Last updated: 3/2018

Econometrics using R

Reinhold Kosfeld

Contents

1. Introduction

1.1 Preliminary remarks

1.2 Calculating with numbers and functions

1.3 Variables, vectors and matrices

1.4 Time series objects and charts

1.5 Importing time series from data files

2. Regression model and least squares method

2.1 Multiple regression analysis and lm function

2.2 Multiple regression analysis and lm function

2.3 Regression analysis and matrix computation

3. Multicollinearity

3.1 Regression using three explanatory variables

3.2 Simple and advanced correlation analysis

3.3 Determinant and condition number

3.4 Method of auxiliary regressions

4. Heteroscedasticity

4.1 Testing for heteroscedasticity

4.1.1 Breusch-Godfrey test

4.1.2 Breusch-Godfrey test

4.2 Weighted least squares (WLS) estimation

4.3 Heteroscedasticity-consistent (HC) standard errors

5. Autocorrelation

5.1 Autocorrelation coefficient and correlogram

5.2 Testing for autocorrelation

5.2.1 Durbin-Watson test

5.2.2 Breusch-Godfrey test

5.3 Cochran-Orcutt method

5.4 Heteroscedasