited approach for a given prediction problem?
  - Which model framework should be chosen?
    univariate vs. multivariate models, linear vs. non-linear models,...)
  - Is it necessary for obtaining good forecasts to have a model with good explanatory power (*'good' model*)?
  - Alternatives to econometric/statistical forecasts that are based on models: judgement, expert opinions,...
  - Forecasting horizon
- **Further issues — not treated here**
  - Combination of forecasts and encompassing forecasts
  - Interval forecasts
  - Forecasting the complete conditional density

3.2. Decomposition of prediction errors

- Some notation
  - **Optimal predictor** if MSEP has to be minimized: conditional expectation

$$y_{T+h|T} \equiv E[y_{T+h}|\mathcal{I}_T]$$

Note that the optimal predictor $y_{T+h|T}$ is in general unfeasible since the conditional expected value of y_{T+h} given the information set is unknown. The conditional expected value (and the conditional density) would be known if the DGP were known.

Supplement (not applied): Proof:

1. Let p_T be any predictor that uses all information given by $\mathcal{I}_T$. Without loss of generality one can set

$$p_T = E[y_{T+h}|\mathcal{I}_T] + a_T$$

where a_T denotes the deviation of the predictor p_T from the conditional expectation.

2. By the law of iterated expectations one has for the MSEP

$$E\left[(y_{T+h} - p_T)^2\right] = E\left[E\left[(y_{T+h} - p_T)^2|\mathcal{I}_T\right]\right]$$

We first focus on the conditional expectation $E\left[(y_{T+h} - p_T)^2|\mathcal{I}_T\right]$.

3. Next we show that the conditional expectation $E\left[(y_{T+h} - p_T)^2 | \mathcal{I}_T\right]$ is minimized for $a_T = 0$:

$$\begin{aligned} E\left[(y_{T+h} - p_T)^2 | \mathcal{I}_T\right] &= E\left[(y_{T+h} - (E[y_{T+h}|\mathcal{I}_T] + a_T))^2 | \mathcal{I}_T\right] \\ &= E\left[(y_{T+h} - E[y_{T+h}|\mathcal{I}_T] - a_T)^2 | \mathcal{I}_T\right] \\ &= E\left[((y_{T+h} - E[y_{T+h}|\mathcal{I}_T]) - a_T)^2 | \mathcal{I}_T\right] \\ &= E\left[(y_{T+h} - E[y_{T+h}|\mathcal{I}_T])^2 | \mathcal{I}_T\right] \\ &- 2E\left[a_T(y_{T+h} - E[y_{T+h}|\mathcal{I}_T]) | \mathcal{I}_T\right] \\ &+ E\left[a_T^2 | \mathcal{I}_T\right] \end{aligned}$$

   Now we have 3 terms: the first one is independent of a_T and can thus be ignored for finding the minimum. Let us look at the second term $E\left[a_T(y_{T+h} - E[y_{T+h}|\mathcal{I}_T]) | \mathcal{I}_T\right]$. Since a_T is known given $\mathcal{I}_T$, it can be taken out of the conditional expectation. Then one can solve the conditional expectation and one obtains

$$\begin{aligned} E\left[a_T(y_{T+h} - E[y_{T+h}|\mathcal{I}_T]) | \mathcal{I}_T\right] &= a_T(E[y_{T+h}|\mathcal{I}_T] - E\left[E\left[y_{T+h}|\mathcal{I}_T\right]|\mathcal{I}_T\right]) \\ &= a_T(E[y_{T+h}|\mathcal{I}_T] - E[y_{T+h}|\mathcal{I}_T]) = 0 \end{aligned}$$

   Finally, the third conditional expectation contains a square and is thus minimized if $a_T = 0$.

4. Since the above step holds for any realization, that is for any history $\mathcal{I}_T$, it holds also for the MSEP. More formally

$$\begin{aligned} E\left[(y_{T+h} - p_T)^2\right] &= E\left[E\left[(y_{T+h} - E[y_{T+h}|\mathcal{I}_T])^2 | \mathcal{I}_T\right]\right] \\ &- 2E\left[a_T 0\right] \\ &+ E\left[E\left[a_T^2 | \mathcal{I}_T\right]\right] \\ &= E\left[(y_{T+h} - E[y_{T+h}|\mathcal{I}_T])^2\right] + E\left[a_T^2\right] \end{aligned}$$

   which again is minimized for $a_T = 0$. q.e.d.

5. We conclude that the MSEP is minimized if we use the predictor $p_T = E[y_{T+h}|\mathcal{I}_T]$, i.e. we use the conditional expectation.

– In order to estimate the conditional expectation one has in general to select/specify a model that contains a set of conditional expectations that vary with respect to a parameter vector θ, say an AR(3) model.

– In general one has a set of model candidates, say all AR(p) models with $p = 0, 1, \ldots, p_{max}$ where the maximum order p_{max} is chosen by the researcher. Let M_j denote the j-th model of the available set of models. The predictor based on model M_j with parameter vector θ_j is

$$E[y_{T+h}|\mathcal{I}_T, M_j, \theta_j].$$

At this point the parameter vector θ_j is unspecified.

The **optimal predictor in the MSEP sense given model** M_j is obtained by minimizing

$$\min_{\theta_j \in M_j} E\left[(y_{T+h} - E[y_{T+h}|\mathcal{I}_T, M_j, \theta_j])^2\right]$$

with respect to all θ_j included in model M_j. This minimizing parameter vector θ_j^0 is frequently called **pseudo-true** parameter vector.

Note that the pseudo-true parameter vector can change if one minimizes another criterion such as e.g. MAEP.

– A feasible predictor of y_{T+h} given information set $\mathcal{I}_T$ and model M_j is obtained by estimating the model parameters θ_j and inserting them into the conditional expectation of model M_j

$$\hat{y}_{T+h|T} \equiv E[y_{T+h}|\mathcal{I}_T, M_j, \hat{\theta}_j].$$

Feasible means that the predictor can be computed once a sample with observations up to time T is available.

- During the process described a number of wrong decisions can be made that all contribute to the total prediction error.
- **Classification of prediction errors**

$$\begin{aligned}\hat{e}_{T+h|T} &= y_{T+h} - \hat{y}_{T+h|T} \\ &= \underbrace{y_{T+h} - E[y_{T+h}|\mathcal{I}_T]}_{\text{unavoidable error}} \\ &+ \underbrace{E[y_{T+h}|\mathcal{I}_T] - E[y_{T+h}|\mathcal{I}_T, M_j, \theta_j^0]}_{\text{approximation error/model misspecification}} \\ &+ \underbrace{E[y_{T+h}|\mathcal{I}_T, M_j, \theta_j^0] - E[y_{T+h}|\mathcal{I}_T, M_j, \hat{\theta}_j]}_{\text{estimation error/parameter uncertainty}}\end{aligned}$$

- **Example**:
  DGP: AR(2) process $y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + u_t$,
  selected model: AR(1) model $y_t = a y_{t-1} + v_t$:

$$\begin{aligned}y_{T+1|T} = E[y_{T+1}|y_T, y_{T-1}, \ldots, y_0] &= \alpha_1 y_T + \alpha_2 y_{T-1} \\ E[y_{T+1}|y_T, y_{T-1}, \ldots, y_0, AR(1), a] &= a y_T \\ \hat{y}_{T+1|T} = E[y_{T+1}|y_T, y_{T-1}, \ldots, y_0, AR(1), \hat{a}] &= \hat{a} y_T\end{aligned}$$

  – Unavoidable error: u_{T+1}

  - Approximation error: $\alpha_1 y_T + \alpha_2 y_{T-1} - a_0 y_T$
    where a_0 denotes the parameter value that minimizes the mean squared approximation error.
  - Estimation error: $(a_0 - \hat{a}) y_T$

- **Notes**:
  - One cannot avoid the genuine prediction error $e_{T+h|T} = y_{T+h} - E[y_{T+h}|\mathcal{I}_T]$.
  - The approximation error is zero if the DGP is in the chosen model M_j, e.g. data are generated by an AR(p) process and we use an AR(q) model for prediction, where $p \leq q$.
  - In order to reduce a possible approximation error one may have to select a model that is 'closer' to the correct model (includes DGP). It may have more parameters.
  - If the correct model has many parameters, there is a tradeoff between facing large parameter uncertainty and/or a considerable approximation error. Thus, for prediction a 'wrong' model may be superior to the correct model that contains the DGP.
  - Model selection procedures are designed to optimize tradeoff between approximation and estimation error by miminizing the mean of the squared approximation and estimation error, see the following section.

- **Further potential sources for prediction errors**:
  - Parameter instability: the parameters of the model may change over time. In this case the stationarity assumption for the model is violated.
  - Variable mis-measurement: data is not exactly measured or is not yet completely known (e.g.: preliminary GNP data)
  - Initial condition uncertainty: the starting value y_0 in a dynamic model is also random.
  - Incorrect exogeneity assumptions.
  - Policy changes: the model may change completely due to policy changes.

3.3. AR Model Specification

See Lütkepohl und Kraetzig (2008, Section 2.5)

- (Economic) theory usually does not give many hints about the possible AR order p. Therefore p has to be determined with statistical methods.
- Model specification procedures are also useful to identify a model for which the assumptions are fulfilled.
- There are three possible procedures:
  - Model selection criteria
  - Sequential testing: Start with a large lag order $p = p_{max}$ and test for significance of the parameter for the p_{max}th lag. If it is significantly different from zero, you are done, if not, reduce the lag order to $p = p_{max} - 1$ and repeat the testing procedure. Continue until you are able to reject the null hypothesis.
  - Box-Jenkins method:
    The Box-Jenkins method requires to define and estimate the partial autocorrelation function. Since this procedure is not so popular any more, we do not discuss it here.

Model selection criteria

- **General structure**:

$$Criterion(p) = \ln \tilde{\sigma}^2(p) + \underbrace{c_T \phi(p)}_{\text{penalty term}}$$

with

$$\tilde{\sigma}^2(p) = \frac{1}{T} \sum_{t=1}^{T} \hat{u}_t^2(p)$$

where the $\hat{u}_t(p)$'s denote the residuals of a fitted AR(p) model.

  - One chooses a maximum order p_{max} and computes $Criterion(p)$ for $p = 0, 1, \ldots, p_{max}$. The selected order $\hat{p}$ is the order for which the selection criterion achieves its *minimum*.
  - $\frac{\partial \phi(p)}{\partial p} > 0$, i.e. adding lags increases the penalty term.
  - c_T can be seen as the weight of $\phi(p)$ in the criterion function. $\frac{\partial c_T}{\partial T} < 0$. i.e. adding lags for short time series (T small) increases the penalty term by a larger amount than for long time series.
  - $\tilde{\sigma}^2(p) \leq \tilde{\sigma}^2(p-1)$ when AR models are fitted by OLS.
  - Note that the maximum likelihood estimator for the error variance is used, i.e. there is no correction for degrees of freedom.

  - For $p = 0, 1, \ldots, p_{max}$ the sample size used for fitting the AR(p) model should be kept fixed, see Ng und Perron (2005) for a simulation study.

- **Standard order selection criteria**: (all have $\phi(p) = p$)
  - **Akaike Information Criterion (AIC)**

$$AIC(p) = \ln \tilde{\sigma}^2(p) + \frac{2}{T}p$$

  - **Hannan-Quinn Criterion (HQ)**

$$HQ(p) = \ln \tilde{\sigma}^2(p) + \frac{2 \ln \ln T}{T}p$$

  - **Schwarz (and Rissanen) Criterion (SC, SIC or BIC)**

$$SC(p) = \ln \tilde{\sigma}^2(p) + \frac{\ln T}{T}p$$

- **Asymptotic properties**
  - Condition I: true order $p_0 < \infty$ and $p_{max} \geq p_0$, , $\hat{p}_T$ denotes the estimated order, the roots of the AR polynomial are all outside or on the unit circle (unit root and stationary processes!)
  - AIC: $P(\hat{p}_T > p_0) > 0$
  - HQ: $\text{plim}_{T \to \infty} \hat{p}_T = p_0$

– SC: $\hat{p}_T \xrightarrow{a.s.} p_0$

– $T \geq 16$: $\hat{p}(SC) \leq \hat{p}(HQ) \leq \hat{p}(AIC)$

– Condition II: true order p_0 infinite: AIC is the best

3.4. Prediction with AR models

See Lütkepohl und Kraetzig (2008, Section 2.8)

- **One-step ahead predictions** for $T+1$ based on an AR(p) model

$$(1 - \alpha_1 L - \cdots \alpha_p L^p) y_t = u_t$$

with **known** AR parameters:

$$\begin{aligned} y_{T+1|T} &= E[y_{T+1}|y_T, y_{T-1}, \ldots, y_1, \ldots, y_{-p+1}] \\ &= \alpha_1 y_T + \alpha_2 y_{T-1} + \cdots + \alpha_p y_{T-p+1} \end{aligned}$$

- One can show that the **prediction based on the conditional expectation** has the smallest **mean squared error of prediction (MSEP) if**

$$u_t \sim i.i.d.(0, \sigma^2).$$

The MSEP, often also abbreviated by MSE, is given for the present case:

$$MSE = E\left[\left(y_{T+1} - y_{T+1|T}\right)^2\right] = E[u_{T+1}^2] = \sigma^2$$

- Predictions for several time periods ahead, $h > 1$, usually abbreviated as h-**step predictions**, can be calculated recursively

$$y_{T+h|T} = \alpha_1 y_{T+h-1|T} + \alpha_2 y_{T+h-2|T} + \cdots + \alpha_p y_{T+h-p|T}, \text{ with } y_{T+j|T} = y_{T+j} \text{ for } j \leq 0.. \quad (3.2)$$

- **Prediction error**:
  A stationary AR(p) process $\alpha(L)y_t = u_t$ can be represented as

$$y_t = \phi(L)u_t = u_t + \phi_1 u_{t-1} + \phi_2 u_{t-2} + \cdots .$$

  This representation is an example of an infinite **Moving Average Process (MA(∞) process)**. It is possible to compute the moving average parameters ϕ_j, $j = 1, 2, \ldots$, by solving $\alpha(L)\phi(L) = 1$ leading to $\phi_s = \sum_{j=1}^{s} \phi_{s-j}\alpha_j$, $s = 1, 2, \ldots$ with $\phi_0 = 1$ and $\alpha_j = 0$ for $j > p$ (see e.g. (2.4) in section 2.1.2).

  One then obtains for the conditional expectation

$$\begin{aligned} y_{T+h|T} &= E\left[y_{T+h}|y_T, y_{T-1}, \cdots\right] \\ &= E\left[u_{T+h} + \phi_1 u_{T+h-1} + \cdots + \phi_{h-1}u_{T+1} + \phi_h u_T + \phi_{h+1}u_{T-1} + \cdots | y_T, y_{T-1}, \cdots\right] \\ &= \phi_h u_T + \phi_{h+1}u_{T-1} + \cdots \end{aligned}$$

and the prediction error is given by

$$\begin{aligned} y_{T+h} - y_{T+h|T} &= u_{T+h} + \phi_1 u_{T+h-1} + \cdots + \phi_{h-1} u_{T+1} + \phi_h u_T + \phi_{h+1} u_{T-1} + \cdots \\ &\quad - (\phi_h u_T + \phi_{h+1} u_{T-1} + \cdots) \\ &= u_{T+h} + \phi_1 u_{T+h-1} + \cdots + \phi_{h-1} u_{T+1}. \end{aligned}$$

- The **mean squared error of prediction** (MSEP) of an h-step prediction is therefore

$$\sigma_y^2(h) = E\left[\left(y_{T+h} - y_{T+j|T}\right)^2\right] = \sigma^2 \sum_{j=0}^{h-1} \phi_j^2,$$

since the errors u_t are assumed to be i.i.d.
Note that *if* $u_t \sim i.i.d.(0, \sigma^2)$, then the h-step predictor $y_{T+h|T}$ is optimal, that is it exhibits among all possible prediction methods the **smallest** MSEP.

- If the errors u_t are just white noise, then equation (3.2) delivers the best *linear* prediction (where then the parameters depend on the goal function and are then pseudo-true values).

- Since the variance σ_y^2 of a stationary process is given by

$$\sigma_y^2 = \sigma^2 \sum_{j=0}^{\infty} \phi_j^2$$

(see Neusser (2009, Abschnitt 2.4, Theorem 2.3)), the variance of the prediction error $\sigma_y^2(h) = \sigma^2 \sum_{j=0}^{h-1} \phi_j^2$ of an h-step prediction approaches σ_y^2 with the number of steps h increasing.

- If the errors u_t are Gaussian white noise, then the prediction errors are also i.i.d. normally distributed and one can compute a **prediction interval** for a given confidence level $(1-\gamma)100\%$

$$[y_{T+h|T} - c_{1-\gamma/2}\sigma_y(h), y_{T+h|T} + c_{1-\gamma/2}\sigma_y(h)]$$

  where $c_{1-\gamma/2}$ denotes the $1-\gamma/2$ quantile of the standard normal distribution.

- One can use the predictor (3.2) also for **autoregressive processes with a unit root**. As a technical remark note that in this case the parameters ϕ_j do not represent the parameters of an MA(∞) process, simply because the latter does not exist.

  Observe that the prediction variance increases in this case with the number of prediction periods h towards infinity since the variance of a random walk also increases unboundedly with t.

- **In empirical work** the parameters of the data generating process are unknown and have to be estimated. One therefore replaces in (3.2) all unknown parameters by their estimates and obtains the feasible predictor

$$\hat{y}_{T+h|T} = \hat{\alpha}_1\hat{y}_{T+h-1|T} + \hat{\alpha}_2\hat{y}_{T+h-2|T} + \cdots + \hat{\alpha}_p\hat{y}_{T+h-p|T}. \tag{3.3}$$

  This leads to the prediction errors

$$\begin{aligned} y_{T+h} - \hat{y}_{T+h|T} &= [y_{T+h} - y_{T+h|T}] + [y_{T+h|T} - \hat{y}_{T+h|T}] \\ &= \sum_{j=0}^{h-1} \phi_j u_{T+h-j} + [y_{T+h|T} - \hat{y}_{T+h|T}]. \end{aligned}$$

The second term on the right hand side through which the estimation uncertainty is captured approaches zero with increasing sample size **if** the parameters are estimated **consistently** (correct order, etc.).

The **variance of the prediction error with estimated parameters** is given by

$$\sigma^2_{\hat{y}}(h) = E\left[\left(y_{T+h} - \hat{y}_{T+h|T}\right)^2\right] = \sigma^2_y(h) + \text{'something that appraoches zero for } T \to \infty\text{'}.$$

Note that in small samples the additional variance due to the estimation uncertainty is non-negligible and should be included when computing prediction intervals. Details can be found e.g. in Lütkepohl (2005, Section 3.5).

3.5. Evaluation of forecasts

- **Properties of optimal forecasts**
  - The optimal predictor (=use model that includes DGP with the true parameter values) is unbiased (by definition).
  - Prediction errors based on optimal 1-step ahead predictions are white noise.
  - Prediction errors based on optimal h-step ahead predictions may be correlated. (If so, they follow a Moving Average Process of order $h-1$ or less).
  - The variance of the prediction errors of h-step predictions converges to the unconditional variance (in case the DGP is stationary).

Thus: Residuals from good forecasts must be unpredictable! This property can be used for **checking the quality of a forecasting procedure by testing** e.g. in

$$\hat{e}_{t+1|t} = \beta_1 + \sum_{i=1}^{k} \beta_i x_{ti} + error_t, \quad t = 1, \ldots, T,$$

whether $\beta_i = 0$, $i = 2, \ldots, k$. If some x_{ti} has significant explaining power, then we should incorporate this variable into our forecasting procedure.

- **Quantities to compare forecasts**
  - **Forecast error**: $\hat{e}_{T+h|T} = y_{T+h} - \hat{y}_{T+h|T}$.
  - **Forecast percent error**: $\frac{y_{T+h} - \hat{y}_{T+h|T}}{y_{T+h}}$.
- **Measures to compare forecast errors**:
  - **Mean error of prediction (MEP)**

$$MEP(h) = E\left[\hat{e}_{T+h|T}\right]$$

  - **Mean squared error of prediction (MSEP)**

$$MSEP(h) = E\left[\hat{e}^2_{T+h|T}\right]$$

  - **Root mean squared error of prediction (RMSEP)**

$$RMSEP(h) = \sqrt{E\left[\hat{e}^2_{T+h|T}\right]}$$

  - **Mean absolute error of prediction (MAEP)**

$$MAEP(h) = E\left[\left|\hat{e}_{T+h|T}\right|\right].$$

Notes:

- These measures can also be defined for the forecast percent error.
- The RMSEP and the MAEP have the same scale as the prediction error.
- The MAEP is not minimized by using the conditional expectation as predictor.
- There also exist measures that weight positive and negative prediction errors differently.
- All measures are generally not observable and have to be estimated. A popular tool for estimation are **out-of-sample forecasts**.

- **Out-of-sample forecasts**:
  - Split the sample into 2 parts with the first T_1 observations in subsample 1 and the second $T - T_1$ observations in subsample 2.
  - Estimate the model on basis of subsample 1 and predict y_{T_1+h} where $h \leq T - T_1$. Denote the prediction error by
$$\hat{e}_{T_1+h|T_1} = y_{T_1+h} - \hat{y}_{T_1+h|T_1}$$
  - Re-estimate the model using all data from $t = 1, \ldots, T_1 + 1$, predict y_{T_1+1+h} and save the prediction error.
  - Repeat the previous step until you predict y_T.
  - At the end you estimate the mean of your measure by averaging over all out-of-sample predictions. For example, the MSEP is estimated by
$$\widehat{MSEP} = \frac{1}{T - T_1 - h + 1} \sum_{j=T_1}^{T-h} \hat{e}^2_{j+h|j}.$$

- Advanced literature to testing and comparing predictive quality of various models: Diebold und Mariano (1995), West (1996), Giacomini und White (2006).

4. More on modeling time series

4.1. Unit root tests

4.1.1. Dickey-Fuller test

- **The simplest case**
  - Consider the estimation of the simple AR(1) model

$$y_t = \alpha y_{t-1} + u_t$$

  - Null and alternative hypothesis:

$$H_0 : \alpha = 1 \quad \text{versus} \quad H_1 : \alpha < 1$$

– t-statistic

$$t_T = \frac{\hat{\alpha} - 1}{\hat{\sigma}_{\hat{\alpha}}}$$

with

$$\hat{\sigma}^2_{\hat{\alpha}} = \frac{\hat{\sigma}^2}{\sum_{t=1}^{T} y^2_{t-1}}, \quad \hat{\sigma}^2 = \frac{1}{T-1} \sum_{t=1}^{T} (y_t - \hat{\alpha} y_{t-1})^2$$

– Asymptotic distribution of the t-statistic in case of a random walk as DGP

$$t_T \xrightarrow{d} \frac{\frac{1}{2}\left([W(1)]^2 - 1\right)}{\left[\int_0^1 [W(r)]^2 dr\right]^{1/2}}$$

A table of selected values of this distribution is given below.

- **Dickey-Fuller tests for various hypotheses**

A: **No deterministic trend**

1. So far:

$$H_0: \text{ random walk} \quad \text{versus} \quad H_1: \text{stationary AR(1) } \textbf{without} \text{ constant} \tag{I}$$

(Case 1 in Hamilton (1994, Section 17.4))

2. More relevant in empirical work:

$$H_0: \text{ random walk} \quad \text{versus} \quad H_1: \text{stationary AR(1) } \textbf{with} \text{ constant} \tag{II}$$

(Case 2 in Hamilton (1994, Section 17.4))

* Remark:
  Two representations of an AR(1) with constant

$$y_t - \mu_0 = \alpha(y_{t-1} - \mu_0) + u_t,$$
$$y_t = \nu + \alpha y_{t-1} + u_t, \quad \text{with } \nu = \mu_0(1-\alpha).$$

  Thus, in case of the null hypothesis $\alpha = 1$ the constant $\nu = \mu_0(1-1) = 0$ drops out!

* The asymptotic distribution changes if a constant is allowed under the alternative. One then has:

$$t_T \xrightarrow{d} \frac{\frac{1}{2}\left([W(1)]^2 - 1\right) - W(1)\int W(r)dr}{\left[\int_0^1 [W(r)]^2 dr - \left(\int W(r)dr\right)^2\right]^{1/2}},$$

  see the table below and e.g. Hamilton (1994, Section 17.4) for mathematical details

B: **Linear deterministic trend in DGP**

3. a) H_0**: Stochastic and deterministic trend (random walk with drift)**

$$y_t = \mu_0 + y_{t-1} + x_t, \quad \Delta y_t = \mu_0 + x_t, \quad x_t \text{ stationary AR(1)}$$

b) H_1: Deterministic trend only (difference stationary)

$$y_t = \mu_0 + \mu_1 t + x_t, \quad x_t \text{ stationary AR(1)}$$

Representation of the null and alternative hypothesis

$$y_t = \mu_0 + \mu_1 t + x_t, \quad x_t = \alpha x_{t-1} + u_t$$

$$H_0 : \alpha = 1 \quad \text{versus} \quad H_1 : \alpha < 1$$

Inserting delivers:

$$y_t - \mu_0 - \mu_1 t = \alpha \left(y_{t-1} - \mu_0 - \mu_1 (t-1) \right) + u_t$$
$$y_t = \mu_0 (1-\alpha) + \mu_1 t (1-\alpha) + \alpha \mu_1 + \alpha y_{t-1} + u_t$$

$$H_0 : \quad y_t = \mu_1 + y_{t-1} + u_t \quad \text{random walk with drift}$$
$$H_1 : \quad y_t = \nu + \delta t + \alpha y_{t-1} + u_t \quad \text{AR(1) with linear trend.}$$

Summary

$$H_0 : \text{random walk with drift} \quad \text{versus} \quad H_1 : \text{AR(1) with linear trend} \tag{III}$$

(Case 4 in Hamilton (1994, Section 17.4))

For this case the asymptotic distribution of the Dickey-Fuller test turns out to be different again (see e.g. Hamilton (1994, Equation 17.4.55)) and the table below.

4. If a linear trend is excluded from the alternative and the pair of hypotheses is the following

$$H_0 : \text{random walk with drift} \quad \text{versus} \quad H_1 : \text{AR(1) with constant}$$

(Case 3 in Hamilton (1994, Section 17.4)),
one finds that the t-statistic is asymptotically normally distributed.

This case is empirically not very relevant since a trend must be of stochastic nature due to the particular choice of the alternative hypothesis.

- **Asymptotic critical values for the Dickey-Fuller unit root tests**

Quantile	1%	2.5%	5%	10%	97.5%
			t-statistic		
Pair of hypotheses (I)	-2.56	-2.23	-1.94	-1.62	1.62
Pair of hypotheses (II)	-3.43	-3.12	-2.86	-2.57	0.24
Pair of hypotheses (III)	-3.96	-3.66	-3.41	-3.13	-0.66
			$T(\hat{\alpha}-1)$		
Pair of hypotheses (I)	-13.7	-10.4	-8.0	-5.7	1.6
Pair of hypotheses (II)	-20.6	-16.9	-14.1	-11.2	0.4
Pair of hypotheses (III)	-29.4	-25.1	-21.7	-18.2	-1.8

Quelle: Davidson und MacKinnon (1993, Table 20.1, p.708)

- **How to proceed in empirical work**
  - If a trend is expected in the data, one chooses the pair of hypotheses (III).
  - If the data should not contain a trend for economic reasons, e.g. interest rate date, one chooses the pair of hypotheses (II).
  - The fewer parameters have to be estimated, that is the smaller the number of the pair of hypothesis (I)-(III), the larger is the power of the test.
  - There is the possibility to test the null hypothesis $\alpha = 1$ und $\delta = 0$ jointly with an F test. The corresponding F statistic has also an asymptotic nonstandard distribution, see e.g. Hamilton (1994, Section 17.4). If this hypothesis is rejected, there is empirical evidence for a deterministic linear trend.

4.1.2. Unit Root Tests in the Presence of Autocorrelation

Augmented Dickey-Fuller Test

- In general, the DGP may have a larger order than 1 and be an **AR(p) process**

$$(1 - \alpha_1 L - \alpha_2 L^2 - \cdots - \alpha_p L^p) x_t = u_t$$
$$\alpha(L) x_t = u_t$$

  with u_t being white noise.

  - The asymptotic distributions of the Dickey-Fuller statistics were derived for null hypotheses based on the AR(1) case, e.g. $x_t = \alpha x_{t-1} + u_t$, $H_0 : \alpha = 1$, $H_1 : \alpha < 1$. **If the DGP is in fact an AR(p) process, then the asymptotic distributions all change!** Then the asymptotic distribution contains unknown parameters that depend on the correlation structure of the DGP! One solution is to estimate these additional parameters *nonparametrically* and adapt the test statistic accordingly $\longrightarrow$ **Phillips-Perron test** (This test will not be discussed in this course. Details are found in Hamilton (1994, Section 17.6))
  - Alternative: one includes the additional lags in the estimation equation $\longrightarrow$ **Augmented Dickey-Fuller test**

- In order to obtain the (Augmented) Dickey-Fuller test statistic for the general AR(p) case, one

has to use the following decomposition

$$x_t = (\alpha_1 + \alpha_2 + \cdots + \alpha_p)x_{t-1} + \alpha_1^*\Delta x_{t-1} + \cdots + \alpha_{p-1}^*\Delta x_{t-(p-1)} + u_t$$

or, after subtracting x_{t-1} on both sides and writing $\phi = (\alpha_1 + \alpha_2 + \cdots + \alpha_p) - 1 = -\alpha(1)$,

$$\Delta x_t = \phi x_{t-1} + \sum_{j=1}^{p-1} \alpha_j^* \Delta x_{t-j} + u_t. \tag{4.1}$$

$$\underbrace{(1 - \alpha_1^* L - \cdots - \alpha_{p-1}^* L^{p-1})}_{\alpha^*(L)} \Delta x_t = \phi x_{t-1} + u_t$$

Try this decomposition for an AR(2) process!

Note that

- the order of the AR model (4.1) in first differences is exactly $p - 1$;
- under H_0 and H_1 the stationarity condition holds for the $\alpha^*(z)$ polynomial, that is, all roots lie outside the unit circle. Thus, the dependent variable Δx_t is both under the null and the alternative hypothesis stationary (H_0: x_t contains random walk component, H_1: x_t stationary). The same holds for the lagged differences Δx_{t-j}, $j = 1, \ldots, p-1$;
- the process $\{x_t\}$ contains exactly one unit root if $\phi = 0$, and thus $\alpha(1) = 0$;
- this version of the Dicky-Fuller test that is augmented by lagged differences is called **Augmented Dickey-Fuller test (ADF test)**;

– Notice that under H_0 the regressor x_{t-1} on the right hand side is **non-stationary**. Since its variance increases with increasing sample size, one observes that the estimate $\hat{\phi}$ converges particularly fast towards the true value 0. This occurs with rate T (instead of $\sqrt{T}$ in the stationary case). One therefore calls the OLS estimator for ϕ in this case **superconsistent**. What is the rate under H_1?

- **The pairs of hypotheses**:

A: No trend

$$\begin{aligned} H_0 &: \text{unit root AR}(p) \quad \text{versus} \quad H_1 : \text{stationary AR}(p) \textbf{ without} \text{ constant} \\ H_0 &: \phi = 0 \quad \text{versus} \quad H_1 : \phi < 0 \end{aligned} \tag{I}$$

$$\begin{aligned} H_0 &: \text{unit root AR}(p) \quad \text{versus} \quad H_1 : \text{stationary AR}(p) \textbf{ with} \text{ constant} \\ H_0 &: \phi = 0 \quad \text{versus} \quad H_1 : \phi < 0 \end{aligned} \tag{II}$$

B: With trend

$$y_t = \mu_0 + \mu_1 t + x_t, \quad \Delta x_t = \phi x_{t-1} + \sum_{j=1}^{p-1} \alpha_j^* \Delta x_{t-j} + u_t$$

$$\begin{aligned} H_0 &: \text{unit root AR}(p) \text{ with drift} \quad \text{versus} \quad H_1 : \text{stationary AR}(p) \text{ with linear trend} \\ H_0 &: \phi = 0 \quad \text{versus} \quad H_1 : \phi < 0 \end{aligned} \tag{III}$$

- **Asymptotic distributions**
  - ϕ: The asymptotic distribution of

$$\frac{T\hat{\phi}}{1-\hat{\alpha}_1^*-\cdots-\hat{\alpha}_{p-1}^*}$$

    corresponds under H_0 with the asymptotic distribution of $T(\hat{\alpha}-1)$ for the corresponding pair of hypotheses in the AR(1) case, see previous table for critical values.
  - t-statistic for ϕ: The asymptotic distribution is given by the asymptotic distribution of the AR(1) case for the corresponding pair of hypotheses, see previous table for critical values.
  - The reason for this property is the superconsistency of the OLS estimator for ϕ under H_0. Further details including derivations can be found in Hamilton (1994, Section 17.7) or Davidson (2000, Chapter 14).
  - α_j^* and corresponding t statistics: like in the stationary AR model.
  - **All the results mentioned so far only hold if the order p is not chosen too small** that is
    - u_t is white noise or
    - u_t is approximately white noise. This means that given the sample size one makes the residuals resemble white noise as much as possible. One can achieve this by letting the

order p increase with sample size T such that the residual process becomes 'whiter'. This case occurs if the true order is infinite. It is important that the p for the estimated model does neither grow too fast nor too slow (Too fast: not enough observations to estimate parameters with small enough variance; too slow: residuals do not become white enough).

- In sum: to obtain a valid asymptotic distribution, the order p has to be large enough. **However**, a too large order leads to a loss in power (since the variance of parameter estimates decreases not fast enough).

  Luckily, it is possible to determine an appropriate order by **using model selection criteria like in the stationary case**: The Hannan-Quinn $HQ(n)$ and the Schwarz criterium $SC(n)$ are consistent if the time series is generated by an AR(p) process with finite p and $p_{max} \geq p$.

- In empirical work it may happen that the degree of integration is larger than $I(1)$. One then can use the *Pantula principle* to determine the appropriate order of integration, see e.g. Lütkepohl und Kraetzig (2008, Section 2.7.1).

- The case of (conditionally) heteroskedastic innovations is treated by Demetrescu (2010) who suggests robust standard errors.

- The power of the ADF test depends on the specification of the deterministic components and increases with sample size:

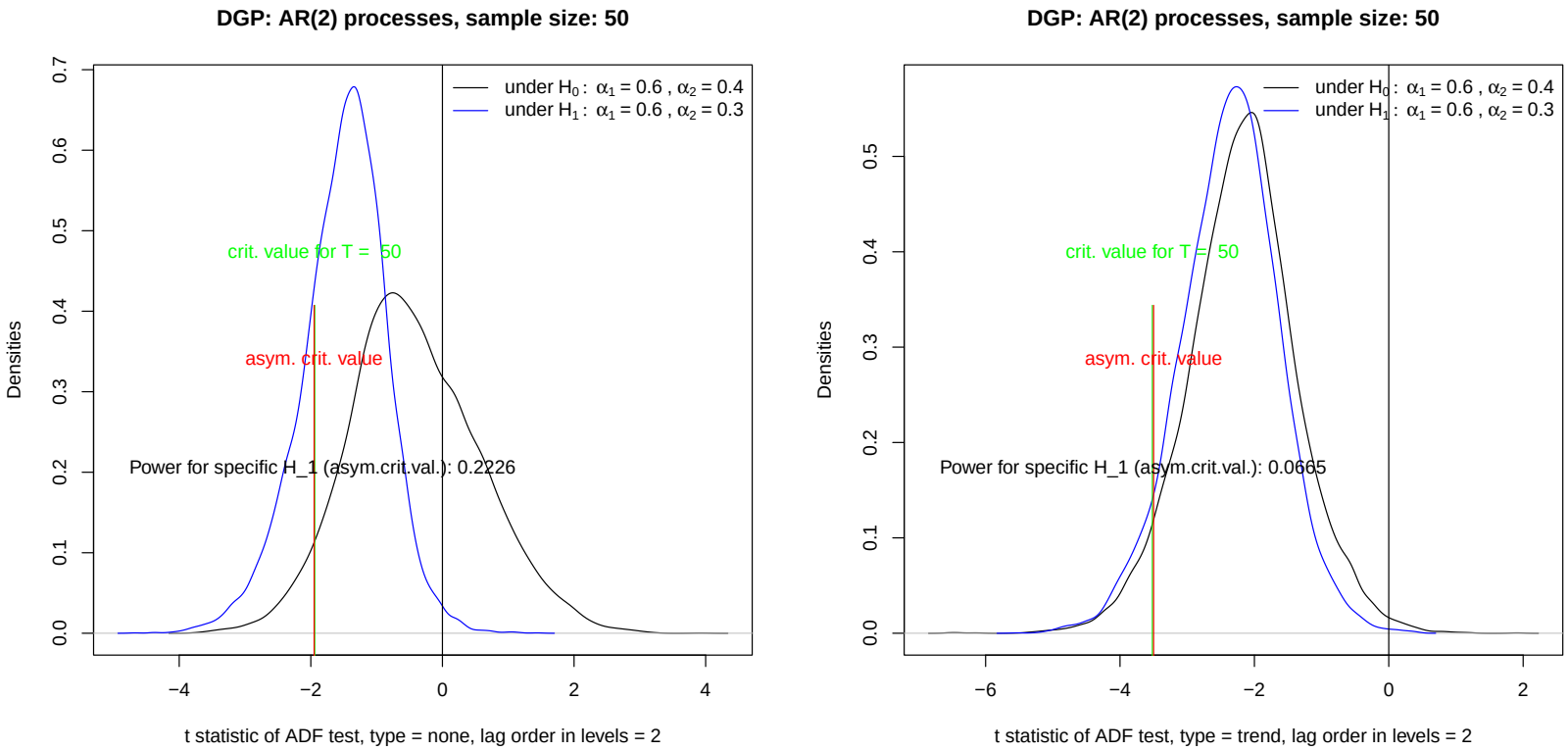


Figure 4.1.: Monte-Carlo simulation of distribution of ADF test statistic given $n = 50$ observations with parameters $\alpha_1 = 0.6$, $\alpha_2 = 0.4$ under H_0 and $\alpha_1 = 0.6$, $\alpha_2 = 0.3$ under H_1. See appendix A.7 for the `R` program.

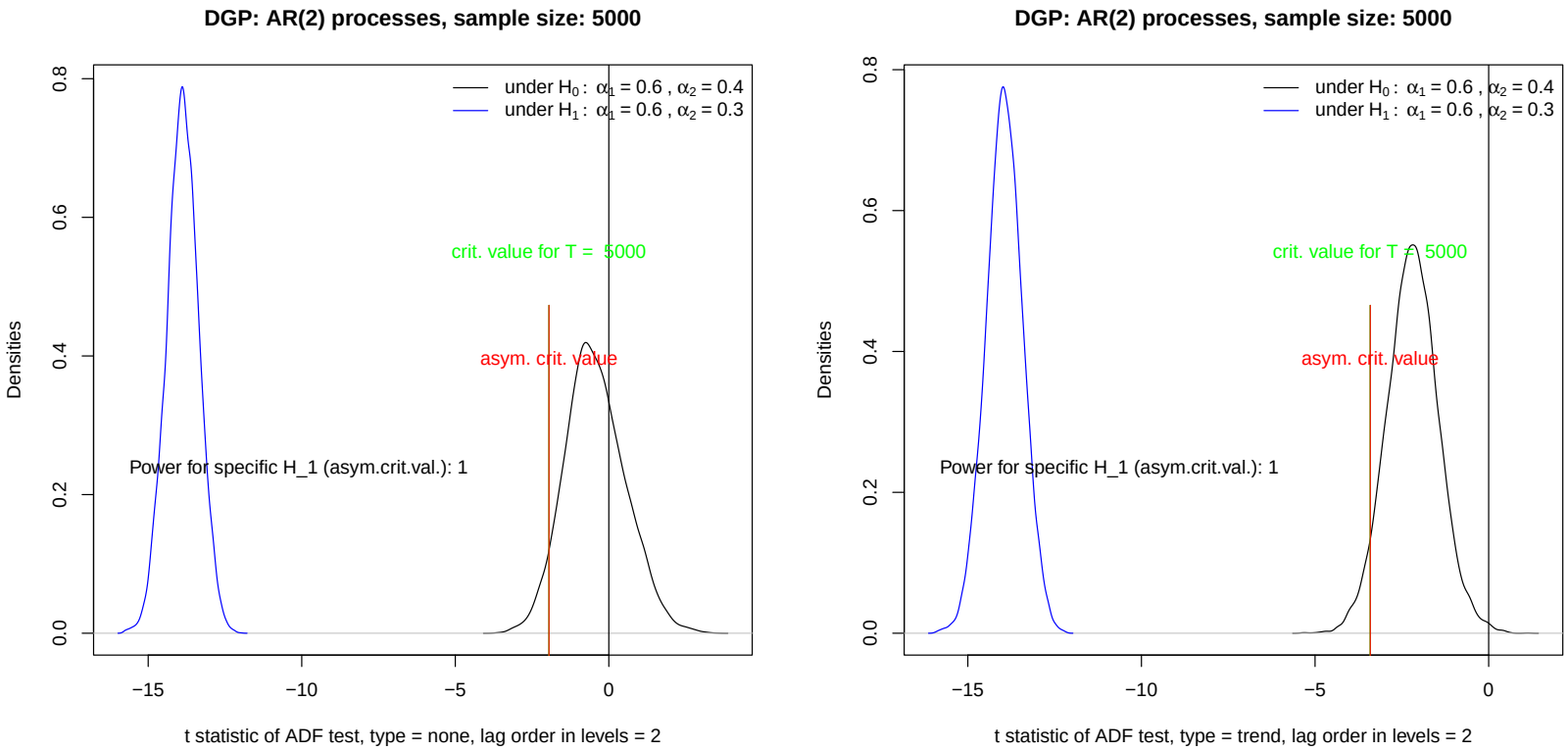


Figure 4.2.: Monte-Carlo simulation of distribution of ADF test statistic given $n = 5000$ observations with parameters $\alpha_1 = 0.6$, $\alpha_2 = 0.4$ under H_0 and $\alpha_1 = 0.6$, $\alpha_2 = 0.3$ under H_1. See appendix A.7 for the R program.

4.1.3. Other Unit Root Tests

- **Phillips-Perron Test**:
  see comments above and for an overview e.g. Kirchgässner, Wolters, und Hassler (2013, Section 5.3.3). A detailed derivation can be found in Hamilton (1994, Section 17.6).

- **KPSS Test**:
  The null hypothesis of all tests discussed so far is that $y_t \sim I(1)$. Kwiatkowski et al. (1992) have developed a test that allows to test stationarity directly using the pair of hypotheses:

$$H_0 : y_t \sim I(0) \quad \text{versus} \quad H_1 : y_t \sim I(1).$$

  - See e.g. Kirchgässner, Wolters, und Hassler (2013, Section 5.3.5) or Lütkepohl und Kraetzig (2008, Section 2.7.4) for an overview.
  - This test is based on the Beveridge-Nelson decomposition (not covered in this course but you may see the slides of **Fortgeschrittene Dynamische Ökonometrie** (available on request) that states that an autoregressive process with a unit root can be decomposed into a random walk and a stationary component

$$\begin{aligned} y_t &= z_t + \eta_t \\ z_t &= z_{t-1} + v_t, \quad v_t \sim iid(0, \sigma_v^2). \end{aligned}$$

– The pair of hypotheses can now be stated more precisely as

$$H_0 : \sigma_v^2 = 0 \quad \text{versus} \quad H_1 : \sigma_v^2 > 0.$$

If H_0 holds, then z_t is a constant and y_t is stationary.

– The test statistic is:

$$KPSS = \frac{1}{T^2} \frac{\sum_{t=1}^{T} \left[\sum_{j=1}^{t} (y_j - \bar{y})^2 \right]}{\hat{\sigma}_{\eta,\infty}^2}$$

where $\sigma_{\eta,\infty}^2$ denotes the so-called long-run variance of the stationary process $\{\eta_t\}$.

If y_t is stationary (H_0 holds), then the numerator converges. If y_t is non-stationary (H_1 holds), then the numerater diverges to plus infinity! Since for both hypotheses $\sigma_{\eta,\infty}^2 < \infty$, one rejects the null hypothesis if the numerator and thus the test statistic is too large.

– The asymptotic distribution of the KPSS test is also non-standard and is given in the tables of Kwiatkowski et al.

– If there is a linear trend under the null hypothesis (trend stationarity), then one considers

$$y_t = \mu_1 t + z_t + \eta_t.$$

Further details can be found e.g. in Lütkepohl und Kraetzig (2008, Section 2.7.4)

• If one uses the ADF or the Phillips-Perron test jointly with the KPSS test, it may happen that

both tests reject or do not reject. This happens if one of the two tests does not have enough power *or* if the alternative does not contain the true data generating process.

- A **structural break** in the data in general reduces the power of the ADF or the Phillips-Perron test (meaning it is less likely to reject the unit root hypothesis if it is not true). The reason is that the standard alternative hypothesis cannot capture the structural break well.

  Thus, if one suspects a structural break, one has to include it in the alternative hypothesis. If one does not know the break point exactly, one may estimate it. This can be done with the freely available menu-driven software JMulTi, see Lütkepohl und Kraetzig (2008, Section 2.7.3) for a description.

- There are a number of other tests, see e.g. other tests in `R` package `urca`.

- Besides the unit root discussed so far it may also happen that the data driven process exhibits **seasonal roots**. If a data generating process contains all seasonal roots, then one can obtain a stationary process by applying the operator $(1 - L^S)$ to the data.

  Depending on the number of seasons S one can decompose this filter differently. Example $S = 4$:

  $$(1 - L^4) = (1 - L)(1 + L)(1 + L^2).$$

  One notices that the (seasonal) unit roots occur at $1, -1, i$. With the **HEGY test** one can test which seasonal unit roots cannot be rejected for a given time series. This test is e.g. implemented

in JMulTi and described in Lütkepohl und Kraetzig (2008, Section 2.7.5).

4.2. Model Checking

4.2.1. Descriptive Analysis of the Residuals

- Outliers or structural breaks frequently show up in the plotted time series of the residuals.

  $\Longrightarrow$

  Plot the standardized residuals $\hat{u}_t^s$:

$$\hat{u}_t^s = \frac{\hat{u}_t - \bar{\hat{u}}_t}{\tilde{\sigma}_u} \quad \text{with } \bar{\hat{u}}_t = T^{-1}\sum_{t=1}^{T}\hat{u}_t \text{ and } \tilde{\sigma}_u^2 = T^{-1}\sum_{t=1}^{T}\left(\hat{u}_t - \bar{\hat{u}}_t\right)^2$$

  (Technical note: the OLS residuals do not sum to 0 if there is no constant in the model; therefore the mean of the residuals has to be subtracted.)

  If the residuals are standard normally distributed (e.g. in case the DGP is Gaussian white noise with variance 1), about 95% of the residuals should be within the interval $[-2, 2]$. If the residuals are approximately normally distributed (e.g. if the DGP has Gaussian white noise errors), then the rule can be applied approximately for identifying outliers and/or structural breaks.

- Analysis of the squared residuals in order to check the homoskedasticity assumption. If the errors are homosekdastic, the squared residuals should not exhibit very large peaks.

- Analysis of the autocorrelation structure in the residuals:
  Let
  $$\rho_{u,i} = \frac{Cov(u_t, u_{t-i})}{\sqrt{Var(u_t)Var(u_{t-i})}}$$
  denote the i-th autocorrelation of the error process $\{u_t\}$. An estimator of the autocorrelation coefficient $\rho_{u,i}$ is
  $$\begin{aligned}\hat{\rho}_{u,i} &= \frac{T^{-1}\sum_{t=i+1}^{T}\left(\hat{u}_t - \bar{\hat{u}}_t\right)\left(\hat{u}_{t-i} - \bar{\hat{u}}_t\right)}{\tilde{\sigma}^2_{\hat{u}}} \\ &= T^{-1}\sum_{t=i+1}^{T}\hat{u}^s_t\hat{u}^s_{t-i}.\end{aligned}$$
  If the error process is white noise, the estimated autocorrelations of the residuals should be in the interval $[-2/\sqrt{T}, 2/\sqrt{T}]$ with roughly 95% probability. In small samples one should correct for the sample size, see e.g. Lütkepohl (2005, Proposition 4.6).

  Observing 'too many' autocorrelations outside the interval $[-2/\sqrt{T}, 2/\sqrt{T}]$ is an indication that relevant lags have been missed in the specified model.

- For the spectral estimation of the residuals, you are referred e.g. to Lütkepohl und Kraetzig (2008, Section 2.2.2).

4.2.2. Diagnostic tests

- **Tests for autocorrelation in the residuals**
  In order to test whether the error process underlying an estimated model is a white noise process it is in principle necessary to check whether *all* its autocorrelations are 0. In empirical work one restricts oneself to test whether the first h autocorrelations are jointly 0. Since the errors are unobservable, the test must be based on the residuals $\hat{u}_t$.

  – **Portmanteau test for residual autocorrelation**
  The Portmanteau (=luggage) test checks whether the first h autocorrelations are jointly 0:

$$\begin{aligned} H_0: &\quad \rho_{u,1} = \cdots = \rho_{u,h} = 0 \\ &\quad \text{versus} \\ H_1: &\quad \rho_{u,i} \neq 0 \text{ for at least one } i = 1, \ldots, h. \end{aligned}$$

  Portmanteau test statistic (Box-Pierce statistic):

$$Q_h = T \sum_{j=1}^{h} \hat{\rho}_{u,j}^2 \tag{4.2}$$

  Under H_0 and for $T \to \infty$ it holds that Q_h follows asymptotically a $\chi^2(h-p-q)$-distribution.

  Note: Test can only be performed for $h > p + q$. Problem: The test is not very reliable in small samples. One therefore uses in practice a modified version described next.

– **Modified Portmanteau test statistic (Ljung-Box statistic)**

$$Q_h^{LB} = T(T+2)\sum_{j=1}^{h}\frac{\hat{\rho}_{u,j}^2}{T-j} \tag{4.3}$$

This test also has an asymptotic $\chi^2(h-p-q)$-distribution under H_0.

– **Lagrange multiplier (LM) test for autocorrelation in the residuals**:

* The **Lagrange multiplier test** is a general testing principle that is derived in the setting of maximum likelihood estimation that will be discussed later during this course. The test statistic of the Lagrange multiplier test varies with the specific model and null hypothesis under consideration.

* The basis for the **LM test for autocorrelation in the residuals of AR models** is an autoregressive model for the errors

$$u_t = \beta_1 u_{t-1} + \cdots + \beta_h u_{t-h} + error_t$$

that allows to test the null hypothesis

$$H_0 : \beta_1 = \cdots = \beta_h = 0 \quad \text{versus} \quad H_1 : \beta_1 \neq 0 \text{ or } \cdots \text{ or } \beta_h \neq 0.$$

This test is not feasible since the errors are not observable. To make the test feasible one uses the residuals $\hat{u}_t$ of the fitted model instead.

* The computation of the test statistic is quite easy:

$$LM_h = TR^2$$

where the coefficient of determination R^2 is obtained (in case of a fitted AR(p) model) from the following auxiliary regression

$$\hat{u}_t = \nu + \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \beta_1 \hat{u}_{t-1} + \cdots + \beta_h \hat{u}_{t-h} + e_t.$$

The Lagrange multiplier test statistic follows asymptotically a χ^2-distribution with h degrees of freedom, thus,

$$LM_h = TR^2 \xrightarrow{d} \chi^2(h).$$

* In small samples, the F version of the LM statistic frequently delivers a better approximation of the finite sample distribution

$$FLM_h = \frac{R^2}{1-R^2}\frac{T-p-h-1}{h} \approx F(h, T-p-h-1).$$

* The LM test is frequently called the **Breusch-Godfrey test** or the **Godfrey test**.

- **Test for nonnormal errors: Lomnicki-Jarcqe-Bera test**

  – If the assumption of normally and identically distributed errors is not violated, then the following holds for the third and fourth moment of the standardized errors $u_t^s = u_t/\sigma$:

$$E\left[(u_t^s)^3\right] = 0, \quad E\left[(u_t^s)^4\right] = 3.$$

The Lomnicki-Jarcqe-Bera test checks whether these two moments correspond to the values implied by a normal distribution:

$$H_0 : E\left[(u_t^s)^3\right] = 0 \quad \text{and} \quad E\left[(u_t^s)^4\right] = 3$$
$$H_1 : E\left[(u_t^s)^3\right] \neq 0 \quad \text{or} \quad E\left[(u_t^s)^4\right] \neq 3$$

– The test statistic is

$$LJB = \frac{T}{6}\left[T^{-1}\sum_{t=1}^{T}(\hat{u}_t^s)^3\right]^2 + \frac{T}{24}\left[T^{-1}\sum_{t=1}^{T}(\hat{u}_t^s)^4 - 3\right]^2 \xrightarrow{d} \chi^2(2).$$

– **Remark**: If H_0 is not rejected, then this does not imply that the errors are normally distributed but merely that their first four moments are compatible with a normal distribution.

- **Testing for (conditional) heteroskedasticity in the error process**
  - **Important definitions**:
    A stochastic process $\{y_t\}$, $t = \ldots, -2, -1, 0, 1, 2, \ldots$, is called
    - **homoskedastic** if
    $$Var(y_t) = \sigma_y^2 \quad \text{for all } t,$$
    - **(unconditionally) heteroskedastic** if
    $$Var(y_t) = \sigma_{y,t}^2 \neq Var(y_s) = \sigma_{y,s}^2 \quad \text{for some } t \neq s,$$
    - **conditionally heteroskedastic** if
    $$Var(y_t|y_{t-1}, y_{t-2}, \ldots,) \neq Var(y_t) \quad \text{for some } t.$$
    - These definitions apply also to the noise process $\{u_t\}$ of a stochastic process $\{y_t\}$.
  - To test for the presence of unconditional heteroskedasticitiy in the noise process one can use tests for structural breaks.

    To test for the presence of conditional heteroskedasticity in the noise process one can use the **ARCH-LM test** that is described in section 5.5.

4.3. Estimating dynamic regression models

- **The dynamic regression model**

$$y_t = X_t'\beta + u_t \tag{4.4}$$

with

$$X_t' = \begin{pmatrix} x_{t1} & x_{t2} & \cdots & x_{tk} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{pmatrix}$$

and X_t' belongs to the information set $\mathcal{I}_t$.

The following assumptions are made about the stochastic processes of the error term and the regressors:

1. **Assumption B** (compare to assumptions (C1) to (C4) in Methods of Econometrics):

   a) The conditional expectation of the error term u_t given the information set up to time t is zero

$$E[u_t|\mathcal{I}_t] = 0.$$

   b) The conditional variance of the error term u_t given the information set up to time t is

constant (= the error term u_t is homoskedastic)

$$E[u_t^2|\mathcal{I}_t] = \sigma^2.$$

c) The regressor matrix behaves nicely in large samples, that is the empirical moment matrix converges asymptotically to a fixed matrix

$$\text{plim}_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} X_t X_t' = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E\left[X_t X_t'\right] = M_{XX}$$

with M_{XX} being positive definite.

d) Strict stationarity of all variables or

$$E\left[\left|\lambda' X_t u_t\right|^{2+\delta}\right] \leq B < \infty, \quad \delta > 0, \quad \text{for all } t, \text{ for all } \lambda \text{ with } \lambda'\lambda = 1.$$

Then it can be shown that the **OLS estimator is asymptotically normally distributed**:

$$\sqrt{T}\left(\hat{\beta} - \beta\right) = \left(\frac{1}{T}\sum_{t=1}^{T} X_t X_t'\right)^{-1} \frac{1}{\sqrt{T}}\sum_{t=1}^{T} X_t u_t \xrightarrow{d} N\left(0, \sigma^2 M_{XX}^{-1}\right)$$

- **Notes to the assumptions**:
  - Note that **Assumption A** for estimating AR(p) models, see subsection 2.5.1, guarantees that **Assumption B** holds. Thus, the dynamic regression model is a generalization of AR(p)

models by allowing also for further explanatory variables that are not lagged endogenous variables.

This implies that parameters associated with lagged endogenous variables have to fulfil the stationarity conditions known from the AR models! This has to be checked in the model analysis!

– The main issue in **B**(c) is that it is excluded that

  * asymptotically the moment matrix M_{XX} becomes singular (Example: $x_{t1} = 1$, $x_{t2} = 1/t$)

  * the empirical second moments do not converge such that M_{XX} does not exist (Example: all regressors contain a random walk).

  If all regressors are stationary, Assumption **B**(c) is fulfilled. It basically restricts the memory of the regressors (remember that a random walk has perfect/infinite memory and therefore violates the requirement of restricted memory).

– As conditioning/explanatory variables one may include

  * deterministic variables (but watch out: deterministic trends can violate **B**(c), see below.)

  * lagged variables

  * current dated (=contemporaneous variables) that are weakly exogenous for β and σ^2. This

requires for example that they are uncorrelated with u_t.

- The homoskedasticity assumption in **B**(b) can be relaxed to heteroskedastic errors. This, however, requires to use a heteroskedasticity-robust estimator for the covariance matrix of the parameter estimates (or the use of FGLS methods, which may be difficult).
- Assumption **B**(d) generalizes Assumption **A**. This assumption is needed such that the variance of $1/\sqrt{T}\sum_{t=1}^{T} x_{ti}u_t$ asymptotically exists. Among other things, this requires that the variances of u_t or x_{ti} are not allowed to increase without bound. It also avoids that the error or the regressor variable have too fat tails.

- A **consistent estimator for the covariance matrix** is

$$\hat{\sigma^2}\left(X'X\right)^{-1}, \quad \hat{\sigma}^2 = \frac{1}{T-k}\sum_{t=1}^{T}\hat{u}_t^2.$$

- Because the OLS parameters are asymptotically normally distributed, it can be shown that the standard t and F tests can be applied, however, only asymptotically (for more details on this, see Methoden der Ökonometrie).
- **Some further (important) notes**:
  - If dummy variables are used, there must be a nonzero fraction of 1's or 0's in any sample. Thus, there is no asymptotic distribution for a dummy that is 1 for a single (or a finite number

of periods) and 0 otherwise in case the sample size increases. Tests for such dummies must be based on other grounds. Including a dummy for a single period means asymptotically that this observation is dropped from the sample.

– A variable is called **trend-stationary** if regressing it on t delivers stationary residuals. If one includes variables that are trend-stationary, then the parameter estimates corresponding to these variables do not have the $\sqrt{T}$ rate of convergence and thus, the standard errors are not correct. Solution: Include a deterministic trend $x_{ti} = t$ as an additional regressor. However: the trend variable has now a different (faster) convergence rate, see also unit root tests.

– **Random walks** and **cointegration**. Will not be discussed here.

4.4. R Code

Testing time series y for a unit root

```
# y denotes data series
install.packages("urca")    # only necessary when used for the first time
library(urca)

# ADF test with up to 5 lags (chosen by AIC) against the alternative of
# a stationary process around a (non-zero) constant (pair of hypotheses (II))
y.adf <- ur.df(y, type = "drift", lags = 5, selectlags = "AIC")
summary(y.adf)
```

```
# KPSS test with number of lags depending on T against the alternative of
# a unit root process without deterministic trend
y.kpss <- ur.kpss(y, type  = "mu", lags = "long")
summary(y.kpss)

# See ur.pp() for the Phillips-Perron Test
?ur.pp

# Other unit root tests in package urca that have not been discussed so far
?ur.ers
?ur.sp
?ur.za
```

Diagnostic tests: serial correlation and non-normality

```
# Analysing residuals from AR(4) regression
ar.est <- ar(y, demean = FALSE, aic = FALSE, order.max = 4)
ar.resid <- ar.est$resid       # contains NA values for first 4 observations

# Autocorrelation structure of residuals (ignoring NA values)
acf(ar.resid, na.action = na.pass)

# Portmanteau test for serial correlation in residuals
# up to order h
h <- 7
Box.test(ar.resid, lag = h, type = "Box-Pierce", fitdf = 4)
# modified version of Portmanteau test
Box.test(ar.resid, lag = h, type = "Ljung-Box", fitdf = 4)

# Test for nonnormal errors
ar.resid        <- na.omit(ar.resid)                    # getting rid of NA values
```

```
T <- length(ar.resid)
ar.resid.mean   <- mean(ar.resid)
ar.resid.demean <- ar.resid - ar.resid.mean
ar.resid.var    <- mean( ar.resid.demean^2 )
ar.resid.stndrd <- ar.resid.demean / sqrt(ar.resid.var) # standardized residuals
ar.resid.skew   <- mean( ar.resid.stndrd^3 )            # skewness
ar.resid.kurt   <- mean( ar.resid.stndrd^4 ) - 3        # (excess) kurtosis

LJB <- T/6 * (ar.resid.skew^2) + T/24 * (ar.resid.kurt^2) # LJB test statistic
1- pchisq(LJB, df = 2)                                    # p-value
```

Dynamic regression

```
# Dynamic regressions are easily performed using the interface provided
# by the dynlm package
library(dynlm)
?dynlm
```

5. Modeling volatility dynamics

A brief introduction:

- Consider the very **simple conditionally heteroskedastic stochastic process** $\{y_t\}$

$$y_t = u_t$$

where the noise process is conditionally heteroskedastic

$$Var(u_t|u_{t-1}, u_{t-2}, \ldots) \neq Var(u_t). \tag{5.1}$$

Note that in the special case considered at the moment

$$Var(u_t|y_{t-1}, y_{t-2}, \ldots) = Var(u_t|u_{t-1}, u_{t-2}, \ldots)$$

since the u_{t-j}'s are known if and only if the y_{t-j}'s are known for all $j = 1, 2, \ldots$.

Note that $E[u_t|y_{t-1}, y_{t-2}, \ldots] = 0$ as in the independent white noise case.

- A more general conditionally heteroskedastic process is the **conditionally heteroskedastic autoregressive process of order** p

$$(1 - \alpha_1 L - \alpha_2 L^2 - \cdots \alpha_p L^p) y_t = u_t$$

where the noise process exhibits conditional heteroskedasticity as given by (5.1). If the autoregressive parameters are known, then

$$Var(u_t | y_{t-1}, y_{t-2}, \ldots) = Var(u_t | u_{t-1}, u_{t-2}, \ldots).$$

If the autoregressive parameters are not known, then the two conditioning sets are different since the errors u_t are **unobservable** and have to be estimated.

- In the following we present models for conditional heteroskedastic noise and for simplicity we assume that the errors are observable. Later on this assumption will be relaxed.

- In order to save notation we frequently write in the following

$$\sigma_t^2 = Var(u_t | u_{t-1}, u_{t-2}, \ldots).$$

- **In all what follows we assume** that

$$u_t = \sigma_t \xi_t, \quad \xi_t \sim IID(0, 1)$$

and ξ_t and u_s are stochastically independent for $t > s$. The distribution for ξ_t may vary, depending on the specific application.

5.1. Standard conditional volatility models

- **Autoregressive Conditionally Heteroskedastic Models (ARCH(m) Models)**
  - Simplest case: **ARCH(1) process**

$$\sigma_t^2 = Var(u_t|u_{t-1}, u_{t-2}, \ldots) = Var(u_t|u_{t-1}) = \gamma_0 + \gamma_1 u_{t-1}^2.$$

The conditional variance only depends on one lagged error.

Properties:

* **Sufficient conditions for positive variance**: $\gamma_0 > 0$ and $\gamma_1 > 0$.
* How can one generate a realization of an ARCH(1) process with conditionally normally distributed innovations?

$$u_t = \sigma_t \xi_t, \quad \xi_t \sim i.i.d.N(0,1)$$

with σ_t^2 given above. You may use the R program in section A.8 for generating ARCH(1) realizations.

* **Unconditional mean** $E[u_t] = 0$ if the unconditional variance exists since

$$E[u_t] = \underbrace{E\left[\sqrt{\gamma_0 + \gamma_1 u_{t-1}^2}\right]}_{D} \underbrace{E[\xi_t]}_{=0 \text{ by assumption}} = D\,0.$$

Thus, D has to be finite which can be shown to be the case if the unconditional variance exists.

* **Unconditional variance**

$$\begin{aligned} Var(u_t) &= E[u_t^2] \overset{\text{law of iterated expectations}}{=} E[E[u_t^2|u_{t-1}]] \\ &= E[\gamma_0 + \gamma_1 u_{t-1}^2] = \gamma_0 + \gamma_1 E[u_{t-1}^2] \\ Var(u_t) &= \gamma_0 + \gamma_1 Var(u_{t-1}) \end{aligned}$$

Under the assumption of stationarity $Var(u_t) = Var(u_{t-1}) = \sigma^2$ we get

$$Var(u_t) = \frac{\gamma_0}{1-\gamma_1}$$

Note the unconditional variance only exists **if** $\gamma_1 < 1$.

* **Conditional fourth moment**

$$E[u_t^4|u_{t-1}] = (\gamma_0 + \gamma_1 u_{t-1}^2)^2 E[\xi_t^4]$$

* **Unconditional fourth moment** (only exists if fourth moment stationary). Then one has $E[u_t^4] = E[u_{t-1}^4] = m_4$ and one can derive under the assumption of normally distributed ξ_t

$$m_4 = \frac{3\gamma_0^2(1+\gamma_1)}{(1-\gamma_1)(1-3\gamma_1^2)}.$$

Note that the unconditional fourth moment only exists **if** the unconditional variance exists and $\gamma_1^2 < 1/3$.

* **Unconditional kurtosis** in case of normally distributed ξ_t:

$$\kappa = \frac{E[u_t^4]}{\left(Var(u_t)\right)^2} = \frac{m_4}{\left(\sigma^2\right)^2} = 3\frac{1-\gamma_1^2}{1-3\gamma_1^2} > 3.$$

Thus, an ARCH(1) process is leptokurtic: the tail distribution of u_t is heavier than that of a normal distribution. Put differently, compared with a normal distribution, 'outliers' are more likely.

– **ARCH(m) process**:

$$\sigma_t^2 = Var(u_t|u_{t-1}, u_{t-2}, \ldots, u_{t-m}) = \gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \gamma_m u_{t-m}^2$$

The conditional variance depends on m lagged errors. The ARCH(m) model was proposed by the nobel prize winner Robert Engle in Engle (1982)

Properties:

* **Sufficient conditions for positive variance**: $\gamma_0 > 0$ and $\gamma_j > 0$, $j = 1, \ldots, m$.

* **Unconditional variance**:
The unconditional variance

$$Var(u_t) = \sigma^2 = \frac{\gamma_0}{1 - \gamma_1 - \cdots - \gamma_m}$$

exists if $1 - \gamma_1 - \cdots - \gamma_m > 0$.

* The ARCH(m) process corresponds to a conditionally heteroskedastic AR(m) process in the squared errors

$$\begin{aligned} u_t^2 &= \sigma_t^2 \xi_t^2 \\ u_t^2 &= \sigma_t^2 + \underbrace{\sigma_t^2(\xi_t^2 - 1)}_{\eta_t} \\ u_t^2 &= \gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \gamma_m u_{t-m}^2 + \eta_t \end{aligned}$$

where $E[\eta_t] = E[E[\eta_t | u_{t-1}, \ldots, u_{t-m}]] = E[\sigma_t^2 E[(\xi_t^2 - 1)]] = E[\sigma_t^2 \, 0] = 0$, since $E[\xi_t^2] = 1$ by assumption.

* Other moments and conditions can also be derived but are messier to present.

Drawbacks of ARCH(m) models:

* Positive and negative shocks have the same impact on volatility. In practice one observes frequently an asymmetric behavior.

* The parameter restrictions are quite severe if one requires (unconditional) fourth moments to be stationary.

* The model may not be very parsimonious if large lags have an important impact (m large).

- **Generalized Autoregressive Conditionally Heteroskedastic Processes (GARCH(m, n) process)**
  - To solve the problem of having too many parameters in an ARCH(m) model, Taylor (2008) and Bollerslev (1986) independently suggested the GARCH(m, n) model that uses lagged conditional variances as explanatory variables in addition to the lagged squared shocks

$$\sigma_t^2 = Var(u_t|u_{t-1}, u_{t-2}, \ldots) = \gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \gamma_m u_{t-m}^2 + \beta_1 \sigma_{t-1}^2 + \cdots + \beta_n \sigma_{t-n}^2.$$

  - A special case is the so-called **integrated generalized autoregressive conditionally heteroskedastic (IGARCH(1,1)) process** for which $\gamma_1 + \beta_1 = 1$.
  - **Properties**:
    - **Unconditional variance of u_t**:
      The unconditional variance is

$$Var(u_t) = \frac{\gamma_0}{1 - \gamma_1 - \ldots - \gamma_m - \beta_1 - \ldots - \beta_n}$$

      and exists if $\gamma_1 + \ldots + \gamma_m + \beta_1 + \ldots + \beta_n < 1$.
    - **Conditional fourth moment**:

$$E[u_t^4]|u_{t-1}, u_{t-2}, \ldots] = \left(\gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \gamma_m u_{t-m}^2 + \beta_1 \sigma_{t-1}^2 + \cdots + \beta_n \sigma_{t-n}^2\right)^2 E\left[\xi_t^4\right]$$

* **Unconditional fourth moment** for a GARCH(1,1) process:
  If $\xi_t \sim IN(0,1)$, then $E[\xi_t^4] = 3$ and then one can show that

$$E[u_t^4] = \frac{3\gamma_0^2(1+\gamma_1+\beta_1)}{(1-\gamma_1-\beta_1)(1-\beta_1^2-2\gamma_1\beta_1-3\gamma_1^2)}.$$

  The fourth moment exists if $(\beta_1^2+2\gamma_1\beta_1+3\gamma_1)^2 < 1$. Note that this condition is stronger than the one for a stationary variance.

* **Kurtosis** for a GARCH(1,1) process:
  If $\xi_t \sim IN(0,1)$, then $E[\xi_t^4] = 3$ and then one can show that

$$\kappa = \frac{E[u_t^4]}{\left(E[u_t^2]\right)^2} = \frac{3(1-\gamma_1-\beta_1)(1+\gamma_1+\beta_1)}{1-\beta_1^2-2\gamma_1\beta_1-3\gamma_1} > 3.$$

  Thus, a conditionally normally distributed GARCH(1,1) process is leptokurtic, meaning that the tails of the unconditional probability density of u_t are thicker than those of a normal density that has a kurtosis of 3: large shocks are more likely to occur than in case of a normally distribution. Note that the type of the distribution is not normal even if ξ_t is IN(0,1).

* Although the unconditional distribution does not correspond to a known standard distribution, Nelson (1990) has shown that a GARCH(1,1) process is **strictly stationary and ergodic if**

$$E\left[\log\left(\beta_1+\gamma_1\xi_t^2\right)\right] < 0.$$

This condition means that one should not observe too often values of $\beta_1 + \gamma_1 \xi_t^2$ that are way beyond 1 and thus getting too much impact.

Note that this condition is weaker than the stationarity condition for the unconditional variance since an IGARCH(1,1) process is strictly stationary but not covariance stationary.

- Using the `R` program in section A.8 one can simulate and estimate an AR(1)-GARCH(1,1) process.
.
- There exist a number of extensions of ARCH and GARCH processes. They will be discussed after the presentation of estimation procedures.

5.2. Maximum Likelihood Estimation

- **Introductory Example**: Consider having thrown a coin 10 times with the results of 9 heads and 1 tail. Do you think that this is a 'fair' coin (a coin for which the probability for observing head is 0.5)?
  - Note that the probability to obtain k times head out of n throws is given by the binomial distribution

$$P('k \text{ heads of out } n \text{ trials}'|p) = \frac{n!}{(n-k)!k!} p^k (1-p)^{(n-k)} \tag{5.2}$$

where p denotes the probability for getting a head in one throw.

– Thus, the probability to observe the outcome stated above is for various p:

$$\begin{aligned} p = 1/2 \quad &\longrightarrow P(\text{'9 heads of out 10 trials'}|p = 1/2) \approx 0.01 \\ p = 3/4 \quad &\longrightarrow P(\text{'9 heads of out 10 trials'}|p = 3/4) \approx 0.19 \\ p = 9/10 \quad &\longrightarrow P(\text{'9 heads of out 10 trials'}|p = 9/10) \approx 0.39 \end{aligned}$$

Hence, the probability of observing the given result is only about 1% if the coin is 'fair'. It seems more likely that the probability of getting head in one throw is far larger than 0.5.

– One may now change the use of the probability function (5.2) and use it to assign a **given** event a probability based on a chosen value for p. In this interpretation one calls (5.2) a **likelihood function** in order to distinguish it from the use of a probability function. One writes for the present case

$$L(p|\text{'}k \text{ heads of out } n \text{ trials'}) = \frac{n!}{(n-k)!k!} p^k (1-p)^{(n-k)} \tag{5.3}$$

– Since one can compute the likelihood $L(p|\text{'}k$ heads of out n trials'$)$ for a given outcome, e.g. '9 heads out of 10 throws', for any p, one can maximize the likelihood $L(p|\text{'}k$ heads of out n trials'$)$ with respect to p. One then obtains an estimate $\hat{p}$ for p that *maximizes the likelihood for which the observed outcome may be observed*. Therefore, this estimator is called **maximum likelihood (ML) estimator**.

- In the current case, one can easily obtain the ML estimator $\hat{p}$ by setting the first derivative of (5.3) to zero and solving for p.
- Very often it is more convenient to maximize the likelihood function after taking logarithms. That does not change the maximum likelihood estimator since taking logarithms is a strictly monotone transformation. However, it makes the analytical or numerical optimization much easier. The **log-likelihood function** in the present case is

$$\begin{aligned} l(p|\text{'}k \text{ heads of out } n \text{ trials'}) &= \ln L(p|\text{'}k \text{ heads of out } n \text{ trials'}) \\ &= \ln\left(\frac{n!}{(n-k)!k!}\right) + k\ln p + (n-k)\log(1-p). \end{aligned}$$

  The first derivative is

$$\frac{\partial \ln L(p|\cdot)}{\partial p} = \frac{k}{p} - \frac{n-k}{1-p} \stackrel{!}{=} 0.$$

  The ML estimate for p is therefore $\hat{p} = k/n = 9/10$. (For completeness one has to check whether the extremum is a maximum. This requires the second derivative to be negative around $\hat{p}$.)

- **Maximum likelihood estimation in the case of continuous random variables**:
  - For a continuous random variable Y it holds that the probability 'Y takes the value y' is zero, that is $P(Y = y) = 0$. This is because there are an infinite number of possible values and a sum of infinitely many positive probabilities cannot sum to 1.

Instead one considers an interval for Y, e.g. $[a, b]$ or frequently $[-\infty, y]$. For the latter interval one obtains the probability distribution function

$$F(Y < y) = P(Y < y)$$

which must be nonzero for some intervals. Thus one can also analyze the change of the probability if one increases the interval by an marginal amount $\delta > 0$. This delivers the absolute change in probability

$$P(Y < y + \delta) - P(Y < y)$$

and the relative change in probability

$$\frac{P(Y < y + \delta) - P(Y < y)}{\delta}.$$

Letting the change in the interval length δ go to 0, one obtains the **probability density function**

$$f(y) = \lim_{\delta \to 0} \frac{P(Y < y + \delta) - P(Y < y)}{\delta}$$

that must be nonzero at some y since otherwise the probability would not change if the interval is increased.

Since

$$P(y \leq Y < y + \delta) = P(Y < y + \delta) - P(Y < y)$$

one obtains, loosely speaking,

$$P(y \leq Y < y + \delta) \approx f(y)\delta.$$

One can approximate the probability that a realization of Y is observed to be in the interval $[y, y+\delta)$ by the density multiplied with the interval length. Of course, this approximation is the better, the smaller δ. **The density is approximately proportional to the probability of Y being observed in a very short interval around y.**

– Equivalently, if the probability and density function depend on a parameter θ one has

$$P(y \leq Y < y + \delta|\theta) \approx f(y|\theta)\delta.$$

Maximizing the likelihood for observing Y (to be in a extremely tiny interval around y) thus can be done by maximizing the density with respect to θ. For continuous random variables the density therefore has the interpretation of the **likelihood function**

$$L(\theta|y) = f(y|\theta).$$

The ML estimator $\hat{\theta}$ for θ is then given by

$$\max_{\theta} L(\theta|y) \quad (= \max_{\theta} f(y|\theta)).$$

– Thus, for deriving the ML estimator for a specific problem one has to choose an appropriate parameterized density function.

– For a sample of T observations $y_1, y_2, \ldots, y_T$ the likelihood function is the joint density with respect to θ

$$L(\theta|y_1, y_2, \ldots, y_T) = f(y_1, y_2, \ldots, y_T|\theta)$$

– The joint density for T i.i.d. observations is the product of T marginal densities. Thus, the likelihood is given by

$$L(\theta|y_1, y_2, \ldots, y_T) = f(y_1, y_2, \ldots, y_T|\theta) = f(y_1|\theta) \cdots f(y_T|\theta)$$

and the **log-likelihood is the sum of the log-likelihood for each observation**

$$l(\theta|y_1, y_2, \ldots, y_T) = \ln f(y_1, y_2, \ldots, y_T|\theta) = \sum_{t=1}^{T} \ln f(y_t|\theta).$$

This property is very convenient for maximizing the (log)-likelihood!

– In case the observations are not i.i.d. one can use the following decomposition

$$\begin{aligned} L(\theta|y_1, y_2, \ldots, y_T) &= f(y_1, y_2, \ldots, y_T|\theta) \\ &= f(y_T|y_{T-1}, \ldots, y_1; \theta) f(y_{T-1}, y_{T-2}, \ldots, y_1|\theta) \\ &= f(y_T|y_{T-1}, \ldots, y_1; \theta) f(y_{T-1}|y_{T-2}, \ldots, y_1|\theta) f(y_{T-2}, \ldots, y_1|\theta) \\ &= f(y_T|y_{T-1}, \ldots, y_1; \theta) f(y_{T-1}|y_{T-2}, \ldots, y_1|\theta) \cdots f(y_2|y_1; \theta) f(y_1|\theta) \end{aligned}$$

Taking logarithms one obtains the sum

$$\begin{aligned} l(\theta|y_1, y_2, \ldots, y_T) &= \ln f(y_1, y_2, \ldots, y_T|\theta) \\ &= \sum_{t=1}^{T} \ln f(y_t|y_{t-1}, \ldots, y_1; \theta) + \ln f(y_1|\theta). \end{aligned}$$

– If the term $\ln f(y_1|\theta)$ is ignored one obtains the **conditional likelihood function** that is conditional on y_1. Its maximization delivers the **conditional maximum likelihood estimator**.

5.3. Estimation of GARCH(m, n) models

- GARCH models are usually estimated with the (conditional) maximum likelihood estimator. This requires to assume a density for the errors ξ_t.
- **Assume independent normally distributed errors**

$$\xi_t \sim N(0, 1)$$

with density

$$f(\xi_t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\xi_t^2\right).$$

Since by definition

$$\xi_t = \frac{u_t}{\sigma_t}$$

one has

$$u_t | u_{t-1}, \ldots \sim N(0, \sigma_t^2)$$

and thus by the properties of the normal density

$$f(u_t | u_{t-1}, \ldots, u_1; \gamma_0, \ldots, \gamma_m, \beta_1, \ldots, \beta_n) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1}{2}\frac{u_t^2}{\sigma_t^2}\right)$$

The log-likelihood function of an GARCH(m, n) model is then given by

$$\begin{aligned} l(\gamma_0, \ldots, \gamma_m, \beta_1, \ldots, \beta_n | u_T, u_{T-1}, \ldots, u_1) &= \sum_{t=1}^{T} \ln f(u_t | u_{t-1}, \ldots; \gamma_0, \ldots, \gamma_m, \beta_1, \ldots, \beta_n) \\ &= \sum_{t=1}^{T} \ln \left\{ \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1}{2}\frac{u_t^2}{\sigma_t^2}\right) \right\} \\ &= -\frac{T}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T} \ln \sigma_t^2 - \frac{1}{2}\sum_{t=1}^{T} \frac{u_t^2}{\sigma_t^2} \end{aligned}$$

which after inserting σ_t^2 becomes

$$-\frac{T}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T} \ln\left(\gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \beta_n \sigma_{t-n}^2\right) - \frac{1}{2}\sum_{t=1}^{T} \frac{u_t^2}{\gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \beta_n \sigma_{t-n}^2}$$

This log-likelihood function has to be maximized for obtaining the ML estimates $\hat{\gamma}_0, \ldots, \hat{\beta}_n$ based on the assumption of conditionally normally distributed errors.

- Frequently residuals of GARCH models of financial time series are leptokurtic (they have fatter tails than normally distributed residuals). In this case the **assumption of conditionally normally distributed errors is wrong** and such a model is **misspecified**.

  Solutions:

  – Use other error distribution, e.g. t-distribution.

  – Use quasi-maximum likelihood estimator.

- **Assume conditionally t-distributed errors**.

  The density of a t-distributed error variable u with ν degrees of freedom and variance σ^2 is

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

  where $\Gamma(h)$ denotes the gamma function $\Gamma(h) = \int_0^\infty x^{h-1}\exp(-x)dx$, $h > 0$.

  The moments are

  – $E[u] = 0$ if $\nu \geq 2$

  – $Var(u) = \frac{\nu}{\nu-2}$ if $\nu \geq 3$

  - all odd moments are zero
  - r-th moment, see Johnson, Kotz, und Balakrishnan (1995),

$$\mu_r(u) = \nu^{\frac{1}{2}r} \frac{1 \cdot 3 \cdots (r-1)}{(\nu-r)(\nu-r+2)\cdots(\nu-2)}$$

  - $E[u^4] = 3\frac{\nu}{\nu-4} Var(u)$
  - kurtosis: $\kappa = 3\frac{\nu-2}{\nu-4}$

- **Assume conditional generalized error distribution (GED)**

$$f(u;\nu) = \frac{\nu\Gamma\left(\frac{3}{\nu}\right)^{1/2}}{2\Gamma\left(\frac{1}{\nu}\right)^{3/2}} \exp\left(-|u|^{\nu}\left(\frac{\Gamma\left(\frac{3}{\nu}\right)}{\Gamma\left(\frac{1}{\nu}\right)}\right)^{\nu/2}\right)$$

  where $\nu > 0$. For $\nu = 2$ one obtains a normal distribution $N(0,\sigma^2)$ and for $\nu < 2$ one has a leptokurtic distribution.

- **Transformation of random variables**:

  How can one obtain the density for a random variable X if one knows the density $f_Z(z)$ of $Z = h(X)$ and the function $h(\cdot)$? Then the density of X is obtained by

$$f_X(x) = f_Z(h(x))|h'(x)|,$$

  see e.g. Davidson und MacKinnon (2004, p. 438-439) for a derivation.

- Setting $U = \frac{Z}{s}$ with $u = h(z) = \frac{z}{s}$ and $h'(z) = 1/s$ one obtains e.g. in case of a t-distributed random variable the density

$$f(z;\nu,s) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{s\sqrt{v\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{z^2}{s^2\nu}\right)^{-\frac{\nu+1}{2}}.$$

- **Numerical optimization**

  There exists a number of algorithms for the optimization of nonlinear functions. For maximizing the log-likelihood function a particularly suited algorithm was developed by Berndt et al. (1974) (BHHH algorithm).

  In the i-iteration you update the estimator $\hat{\theta}_i$ by

$$\hat{\theta}_i = \hat{\theta}_{i-1} + \phi\left(\sum_{i=1}^{T}\frac{\partial l_t}{\partial\theta}\frac{\partial l_t}{\partial\theta'}\bigg|_{\theta=\hat{\theta}_{i-1}}\right)^{-1}\sum_{t=1}^{T}\frac{\partial l_t}{\partial\theta}\bigg|_{\theta=\hat{\theta}_{i-1}}$$

  where ϕ denotes the step length that also can be automatically chosen.

  For running these iterations **you have to choose**

  – starting values θ_0,

  – a maximum number of iterations i_{max},

  – a stopping rule, e.g. $|\hat{\theta}_i - \hat{\theta}_{i-1}| < \epsilon$ where the precision ϵ is a priori chosen,

– a numerical method or the analytical derivatives to compute the various derivatives,

– possibly an algorithm to determine the step length.

Attention: the choice of the starting values matters whatever numerical algorithm one chooses. Badly chosen starting values may result in a **local** instead of a **global** maximum! Thus, it is often useful to try several starting values, e.g. even by drawing random numbers / vectors. If possible, one may get the starting values from an auxiliary model (e.g. for estimating an ARCH model from the parameter estimates of the ARCH-LM test, see below.)

- Under the regularity conditions, see below, the **ML estimator is asymptotically normally distributed**

$$\sqrt{T}\left(\hat{\theta}-\theta\right) \xrightarrow{d} N\left(0, S^{-1}\right)$$

with

$$\begin{aligned} S &= \lim_T \frac{1}{T}\sum_{t=1}^{T} E\left[\frac{\partial l_t}{\partial \theta}\frac{\partial l_t}{\partial \theta'}\right] \\ &= \lim_T \frac{1}{T} E\left[\frac{\partial l}{\partial \theta}\frac{\partial l}{\partial \theta'}\right] \\ &= \lim_T \frac{1}{T} Var\left(\frac{\partial l}{\partial \theta}\right) \end{aligned}$$

denoting the **asymptotic Fisher information matrix**. The Fisher information matrix or the covariance matrix of the gradient of the log-likelihood function is given by $E\left[\frac{\partial l_t}{\partial \theta}\frac{\partial l_t}{\partial \theta'}\right]$.

Note that the expectation of the score (=first derivative of the log-likelihood function) computed at the true parameter vector is zero,

$$E\left[\left.\frac{\partial l}{\partial \theta}\right|_{\theta=\theta_{true\ value}}\right] = 0$$

if the model is correctly specified! For details see e.g. Methoden der Ökonometrie.

- Let $\mathcal{I}_t$ denote the information set that is available for computing the log-likelihood function at time t. In case of an AR(p) model, the information set is given by all past observations of the endogenous variable.

  The **regularity conditions** are

  - $E[u_t|\mathcal{I}_t] = 0$
    The conditional mean of the error is zero (standard assumption)
  - $E[u_t^2|\mathcal{I}_t] = \sigma_t^2$
    Conditional heteroskedasticity is correctly modeled.
  - plim$\frac{1}{T}\sum_{t=1}^{T} Y_{t-1}Y_{t-1}' =$ lim$\frac{1}{T}\sum_{t=1}^{T}\left[Y_{t-1}Y_{t-1}'\right] = M < \infty$ and M is invertible.
    This means that even for very large samples the regressors are well behaved, e.g. not becoming perfectly linearly dependent.
  - $E\left[|u_t|^{4+\delta}\right] < \infty$

The fourth moments of the errors exist.

- In case no error distribution is found to be appropriate, one may use **quasi-maximum likelihood estimation (QML estimation)** : one uses the maximum likelihood estimator based on a normal error density although the true errors may follow a different distribution.

  Properties:

  – The QML estimator is **consistent**.

  – The QML estimator requires the computation of a more complicated covariance matrix of the parameter estimates. Denote the limit of the negative expectation of the Hessian matrix (matrix of all second partial derivatives) divided by T by

  $$D = -\lim_T \frac{1}{T} E\left[\frac{\partial^2 l}{\partial\theta\partial\theta'}\right].$$

  Then the **covariance matrix of the QML estimator** is $D^{-1}SD^{-1}$. Thus, the QML estimator is not efficient since in general the covariance matrix is 'larger' than the covariance matrix based on the correct error density.

  – The QML estimator is **asymptotically normally distributed**

  $$\sqrt{T}\left(\hat{\theta} - \theta\right) \xrightarrow{d} N\left(0, D^{-1}SD^{-1}\right).$$

– If the **information matrix equality**

$$D^{-1} = S$$

holds asymptotically, then the asymptotic distribution of the QML estimator is identical with that of the ML estimator. Note, however, that the matrices S and D have to be estimated in practice. Thus, in general $\hat{D}^{-1}\hat{S}\hat{D}^{-1} \neq \hat{S}^{-1}$ and one should use the ML estimator if the errors are indeed normal.

For details see e.g. Methoden der Ökonometrie.

Warnings:

– The QML principle only works with the normal distribution. Using another 'wrong' distribution does not lead to a consistently and asymptotically normally distributed estimator.

– If one has estimated the covariance matrix by QML, one cannot produce forecasting intervals since the error distribution is still unknown.

- **Tests based on ML estimators in case of nonnormal errors are misleading** since the standard errors of the parameter estimates are incorrectly estimated!

- **Estimation of conditional mean and conditional variance functions**:

– The information matrix is **block diagonal** if the covariances between all parameter estimators for the conditional mean function and all parameter estimators for the conditional variance

function are zero.

– **Sequential estimation** is possible if the information matrix is block diagonal. One then can consistently estimate in the first step the conditional mean function (e.g. an AR(p) model) and in the second step based on the residuals from the first step the parameters of the conditional variance function (e.g. GARCH(1,1) parameters). Nevertheless, both the conditional mean and the conditional variance have to be specified correctly for obtaining correct standard errors for the parameters of the conditional mean estimates. Otherwise, the information matrix equality is violated as well.

  Models with a block diagonal information matrix:
  ARCH, GARCH, TGARCH with symmetric errors (see next section)

– **Models without a block diagonal information matrix**:
  EGARCH, GARCH-in-mean and TGARCH models with skewed (=asymmetric) errors (see next section).

5.4. Asymmetry and leverage effects

In many cases it has been observed that negative shocks have a larger impact on the volatility than a positive shock of the same magnitude. Models that allow for such asymmetric effects are e.g.:

- **Threshold GARCH (TGARCH model)**

$$\sigma_t^2 = \gamma_0 + \gamma_1 u_{t-1}^2 + \gamma_1^- I(u_{t-1} < 0) u_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

where $I(A)$ is the indicator function, taking the value 1 if the argument A is true and the value 0 otherwise.

Properties:

  - if the parameter γ_1^- has a positive sign, then negative shocks increase volatility more than positive shocks. One also calls this phenomenon **leverage effect**.

- **Exponential GARCH (EGARCH model)**

$$\log \sigma_t^2 = \tilde{\gamma}_0 + \tilde{\gamma}_1 \left(|\xi_{t-1}| - E\left[|\xi_{t-1}| \right] \right) + \tilde{\gamma}_1^- \xi_{t-1} + \tilde{\beta}_1 \log \sigma_{t-1}^2$$

Properties:

  - If $\xi_t \sim N(0,1)$, then $E\left[|\xi_t| \right] = \sqrt{2/\pi}$
  - $\tilde{\gamma}_1 > 0$ causes volatility clustering
    $\tilde{\gamma}_1^- < 0$ causes a leverage effect
  - By construction one always has a positive variance since $\log \sigma_t^2$ implies $\sigma_t^2 > 0$.
  - In practice it was found that outliers get too much impact on the estimation.

- **ARCH-in-mean model**

  It may happen that the conditional risk also influences the return itself, e.g. if there is a nonlinear relationship between risk and return. A potentially useful model is an AR model that also has an ARCH effect in the conditional mean function.

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \rho\sigma_{t-1} + u_t, \quad u_t = \sigma_t \xi_t$$
$$\sigma_t^2 = \gamma_0 + \gamma_1 u_{t-1}^2 + \cdots + \gamma_m u_{t-m}^2$$

5.5. Testing for the presence of conditional heteroskedasticity

- Before one enters into the stage of modeling conditional heteroskedasticity, one should test for the presence of (conditional) heteroskedasticity. A good test is the ARCH-LM test.
- **ARCH-LM test**:

  One considers the following autoregression model for the squared errors

$$u_t^2 = \beta_0 + \beta_1 u_{t-1}^2 + \cdots + \beta_q u_{t-q}^2 + errors_t$$

  In case of homoskedastic errors all $\beta_j = 0$, $j = 1, 2, \ldots, q$. Thus, one has the pair of hypotheses

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_q = 0 \quad \text{versus} \quad H_1 : \text{at least one } \beta_j \neq 0,\ j = 1, 2, \ldots, q$$

Since the errors are not directly observable, they are replaced the residuals $\hat{u}_t$. One then estimates

$$\hat{u}_t^2 = \beta_0 + \beta_1 \hat{u}_{t-1}^2 + \cdots + \beta_q \hat{u}_{t-q}^2 + errors_t$$

and computes the resulting R^2. Then the test statistic and its asymptotic distribution are given by

$$ARCH_{LM}(q) = TR^2 \xrightarrow{d} \chi^2(q).$$

Choice of q: In long time series one easily should choose q large enough. However, in short time series testing jointly too many parameters may lead to a loss of power since the estimated variances of the parameter estimates are quite large. Very often, choosing q small is already sufficient to reject the null of homoskedasticity.

- **Inspection of autocorrelation of squared residuals** If there are many estimated autocorrelations of the squared residuals outside the 95% confidence interval, one definitely should conduct the ARCH-LM test of adequate order.
- **Important**: If the null hypothesis is rejected, the asymptotic distribution of the OLS estimator for the conditional mean function is invalid!

  Solutions:

  – Use heteroskedasticity consistent standard errors, e.g. the White procedure

  – Specify model for conditional heteroskedasticity, e.g. ARCH, GARCH etc.

5.6. Model selection

In general the type and order of the conditional mean and the conditional volatility function are unknown and have to be estimated as well. The typical procedure is:

1. test for the presence of stochastic or deterministic trends or other kinds of nonstationarity (e.g. using ADF tests)
2. specify the conditional mean function (e.g. select order and terms in AR model)
3. check residuals of conditional mean model for remaining autocorrelation and potential (conditional) heteroskedasticity (e.g. using the LM test for autocorrelation and the ARCH-LM test).

   Again: ignoring (conditional) heteroskedasticity implies false inference!
4. specify a conditional heteroskedasticity model (e.g. ARCH or TGARCH) and choose a ML or the QML estimator
5. check standardized residuals

$$\hat{\xi}_t = \frac{\hat{u}_t}{\hat{\sigma}_t}$$

   of full model for remaining autocorrelation and check choice of error distribution in the previous step (problem: asymptotic distribution of ARCH-LM test not known except in the case of H_0: "no remaining ARCH´´); visual inspection of density estimate (e.g. histogram) of standardized

residuals

5.7. Prediction of conditional volatility

For simplicity assume a simple GARCH(1,1) process

$$\sigma_t^2 = Var(u_t|u_{t-1}, u_{t-2}, \ldots) = \gamma_0 + \gamma_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \quad \xi_t \sim i.i.d.(0,1)$$

Derivation of the h-**step ahead forecast of** σ_{T+h}^2 given information up to time T:

- Rewriting the GARCH(1,1) for various periods $T+j$, $j = 1, 2, 3$ one obtains

$$\begin{aligned}
\sigma_{T+1}^2 &= \gamma_0 + \gamma_1 u_T^2 + \beta_1 \sigma_T^2 \\
\sigma_{T+2}^2 &= \gamma_0 + \gamma_1 u_{T+1}^2 + \beta_1 \sigma_{T+1}^2 \\
\sigma_{T+3}^2 &= \gamma_0 + \gamma_1 u_{T+2}^2 + \beta_1 \sigma_{T+2}^2
\end{aligned}$$

- It is well known that the conditional expectation is the best predictor for minimizing the mean squared error of prediction (here for σ_{T+h}^2). Therefore, the **optimal predictor** given information up to time T is

$$\sigma_{T+h|T}^2 \equiv E[\sigma_{T+h}^2|\mathcal{I}_T]$$

where $\mathcal{I}_T$ denotes the available information up to time T.

- Note that by the law of iterated expectations one has

$$E[\sigma^2_{T+h}|\mathcal{I}_T] = E[E[\sigma^2_{T+h}|\mathcal{I}_{T+h-1}]|\mathcal{I}_T]$$

Applying to $h = 2$ one obtains

$$\begin{aligned}\sigma^2_{T+2|T} &= E[E[\sigma^2_{T+2}|\mathcal{I}_{T+1}]|\mathcal{I}_T] \\ &= E[\gamma_0 + \gamma_1 u^2_{T+1} + \beta_1 \sigma^2_{T+1}|\mathcal{I}_T] \\ &= \gamma_0 + \gamma_1 E[u^2_{T+1}|\mathcal{I}_T] + \beta_1 E[\sigma^2_{T+1}|\mathcal{I}_T]\end{aligned}$$

Since $u^2_t = \sigma^2_t \xi^2_t$ and therefore

$$E[u^2_{T+1}|\mathcal{I}_T] = \underbrace{E[\sigma^2_{T+1}|\mathcal{I}_T]}_{=\sigma^2_{T+1}} \underbrace{E[\xi^2_{T+1}|\mathcal{I}_T]}_{=1 \text{ since } \xi_t \text{ iid}}$$

one obtains

$$\begin{aligned}\sigma^2_{T+2|T} &= \gamma_0 + \gamma_1 \sigma^2_{T+1} + \beta_1 \sigma^2_{T+1} \\ &= \gamma_0 + (\gamma_1 + \beta_1)\sigma^2_{T+1}\end{aligned}$$

- Similarly one can compute the optimal predictor for $h = 3$:

$$\begin{aligned}\sigma^2_{T+3|T} &= E[E[\sigma^2_{T+3}|\mathcal{I}_{T+2}]|\mathcal{I}_T] \\ &= E[\gamma_0 + \gamma_1 u^2_{T+2} + \beta_1 \sigma^2_{T+2}|\mathcal{I}_T] \\ &= \gamma_0 + \gamma_1 E[u^2_{T+2}|\mathcal{I}_T] + \beta_1 E[\sigma^2_{T+2}|\mathcal{I}_T] \\ &= \gamma_0 + (\gamma_1 + \beta_1)\sigma^2_{T+2|T}\end{aligned}$$

Inserting $\sigma^2_{T+2|T}$ delivers

$$\begin{aligned}\sigma^2_{T+3|T} &= \gamma_0 + (\gamma_1 + \beta_1)\left[\gamma_0 + (\gamma_1 + \beta_1)\sigma^2_{T+1}\right] \\ &= \gamma_0\left[1 + (\gamma_1 + \beta_1)\right] + (\gamma_1 + \beta_1)^2\sigma^2_{T+1}\end{aligned}$$

- One can continue in this way in order to derive the h-**step ahead predictor**, $h \geq 2$,

$$\sigma^2_{T+h|T} = \gamma_0 \sum_{i=1}^{h-1}(\gamma_1 + \beta_1)^{i-1} + (\gamma_1 + \beta_1)^{h-1}\sigma^2_{T+1}$$

- In practice the optimal predictor is usually not feasible since the parameters are unknown. They have then to be replaced by consistent estimators.

- In `R` it is possible to compute these forecasts by applying method `predict()` to an object of class `garch`. An object of class `garch` is returned by function `garch()`, which estimates a GARCH(m,n) model by maximum likelihood. This function is part of package `tseries`.

5.8. Supplement: Organisation of an empirical project

An R program containing a complete empirical analysis is given in section A.9.

- **Purpose of project and data definition**
  - goal of project / model
  - definition of original data what is measured, how is measured (e.g. construction of financial indices/redefinitions, etc.)
  - period, sampling frequency
  - data source
  - specific data choice, e.g. opening/closing price...
  - data transformations (taking logarithms, etc.)
- **Descriptive statistics and first analysis of data properties**
  - plot of original and transformed data
  - mean
  - median

- max/min
- potential (conditional) heteroskedasticity
- skewness
- kurtosis
- normality
- correlations
- unit root testing:
  - choice of lags
  - choice of deterministic components (trend, constant, seasonal dummies)
  - choice of test(s) and required assumptions
  - level, differences
- seasonal components
- structural breaks, etc.
- Summary of findings and guess of model class(es)

- **Model choice and diagnostics**
  - choice of model class: e.g. AR or ARMA
  - choice of lag orders
  - choice of deterministic components: constant, time trend, seasonal dummies
  - choice of estimator and required assumptions
  - residual diagnostics
    - residual (auto)correlation
    - (conditional) heteroskedasticity
    - normality
    - stationarity
  - if necessary: model modification: e.g.: ARCH, GARCH, TGARCH
    - model choice
    - lag choice
    - residual check, see above plus correspondence of assumption about error distribution with

residual distribution

- **Statistical results and interpretation**
- **Forecasting or other uses**
  - point forecasts
  - interval forecasts
  - take into account prior data transformations
- **Check spelling**

- Offizielle Beschreibungen der 'DAXe' von der Deutschen Börse
  - **DAX**

    Aktienindex, der die Wertentwicklung der 30 nach Marktkapitalisierung größten und umsatzstärksten deutschen Aktien im Prime Standard der FWB® Frankfurter Wertpapierbörse abbildet.

    Der DAX-Index (Deutscher Aktienindex) wird von der Deutschen Börse aus den Kursen der 30 umsatzstärksten deutschen Aktien berechnet und ist der meist beachtete Indikator für die Entwicklung des deutschen Marktes.

    Die DAX-Werte notieren im Prime Standard. Kriterien für die Gewichtung der Aktien in DAX sind Börsenumsatz und Marktkapitalisierung des Streubesitzes. DAX wird als Kurs- und Performance-Index aus Xetra®-Kursen sekündlich berechnet und aktualisiert.

    Kursdaten zu DAX und den enthaltenen Werten sowie die Termine der Neuzusammensetzung finden Sie auf boerse-frankfurt.de/indizes.

  - **MDAX**

    Index, der die Wertentwicklung der 50 größten auf die DAX®-Werte folgenden Unternehmen der klassischen Branchen im Prime Standard abbildet.

MDAX® wird seit dem 19. Januar 1996 berechnet. Der Index enthält die 50 nach Marktkapitalisierung und Börsenumsatz größten Unternehmen der klassischen Branchen im Prime Standard unterhalb der DAX-Werte. Basis der Berechnung ist der 30. Dezember 1987 mit einem Wert von 1.000 Punkten.

Die Indexzusammensetzung wird üblicherweise halbjährlich überprüft und mit Wirkung zum März und September angepasst. Kriterien für die Gewichtung der Aktien in MDAX sind Börsenumsatz und Marktkapitalisierung auf Basis des Streubesitzes (Freefloats).

Ein Unternehmen kann außerhalb der ordentlichen überprüfungstermine aus dem Index genommen werden, wenn es beim Kriterium Marktkapitalisierung oder Börsenumsatz nicht mehr zu den 75 größten Unternehmen zählt, bzw. in den Index aufgenommen werden, wenn es bei den Kriterien Marktkapitalisierung und Börsenumsatz eines der 40 größten Unternehmen ist. Der Austausch findet zum nächsten Verkettungstermin statt. über Veränderungen in MDAX entscheidet der Vorstand der Deutsche Börse AG. Er wird dabei vom Arbeitskreis Aktienindizes beraten.

– **SDAX**

Index der 50 größten auf die MDAX®-Werte folgenden Unternehmen der klassischen Branchen des Prime Standard.

SDAX® startete am 21. Juni 1999. Er umfasst die 50 nach Marktkapitalisierung und

Börsenumsatz größten Unternehmen der klassischen Branchen unterhalb der MDAX-Werte. Der Index wird als Kurs- und Performance-Index von der Deutschen Börse berechnet. Basis der Berechnung ist der 30. Dezember 1987 mit einem Wert von 1.000 Punkten.

Die Indexzusammensetzung wird üblicherweise halbjährlich überprüft und mit Wirkung zum März und September angepasst. Kriterien für die Gewichtung der Aktien in SDAX sind: Börsenumsatz und Marktkapitalisierung auf Basis des Streubesitzes sowie Branchenrepräsentativität. über Veränderungen in SDAX entscheidet der Vorstand der Deutschen Börse. Er wird dabei vom Arbeitskreis Aktienindizes beraten.

– **TecDAX**

Index für die Wertentwicklung der 30 größten Technologieaktien im Prime Standard unterhalb der DAX®-Titel.

TecDAX® startete am 24. März 2003. Er umfasst die 30 nach Marktkapitalisierung und Börsenumsatz größten Unternehmen der Technologiebranchen im Prime Standard unterhalb des Leitindex DAX. Der Index wird als Kurs- und als Performance-Index berechnet. Basis der Berechnung ist der 30. Dezember 1997 mit einem Wert von 1.000 Punkten.

Die Indexzusammensetzung wird üblicherweise halbjährlich überprüft und mit Wirkung

zum März und September angepasst. Kriterien für die Gewichtung der Aktien in TecDAX sind: Börsenumsatz und Marktkapitalisierung auf Basis des Streubesitzes (Freefloats).

Ein Unternehmen kann außerhalb der ordentlichen überprüfungstermine aus dem Index genommen werden, wenn es beim Kriterium Marktkapitalisierung oder Börsenumsatz nicht mehr zu den 45 größten Unternehmen zählt, bzw. in den Index aufgenommen werden, wenn es bei den Kriterien Marktkapitalisierung und Börsenumsatz eines der 25 größten Unternehmen ist. Ein Austausch findet zum nächsten Verkettungstermin statt. über Veränderungen in TecDAX entscheidet der Vorstand der Deutsche Börse AG. Er wird dabei beraten vom Arbeitskreis Aktienindizes.

6. Long-run forecasting

6.1. Estimating/predicting unconditional means

- Time series and (dynamic) regression models are normally used to compute **conditional predictions**.
- For long-term investment decisions reliable forecasts of the **unconditional mean** of stock returns, interest rates, etc. are important as well: for large h, the h-step ahead forecast for a stationary time series is the unconditional mean of the series.
- Consider a stationary process $\{y_t\}$ (of returns) that has mean μ and variance σ_y^2,

$$y_t \sim (\mu, \sigma_y^2).$$

  Note that we did not assume that y_t is i.i.d., thus the covariances $Cov(y_t, y_s)$ may be non-zero.

The traditional **mean estimator** of a realization with T observations is

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

Properties:

– **Expected Value**

$$E(\hat{\mu}) = \frac{1}{T}\sum_{t=1}^{T} E(y_t) = \frac{1}{T}\sum_{t=1}^{T} \mu = \mu$$

– **Variance**

$$
\begin{aligned}
Var(\hat{\mu}) &= E\left[\left(\frac{1}{T}\sum_{t=1}^{T} y_t - \mu\right)^2\right] \\
&= E\left[\left(\frac{1}{T}\sum_{t=1}^{T} y_t - \frac{1}{T}T\mu\right)^2\right] = \frac{1}{T^2}E\left[\left(\sum_{t=1}^{T}(y_t - \mu)\right)^2\right] \\
&= \frac{1}{T^2}E\left[\sum_{t=1}^{T}\sum_{s=1}^{T}(y_t - \mu)(y_s - \mu)\right] \\
&= \frac{1}{T^2}\sum_{t=1}^{T} Var(y_t) + \frac{1}{T^2}\sum_{t=1}^{T}\sum_{s=1,s\neq t}^{T} Cov(y_t, y_s) \\
&= \frac{1}{T}\sigma_y^2 + \frac{1}{T^2}\sum_{t=1}^{T}\sum_{s=1,s\neq t}^{T} Cov(y_t, y_s)
\end{aligned}
$$

This expression simplifies **if** $\{y_t\}$ **is uncorrelated**. Then

$$Var(\hat{\mu}) = \frac{\sigma_y^2}{T}.$$

– **Asymptotic normality**

$$\sqrt{T}\left(\hat{\mu} - \mu\right) \xrightarrow{d} N\left(0, \sigma_y^2 + \gamma_\infty\right)$$

with

$$\gamma_\infty = \lim_{T\to\infty} \frac{1}{T^2} \sum_{t=1}^{T} \sum_{s=1, s\neq t}^{T} Cov(y_t, y_s).$$

Note that $\gamma_\infty = 0$ if y_t is uncorrelated.

The term γ_∞ converges if y_t is a stationary autoregressive process.

- **Estimating annual returns**
  - Given a time series of annual observations one has in case of uncorrelated returns the estimation variance $\frac{\sigma_y^2}{T}$.

– **Does it help to collect a time series of daily observations of daily returns**?

Let s denote the day of the year and t the year with 250 working days per year. Then

$$R_t = \prod_{s=1}^{250} R_{ts}$$

and in log returns

$$r_t \approx \sum_{s=1}^{250} r_{ts}.$$

For simplicity, assume i.i.d. daily returns with

$$\mu_{day} = \mu_{year}/250$$

and

$$\sigma_{day} = \sigma_{year}/\sqrt{250}.$$

Then

$$E[r_t] = E\left[\sum_{s=1}^{250} r_{ts}\right] = 250\mu_{day} = \mu_{year}$$

$$Var(r_t) = Var\left(\sum_{s=1}^{250} r_{ts}\right) \overset{iid}{=} \sum_{s=1}^{250} Var(r_{ts}) = 250\sigma_{day}^2 = 250\left(\frac{\sigma_{year}}{\sqrt{250}}\right)^2$$
$$= \sigma_{year}^2$$

* Thus, one has to multiply the daily mean by 250 in order to get the annual mean!

* And the **estimation variance for the annual mean stays the same** since

$$\begin{aligned} Var(250\hat{\mu}_{day}) &= Var\left(250\frac{1}{250T}\sum_{t=1}^{T}\sum_{s=1}^{250} r_{ts}\right) \\ &= Var\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{250} r_{ts}\right) \\ &\overset{iid}{=} \frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{250} Var(r_{ts}) \\ &= 250\sigma^2_{day} = \sigma^2_{year}. \end{aligned}$$

  Therefore, sampling at higher frequencies does not help to reduce the estimation variance!

* This conclusion remains qualitatively unchanged if returns are correlated.

- **Estimating long-horizon annualized returns**
  - If one is interested in estimating the 20-year return, then the same argument holds as above: sampling at a yearly frequency does not help to reduce the 20-year return variance.
  - However, if one is interested in the **annualized h-year return**? Let R_{h-year} denote the

h-year return and r_{h-year} the corresponding log return. The annualized return is then

$$R_{h-year,annualized} = \sqrt[h]{R_{h-year}} = (R_{h-year})^{\frac{1}{h}}$$

and the annualized log return

$$r_{h-year,annualized} = \frac{1}{h} r_{h-year}.$$

Therefore,

$$\mu_{h-year,annualized} = \mu_{h-year}/h$$

and

$$\sigma^2_{h-year,annualized} = \sigma^2_{h-year}/h^2.$$

Note that one has here the same scaling effect as for yearly and daily returns. Thus, the estimation variance for estimating h-year returns cannot be reduced by this trick.

- To make it completely clear: If one is interested in the 100-year overall return and has a sample of 100 annual returns, then the estimation variance is just $100\sigma^2_{year}$ (while for estimating the 1-year return it is $\sigma^2_{year}/100$).

- **Don't look only at the estimated mean alone**! It always pays to compute a confidence interval or even estimate the probability distribution of the returns. The latter can be done by looking at the

  - histogram or

– quantiles

of the empirical distribution.

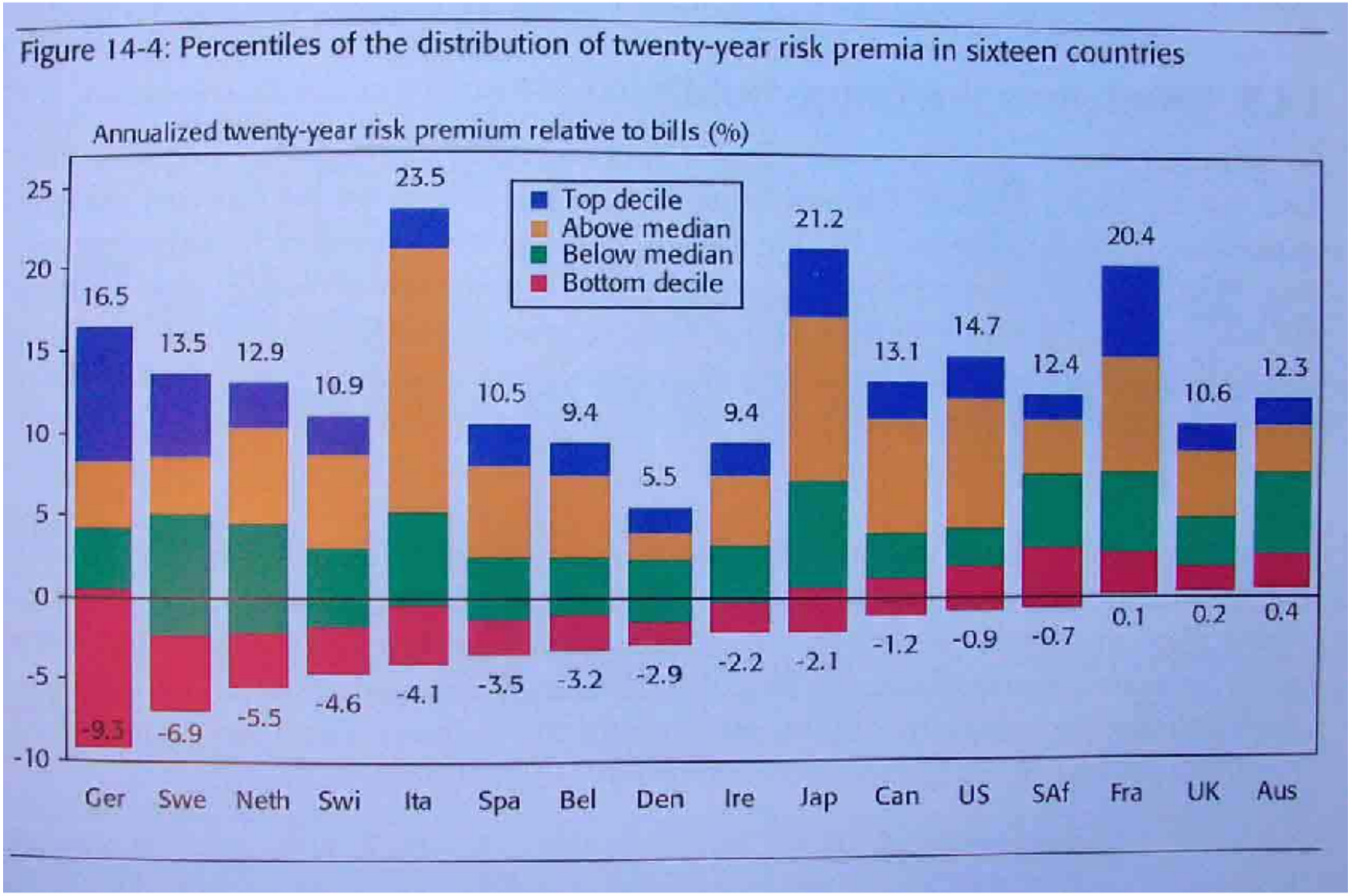
- **Overlapping observations**
  If one observes only T/h observations for the h-year mean return but one has yearly observations, then one may compute **overlapping** h-year returns: $r_1 + \cdots + r_h$, $r_2 + \cdots + r_{h+1}$, ..., $r_{T-h+1} + \cdots + r_T$. Note that computing a histogram/quantiles with overlapping observations does not deliver much more information since each observation is used h times! Nevertheless doing this delivers nicer pictures.

  And is done in practice. In the book of

  > Dimson, E., Marsh, P. and Staunton, M. (2002), *Triumph of the Optimists*, Princeton University Press

  one finds a large collection of results on the performance of various assets in OECD countries over the last 100 years. The following graph from this book shows the quantiles for the estimated 20-year return minus the risk free rate using overlapping observations:

Figure 14-4: Percentiles of the distribution of twenty-year risk premia in sixteen countries



6.2. Predicting long-term wealth: the role of arithmetic and geometric means

Timing schedule for **discrete payments / returns** on investments

Time	period 0		period 1		period 2		period 3
	begin	end	begin	end	begin	end	begin
Payments				$W_0(R_1 - 1)$		$W_0(R_2 - 1)$	
Wealth		W_0		$\underbrace{W_0 \cdot R_1}_{W_1}$		$\underbrace{W_0 \cdot R_1 \cdot R_2}_{W_2}$	

- Average of a given flow of discrete returns $\bar{R}_h$ over h periods if received payments are reinvested:

$$\begin{aligned} W_h &= W_0 \cdot R_1 \cdots R_h \\ &= W_0 \cdot \bar{R} \cdots \bar{R} \\ &= W_0 \cdot (\bar{R})^h. \end{aligned}$$

$\Longrightarrow$ **Average of a given flow of discrete return is given by the geometric mean of returns**

$$\bar{R} = [R_1 \cdots R_h]^{\frac{1}{h}}$$

- **Uncertain flow of discrete returns**:

  Consider the expected value of the uncertain final wealth W_h

$$E\left[W_h\right] = E\left[W_0 \prod_{t=1}^{h}(R_t)\right] \tag{6.1}$$

  – If discrete returns are **independently distributed**, then the expectation of the returns is the product of the expectations and one obtains

$$\mu_W \equiv E\left[W_h\right] \overset{indep.}{=} W_0 \prod_{t=1}^{h} E\left[R_t\right] \tag{6.2}$$

$$= W_0 \prod_{t=1}^{h} E[R_t]. \tag{6.3}$$

  If the means of the R_t's are known, then the expected final wealth is easily computed.

  – If discrete returns are **independently and identically distributed**, then this simplifies with $\mu_R = E[R_t]$ to

$$\mu_W = W_0(\mu_R)^h.$$

- **Additional estimation risk**:

  If the means are unknown, one faces in addition estimation risk. Now one has to be careful how to estimate $E[W_h]$!

  For simplicity, we continue to assume i.i.d. returns

$$R_t = \mu_R + \varepsilon_t, \quad \varepsilon_t \sim i.i.d.(0, \sigma_R^2).$$

  It is well known from statistics that one then can estimate an expected value by the sample mean, thus estimate

$$\hat{\mu}_R = \frac{1}{T} \sum_{t=1}^{T} R_t.$$

  This estimator for μ_R is unbiased since

$$E[\hat{\mu}_R] = \frac{1}{T} \sum_{t=1}^{T} E[\mu_R + \varepsilon_t] = \mu_R = E[R_t].$$

  In order to estimate μ_W

$$\mu_W = W_0 \prod_{j=1}^{h} \mu_R = W_0 (\mu_R)^h$$

  one has to replace μ_R by some estimate:

– **Arithmetic mean**: Replacing in μ_W the unknown μ_R by its arithmetic mean estimate delivers

$$\hat{\mu}_W = W_0 \left(\underbrace{\frac{1}{T}\sum_{t=1}^{T} R_t}_{=\hat{\mu}_R} \right)^h .$$

However, $\hat{\mu}_W$ is **biased** since (for simplicity consider $h = 2$)

$$\begin{aligned} E\left[\hat{\mu}_W\right] &= W_0 E\left[(\hat{\mu}_R)(\hat{\mu}_R)\right] \\ &= W_0 E\left[\hat{\mu}_R^2\right] \\ &= W_0 \left(\underbrace{E\left[(\mu_R + \bar{\varepsilon})^2\right]}_{=E\left[\mu_R^2 + 2\mu_R\bar{\varepsilon} + \bar{\varepsilon}^2\right]} \right) \\ &= W_0 \left(\mu_R^2 + \frac{1}{T}\sigma^2 \right) \\ &> W_0 \left(\mu_R^2\right) = \mu_W . \end{aligned}$$

Thus, **using the arithmetic mean estimator results in overestimating the expected final wealth**.

Exception: For very large T, the arithmetic mean is fine because the nonlinearity does no

longer matter. Therefore, the arithmetic mean estimator is **consistent**.

– **Geometric mean**: Replacing in μ_W the unknown μ_R by its geometric mean

$$\tilde{\mu}_R = \left(\prod_{t=1}^{T} R_1 \cdots R_T \right)^{1/T}$$

delivers

$$\tilde{\mu}_W = W_0 \left(\prod_{t=1}^{T} R_1 \cdots R_T \right)^{h/T}.$$

Thus, the expected value is

$$\begin{aligned} \tilde{\mu}_W &= W_0 E \left[\left(\prod_{t=1}^{T} R_1 \cdots R_T \right)^{h/T} \right] \\ &\overset{iid}{=} W_0 E \left[R_1^{h/T} \right] \cdots E \left[R_T^{h/T} \right]. \end{aligned}$$

If $h = T$ and $E[R_t] = \mu_R$, one obtains

$$\tilde{\mu}_W = W_0 \mu_R^h.$$

Only in this special case, $h = T$, the geometric mean estimator delivers an unbiased wealth estimate because there is no averaging effect.

Since both mean estimators are unbiased only in extreme situations, recently a combination of both was suggested to obtain an unbiased predictor (Eric Jacquier, Alex Kane, and Alan J. Marcus (2003). "Geometric or Arithmetic Mean: A Reconsideration", *Financial Analysts Journal*, 59, 46-53):

$$\breve{\mu}_W = W_0 \left(\text{Arithmetic average } \times (1 - h/T) + \text{ Geometric average } \times h/T\right)^h$$

(see Bodie, Kane & Marcus (2005), *Investments*, McGraw-Hill, 8th edition, p. 866).

- Note that the correction is horizon dependent!
- The correction can be large if the difference between the arithmetic and the geometric mean is large!
- Note that while the above estimator delivers an unbiased estimator of the annualized rate of return over h periods, this estimator does not exhibit the smallest mean squared error of prediction. For obtaining this, yet another estimator for μ_R has to be used, see Bodie, Kane & Marcus (2002). *Optimal Forecasts of Long-Term Returns and Asset Allocation: Geometric, Arithmetic, or Other Means?*, Working Paper or a paper of the same authors in *Financial Analysts Journal*, 2003.
- This case is a nice example that one has to consider the goal function explicitly in case of nonlinear functions.

6.2.1. Log or continuously compounding returns

- Using the definition of **continuously compounding returns**

$$r_t \equiv \log(R_t) \approx R_t - 1$$

and therefore

$$e^{r_t} = R_t$$

one can rewrite the wealth equation $W_h = W_0 \cdot R_1 \cdots R_h$ as

$$\begin{aligned} W_h &= W_0 e^{r_1} \cdots e^{r_h} \\ &= W_0 e^{r_1 + \cdots + r_h} \\ &= W_0 e^{\sum_{j=1}^{h} r_j} \\ &= W_0 e^{h\bar{r}_h} \end{aligned}$$

where

$$\bar{r}_h = \frac{1}{h} \sum_{j=1}^{h} r_j$$

defines the **arithmetic mean of the continuously compounding returns**.

- **Summary**:

$$\frac{W_h}{W_0} = \prod_{j=1}^{h}(R_j) = e^{\sum_{j=1}^{h} r_j}$$

and

$$\bar{R}_h \equiv \left[\prod_{j=1}^{h} R_j\right]^{\frac{1}{h}} = e^{\frac{1}{h}\sum_{j=1}^{h} r_j} \equiv e^{\bar{r}_h}.$$

The geometric mean of discrete returns translates into the arithmetic mean of log returns.

- In case **(log) returns are uncertain**, one obtains
  -

$$E\left[W_h\right] = W_0 E\left[e^{\sum_{j=1}^{h} r_j}\right] \tag{6.4}$$

  - **If stock returns are independent**, then this equation simplifies to

$$E\left[W_h\right] = W_0 \prod_{j=1}^{h} E\left[e^{r_j}\right]$$

  - **If stock returns are i.i.d.**, this equation further simplifies to

$$E\left[W_h\right] = W_0 \left(E\left[e^{r_j}\right]\right)^h$$

– To check the equality of both expectations indicates whether the returns are i.i.d.! This property can be used for testing the i.i.d. hypothesis.

– In order to compute the expected values one has to know the probability distribution for R_j. If it is unknown, in many cases a reasonable assumption is that

R_j **is i.i.d.lognormally distributed**.

This means assuming that

$$r_t = \mu + \varepsilon_t, \quad \varepsilon_t \sim i.i.d.N(0, \sigma^2)$$

and applying statistical rules to $R_t = e^{r_t}$ so that

$$E[R_t] = E\left[e^{r_t}\right] = e^{E[r_t]+\frac{1}{2}Var(r_t)} = e^{\mu+\frac{1}{2}\sigma^2}.$$

Thus, if the discrete returns are i.i.d.lognormally distributed, then the expected value of final wealth is

$$E[W_h] = W_0 \prod_{j=1}^{h} \left(e^{\mu+\frac{1}{2}\sigma^2}\right) = W_0 e^{h\left(\mu+\frac{1}{2}\sigma^2\right)} \neq W_0 e^{hE[r_t]}.$$

The inequality sign should not surprise since the expected value cannot be switched with nonlinear functions!

– If the mean μ and the variance σ^2 of the log returns r_t have to be estimated, one faces the same problems as for estimating the means of the R_j's. One cannot simply insert estimators

for μ in the exponent. This leads to a biased estimate. The reason for these difficulties is the same as above, the nonlinearity of the exponential function.

6.3. Are long-term returns predictable?

- This is an ongoing debate in current research. Some recent literature is:
  - Campbell, J.Y., A.W. Lo, and A.C. MacKinlay (1997). *The Econometrics of Financial Markets*, Princeton University Press
  - Campbell, J.Y. and Shiller, R.J. (2004). Valuation ratios and the long-run stock market outlook: an update, in: Barberis, N. und Thaler, R. (eds.). *Advances in Behavioral Finance, Volume II*, Russel-Sage Foundation. Working paper can be downloaded.
  - Cochrane, J.H. (2006). The dog that did not bark: a defense of return predictability, University of Chicago.
  - Goyal, A. and Welch, I. (2003),
    Predicting the Equity Premium with Dividend Ratios, *Management Science*, 49, 639-654.
  - Goyal, A. and Welch, I. (2006),
    A Comprehensive Look at the Empirical Performance of Equity Premium Prediction (January

11, 2006). Yale ICF Working Paper No. 04-11 Available at SSRN:

– Pesaran, M.H. und Timmermann, A.G. (2000). A recursive modelling approach to predicting UK stock returns, *Economic Journal*, 110, 159-191.

7. Explaining returns and estimating factor models

7.1. The basics of the theory of finance

- Literature:
  - Cochrane, J. (2005). *Asset Pricing*, Princeton University Press (book is used in graduate courses and succeeds in providing a unifying approach to finance using stochastic discount factors)
  - Bodie, Kane, Marcus (2005). *Investments*, MacGrawHill (an undergraduate book, technically much less demanding with a very broad focus and very up-to-date with empirical and theoretical develepments)
- **The basic problem of an investor**
  - The setup is made as simple as possible. There are many, more complicated extensions that do

not alter the basic message here. The presentation follows largely Cochrane (2005, Sections 1.1-1.4).

– Assumptions:

  * The investor is exclusively interested in consumption. Additional consumption is viewed less important if the consumption level is high $\longrightarrow$ utility function is concave

$$\frac{du(c_t)}{dc_t} \text{ declines in } c_t.$$

  * Future consumption is uncertain and future expected consumption is discounted using the discount factor β.

  * There is one investor who lives 2 periods having utility

$$U(c_t, c_{t+1}) = u(c_t) + \beta E[u(c_{t+1})].$$

  * In each period the investor receives a fixed endowment e_t and e_{t+1}.

  * The investor can invest some of his endowment in period t into an asset with price p_t and risky payoff x_{t+1} in period $t+1$ by buying ξ items of the asset.

$$c_t = e_t - p_t\xi$$
$$c_{t+1} = e_{t+1} + x_{t+1}\xi.$$

Thus, his control variable is ξ which allows her to maximize her utility

$$\max_{\xi} U(c_t, c_{t+1}) = \max_{\xi} u(c_t) + \beta E[u(c_{t+1})|\mathcal{I}_t].$$

* There are no buying or selling restrictions.

* There are no trading costs, taxes, etc....

– **Solving the model**:

Insert the budget constraints into the utility equation

$$U(c_t, c_{t+1}) = u(e_t - p_t\xi) + \beta E[u(e_{t+1} + x_{t+1}\xi)|\mathcal{I}_t]$$

and maximize with respect to ξ. The first-order condition (FOC) is

$$\frac{\partial U(c_t, c_{t+1})}{\partial \xi} = -u'(c_t)p_t + \beta E\left[u'(c_{t+1})x_{t+1}|\mathcal{I}_t\right] \stackrel{!}{=} 0. \tag{7.1}$$

Note that taking the derivative of the expectation works because

$$\frac{\partial E[g(x;\alpha)]}{\partial \alpha} = \frac{\partial}{\partial \alpha}\int g(x;\alpha)f(x)dx = \int \frac{\partial}{\partial \alpha} g(x;\alpha)f(x)dx = E\left[\frac{\partial g(x;\alpha)}{\partial \alpha}\right]$$

which can be applied here as well by defining

$$g(x_{t+1};\xi) \equiv u(c(x_{t+1},\xi)), \quad c(x_{t+1},\xi) = e_{t+1} + x_{t+1}\xi.$$

Rearranging the FOC and using the notation

$$E_t[\cdot] \equiv E[\cdot|\mathcal{I}_t]$$

delivers the central asset pricing formula:

- **Central asset pricing formula**

$$p_t = E_t\left[\frac{\beta u'(c_{t+1})}{u'(c_t)} x_{t+1}\right] \tag{7.2}$$

  - Note that c_t and c_{t+1} depend on ξ. Thus finding the optimal ξ requires the nonlinear solution of the asset pricing equation.
  - The theory of asset pricing can be viewed as providing specializations and manipulations of the asset pricing equation.
  - **Interpretation**:

    The investor is willing to buy an extra unit of the asset at price p_t if the marginal loss in utility due to less consumption is lower than the expected discounted marginal gain in future utility. The marginal gain in future utility is just the payoff times the marginal utility in consumption.
  - The term in front of the future payoff has a special function and is called stochastic discount factor. It will be discussed next.

- **Stochastic discount factor**:
  - Note that the random variable
$$m_{t+1} \equiv \frac{\beta u'(c_{t+1})}{u'(c_t)}$$
plays the function of a discount factor of the future payoff x_{t+1}. Since it is random, it is called **stochastic discount factor**.
  - Since in general $\beta < 1$ one has in general also $m_{t+1} < 1$ except if c_{t+1} is much smaller than c_t due to the decreasing marginal utility of consumption. In other words, **given current consumption c_t, the stochastic discount factor is the smaller, the larger future consumption c_{t+1} is, implying larger discounting**!
  - Thus, discounting depends on the uncertain level of future consumption. Therefore, the discount factor must be stochastic.
  - **Small example**: Imagine that there are 3 different states of the world in period 2 that imply 3 different payoffs. The larger the payoff, the larger the consumption in period 2, discounting is largest in the state with the largest payoff. Since in period 1 it is unknown which state will realize in period 2, the investor takes expectations. Note that by using the stochastic discount factor, the investor already takes care about her future utility of utility/consumption.
  - Thus, the stochastic discount factor does not only take into account the risk with respect to

future payoff but directly the risk with respect to future consumption.

- The stochastic discount factor is independent of a specific asset. Its value for a current state only depends on the consumption level of that state. **The function of the stochastic discount factor is the same for all assets**!
- However, the correlation between the values of the stochastic discount factor and the payoffs can be different among assets. It therefore provides an asset-specific risk correction w.r.t. utility maximization, the ultimate goal of investment.
- The specific function of the stochastic discount factor changes if the utility function is changed.
- Note that future payoff is linearly correlated with future marginal utility of consumption. In contrast, the dependence of future payoff with future consumption is in general highly nonlinear and invokes all moments!

- **Special cases**:
  - **No uncertainty**: There is only one future state and

$$p_t = m_{t+1}x_{t+1} = \frac{1}{R^f}x_{t+1}.$$

    Thus, in a risk-free world the stochastic discount factor corresponds to the inverse of the risk-free rate R^f.

  - **Uncertainty w.r.t. pay-off**: relationship between future consumption and payoff is ignored

$$p_t = \frac{1}{R^i}E_t[x_{t+1}].$$

    Here, the risk-adjustment is done by using the asset-specific discount factor $1/R^i$ for asset i.

  - **Uncertainty w.r.t. consumption**: relationship between future consumption and payoff taken into account $\rightarrow$ asset pricing equation

$$p_t = E_t[m_{t+1}x_{t+1}]$$

- **Other names** for the stochastic discount factor: marginal rate of substitution, pricing kernel, change of measure, state-price density.

- **Relating payoff risk and utility/consumption risk**
  - Since $Cov(X, Y) = E[XY] - E[X]E[Y]$ the asset pricing equation can be rewritten as

$$p_t = E_t[m_{t+1}]E_t[x_{t+1}] + Cov_t(m_{t+1}, x_{t+1}).$$

  - Assume for the moment that future consumption is stochastically independent with future payoff. Then $Cov_t(m_{t+1}, x_{t+1}) = 0$. Therefore, future payoff cannot help to control future consumption but it still may improve future consumption on average. In this case one can obtain the **risk-free rate**

$$R^f = \frac{1}{E[m_{t+1}]}$$

    as the inverse of expected stochastic discount factor. The price of such assets is

$$p_t = \frac{E_t[x_{t+1}]}{R^f}.$$

  - **Interpretation of the risk adjustment term** $Cov_t(m_{t+1}, x_{t+1})$:
    The asset price increases if an increase in payoff is associated with an increase in the stochastic discount factor $\sim$ increase in future marginal utility $\sim$ decrease future consumption and vice versa. Thus, a positive covariance helps to reduce the variation between current and future consumption (or in the level of marginal utility). **A positive covariance between the stochastic discount factor and payoff smoothes the consumption stream**. In

contrast, if the covariance is negative, the volatility of the consumption stream is increased and the investor wants to be payed for this by a higher return.

- **Systematic and idiosyncratic risk**
  If $Cov(m_{t+1}, x_{t+1}) \neq 0$, then there is correlation between the fluctuations in utility and in payoff. This joint risk is called **systematic** risk while the risk in payoff that is without impact on future consumption/utility is called **idiosyncratic**. From the asset pricing formula it follows that only the systematic risk influences the asset price. The idiosyncratic risk does not contribute to smoothing/aggrevating the consumption stream and is therefore not priced.

- **Important special cases of the asset pricing equation**:
  - The asset pricing formula (7.2) can be applied to a wide range of assets, e.g. to returns. This delivers the mean-variance frontier.
  - Be aware that the formulas resulting from the asset pricing formula (7.2) hold for an investor that optimized her portfolio in period 1 by buying the optimal number ξ of the asset.
  - A 'return' of an asset i costs in period t one unit and pays in period $t+1$ R^i units. Thus, dropping time indices,

$$1 = E[mR^i]$$
$$1 = \frac{E[R^i]}{R^f} + Cov(m, R^i).$$

Rewriting delivers an explanation for **expected excess returns**

$$\underbrace{E[R^i] - R^f}_{\text{expected excess return}} = \underbrace{-R^f Cov(m, R^i)}_{\text{risk adjustment}}. \tag{7.3}$$

Note that the stochastic discount factor m depends on c_t, c_{t+1} and thus on ξ. Interestingly, m is the quantity that is influenced by the investor by choosing ξ. Thus, by choosing ξ one influences the expected value or the covariance in the equation above.

- **Expected return-beta representation and the mean-variance frontier**
  - Rewriting equation (7.3) for expected excess returns delivers the **expected return-beta representation** or **beta-pricing model**

$$E[R^i] - R^f = \underbrace{\frac{Var(m)}{E[m]}}_{\lambda_m} \underbrace{(-1)\frac{Cov(m, R^i)}{Var(m)}}_{\beta_{i,m}}$$

$$E[R^i] - R^f = \lambda_m \beta_{i,m}. \tag{7.4}$$

In the representation (7.4) the coefficient λ_m represents the **price of risk** that is independent of the specific return considered and depends exclusively on the type and volatility of the stochastic discount factor. The beta-parameter $\beta_{i,m}$ denotes the quantity of systematic risk associated with return i.

  - Rewriting equation (7.3) for expected excess returns in a slightly different way, using $Cov(X, Y) =$

$Corr(X,Y)\sigma(X)\sigma(Y)$ with $\sigma(X) = \sqrt{Var(X)}$, delivers

$$E[R^i] - R^f = -\frac{Corr(m, R^i)\sigma(m)\sigma(R^i)}{E[m]}$$
$$\frac{E[R^i] - R^f}{\sigma(R^i)} = -Corr(m, R^i)\frac{\sigma(m)}{E[m]}.$$

The expression

$$\frac{E[R^i] - R^f}{\sigma(R^i)}$$

is called **Sharpe ratio**. Since $|Corr(m, R^i)| \leq 1$, one has

$$\left|\frac{E[R^i] - R^f}{\sigma(R^i)}\right| \leq \frac{\sigma(m)}{E[m]}.$$

The Sharpe ratio that is the slope of the mean-volatility line of any return i cannot be larger in absolute value than the coefficient of variation $\frac{\sigma(m)}{E[m]}$ of the stochastic discount factor m. The slope of the coefficient of variation is said to provide the **mean-variance frontier** for **all possible** returns. The mean-variance frontier is also called **capital market line**. Note that in the present context the **capital market line** depends on the decision of the investor through ξ.

Properties of the mean-variance frontier:

* all returns on the frontier are perfectly correlated with the stochastic discount factor since $|Corr(m, R^i)| = 1$.

* all returns on the upper frontier are maximally risky since consumption is highest when payoffs are highest and thus there is the least possible smoothing of consumption. Therefore the investor wants to be paid for taking this risk and the expected excess return is highest.

* all returns on the lower frontier are minimally risky since consumption is lowest when payoffs are highest and thus there is the most possible smoothing of consumption. Thus, the expected excess return is the lowest.

* any pair of two frontier returns are perfectly correlated and therefore spans/synthesizes any frontier return.

* any **mean-variance efficient return**, abbreviated by R^{mv} carries all pricing information since $Corr(m, R^{mv}) = -1$ and $R^{mv} = a + bm$. Then

$$Cov(m, R^{mv}) = -\sigma(m)\sigma(R^{mv}) = -b\sigma(m)\sigma(m).$$

For the **expected excess return of a mean-variance efficient return** the beta pricing model delivers

$$E[R^{mv}] - R^f = \lambda_m \frac{b\sigma^2(m)}{\sigma^2(m)} = b\lambda_m.$$

Inserting

$$\lambda_m = \frac{E[R^{mv}] - R^f}{b}$$

into the expected return-beta representation (7.4) for return i finally produces

$$E[R^i] - R^f = \beta_{i,m} \frac{E[R^{mv}] - R^f}{b}$$

$$\text{or} \quad E[R^i] - R^f = \beta_{i,mv} \left(E[R^{mv}] - R^f \right) \tag{7.5}$$

since

$$\beta_{i,mv} \equiv \frac{Cov(R^{mv}, R^i)}{Var(R^{mv})} = \frac{b\, Cov(m, R^i)}{b^2 Var(m)} = \frac{Cov(R^i, m)}{b Var(m)} \equiv \frac{\beta_{i,m}}{b}.$$

- **Example: power utility function**
  - Power utility:

$$u(c) = \frac{1}{1-\gamma} c^{1-\gamma}$$

$$u'(c) = c^{-\gamma}.$$

  - Stochastic discount factor:

$$m_{t+1} = \beta \left(\frac{c_{t+1}}{c_t} \right)^{-\gamma}$$

– **Case I**: no uncertainty

$$R_{t+1}^f = \frac{1}{m_{t+1}} = \frac{1}{\beta}\left(\frac{c_{t+1}}{c_t}\right)^\gamma$$

Real interest rates are high if

* people are impatient ($=\beta$ low)
* consumption growth is high
* the desire for smooth consumption is large (γ large)

– **Case II**: uncertainty

Assumption: consumption growth is log-normally distributed

$$\ln \frac{c_{t+1}}{c_t} = \mu_t + u_t, \quad u_t \sim N(0, \sigma_t^2).$$

The risk-free rate is

$$R_{t+1}^f = \frac{1}{E[m_{t+1}]}.$$

Using the assumption of log-normal consumption growth one has, $e^{-\delta} \equiv \beta$,

$$\beta \left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} = e^{-\delta} \left(e^{\ln c_{t+1} - \ln c_t} \right)^{-\gamma} = e^{-\delta} e^{-\gamma(\Delta \ln c_{t+1})}$$

$$E_t \left[\beta \left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} \right] = E_t \left[e^{-\delta} e^{-\gamma(\Delta \ln c_{t+1})} \right] = e^{-\delta} e^{-\gamma \mu_t + \frac{\gamma^2}{2} \sigma_t^2}$$

$$R_{t+1}^f = \left[e^{-\delta} e^{-\gamma \mu_t + \frac{\gamma^2}{2} \sigma_t^2} \right]^{-1}.$$

Real interest rates are high if

* impatience δ is high
* mean consumption growth μ_t is high
* more smoothing γ needed

and

* consumption variance σ_t^2 is small.

Note that the parameter γ in the power utility function controls

* the degree of intertemporal substitution (=aversion to a consumption stream that varies over time)

  * risk aversion (=aversion to a consumption stream that varies over states of nature)
  * the degree of precautionary savings ($\sigma_t^2\gamma^2/2$ term).

  These links can be broken up by using more flexible utility functions.

- Using the power utility function one can also express the beta pricing model directly in terms of changing consumption. Applying Taylor expansions to λ_m and $\beta_{i,m}$ in (7.4), one obtains

$$E[R^i] - R^f \approx \beta_{i,\Delta c} \underbrace{\gamma Var(\Delta c_{t+1})}_{\lambda_{\Delta c}}.$$

  Thus, the price of risk increases with

  * an increase in the variance of consumption growth
  * a larger risk aversion/degree of intertemporal substitution.

- **The equity premium puzzle**
  - If one computes the mean-variance frontier for the power utility function, then the Sharpe ratio of a mean-variance efficient return can be approximated by

$$\left|\frac{E[R^{mv}] - R^f}{\sigma(R^{mv})}\right| \approx \gamma\sigma(\Delta \ln c_{t+1})$$

  - Over the last 100 years the real stock return of the stock market index in the US was about 9% with standard deviation 16% while the real return of treasury bills was 1%. This delivers

a Sharpe ratio of about 0.5. For the same period consumption growth exhibited mean 1% and standard deviation 1%. Thus the risk aversion parameter γ has to be 50! This number is way too large given empirical evidence on risk aversion from economic experiments. Thus the expected excess return of the market portfolio was too large given the risk-aversion of people. This is the **equity premium puzzle**.

- Note that equation (7.5) is almost the capital asset pricing model. The **capital asset pricing model (CAPM)** states that the mean-variance efficient return is given by return on the market or total wealth portfolio

$$E[R^i] - R^f = \beta_{i,R^W} \left(E[R^W] - R^f \right). \tag{7.6}$$

Note that going from (7.5) to (7.6) requires additional assumptions. One of the following set of assumptions is enough, see e.g. Cochrane (2005, section 9.1),

  - quadratic utility function $u(c) = (c - c^*)^2$, c^* fixed, and no labor income or
  - exponential utility $u(c) = -e^{-ac}$, a fixed, and normally distributed returns

- **Random Walks and Efficient Markets**
  - A wide spread hypothesis: Financial markets are efficient if prices follow a random walk

$$\begin{aligned} p_t &= E_t[p_{t+1}] \\ p_{t+1} &= p_t + u_t, \quad u \sim (0, \sigma^2). \end{aligned}$$

This says: prices tomorrow are unpredictable given prices and other information until today.

– A generalization of a random walk is a **martingale**

$$y_{t+1} = y_t + u_t, \quad u_t \sim (0, \sigma_t^2)$$

where the variance of the error term may be (conditionally) heteroskedastic.

– Consider the asset pricing formula:

$$p_t = E_t[m_{t+1} x_{t+1}].$$

This relationship only reduces to a random walk if $m_{t+1} = 1$ and $x_{t+1} = p_{t+1}$. This implies that

* one considers a *short* time horizon such that $\beta \approx 1$,
* investors are risk neutral $\Leftrightarrow$ $u(\cdot)$ linear or $\sigma_c^2 = 0$ and
* there is no dividend payment in $t + 1$.

– How to view the asset pricing formula as a random walk? Prices follow a random walk *after scaling and adjusting by discounted marginal utility*.

– Since changes in marginal utility may not matter in the very short run, the random walk hypothesis is likely to hold in the very short run. But in the long run?

Reconsider the beta-pricing model (7.4) with power utility that was used to explain the equity premium puzzle

$$E_t[R^{mv}] - R^f \approx \gamma_t \sigma_t(\Delta \ln c_{t+1}) \sigma_t(R^{mv}).$$

A time index was added to each quantity since the asset pricing formula was derived conditional on information up to time t. The **expected return can vary if**

* the risk aversion γ_t changes,
* the conditional variance of the mean-variance return changes,
* the conditional variance of consumption growth changes

Note that all quantities may vary in the long-run but not in the short-run. Thus, **prices should only be predictable in the long-run if at all!**

- **Infinite horizon models**
  - Imagine an investor with infinite horizon that can purchase a stream of dividends $\{d_{t+j}\}$ at price p_t. The asset pricing formula continues to hold

$$p_t = E_t[m_{t+1}(p_{t+1} + d_{t+1})]$$

with $x_{t+1} = p_{t+1} + d_{t+1}$ being the payoff in period $t+1$. It may also be written as

$$p_t = E_t \left[\sum_{j=0}^{\infty} m_{t,t+j} d_{t+j} \right], \quad m_{t,t+j} = \beta^j \frac{u'(c_{t+j})}{u'(c_t)}$$

by iterating the asset pricing formula forward (=iteratively inserting) and using the assumption (=transversatility condition) $\lim_{j\to\infty} E_t[m_{t+j} p_{t+j}] = 0$.

Thus, **asset prices are expected future dividends stochastically discounted**.

- **Important remarks**:
  The asset pricing formula
  1. is no equilibrium condition,
  2. does not require complete markets or a representative investor,
  3. does not require specific assumptions on future prices/returns,
  4. does not require a specific utility function,
  5. does not exclude other income sources,
  6. allows for stochastic discount factors that vary across individuals,
  7. makes a statement about optimal investment for a marginal investment for an individual

investor given current price,

8. states the price *given* the joint distribution of the (subjective) stochastic discount factor (or consumption) and the asset payoff or
9. states current consumption $u'(c_t)$ *given* the joint distribution of the (subjective) stochastic discount factor (or consumption) and the asset payoff and today's price
10. does not specify truly exogenous stochastic processes that drive consumption and payoffs.

See Cochrane (2005, Section 2.1). In the following some of the items above are treated in more detail.

- **Finding the individual willingness to pay for an asset using the asset pricing equation**
  - This amounts to finding the price for investing an extra marginal amount ξ into an asset. This increases the investor utility from

$$u(c_t) + \beta E_t[u(c_{t+1})]$$

to

$$u(c_t - p_t\xi) + \beta E_t[u(c_{t+1} + x_{t+1}\xi)].$$

This implies the

$$\begin{aligned}\text{loss} &: u(c_t - p_t\xi) - u(c_t)\\ \text{gain} &: \beta E_t[u(c_{t+1} + x_{t+1}\xi) - u(c_{t+1})].\end{aligned}$$

Using first-order Taylor expansions taken at c_t and c_{t+1} one obtains approximatively

$$\begin{aligned}\text{loss} &: u(c_t) + u'(c_t)p_t\xi - u(c_t) = u'(c_t)p_t\xi\\ \text{gain} &: \beta E[u(c_{t+1}) + u'(c_{t+1})x_{t+1}\xi - u(c_{t+1})] = \beta E[u'(c_{t+1})x_{t+1}\xi].\end{aligned}$$

Comparing the loss and the gain one obtains

$$p_t = \left\{\begin{array}{c} > \\ = \\ < \end{array}\right\} \frac{\beta E[u'(c_{t+1})x_{t+1}\xi]}{u'(c_t)\xi} = E\left[\beta\frac{u'(c_{t+1})}{u'(c_t)}x_{t+1}\right] \equiv v_t.$$

On the left hand side is the market price p_t. On the right hand side one has the private valuation v_t of an extra unit of the asset given everything else. Now the investor faces three situations

* $v_t > p_t$ (private valuation larger than market valuation) the investor buys more of the asset. This increases future consumption on average leading to an average decline of future marginal utility and thus a decrease in the private valuation of additional units of the asset. The investor stops buying once $v_t = p_t$.

  * $v_t = p_t$ nothing happens. The investor is in post-trade or in the equilibrium situation. In the optimum for the investor loss equals gain.

  * $v_t < p_t$ (private valuation smaller than market valuation) the investor sells more of the asset. This decreases future consumption on average leading to an average increase of future marginal utility and thus an incrase in the private valuation of additional units of the asset. The investor stops selling once $v_t = p_t$.

- **Complete markets, the stochastic discount factor, and risk-neutral probabilities**

  – The stochastic discount factor can be related to the very important concept of complete markets. First we need a class of specific assets:

  – **Contingent claims**:

  Consider a simple world with finitely many states in the future, let's say S states. A **contingent claim for state** s is an asset that pays exactly 1 unit in state s and nothing in any other state. Thus, by buying a contingent claim for all S states one can ensure to always obtain 1 unit in the future period. This certain payoff costs simply the sum of the prices of all contingent claims.

  Note that this insurance type of investment only works if there exists a contingent claim for every state. In this case, a market is called **complete**. Otherwise, a market is called

incomplete. In reality, markets are in general incomplete.

For a market to be complete it is not necessary that contingent claims themselves exist. It is sufficient that there exist enough assets such that a contingent claim for each state can be constructed by choosing an appropriate portfolio of available assets. One says **the assets span/synthezise all contingent claims**.

– **Relating complete markets with the stochastic discount factor**

* Price of contingent claim for state s: $pc(s)$
* Recall that the price of a payment that pays $x(1), \ldots, x(S)$ units in states 1 to S in the next period is

$$p(x) = \sum_{s=1}^{S} x(s) pc(s).$$

In order to write this as an expected value write

$$\begin{aligned} p(x) &= \sum_{s=1}^{S} x(s) \frac{pc(s)}{\pi(s)} \pi(s) \\ &= E\left[x(s) \frac{pc(s)}{\pi(s)} \right] \end{aligned}$$

where $\pi(s)$ denotes the probability that state s occurs in the next period.

* Now observe that the ratio of contingent claims prices and probabilities play the role of the stochastic discount factor! Thus, define the stochastic discount factor for state s

$$m(s) \equiv \frac{pc(s)}{\pi(s)}, \quad s = 1, \ldots, S$$

and obtain the asset pricing equation

$$p(x) = E\left[x(s)m(s)\right].$$

* Thus, if markets are complete, a stochastic discount factor exists. If the state space is continuous a similar result can be shown but this is more difficult.

* Now consider the price for a certain payout of 1 unit

$$\sum_{s=1}^{S} pc(s) = \sum_{s=1}^{S} \underbrace{\frac{pc(s)}{\pi(s)}}_{m(s)} \pi(s) = E\left[m(s)\right].$$

Thus, the risk-free rate of return is

$$R^f = \frac{1}{\sum_{s=1}^{S} pc(s)} = \frac{1}{E\left[m(s)\right]}$$

as shown before.

– **Risk-neutral probabilities**

* The concept of risk-neutral probabilities is very important for pricing options or computing term structure models for interest rates. It can be very nicely explained using the asset pricing equation.

* Note that **if the investor were risk-neutral**, then marginal utility would be constant and the asset pricing equation would simplify to

$$p(x) = \beta E[x(s)].$$

* Now we expand the asset pricing equation once more. This time by $E[m(s)]$ which leads to

$$\begin{aligned} p(x) &= E\left[x(s)\frac{m(s)}{E[m(s)]}E[m(s)]\right] \\ &= E[m(s)]\sum_{s=1}^{S} x(s)\underbrace{\frac{m(s)}{E[m(s)]}\pi(s)}_{\pi^*(s)}. \end{aligned}$$

Note that $\pi^*(s)$ are also probabilities since they add up to 1 (because $\frac{m(s)}{E[m(s)]}$ is a weight

function that sums to 1). Thus, one can write in short

$$\begin{aligned} p(x) &= E[m(s)] \sum_{s=1}^{S} x(s)\pi^*(s) \\ &= \frac{1}{R^f} E^*[x(s)] \end{aligned}$$

where $E^*[\cdot]$ denotes the expectation using the probabilities $\pi^*(s)$. The probabilities $\pi^*(s)$ are called **risk-neutral probabilities** since now the price corresponds to the formula of a risk-neutral investor. Thus, one was able to move the risk-aversion into the probabilities by changing them.

* A few remarks:
  - Agents are risk-neutral with respect to the probability distribution π^*.
  - Note that π^* gives a greater weight to states with higher than average future marginal utility $u'(c(s))$, see example below.
  - Risk aversion is equivalent to paying more attention to unpleasant states relative to their actual probability of occurrence.
  - The combination of $\pi \times m$ is the most important piece of information for many decisions!

· There is also technical jargon (not needed in this course but for later reference)

$$\pi^*(s) = \underbrace{\frac{m(s)}{E[m(s)]}}_{\text{change of measure}} \pi(s).$$

The expression $\frac{m(s)}{E[m(s)]}$ is called **change of measure** or **derivative** since it allows to switch from the measure (=probability distribution) π to the risk-neutral measure (=probability distribution) π^*.

– **Example**

* Power utility: $u(c) = \frac{1}{1-\gamma}c^{1-\gamma}$ with $\gamma = \frac{1}{2}$. Thus, marginal utility is $u'(c) = c^{-1/2}$.
* For simplicity, the discount rate $\beta = 1$.
* Consumption in period t is already chosen and $c_t = 9$. Thus, current marginal utility is $u'(c_t) = \frac{1}{3}$.
* Endowment in period $t+1$ is independent of the state and is $e_{t+1} = 6$. The only other income in period $t+1$ is the payoff $x_{t+1}(s)$ of the asset.
* In the table below the payoff for each of the $S = 3$ states is specified and the resulting future marginal utility, the discount factor for each state, the risk-neutral probability and the price of a contingent claim that pays 1 unit in one state.

state	probability	payoff	future consumption	future marginal utility	stochastic discount factor	risk-neutral probability	price of contingent claims
s	$\pi(s)$	$x_{t+1}(s)$	$c_{t+1}(s)$	$u'(c_{t+1}(s))$	$m_{t+1}(s)$	$\pi^*(s)$	$pc(s)$
1	$\frac{1}{2}$	3	9	$\frac{1}{3}$	1	$\frac{1}{2}$	$\frac{1}{2}$
2	$\frac{1}{3}$	10	16	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
3	$\frac{1}{6}$	-2	4	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

Notice that the risk-neutral probability for the state with negative payoff is much higher than the probability itself, indicating the risk aversion of the investor!

* Price of asset with payoff x_{t+1}F

$$p_t = E\left[m_{t+1}x_{t+1}\right] = \sum_{s=1}^{3} m_{t+1}(s)x_{t+1}(s)\pi(s) = 3\frac{1}{2} + \frac{30}{4}\frac{1}{3} - 3\frac{1}{6} = \frac{7}{2}.$$

Price of asset using risk-neutral probabilities

$$p_t = E^*\left[x_{t+1}\right] = \sum_{s=1}^{3} x_{t+1}(s)\pi^*(s) = 3\frac{1}{2} + 10\frac{1}{4} - 2\frac{1}{4} = \frac{7}{2}.$$

7.2. Is the asset pricing equation empirically relevant?

- In order to check the asset pricing equation empirically, one has to assume something about the stochastic discount factor and to model its joint distribution with future payoff. Moreover one also has to assume that there exists something like an average utility function or a representative agent.
  - Simplest case: power utility function, see above. This assumption is not very well supported by the data.
  - Alternatives:
    - **Other utility functions**: variables driving utility; separating intertemporal substitution and risk aversion, etc.
    - **General equilibrium models**: link consumption to other variables e.g. income, interest rates; model covariance in beta explicitly
    - **Factor pricing models**: model stochastic discount factor (ratio of marginal utilities) in terms of other variables

$$m_{t+1} = a + b_A \underbrace{f^A_{t+1}}_{\text{factor } A} + b_B f^B_{t+1} + \cdots,$$

Special cases:

- **Capital Asset Pricing Model (CAPM)**

$$m_{t+1} = a + bR^W_{t+1}$$

where R^W_{t+1} is the rate of return on a claim to total wealth.

- **Arbitrage Pricing Theory (APT)**
- **Intertemporal Capital Asset Pricing Model (ICAPM)**
- **Term structure models**

* **Arbitrage or near-arbitrage pricing**: use the no-arbitrage condition to determine the price of one payoff in terms of the prices of other payoffs (most famous: Black-Scholes option prices). Works because of the existence of the asset pricing equation and positive marginal utility.

7.3. Factor-pricing models

- Idea: link stochastic discount factor to other (observable) data

Easiest: use a link that linearly combines several factors: **linear factor pricing models**. This

is mostly used in empirical work, see below. First, we discuss its theoretical properties.

- **Factor pricing models**
  - replace consumption-based expression for marginal utility growth version of stochastic discount factor with a linear model

    $$m_{t+1} = a + b' f_{t+1}$$

    where f_{t+1} is a vector of factors that have to be determined.
  - are equivalent to the multivariate/multiple version of the **expected return-beta representation** or **beta-pricing model**

    $$E[R^i_{t+1}] = \gamma + \beta' \lambda$$

    where λ is a vector that captures the various prices of risks that influence expected returns.
  - **Main issue**: which factors to choose?
    - If one starts again from the simple consumption-based model, then the factors must proxy **aggregate** marginal utility growth

      $$\beta \frac{u'(c_{t+1})}{u'(c_t)} \approx a + b' f_{t+1}$$

    - Why is this view useful?

> "the essence of asset pricing is that there are special states of the world in which investors are especially concerned that their portfolios not do badly. They are willing to trade off some overall performance—average return—to make sure that portfolios do not do badly in those particular states of nature. The factors are variables that indicate that these "bad states" have occured." Cochrane (2005, p.149)

* One needs **factors** that
  - **measure the state of the economy**: $u'(c_t)$
  - **forecast the state of the economy**: $u'(c_{t+1})$
* **Factors that measure** the state of the economy:
  - returns on broad-based portfolios
  - interest rates
  - growth in GDP
  - investment
  - other macroeconomic variables
  - returns to real production processes

    - etc.

  * **Factors that forecast** the state of the economy must be related to **news**
    - variables that are correlated with "changes in the investment opportunity set"
    - term premium
    - dividend price ratio
    - stock returns
    - etc.

– **Should factors be predictable over time**?

  * Simple case: assume constant real interest rate

$$R^f = \frac{1}{E_t[m_{t+1}]}.$$

Then using the consumption based stochastic discount factor

$$m_{t+1} = \beta \frac{u'(c_{t+1})}{u'(c_t)}$$

one obtains

$$1 = R^f \beta E_t \left[\frac{u'(c_{t+1})}{u'(c_t)} \right], \quad E_t \left[\frac{u'(c_{t+1})}{u'(c_t)} \right] = \frac{1}{\beta R^f}.$$

Thus

$$\frac{u'(c_{t+1})}{u'(c_t)} = \frac{1}{\beta R^f} + \varepsilon_{t+1}, \quad E[\varepsilon_{t+1}] = 0$$

* In reality, the real interest rate varies, however, not too much. Thus, marginal utility growth cannot be expected to be highly predictable! This carries over to **all** factors that should proxy marginal utility growth!

* Thus, one may choose factors that represent changes themselves: GNP growth, portfolio returns, price-dividend ratios, etc.

- **Capital Asset Pricing Model (CAPM)**
  - Most famous asset pricing model
  - A one factor model with the factor **return on the "wealth portfolio"**

$$m_{t+1} = a + bR^W_{t+1}.$$

  - how to proxy the wealth portfolio: in practice S&P500, value- or equally-weighted NYSE, DAX(?)

- One simple **derivation** based on two-period quadratic utility:

  Assumptions:

- quadratic utility $u(c) = -\frac{1}{2}(c_t - c^*)^2$.
- one investor, living two periods, maximizing the utility

$$U(c_t, c_{t+1}) = -\frac{1}{2}(c_t - c^*)^2 - \frac{1}{2}\beta E\left[(c_{t+1} - c^*)^2\right].$$

- Initial endowment/wealth W_t exogenously given at the beginning of period 1
- No labor income
- Investment opportunities: N assets with price p_t^i for asset i with payoff x_{t+1}^i. Thus, asset i has return R_{t+1}^i.

The optimization problem is to maximize intertemporal utility given the budget constraint by selecting the portfolio weights w_i and implicitly c_t, c_{t+1}:

$$\begin{aligned} c_{t+1} &= W_{t+1} \\ W_{t+1} &= R_{t+1}^W (W_t - c_t) \\ R_{t+1}^W &= \sum_{i=1}^N w_i R_{t+1}^i, \quad \sum_{i=1}^N w_i = 1 \end{aligned}$$

where R_{t+1}^W and R_{t+1}^i denote the rate of return on total wealth and asset i, respectively. Note

that shortselling is allowed since w_i maybe negative.

$$
\begin{aligned}
m_{t+1} &= \beta \frac{R_{t+1}^W \left(W_t - c_t\right) - c^*}{c_t - c^*} \\
&= \underbrace{\frac{\beta c^*}{c^* - c_t}}_{a_t} - \underbrace{\frac{\beta \left(W_t - c_t\right)}{c^* - c_t}}_{b_t} R_{t+1}^W \\
m_{t+1} &= a_t - b_t R_{t+1}^W
\end{aligned}
$$

Note that the denominator was rearranged such that a_t and b_t are non-negative for $c_t \leq c^*$ which is the relevant part of the utility function. Note also that the parameters a_t and b_t are time-dependent since they depend on the situation at time t. Inserting $m_{t+1} = a_t - b_t R_{t+1}^W$ into the expected returns - beta representation results in a time-varying price of risk $\lambda_{W,t}$ and beta

$\beta_{W,i,t}$

$$
\begin{aligned}
E_t\left[R^i_{t+1}\right] - R^f_{t+1} &= \frac{Var_t(m_{t+1})}{E_t[m_{t+1}]} \frac{-Cov_t(m_{t+1}, R^i_{t+1})}{Var_t(m_{t+1})} \\
&= \underbrace{\frac{Var_t(R^W_{t+1})b_t^2}{a_t - b_t E_t[R^W_{t+1}]}}_{\lambda_{W,t}} \underbrace{\frac{(-1)^2 b_t Cov_t(R^W_{t+1}, R^i_{t+1})}{Var_t(R^W_{t+1})b_t^2}}_{\beta_{W,i,t}} \\
E_t\left[R^W_{t+1}\right] - R^f_{t+1} &= \lambda_{W,t} \cdot 1 \\
&= \text{and therefore} \\
\lambda_{W,t} &= E_t\left[R^W_{t+1}\right] - R^f_{t+1}
\end{aligned}
$$

- Other factor models are the Intertemporal Capital Asset Pricing Model (ICAPM) or the Arbitrage Pricing Theory (APT), see e.g. Cochrane (2005, Sections 9.2 - 9.5)

7.4. Regression based tests of linear factor models

- **Using time series regressions**
  - Recall the linear single factor model using the notation of excess returns $R^{ei} = R^i - R^f$ and

$R^{eW} = R^W - R^f$:

$$E[R_t^{ei}] = \beta_{i,W}\lambda$$

- Note that the excess return-beta representation is also obtained by taking expectations of

$$R_t^{ei} = \beta_{i,W} R_t^{eW} + \varepsilon_t^i, \quad E[\varepsilon_t^i] = 0, \quad i = 1, \ldots, N, t = 1, \ldots, T.$$

- This model is nested in the standard time series regression model (including a constant)

$$R_t^{ei} = \alpha_i + \beta_{i,W} R_t^{eW} + \varepsilon_t^i. \tag{7.7}$$

In practice, all **pricing errors** α_i should be zero.

- **Estimation procedure**
  - Estimate price of risk λ by $\frac{1}{T}\sum_{t=1}^T R_t^{eW}$
  - Run N OLS regressions of (7.7) and test for each

$$H_0 : \alpha_i = 0 \quad \text{versus} \quad H_1 : \alpha_i \neq 0.$$

Example: Estimation of a CAPM model

– **Data**: monthly data January 2003 to June 2012 of `DAX`, `Bayer`, `Daimler`, `E.ON`, `Deutsche Post`, `Deutsche Lufthansa`, `SAP`, `Siemens` and short-term interest rates; see file capm.RData.

– Computation of excess returns of stock i

$$R_t^{ei} = \ln P_t - \ln P_{t-1} - R_t^f/(12 * 100),$$

since the three-month interest rates are given annualized in percent.

– **Estimation of time series regressions in R**

Regressing excess returns for the `Bayer` stock on the excess returns of the `DAX` yields

```
Call:
lm(formula = excess.stock ~ +1 + excess.market)

Residuals:
      Min        1Q    Median        3Q       Max
-0.217861 -0.032947 -0.001235  0.030595  0.125766

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.005474   0.005143   1.065    0.289
excess.market 0.992579   0.086037  11.537   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05443 on 111 degrees of freedom
```

```
Multiple R-squared: 0.5453,Adjusted R-squared: 0.5412
F-statistic: 133.1 on 1 and 111 DF,  p-value: < 2.2e-16
```

Hence, the constant term is not significant and beta is close to unity.

- **Conclusion:** Do you think that the CAPM captures most of the variation of the equity premia for the considered period and stocks?

  In general, adding additional factors play a role in explaining equity premia such as the difference between returns of big and small companies or the difference in returns between companies with a high book-to-market ratio and those with a low ratio. For a recent survey on the CAPM and its extensions, see e.g.

  Fama, Eugene F., French, Kenneth R. (2004). The Capital Asset Pricing Model: Theory and Evidence, *Journal of Economic Perspectives* **18**, 25 – 4

  Perold, André F. (2004). The Capital Asset Pricing Model Source, *Journal of Economic Perspectives* **18**, 3 – 24

A. R programs for the empirical examples

Note:
The data are available from GRIPS. You need to create the subdirectory Data within the folder of your R programs and save the data in the subdirectory. You also need to set the working directory in R on the directory that contains your R programs.

A.1. Real Prices of the S&P 500

Used in section 1.2.

Data source: homepage of Robert Shiller: www.econ.yale.edu/~shiller/.

R program:

```
##############################################################################
#                    AOE_Ch1-4_Shiller_AR1.R
```

```
################################################################################

# Plot graph of real prices and real earnings of the Shiller S&P 500 data
# shown in the lecture notes and estimates an AR(1) model for the
# real_price data series from Shiller's dataset ie_data.xls
#
# the dataset "ie_data_2021_04_12.xls" has to be in the subdirectory "data".

# written by Rolf Tschernig (RT)
# version: 2021_04_15, 2022_04_28, 2023_04_10 (Shiller data updated)


# ------------------------------------------------------------------------------
#   Specify parameters for analysis
# ------------------------------------------------------------------------------

Shiller_series<- "RealPrice"
# available are: SPCompPrice", "Dividend", "Earnings",
# "CPI", "DateFraction", "LongInterest",
# "RealPrice", "RealDividend",
# "RealTotalReturnPrice", "RealEarnings",
# "RealScaledEarnings", "P/E10"

transformdata = "log"   # must be: level, diff, log, or difflog

# ------------------------------------------------------------------------------
#   load and check data, create zoo and time series objects
# ------------------------------------------------------------------------------

if (!require(zoo)) install.packages("zoo") # requires library zoo
```

```
library(zoo)
if (!require(tseries)) install.packages("tseries") # requires library zoo
library(tseries)
if (!require(readxl)) install.packages("readxl") # for reading .xls or
                          #.xlsx files
library(readxl)

    # if you use all available data
ie_data <- data.frame(
    read_excel(path="Data/ie_data_2023_04_10.xls",
               range="Data!B9:M1834",
               col_names=c("SPCompPrice", "Dividend", "Earnings",
                           "CPI", "DateFraction", "LongInterest",
                           "RealPrice", "RealDividend",
                           "RealTotalReturnPrice", "RealEarnings",
                           "RealScaledEarnings", "P/E10")))
head(ie_data)
tail(ie_data)

#  create ts object
#     Note that ts cannot handle missing values. This requires library zoo
ie_data_ts            <- ts(ie_data, start = c(1871, 1), frequency = 12)
ie_data_zoo             <- zoo(ie_data_ts)
series_to_check_all_ts  <- window(ie_data_ts[,Shiller_series], start=NULL, end=NULL)
series_to_check_all_zoo <- window(ie_data_zoo[,Shiller_series, drop=FALSE],
                                  start=NULL, end=NULL)
# drop=FALSE keeps series_to_check_all_zoo as
# a matrix object and stores therefore also the
# the series name
head(series_to_check_all_zoo)
```

```
head(series_to_check_all_ts)
tail(series_to_check_all_zoo)
tail(series_to_check_all_ts)

# transform data if desired

if (transformdata=="level"){
    series_to_check_zoo       <- series_to_check_all_zoo
    series_to_check_ts        <- series_to_check_all_ts
} else if (transformdata=="diff"){
    series_to_check_zoo       <- diff(series_to_check_all_zoo,lag=1)
    series_to_check_ts        <- diff(series_to_check_all_ts,lag=1)
} else if (transformdata=="log"){
    series_to_check_zoo       <- log(series_to_check_all_zoo)
    series_to_check_ts        <- log(series_to_check_all_ts)
} else if (transformdata=="difflog"){
    series_to_check_zoo       <- diff(log(series_to_check_all_zoo),lag=1)
    series_to_check_ts        <- diff(log(series_to_check_all_ts),lag=1)
} else {
    print("transformdata has to be level, diff, log, or difflog")

}
nobs <- length(series_to_check_zoo)

# ------------------------------------------------------------------------
#           Plot time series
# ------------------------------------------------------------------------

plot.zoo(series_to_check_zoo, xlab = "Time",
         ylab = paste(names(series_to_check_zoo),
```

```
                    ", data transformation: ", transformdata),
        main = "S&P 500 Real Prices" )
plot.ts(series_to_check_ts, xlab = "Time",
        main = "S&P 500 Real Prices" )

# ------------------------------------------------------------------------
#           Estimate AR(1) model
# ------------------------------------------------------------------------

# Method A)

RP <- series_to_check_zoo
RP_0to1 <- embed(RP, 2)             # matrix containing contemporaneous and (once) lagged data

RP_t    <- RP_0to1[,1]              # contemporaneous series
RP_tm1  <- RP_0to1[,2]              # lagged series

est_ar1 <- lm(RP_t ~ 1 + RP_tm1)    # ols regression with constant
summary(est_ar1)                    # summary output of linear regression

# Method B) most convinient
# requires zoo object and dynlm package

est_ar1_dynlm <- dynlm(RP ~ L(RP))
summary(est_ar1_dynlm)

# Method C)
p = 1
ar.ols(series_to_check_ts, order.max = p,
       aic = FALSE, deman = FALSE, intercept = TRUE)
```

```
# Method D)
series_to_check_ar2_lm <- lm(y ~ y.l1 + y.l2, data =
                                ts.intersect(y = series_to_check_ts,
                                             y.l1 = lag(series_to_check_ts,-1),
                                             y.l2 = lag(series_to_check_ts,-2)))



#============================ End ================================================
```

Listing A.1: .././R_code/AOE_Ch1–4_Shiller_AR1.R

A.2. Generate AR(1) trajectories

Used in section 2.1.1.

R program:

```
##############################################################################
#                      AOE_Ch_2-1_AR1_Traj.R
##############################################################################
#
# Creates a plot with R realisations of an AR(1) process, plot them, and compute
# the ensemble means.
# It allows
```

```
#     - to choose between deterministic or random presample values and
#     - to add a random number to each time series value to create non-ergodic
#       time series.
# German source: 12_1_Traj_RW.R of Methoden der Ökonometrie
#
# Last change: 2021_04_22, 2023_04_12 (translation to English), RT

save.pdf <- 0      # 1 = create and save PDF of plots, 0 = do not

# Parameters of the DGPs and for the Monte-Carlo simulation
n       <- 200      # number of observations

alpha_0 <- 0.2      # constant of AR(1) process
alpha_1 <- 0.9      # AR parameter of AR(1) process
                            # 0:     white noise
                            # 1:     random walk
                            # 0 < |alpha| < 1: stationary process
y_0_type <- "random"  # "null" = use zero, "determ" = value chosen by y_0_value,
                    # "random" = normally distributed with mean y_0_value and sd = 1
y_0_value<- -2      # if y_0_type = "deterministic": presample value
                    # if y_0_type = "random": mean of for y_0 ~ N(y_0_value, 1)

R       <- 1000     # number of trajectories /  realisations

var_z   <- 0        # variance of z to illustrate non-ergodicity:
                            # 0 => ergodic
                            # >0 => non-ergodic
# Parameters for plots
lwd     <- 3
cexmu   <- 2
```

```
set.seed(42)       # seed value

# Initialize output matrices
u       <- matrix(rnorm(n*R), nrow = n) # generate Gaussian white noise
y       <- matrix(NA, nrow = n, ncol=R) # initialize matrix to contain time series
                                        #   using NAs
z       <- rnorm(R) * var_z             # Draw random number which is the same for all t

# Initialize presample values
if (y_0_type == "null") {
    y_0 = rep(0, R)
} else if (y_0_type == "deterministic") {
    y_0 = rep(y_0_value, R)
} else if (y_0_type == "random") {
    y_0 = rnorm(R, mean = y_0_value, sd = 1)
}

# Create all R trajectories of the AR(1) process
for (i in 1:R) y[,i] <- filter(u[,i] + alpha_0, alpha_1,
                               method = "recursive", init = y_0[i])

# Simuated ensemble mean
(y_means_ensemble <- rowMeans(y) + mean(z))

# Plot of all R time series showing ensemble means
if (save.pdf) pdf("Traj_AR1_points.pdf")
# first trajectory
plot(y[,1] + z[1], cex.lab = cexmu, cex.axis = cexmu, lwd = lwd,
        ylim = c(min(y+min(z)), max(y) + max(z)),
```

```
        ylab = expression(y[t]), xlab = "t", main =paste(R, "Time Series with their ensemble means"))
# 2nd to R-th trajectories
for (i in 2:R) points(y[,i] + z[i], col = i, lwd = lwd)
    # plot the simulated ensemble mean
    lines(y_means_ensemble, col = "black", lwd = lwd)
if (save.pdf) dev.off()

# Plot of all R time series as trajectories
if (save.pdf) pdf("Traj_AR1_lines.pdf")
    # first trajectory
plot(y[,1] + z[1], cex.lab = cexmu, cex.axis = cexmu, lwd = lwd,
     type = "l", ylim = c(min(y), max(y)),
     ylab = expression(x[t]), xlab = "t")
    # 2nd to R-th trajectories
for (i in 2:R) lines(y[,i] + z[i], col = i, lwd = lwd)
    # plot the simulated ensemble mean
    lines(y_means_ensemble, col = "black", lwd = 2*lwd)
if (save.pdf) dev.off()
# ======================== End ===============================================
```

Listing A.2: .././R_code/AOE_Ch2–1_AR1_Traj.R

A.3. Generate ARMA(1,1) trajectories

Used in sections 2.2.1 and 2.3.

R program:

```
##############################################################################
#                             AOE_Ch2-3_ARMA11_SIM.R
##############################################################################
# Program for generating one realisation of an ARMA(1,1) process.
# created by: RT, 2020_05_09, 2023_04_13 correection and arima.sim added

graphics.off()      # close all graphic windows

# Determine parameters of ARMA(1,1) model and of simulation study
set.seed(42)        # random seed
n       <- 500      # sample size
n_burn  <- 100      # burn-in phase

alpha_0 <- 0
alpha_1 <- 0.8      # parameter for autoregressive polynomial for AR(1)
m_1     <- 0.9      # parameter for moving average polynomial for MA(1)
sigma   <- 1        # standard deviation of the innovations
y0      <- 0        # presample value for of AR(1) process

# First method: for loop
u   <- rnorm(n + n_burn, mean = 0, sd = sigma) # generate one time series of innovations
y   <- rep(1, n + n_burn) *  y0              # initialize time series of y
for (t in (2:(n + n_burn)))    # compute y_t via iteration
{
    y[t] <- alpha_0 + y[t-1] * alpha_1 + u[t]  + m_1 * u[t-1]
}
y   <- y[(n_burn + 1):(n + n_burn)]

plot(y, type = "l", main = "Realisation of an ARMA(1,1) process", xlab = "Time")
```

```
# Second method: arima.sim()
# using arima.sim()
yy <- arima.sim(model = list( ar = c(alpha_1), ma = c(m_1)),
                n = n, innov = u[(n_burn + 1):(n_burn + n)], n.start = n_burn, start.innov = u[1:n_burn])
# plot(yy, type = "l", main = "Realisation of an ARMA(1,1) process", xlab = "Time")
lines(yy, col = "blue")

# Third method: filter():
e <- filter(u, c(1, m_1), method = "convolution", sides = 1)
yyy <- filter(e[2:length(e)], c(alpha_1), method = "recursive") # apply AR filter
yyy <- yyy[(length(yyy)-n+1): length(yyy)]

lines(yyy, col = "red")

head(cbind(y, yy, yyy))
tail(cbind(y,yy, yyy))


# ======================== End ================================================
```

Listing A.3: .././R_code/AOE_Ch2–3_ARMA11_SIM.R

A.4. Useful commands for ARMA(p, q) processes

Used in section 2.3 and before.

R program:

```
##############################################################################
#                         AOE_Ch2-3_ARMA_commands.R
##############################################################################
#
# Examples for useful commands for AR, MA and ARMA processes
# 2022_05_12, RT

# The commands are illustrated for a stationary AR(2) process with parameters
alpha_0 <- 2
alpha_1 <- 0.5
alpha_2 <- 0.3
sigma_u <- 2

# Compute infinite moving average representation of a stationary AR(2) process
# command ARMAtoMA - get help by ?ARMAtoMA
AR2_rep_MA <- ARMAtoMA(ar = c(alpha_1, alpha_2),  ma = 0, lag.max = 20 )
AR2_rep_MA

# Compute autocorrelation function of a stationary AR(2) process
AR2_corr <- ARMAacf(ar = c(alpha_1, alpha_2), ma = 0, lag.max = 20 )
AR2_corr

# Compute partial autocorrelation function of a stationary AR(2) process
AR2_pacf <- ARMAacf(ar = c(alpha_1, alpha_2), ma = 0, lag.max = 20, pacf = TRUE )
AR2_pacf

# Simulate / generate realzation of the AR(2) process with Gaussian white noise and 100 observations
# Alternatives to simulations: the filter command or a for loop as in AOE_Ch2_ARMA11_SIM
set.seed(131411)
```

```
y <- arima.sim(model = list(ar = c(alpha_1, alpha_2), ma = 0), n = 100, rand.gen = rnorm, sd = sigma_u)
plot(y)
```

Listing A.4: ../R_code/AOE_Ch2–3_ARMA_commands.R

A.5. Generate finite-sample distribution of OLS estimator for a stationary AR(1) process

Used in section 2.5.1

R program:

```
##############################################################################
#                          AOE_Ch2-5_bias_skewness_ar1.R
##############################################################################
# Simulate 10000 replications of an AR(1) process with alpha and
# n observations to show how the density of the OLS estimator is skewed and
# biased given a small sample size.

alpha <- 0.8    # AR parameter
n <- 30         # sample size
n_burn <- 120   # burn-in observations

set.seed(424)   # seed value

print_pdf <- 0  # 1 = save plot as PDF, 0 = no
```

```
# Generate all AR realisations at once
Y <- replicate(10000, filter( rnorm(n + n_burn), alpha, "recursive"))

# Extract sample
Y <- Y[(n_burn + 1):(n_burn + n),]

# Estimate alpha by ols
alpha_hat <- apply(Y, 2, function(x) ar.ols(x,
                            aic = FALSE, order.max = 1, demean = FALSE)$ar)

# Compute densities and highest value of density
dens <- density(alpha_hat)
dens.max <- max(dens$y)

# Produce plots of densities
if (print_pdf) pdf(file = "Density_OLS_AR1_08.pdf", width = 16, height = 9)
plot(dens, xlim = c(0, 1.1), ylim = c(0, dens.max),
     main = "Density of OLS estimator (true alpha = 0.8) and normal density",
     xlab = "alpha_hat", bty = "l")
x = seq(0, 1.1, length = 2000)
y = dnorm(x, mean = mean(alpha_hat), sd=sd(alpha_hat))
points(x, y, type = "l", lwd = 2, col = "red")
legend("topright", lty = 1, lwd = 3, bty = "n",
       legend = c("Finite-sample distribution", "Normal distribution" ),
       cex = 1.5, col = c(1,2))
text(x = 0, y = dens.max, labels = paste("Sample Size = ", n,
                                         "\nMean = ", mean(alpha_hat),
                                         "\nVariance = ", var(alpha_hat),
                                         sep = ""), adj = c(0,1), cex = 1.5)
```

```
if (print_pdf) dev.off()
# ======================== End ============================================
```

Listing A.5: ../../R_code/AOE_Ch2–5_bias_skewness_AR1.R

A.6. Generate Dickey-Fuller distribution

Used in section 2.5.1.

R program:

```
##############################################################################
#                   AOE_Ch2-5_df_distribution
##############################################################################
# Finite-sample distribution of the OLS estimator for an AR1 process
# in case of a random walk vs. normal distribution.
# Simulate 10000 replications of an AR(1) process with alpha = 1 and
# 100 observations.
# 2023_04_12, RT

n <- 100        # sample size
                # generate all AR(1) realisation at once
set.seed(424)

print_pdf <- 0  # 1 = save plot as PDF, 0 = no

Y <- replicate(10000, filter( rnorm(n), 1, "recursive"))
```

```
                # Estimate alpha by ols
alpha_hat <- apply(Y, 2, function(x) ar.ols(x, aic = FALSE,
                                    order.max = 1, demean = FALSE)$ar)


                # Compute density and highest value of density
alpha_hat.std <- (alpha_hat - 1 ) * n
dens <- density(alpha_hat.std)
dens.max <- max(dens$y)

if (print_pdf) pdf(file = "Density_OLS_AR1_1.pdf", width = 16, height = 9)
plot(dens, xlim = c(-10, 10), ylim = c(0, dens.max),
     main = paste0("Density of standardized of ", n,
                   " *(hat alpha - 1) and normal density"),
     xlab = paste0(n, " *(hat alpha - 1)"), bty = "l")
x = seq(-10, +10, length = 2000)
y = dnorm(x, mean = 0, sd = sd(alpha_hat.std))
points(x, y, type = "l", lwd = 2, col = "red")
legend("topright", lty = 1, lwd = 3, bty = "n",
       legend = c("standardized and scaled \nOLS estimator",
                  "Normal distribution" ), cex = 1.5, col = c(1,2))
text(x = -10, y = dens.max, labels = paste("True alpha = 1 \nSample size = ", n,
                                           "\nMean of hat alpha = ", mean(alpha_hat),
                                           "\nVariance of hat alpha = ", var(alpha_hat),
                                           sep = ""),
                        adj = c(0,1), cex = 1.5)
if (print_pdf) dev.off()
# ========================== End ================================================
```

Listing A.6: ../../R_code/AOE_Ch2–5_df_distribution.R

A.7. Power of ADF test

Used in section 4.1.2.

R program:

```
# ======================== AOE_Ch4_MC_DF_H_0_H_1.R ==============================
# Program for conducting Monte Carlo simulation to obtain densities of
# t statistic that is used in the Augmented Dickey-Fuller test where
# a AR(2) process is used as DGP:
#       Under H_0 the AR(2) process has a unit root.
#       Under H_1 the AR(2) process is stable where a specific DGP is used,
#                                 see below.
# No deterministic terms are included.
# If one chooses T_obs, the sample size, large, then the Dickey-Fuller
# distribution under H_0 is well approximated. This can be seen by
# comparing the critical value obtained from the Monte Carlo distribution
# and the asymptotic critical value delivered by the ur.df() command.
# Furthermore, the size under H_0 and the power under the specific H_1
# is reported.
#
# Note: under H_0, the density is always skewed while
#       under H_1 it is (asymptotically) symmetric as it is asymptotically
```

```
#                   normal
# created by: RT, 2022_06_23, 2023_04_12 (added: library(urca))

# rm(list = ls()) # clear workspace
dev.off()   # close all open graphic windows

if (!require(urca)) install.packages("urca")
library(urca)

print_pdf <- 1              # 1 = save plot as PDF, 0 = no

set.seed(26062011)
ar2_H0      <- c(0.6, 0.4)  # AR(2) parameters under H_0
ar2_H1      <- c(0.6, 0.3)  # AR(2) parameters under H_1
sigma_sq    <- 5            # white noise variance
T_obs       <- 50           # number of observations

                            # specify ADF equation
df_type     <- "none"       # kind of deterministic terms
                            # "none", "drift", "trend"
df_lags     <- 1            # lag order in ADF equation
                            # therefore an AR(df_lags + 1)
                            # is modelled for the level series

reps        <- 10000        # number of replications in Monte Carlo simulation

# ---- distribution of t statistic under H_0:
#      generate  AR(2) with unit root and without drift

        # generate reps realizations of AR(2) under H_0
```

```
Y.obs.H0    <- replicate(reps,
                      filter( rnorm(T_obs, mean = 0, sd = sqrt( sigma_sq )),
                           filter = ar2_H0, method = "recursive" ))

        # compute t statistic for each realization using correct lag specification
        # ur.df estimates ADF test equation and returns t statistic for y_{t-1}
ADF.A.obs.H0 <- apply(Y.obs.H0, 2, function(x) ur.df(x, type = df_type,
                                        lags = df_lags, selectlags = "Fixed")@teststat[1])
        # obtain critical value for given sample size
crit_val_H0_T   <- quantile(ADF.A.obs.H0, probs = c(0.01, 0.05, 0.1))
        # obtain asymptotic critical values
crit_val_H0_as  <- ur.df(Y.obs.H0[ ,1], type = df_type, lags = df_lags,
                     selectlags = "Fixed")@cval[1,]

# ---- distribution of t statistic under one possible H_1:
#      generate stable AR(2)
        # generate reps realizations of AR(2) under H_1
Y.obs.H1    <- replicate(reps,
                      filter( rnorm(T_obs, mean = 0, sd = sqrt( sigma_sq )),
                           filter = ar2_H1, method = "recursive" ))

        # compute t statistic for each realization using correct lag specification
        # ur.df estimates ADF test equation and returns t statistic for y_{t-1}
ADF.A.obs.H1 <- apply(Y.obs.H1, 2, function(x) ur.df(x, type = df_type,
                                        lags = df_lags, selectlags = "Fixed")@teststat[1])

# plot densities based on reps realizations under H_0 and specific coice of H_1
    # use default density estimator R
ADF.A.obs.H0.den <- density(ADF.A.obs.H0)
ADF.A.obs.H1.den <- density(ADF.A.obs.H1)
```

```
    # obtain ranges of x and density values
xlim <- range(ADF.A.obs.H0.den$x, ADF.A.obs.H1.den$x)
ylim <- range(ADF.A.obs.H0.den$y, ADF.A.obs.H1.den$y)
    # plot densities
if (print_pdf) pdf(paste0("ADF_dens_H_0_H_1_T", T_obs,"_", df_type, ".pdf"))
plot(ADF.A.obs.H0.den, xlim = xlim, ylim = ylim,
     main = paste0("DGP: AR(2) processes, sample size: ", T_obs),
     ylab = "Densities",
     xlab = paste0("t statistic of ADF test, type = ", df_type,
                   ", lag order in levels = ", df_lags + 1))
lines(density(ADF.A.obs.H1), col = "blue")
abline(v = 0)
    # draw vertical line without arrow heads to indicate
    # finite sample 5%-critical value
arrows(x0 = crit_val_H0_T[2], y0 = -1, x1 = crit_val_H0_T[2], y1 = 0.6*ylim[2],
       length = 0, col = "green")
text(crit_val_H0_T[2], 0.7*ylim[2],
     labels = paste0("crit. value for T =  ", T_obs), col = "green")

    # asymptotic 5%-critical value
arrows(x0 = crit_val_H0_as[2], y0 = -1, x1 = crit_val_H0_as[2], y1 = 0.6*ylim[2],
       length = 0, col = "red")
text(crit_val_H0_as[2], 0.5*ylim[2],
     labels = "asym. crit. value", col = "red")

    # power under specific H_1
text(xlim[1], 0.3*ylim[2], pos = 4,
     labels = paste0("Power for specific H_1 (asym.crit.val.): ", mean(ADF.A.obs.H1 < crit_val_H0_as[2])
     ))
    # legend with AR(2) parameters
```

```
legend("topright", legend = c(bquote(under ~ H[0]: ~ alpha[1] ~ "=" ~ .(ar2_H0[1]) ~
                                      "," ~ alpha[2] ~ "=" ~ .(ar2_H0[2])),
                              bquote(under ~ H[1]: ~ alpha[1] ~ "=" ~ .(ar2_H1[1]) ~
                                      "," ~ alpha[2] ~ "=" ~ .(ar2_H1[2]))),
                            lty = 1,
       col = c("black", "blue"),
       bty = "n")
if (print_pdf) dev.off()

# 5% critical value
crit_val_H0_as[2]
# compute size of ADF test using asymptotic critical value
mean(ADF.A.obs.H0 < crit_val_H0_as[2])
# compute power for given DGP under H_1
mean(ADF.A.obs.H1 < crit_val_H0_as[2])
```

Listing A.7: .././R_code/AOE_Ch4_MC_DF_H_0_H_1.R

A.8. AR(1)-GARCH(1,1) process: generation and estimation

Used in section 5.1.

R program:

```
##############################################################################
```

```
#                        AOE_Ch5-8_R-Code_AR_GARCH_block_sim.R
##############################################################################
# Simulation of one realizsation of
# AR(1)-(G)ARCH processes and one-step and two-step estimation
# to compare standard errors for conditional mean estimation
# RT, 2022_07_10

library(dynlm)
library(rugarch)

set.seed(1414132)
T       <- 1e3  # number of observations for 1e6 it takes quite long
ar.1    <- 0.3  # AR_1 parameter
gamma.0 <- 2
gamma.1 <- 0.3
beta.1  <- 0.6
m       <- 1
n       <- 1
plot.yes <-0

### Generate GARCH errors

# for ARCH processes fourth moment exists up to gamma.1 < sqrt(1/3) = 0.577
# for GARCH processes fourth moment exists up to e.g. gamma.1 <= 0.3,
# beta.1 <= 0.6
# unconditional variance (under the assumption of stationarity)
sigma.sq <- gamma.0 / (1 - gamma.1 - beta.1)
sigma.t.sq <- numeric(T)
xi <- rnorm(T)
u  <- numeric(T)
```

```
# starting value of process u (variance set to unconditional variance)
u.0 <- sqrt( sigma.sq ) * rnorm(1)
# computing conditional variance and realization of u for period 1
sigma.t.sq[1] <- gamma.0 + gamma.1 * u.0^2 + beta.1 * sigma.sq
u[1]          <- sqrt( sigma.t.sq[1] ) * xi[1]
# computing conditional variance and u for all following periods
for(t in 2:T){
    sigma.t.sq[t] <- gamma.0 + gamma.1 * u[t-1]^2 + beta.1 * sigma.t.sq[t-1]
    u[t]          <- sqrt( sigma.t.sq[t] ) * xi[t]
}
# plot u and its conditional variance
# par( mfrow =c(2,1))
if (plot.yes) ts.plot(u)
if (plot.yes) ts.plot(sigma.t.sq)

# theoretical fourth moment
if (plot.yes) ts.plot(u^4)
( m.4 <- (3 * gamma.0^2 * (1 + gamma.1 + beta.1)) /
        ((1 - gamma.1 - beta.1) * (1 - beta.1^2 - 2 * gamma.1 * beta.1 - 3 * gamma.1^2)) )
# compute fourth moment by mean estimator
# this, however, does not seem to work well if fourth moment is very large.
mean(u^4)

### Generate AR1-GARCH process

y <- filter(u, filter = ar.1, method = "recursive")

## Two-step estimation

# 1. OLS estimator of AR(1) model
```

```
if (plot.yes) ts.plot(y)
y_ar1_lm <- dynlm(y ~ L(y))
summary(y_ar1_lm)

# 2 GARCH estimation based on residuals
u_hat <- residuals(y_ar1_lm)

mean  <- FALSE # since residuals are used and constant in AR(1) model

res_ugarch_spec <- ugarchspec(variance.model = list(garchOrder = c(m,n)),
                              mean.model = list(armaOrder = c(0,0),
                                                include.mean = mean ))

res_ugarch_fit <- ugarchfit(spec = res_ugarch_spec, # specification
                            data = u_hat, # time series
                            solver.control=list(trace = 0),
                            solver = "hybrid")
res_ugarch_fit

## One-step estimation using ML
# specify ARMA(p,q)-GARCH(m,n) model for series_to_check assuming normality
print("Estimate AR(1)-GARCH model using normal errors")
mean  <- TRUE # since residuals are used

ugarch_spec <- ugarchspec(variance.model = list(garchOrder = c(m,n)),
                          mean.model = list(armaOrder = c(1,0),
                                            include.mean = mean ))

ugarch_fit <- ugarchfit(spec = ugarch_spec, # specification
                        data = y, # time series
```

```
                        solver.control=list(trace = 0),
                        solver = "hybrid")
ugarch_fit

### Comparison
# one-step
ugarch_fit@fit$coef
ugarch_fit@fit$se.coef

# two-step
summary(y_ar1_lm)

res_ugarch_fit@fit$coef
res_ugarch_fit@fit$se.coef


### ARCH-LM test
u_hat_sq <- u_hat^2
u_hat_sq_lm <- dynlm(u_hat_sq ~ L(u_hat_sq, 1:5))
summary(u_hat_sq_lm)
```

Listing A.8: .././R_code/AOE_Ch5–8_R–Code_AR_GARCH_block_sim.R

A.9. Empirical example

Used in sections 2.5.1, 5.8 and throughout the course.

R program:

```
################################################################################
#                     Series_to_check_complete_2023.R
################################################################################
# R program accompanying the course "Applied Financial Econometrics"
# at the Universität Regensburg
# Task: Analysis of univariate time series data using two examples:
#   - monthly data of the S&P 500 index of Robert Shiller
#   - daily stock prices taken from Yahoo - Finance
# and possible transformations.
# Modelling steps:
#     1) Unit root testing and specifying deterministic trends,
#     2) Specifying and estimating the conditional mean function
#     3) Residual diagnostics
#     4) Specifying and estimating the conditional volatility function if needed
#     5) Diagnostics of standardized residuals if needed
#     6) Out-of-sample predictions
#
# written by Rolf Tschernig (RT) with parts by Stefan Rameseder (SR)
# Version: 2017-08-02, 2017-08-04 (kurtosis() corrected in lines 406 and 431),
#          2018-04-19, 2020-05-09 (corrections in text), 2021-04-15 (Shiller data as of 2021),
#          2022-04-28 (Shiller data as of 2022)
#          2022-06-19 predictions corrected: ar() replaced by ar.ols() and n_pred adjusted
#          2023-04-13 Shiller data updated, AR lags set to 5
#
# Notes:
# Mac-User have to install XQuartz if the package rugarch is used, see
#   https://ftp.gwdg.de/pub/misc/cran/
# and
#   https://www.xquartz.org/
```

```
################################################################################
#         Function SelectCritEviews
################################################################################
# Function to compute model selection criteria like in EViews
# RT, 2011_01_26, 2017_05_18

SelectCritEviews <- function(model)
{
  n   <- length(model$residuals)
  k   <- length(model$coefficients)
  fitmeasure  <- -2*logLik(model)/n

  aic <- fitmeasure + k * 2/n
  hq  <- fitmeasure + k * 2*log(log(n))/n
  sc  <- fitmeasure + k * log(n)/n
  sellist <- list(aic=aic[1],hq=hq[1],sc=sc[1])
  return(t(sellist))
}

################################################################################
#               End Function SelectCritEviews
################################################################################

################################################################################
#       Function ARCH.LM
################################################################################
# Function to compute ARCH-LM Test
# SR, RT, 2017_07_26
```

```
ARCH.LM <- function(residuals,q_ARCH){
  res_sq    <- ts(residuals)^2
  r_squared <- summary(dynlm(res_sq ~ 1 + L(res_sq, 1:q_ARCH)))$r.squared
  test_stat <- length(res_sq) * r_squared # T * R^2 which is chi^2(q_ARCH) distributed
  p_value   <- 1 - pchisq(test_stat, df=q_ARCH)
  print(paste0("ARCH-LM test statistic: ",round(test_stat,4), ", p value: ", p_value))
  result_list <- list(test_stat=test_stat, p_value=p_value)
  return(result_list)
}

################################################################################
#               End Function ARCH.LM
################################################################################

################################################################################
#                   Main program
################################################################################


# ------------------------------------------------------------------------
#   Specify parameters for analysis
# ------------------------------------------------------------------------

                          # set working directory
#setwd("....")      # fill in .... or use in RStudio:
                    # Session -> Set Working Directory

data_source   <- "yahoo"  # must be: "Shiller", "yahoo"
                          # Notes:
                          # for "yahoo" you need an internet connection;
```

```
                          # for "Shiller" you need to set the working
                          # directoryto the directory where the Excel data are
inst_yahoo    <- "bmw.de"   # see finance.yahoo.de
start_yahoo   <- "2001-01-13" # date on which series starts
series_yahoo  <- "Close"  # name of series to analyse; note Adj.Close not always avaialble

Shiller_series<- "RealPrice"
                          # available are: SPCompPrice", "Dividend", "Earnings",
                          # "CPI", "DateFraction", "LongInterest",
                          # "RealPrice", "RealDividend",
                          # "RealTotalReturnPrice", "RealEarnings",
                          # "RealScaledEarnings", "P/E10"

transformdata = "log"   # must be: level, diff, log, or difflog

# used in ur.df (urca package):
adf_determ    <- "trend"    # choose: "none", "drift", "trend"
adf_maxlags   <- 10         # maximal number of lags in lag selction in ADF testing
adf_selcrit   <- "BIC"      # lag selection criterion: "AIC", "BIC"
                          # if "Fixed" adf_maxlags determines AR order of test equation

p             <- 5          # AR order
q             <- 0          # MA order - currently only 0 allowed
                          # except for ARMA(p,q)-GARCH(m,n) model

ACF_num_lags  <- 35         # number of lags for Box-Pierce / Ljung/Box statistic
q_ARCH        <- 5          # order for ARCH-LM test

m             <- 1          # ARCH order
n             <- 1          # GARCH order
```

```
# parameters for prediction

# for Shiller data
  # subperiod for estimation
subperest_beg <- NULL       # used for ts object with window comannd, e.g. c(2010,1)
subperest_end <- c(2015,11)  # used for ts object with window comannd, e.g. c(2017,1)
  # subperiod for prediction

subperfor_end <- c(2023,2)

# for yahoo data
n_est_ratio   <- 0.99        # ratio of number of observations used for estimation
                             # for out-of-sample prediction, 0<. < 1
h_pred        <- 12          # number of horizons for prediction




# ------------------------------------------------------------------------
#   load and check data, create zoo and time series objects
# ------------------------------------------------------------------------

if (!require(zoo)) install.packages("zoo") # requires library zoo
library(zoo)
if (!require(tseries)) install.packages("tseries") # requires library zoo
library(tseries)
# ------------ Shiller data
if (data_source=="Shiller"){
```

```
  # use pacakge "readxl" that does not need Java
  # it returns a data format "tibble" that the command
  # "data.frame" turns into a dataframe

if (!require(readxl)) install.packages("readxl")
library(readxl)

  ie_data <- data.frame(
      read_excel(path="Data/ie_data_2023_04_10.xls",
                 range="Data!B9:M1834",
                 col_names=c("SPCompPrice", "Dividend", "Earnings",
                             "CPI", "DateFraction", "LongInterest",
                             "RealPrice", "RealDividend",
                             "RealTotalReturnPrice", "RealEarnings",
                             "RealScaledEarnings", "P/E10")))

  # ie_data <- data.frame(
  #     read_excel(path="Data/ie_data_2021_04_12.xls",
  #                range="Data!B9:M1812",
  #                col_names=c("SPCompPrice", "Dividend", "Earnings",
  #                            "CPI", "DateFraction", "LongInterest",
  #                            "RealPrice", "RealDividend",
  #                            "RealTotalReturnPrice", "RealEarnings",
  #                            "RealScaledEarnings", "P/E10")))

#ie_data                 <- data.frame(read_excel(path="ie_data_2017_05_04.xls",
#                                 range="Data!H9:K1764",col_names=FALSE))
#names(ie_data)           <- c("REAL_PRICE","REAL_DIV","REAL_EARN","P/E10")
head(ie_data)
tail(ie_data)
```

```
  #  create ts object
  #     Note that ts cannot handle missing values. This requires library zoo
  ie_data_ts              <- ts(ie_data, start = c(1871, 1), frequency = 12)
  ie_data_zoo             <- zoo(ie_data_ts)
  series_to_check_all_ts  <- window(ie_data_ts[,Shiller_series], start=NULL, end=NULL)
  series_to_check_all_zoo <- window(ie_data_zoo[,Shiller_series, drop=FALSE],
                                    start=NULL, end=NULL)
                                    # drop=FALSE keeps series_to_check_all_zoo as
                                    # a matrix object and stores therefore also the
                                    # the series name

  # write data in ascii file that can be read by JMulTi
  write("/* Shiller S&P 500 data, downloaded from http://www.econ.yale.edu/~shiller/ */", file="Data/data_
   Shiller_jmulti.dat")
  write("<1871 Q1>", file="data_Shiller_jmulti.dat", append=TRUE)
  write(Shiller_series, file="data_Shiller_jmulti.dat", append=TRUE)
  write(as.vector(series_to_check_all_zoo), file="data_Shiller_jmulti.dat", ncolumns=1, append=TRUE )

} else if (data_source=="yahoo"){

# ------------ daily stock price data
  # if from yahoo: one may use get.hist.quote from library(tseries)
  stock_data_zoo <- get.hist.quote(instrument = inst_yahoo,  start = start_yahoo) # note help for choice of
   instrument

  # alternatively via downloaded xls-File
    ## stock_data      <- data.frame(read_excel(path="bmw.de.xlsx",range="Tabelle1!A1:F4095",na="null",col_
   names=TRUE))
                                                  # NA are coded as null in the excel sheet
```

```
    # create zoo object; requires to change date format from POSIX to Date such that
    # extraction of subperiods in prediction section works
    ## stock_data_zoo  <- zoo(x=stock_data[,-1],order.by=as.Date(stock_data[,1]))

  head(stock_data_zoo)
  tail(stock_data_zoo)

  stock_data_naomit_zoo     <- na.omit(stock_data_zoo)       # eliminate all rows which contain NAs

  # select univariate series that does not contain missing values
  series_to_check_all_zoo   <- stock_data_naomit_zoo[,series_yahoo]

  # write data in ascii file that can be read by JMulTi
  write("/* daily stock data from yahoo finance ", file="data_yahoo_jmulti.dat")
  write(paste0("Beginn: ", time(series_to_check_all_zoo)[1],
               ", End: ", time(series_to_check_all_zoo)[length(series_to_check_all_zoo)], " */"),
        file="data_yahoo_jmulti.dat", append=TRUE)
  write("<1>", file="data_yahoo_jmulti.dat", append=TRUE)
  write("Close", file="data_yahoo_jmulti.dat", append=TRUE)
  write(as.vector(series_to_check_all_zoo), file="data_yahoo_jmulti.dat", ncolumns=1, append=TRUE )

  # compute ts object that is occasionally needed
  series_to_check_all_ts    <- ts(series_to_check_all_zoo)
}
print("Check data before your analysis\n")
head(series_to_check_all_zoo)
head(series_to_check_all_ts)
tail(series_to_check_all_zoo)
tail(series_to_check_all_ts)
```

```
# transform data if desired

if (transformdata=="level"){
  series_to_check_zoo         <- series_to_check_all_zoo
  series_to_check_ts          <- series_to_check_all_ts
} else if (transformdata=="diff"){
  series_to_check_zoo         <- diff(series_to_check_all_zoo,lag=1)
  series_to_check_ts          <- diff(series_to_check_all_ts,lag=1)
} else if (transformdata=="log"){
  series_to_check_zoo         <- log(series_to_check_all_zoo)
  series_to_check_ts          <- log(series_to_check_all_ts)
} else if (transformdata=="difflog"){
  series_to_check_zoo         <- diff(log(series_to_check_all_zoo),lag=1)
  series_to_check_ts          <- diff(log(series_to_check_all_ts),lag=1)
} else {
  print("transformdata has to be level, diff, log, or difflog")

}
nobs <- length(series_to_check_zoo)

# ------------------------------------------------------------------------
#           Plot time series
# ------------------------------------------------------------------------

plot.zoo(series_to_check_zoo, xlab = "Time",
         ylab = paste(names(series_to_check_zoo),
                      ", data transformation: ", transformdata),
         main = paste0("Data source: ",data_source) )
plot.ts(series_to_check_ts, xlab = "Time",
        main = paste0("Data source: ",data_source) )
```

```
# ------------------------------------------------------------------------
#           1.  Unit Root Tests
# ------------------------------------------------------------------------

# There are several R packages for conducting unit root tests, among them:
# urca, tseries, aTSA
# In the following the ADF test is carried out with all three packages as they
# differ with respect to the output they provide.
# All three packages also contain commands for the Phillips-Perron test
# (urca::ur.pp, tseries::pp.test, aTSA::pp.test)
# and the Kwiatkowski-Phillips-Schmidt-Shin test
# (urca::ur.kpss, tseries::kpss.test, aTSA:: kpss.test )
# but the urca package also contains further unit root tests

# ADF test
#   1) using package urca
if (!require(urca)) install.packages("urca")
library(urca)
# note that for all estimations only n-lags (=lag.max) observations are used.
# Thus, changing lags typically changes the test statistic due to changing DF.
series_to_check.adf     <- ur.df(series_to_check_zoo, type = adf_determ,
                                 lags = adf_maxlags, selectlags = adf_selcrit)
summary(series_to_check.adf)
length(series_to_check.adf@testreg$aliased) # number of estimated parameters in test reqression


#   2) using package tseries
# provides p-value, only available for det+trend!
# alternative = "stationary" corresponds to alpha<1,
```

```
# "explosive" corresponds to H_1: alpha>1
# also note: if test statistics differs from ur.df, then this is because a
# differing number of observations used in the estimation due to lag selection
# in the ur.df case.
if (adf_determ=="trend"){
    # compute p value using different package
  print("Note that test statistic may slightly differ from the one that includes lag selection
      due to different number of initial values used in test regression")
  print(summary(ur.df(series_to_check_zoo, type = adf_determ, lags = p-1, selectlags = "Fixed")))
  tseries::adf.test(series_to_check_zoo, alternative="stationary",k=p-1)
}

#    3) using package aTSA
# includes adf.test() which provides p-values
# for all combinations of lag and trend specifications up to
# maximum lag either computed by default or given by the user
# added 2020_05_09, RT

if (!require(aTSA)) install.packages("aTSA")
library(aTSA)

aTSA::adf.test(as.vector(series_to_check_zoo)) #, nlag = adf_maxlags, output = TRUE)



# ------------------------------------------------------------------------
#             AR estimation with different methods
# ------------------------------------------------------------------------

# ------------------------ autoregressions with ts objects -----------------
```

```
#  A) with standard lm command with ts object and ts.intersect
#         ts.intersect does not work for zoo object

#     1) using lm command illustrated for an AR(2)
series_to_check_ar2_lm <- lm(y ~ y.l1 + y.l2, data =
                                ts.intersect(y = series_to_check_ts,
                                             y.l1 = lag(series_to_check_ts,-1),
                                             y.l2 = lag(series_to_check_ts,-2)))
summary(series_to_check_ar2_lm)
series_to_check_ar2_res <- series_to_check_ar2_lm$residuals
plot(series_to_check_ar2_res, type = "l", main = "Residuals of AR(2) model")

#   2) with ar or ar.ols or arma command - requires ts object,
#           zoo object does not work

#     AR(p)
(series_to_check_ar    <- ar.ols(series_to_check_ts, order.max = p,
                                aic = FALSE, deman = FALSE, intercept = TRUE))
sqrt(series_to_check_ar$var.pred)   # shows standard error of AR regression
sort(abs( polyroot(c(1, - series_to_check_ar$ar)) )) # check roots
series_to_check_ar_res  <- series_to_check_ar$resid
                                   # define residuals of autoregression

# ----------------------- autoregressions with zoo or ts objects -----------------

#  B) with dynlm library
#     Note (from help of dynlm):
#       d(x, k) is diff(x, lag = k) and L(x, k) is lag(x, lag = -k),
#       note the difference in sign.
```

```
#       The default for k is in both cases 1.
#       For L(), it can also be vector-valued, e.g., y ~ L(y, 1:4).

if (!require(dynlm)) install.packages("dynlm")
library(dynlm)

series_to_check_dynlm <- dynlm(series_to_check_zoo ~ L(series_to_check_zoo,1:p) )
summary(series_to_check_dynlm)


    # compute model selection criteria
    # equation numbers below refer to handout "Methoden der Ökonometrie"
extractAIC(series_to_check_dynlm)               # uses (10.4) multiplied by n
if (!require(MASS)) install.packages("MASS")    # required for stepAiC
stepAIC(series_to_check_dynlm)                  # as extractAIC
AIC(series_to_check_dynlm)                      # uses (10.10)
SelectCritEviews(series_to_check_dynlm)         # uses (10.4)

    # check stability of AR polynomial
abs( polyroot(c(1, - coef(series_to_check_dynlm)[2:(p+1)])) )

    # plot residuals
series_to_check_dynlm_res <- series_to_check_dynlm$residuals
plot(series_to_check_dynlm_res, xlab = "Time")

# ------------------------------------------------------------------------
#             Diagnostics of residuals
# ------------------------------------------------------------------------

# Residuals
```

```
    # plot standardized residuals
plot((series_to_check_dynlm_res - mean(series_to_check_dynlm_res)) /
      sd(series_to_check_dynlm_res) , xlab = "Time")
abline(h = 1.96, col = "gray", lty = 2)
abline(h = -1.96, col = "gray", lty = 2)

    # plot autocorrelation funcion
acf(ts(series_to_check_dynlm_res), type = "correlation",
    na.action = na.omit)    # acf() requires ts object,
                            # missing values are excluded

if (ACF_num_lags <= (p + q)){
    print("Number of lags for ACF tests has to be larger than (p+q)")
} else {
  # compute Box-Pierce statistic
    print(Box.test(series_to_check_dynlm_res, lag = ACF_num_lags, type = "Box-Pierce",
            fitdf = p+q))
        # print() needed since command Box.test within else condition

  # compute Ljung-Box statistics
    print(Box.test(series_to_check_dynlm_res, lag = ACF_num_lags, type = "Ljung-Box",
            fitdf = p+q))

  # compute LM test
    dynlm_res_lm_test <- dynlm(series_to_check_dynlm_res ~
                                  L(series_to_check_zoo, 1:5)
                                  + L(series_to_check_dynlm_res, 1:ACF_num_lags ))
    print(paste("LM statistic: ",
          LM_statistic <- summary(dynlm_res_lm_test)$r.squared * nobs) )
```

```
    print(paste("p-value: ",1-pchisq(LM_statistic, df = ACF_num_lags)) )
    print(paste("critical value at 5%: ", qchisq(0.95, df = ACF_num_lags)) )

  # compute Lomnicki-Jarque-Bera test, requires package moments
    if (!require(moments)) install.packages("moments")
    library(moments)
    jarque.test(na.omit(as.vector(series_to_check_dynlm_res)))


    # alternative package for using Jarque-Bera test
    # if (!require(fBasics)) install.packages("fBasics")
    # library(fBasics)
    # jarqueberaTest(na.omit(ts(series_to_check_dynlm_res)))
}
# RESET test
# Note resettest() of package lmtest only works with standard lm objects,
# with dynlm objects

# Squared residuals

  # plot acf of squared residuals
plot(series_to_check_dynlm_res^2, xlab = "Time")

  # plot autocorrelation funcion of squared residuals
acf(ts(series_to_check_dynlm_res)^2, na.action = na.omit)

  # ARCH-LM test
result_ARCH_LM <- ARCH.LM(series_to_check_dynlm_res, q_ARCH)

# ------------------------------------------------------------------------
```

```
#           Dynamic Regression Models
# ------------------------------------------------------------------------

# estimate time series with deterministic trend if level/log data
if ((transformdata == "level") || (transformdata == "log")){

    dyn_reg_mod_dynlm <- dynlm(series_to_check_zoo
                               ~ 1 + trend(series_to_check_zoo)
                               + L(series_to_check_zoo,1:p)
                                )
    summary(dyn_reg_mod_dynlm)

    # check stability of AR polynomial
    abs( polyroot(c(1, - coef(dyn_reg_mod_dynlm)[3:(p+2)])) )

    # plot residuals
    dyn_reg_mod_dynlm_res <- dyn_reg_mod_dynlm$residuals
    plot(dyn_reg_mod_dynlm_res, xlab = "Time")

    "how do you interpret the results? "
    "Take into account your results of the unit root tests."
}

# ------------------------------------------------------------------------
#           GARCH models
# ------------------------------------------------------------------------
# estimation methods in libraries:
# - rugarch (encompasses many different GARCH variants),
# - fGarch (more restricted, so code commented out)
```

```
# if (!require(fGarch)) install.packages("fGarch")
# library(fGarch)
#                                   # run GARCH(1,1) for residuals
# garchFit(~garch(1,1), data=na.omit(series_to_check_dynlm_res))
#
# # run AR(p)-GARCH(m,n) for levels
# AR_GARCH <- garchFit(AR2GARCH11 ~arma(2,0)+garch(1,1),data=series_to_check_zoo)
# res_stand_AR_GARCH <- AR_GARCH@residuals / sqrt(AR_GARCH@h.t)
# plot( res_stand_AR_GARCH ,type="l") # plot standardized residuals
#
# jarqueberaTest(res_stand_AR_GARCH)    # for using Jarque-Bera-Test
#
# plot(rnorm(length(res_stand_AR_GARCH)),type="l")
                                # plot standard normal noise for comparison

if (!require(rugarch)) install.packages("rugarch")
library(rugarch)

# specify ARMA(0,0)-GARCH(m,n) model for residuals

mean  <- FALSE # since residuals are used

res_ugarch_spec <- ugarchspec(variance.model = list(garchOrder = c(m,n)),
                          mean.model = list(armaOrder = c(0,0),
                                            include.mean = mean ))

res_ugarch_fit <- ugarchfit(spec = res_ugarch_spec, # specification
                        data = series_to_check_dynlm_res, # time series
                        solver.control=list(trace = 0),
                        solver = "hybrid")
```

```
res_ugarch_fit

# specify ARMA(p,q)-GARCH(m,n) model for series_to_check assuming normality
print("Estimate ARMA-GARCH model using normal errors")
mean  <- TRUE # since residuals are used

ugarch_spec <- ugarchspec(variance.model = list(garchOrder = c(m,n)),
                          mean.model = list(armaOrder = c(p,q),
                                            include.mean = mean ))

ugarch_fit <- ugarchfit(spec = ugarch_spec, # specification
                        data = series_to_check_zoo, # time series
                        solver.control=list(trace = 0),
                        solver = "hybrid")
ugarch_fit
GARCH_res_std   <- ugarch_fit@fit$z
                # identical to ugarch_fit@fit$residuals/ugarch_fit@fit$sigma
plot(GARCH_res_std, type = "l")
plot(density(GARCH_res_std))
plot.function(dnorm, from = -6, to = 6, col = "red")
lines(density(GARCH_res_std))
print( paste("Estimated kurtosis", kurtosis(GARCH_res_std)) )
                # kurtosis() of package moments


# specify ARMA(p,q)-GARCH(m,n) model for series_to_check assuming t-distributed errors
print("Estimate ARMA-GARCH model with t-distributed errors")
mean  <- TRUE # since residuals are used

tdist_ugarch_spec <- ugarchspec(variance.model = list(garchOrder = c(m,n)),
```

```
                          mean.model = list(armaOrder = c(p,q), include.mean = mean ),
                          distribution.model = "std")

tdist_ugarch_fit <- ugarchfit(spec = tdist_ugarch_spec, # specification
                         data = series_to_check_zoo, # time series
                         solver.control=list(trace = 0),
                         solver = "hybrid")
tdist_ugarch_fit

  # compute and plot standardized residuals
GARCH_tdist_res_std <- tdist_ugarch_fit@fit$residuals/tdist_ugarch_fit@fit$sigma
                    # identical to std_ugarch_fit@fit$z
plot(GARCH_tdist_res_std,type="l", xlab = "Time", main = "Standardized Residuals")
  # plot standard normal noise for comparison
plot(rnorm(length(GARCH_tdist_res_std)), type = "l",
     main = "Standardized Normal Variables")


  # compute kurtosis of standardized residuals
kap <- kurtosis(GARCH_tdist_res_std) # kurtosis() of package moments

  # compute kurtosis implied by t distribution using estimated deg. of freedom
shape <-  tdist_ugarch_fit@fit$coef[length(tdist_ugarch_fit@fit$coef)]
kap_t_dist <- 3*(shape-2)/(shape-4)

print(paste("Estimated kurtosis",kap))
print(paste0("Kurtosis implied by t distr.: ",kap_t_dist))
print(paste0("Est. degrees of freedom of t distr.: ",shape))

# ------------------------------------------------------------------------
```

```
#   Diagnostics of standardized GARCH residuals with assumed t distribution
# ------------------------------------------------------------------------

# Residuals
print("Check standardized residuals of ARMA-GARCH model with t-distributed errors:")
if (ACF_num_lags<=(p+q)){
    print("Number of lags for ACF tests has to be larger than (p+q)")
}

  # plot autocorrelation funcion
acf(ts(GARCH_tdist_res_std), na.action = na.omit)   # requires ts object,

  # compute Box-Pierce statistic
Box.test(GARCH_tdist_res_std, lag = ACF_num_lags, type = "Box-Pierce",
         fitdf = p+q)

  # Compute Ljung-Box statistics
Box.test(GARCH_tdist_res_std, lag = ACF_num_lags, type = "Ljung-Box", fitdf = p+q)

  # Lomnicki-Jarque-Bera test
jarque.test(na.omit(as.vector(GARCH_tdist_res_std)))    # for using Jarque-Bera test
print("Note: Jarque-Bera test should reject when shape parameter is small")


# Squared residuals

  # plot acf of squared residuals
plot(GARCH_tdist_res_std^2, type = "l")

  # plot autocorrelation funcion of squared residuals
```

```
acf(ts(GARCH_tdist_res_std)^2, na.action = na.omit)

  # compute ARCH-LM test
GARCH_tdist_results_ARCH_LM <- ARCH.LM(GARCH_tdist_res_std, q_ARCH)


# ------------------------------------------------------------------------
#           Prediction with AR and ARIMA models
# ------------------------------------------------------------------------
if (data_source=="Shiller"){

  # compute first month for prediction horizon
  if (subperest_end[2]<12) {
      subperfor_beg <- subperest_end+c(0,1)
  } else {
      subperfor_beg <- c(subperest_end[1]+1,1)
  }
  # compute largest prediction horizon
  n_pred <- (subperfor_end[1] - subperfor_beg[1])*12 +
            (subperfor_end[2] - subperfor_beg[2]) + 1
                # RT 2022_06_19 "+ 1 " added

  # estimate AR model for subperiod
  series_to_check_ar_sub <- ar.ols(window(series_to_check_ts, start = NULL,
                                          end = subperest_end),
                                   order.max = p, aic = FALSE, deman = FALSE,
                                   intercept = TRUE)
        # RT 2022_06_19 correction:
        # ar.ols needed (or ar(..., method = "ols",))
        # otherwise Yule-Walker estimator is used by default which ignores
```

```
      # the parameter intercept and attempts to estimate a stationary series
      # Then predictions go awry if the data contain a unit root componet
  # geschätzte Konstante
  series_to_check_ar_sub$x.intercept
  (series_to_check_ar_sub)
  # check stability of AR polynomial
  sort(abs( polyroot(c(1, - series_to_check_ar_sub$ar)) ))
  mean(series_to_check_ts)
  str(series_to_check_ar_sub)

  # compute predictions
  series_to_check_ar_predict <- predict(series_to_check_ar_sub, n.ahead = n_pred)

  # plot observed and predicted values
  plot.ts( cbind(window(series_to_check_ts, start = subperfor_beg,
                        end = subperfor_end),
                 series_to_check_ar_predict$pred), plot.type = "single", lty = 1:2,
                 ylab = "observed values (solid), predictions (dotted)" )

  # plot observed and predicted values for full period
  plot.ts( cbind(window(series_to_check_ts, start = NULL,
                        end = subperfor_end),
                  c(window(series_to_check_ts, start = NULL,
                           end = subperest_end),
                    series_to_check_ar_predict$pred)),
           plot.type = "single", lty = 1:2,
           ylab = "observed values (solid), predictions (dotted)" )


} else if (data_source=="yahoo"){
```

```
# get dates for estimation sample
dates_of_obs <- index(series_to_check_zoo)  # get dates of time series in object zoo
numb_of_obs  <- length(dates_of_obs)        # number of available observations
numb_for_est <- round(numb_of_obs*n_est_ratio)  # number of observations for estimation
                                            # depends on n_est_ratio
# selects all observations for estimation
series_to_check_est_zoo <- series_to_check_zoo[as.Date(dates_of_obs[1:numb_for_est])]

# selects all observations that will be predicted used for comparison
series_to_check_pred_zoo <- series_to_check_zoo[
                      as.Date(dates_of_obs[(numb_for_est+1):(numb_for_est+h_pred)]) ]

# estimate AR model for subperiod using ar command and ts object
series_to_check_ar_sub <- ar.ols(ts(series_to_check_est_zoo),
                              order.max = p, aic = FALSE, deman = FALSE,
                              intercept = TRUE)
      # RT 2022_06_19 correction: ar.ols instead of ar as
(series_to_check_ar_sub)

# compute predictions
series_to_check_ar_predict <- predict(series_to_check_ar_sub, n.ahead = h_pred)

# put the predicted values back into a zoo object
series_to_check_pred_hat_zoo <- zoo(series_to_check_ar_predict$pred,
                                    as.Date(index(series_to_check_pred_zoo)))

# plot observed and predicted values using zoo objects
plot.zoo(cbind(series_to_check_pred_zoo,series_to_check_pred_hat_zoo),
         plot.type = "single" , lty = c(1,2),
```

```
            ylab="observed values (solid), predictions (dotted)" )
}

#============================== End ================================================
```

Listing A.9: ../R_code/AOE_series_to_check_complete_2023.R

Bibliography

Berndt, E., H. Hall, R. Hall, und J. Hausman (1974), "Estimation and inference in nonlinear structural models," *Annals of Economic and Social Measurement*, 3, 653–665.

Bollerslev, T. (1986), "Generalized Autoregressive Conditional Heteroskedasticity," *Journal of Econometrics*, 31, 307 – 327.

Brockwell, P. J., und R. A. Davis (1991), *Time Series: Theory and Methods*, 2nd edn., Springer, New York, NY.

Brooks, C. (2008), *Introductory econometrics for finance*, 2nd edn., Cambridge University Press, Cambridge.

Brooks, C. (2019), *Introductory econometrics for finance*, 4th edn., Cambridge University Press, Cambridge.

Campbell, J., A. Lo, und A. MacKinlay (1997), *The Econometrics of Financial Markets*, Princeton

University Press, Princeton.

Cochrane, J. (2005), *Asset Pricing*, Princeton University Press, Princeton.

Davidson, J. (2000), *Econometric Theory*, Blackwell Publishers.

Davidson, R., und J. MacKinnon (1993), *Estimation and Inference in Econometrics.*, Oxford University Press, URL http://www.oup.com/uk/catalogue/?ci=9780195060119.

Davidson, R., und J. G. MacKinnon (2004), *Econometric Theory and Methods*, Oxford University Press, Oxford.

Deistler, Manfred und Scherrer, W. (2018), *Modelle der Zeitreihenanalyse*, Mathematik Kompakt, Birkhäuser, doi:https://doi.org/10.1007/978-3-319-68664-6.

Demetrescu, M. (2010), "On the Dickey-Fuller test with White standard errors," *Statistical Papers*, 51(1), 11–25, doi:10.1007/s00362-007-0112-1, URL http://dx.doi.org/10.1007/s00362-007-0112-1.

Diebold, F. (2007), *Elements of forecasting*, 4th edn., Thomson/South-Western.

Diebold, F. X., und R. S. Mariano (1995), "Comparing Predictive Accuracy," *Journal of Business and Economic Statistics*, 13(3), 253–263.

Enders, W. (2015), *Applied Econometric Time Series*, 4th edn., Wiley.

Engle, R. F. (1982), "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, 50(4), 987-1007.

Franke, J., W. Härdle, und C. Hafner (2004), *Einführung in die Statistik der Finanzmärkte*, 2nd edn., Springer.

Franke, J., W. Härdle, und C. Hafner (2011), *Statistics of Financial Markets*, 3rd edn., Springer, doi:10.1007/978-3-642-16521-4.

Franke, J., W. Härdle, und C. Hafner (2019), *Statistics of Financial Markets: An Introduction*, 5th edn., Springer, doi:10.1007/978-3-030-13751-9.

Giacomini, R., und H. White (2006), "Tests of Conditional Predictive Ability," *Econometrica*, 74(6), 1545–1578.

Hamilton, J. D. (1994), *Time Series Analysis*, Princeton University Press.

Hassler, U. (2019), *Time Series with Long Memory in View*, John Wiley and Sons.

Johnson, N. L., S. Kotz, und N. Balakrishnan (1995), *Continuous univariate distributions*, *Wiley series in probability and mathematical statistics*, vol. 2, Mifflin / Wiley.

Kirchgässner, G., J. Wolters, und U. Hassler (2013), *Introduction To Modern Time Series Analysis*, 2nd edn., Springer, Berlin, [u.a.].

Kleiber, C., und A. Zeileis (2008), *Applied Econometrics with R*, Springer, doi:10.1007/978-0-387-77318-6.

Kreiss, J.-P., und G. Neuhaus (2006), *Einführung in die Zeitreihenanalyse*, Springer.

Kwiatkowski, D., P. C. B. Phillips, P. Schmidt, und Y. Shin (1992), "Testing the null of stationarity against the alternative of a unit root: How sure are we that the economic time series have a unit root?" *Journal of Econometrics*, 54, 159–178.

Ligges, U. (2008), *Programmieren mit R*, Statistik und ihre Anwendungen, Springer, doi:10.1007/978-3-540-79998-6.

Lütkepohl, H., und M. Krätzig (eds.) (2004), *Applied Time Series Econometrics*, Cambridge University Press, Cambridge.

Lütkepohl, H. (2005), *New Introduction to Multiple Time Series Analysis*, Springer.

Lütkepohl, H., und M. Kraetzig (2008), *Applied Time Series Econometrics*, Cambridge University Press.

Neusser, K. (2009), *Zeitreihenanalyse in den Wirtschaftswissenschaften*, 2nd edn., Teubner, Wiesbaden.

Neusser, K., und M. Wagner (2022), *Zeitreihenanalyse in den Wirtschaftswissenschaften*, 4th edn., Springer Gabler, Berlin Heidelberg.

Ng, S., und P. Perron (2005), "A Note on the Selection of Time Series Models," *Oxford Bulletin Of Economics And Statistics*, 67, 115–134.

Taylor, S. (2008), *Modelling Financial Time Series*, World.

Tsay, R. S. (2010), *Analysis of Financial Time Series*, 3rd edn., John Wiley & Sons.

Tschernig, R. (1994), *Wechselkurse, Unsicherheit und Long Memory*, Physica-Verlag, Heidelberg, URL http://epub.uni-regensburg.de/6928/.

West, K. D. (1996), "Asymptotic inference about predictive ability," *Econometrica*, 64, 1067–1084.

Wooldridge, J. M. (2020), *Introductory Econometrics. A Modern Approach*, 7th edn., Cengage, Boston.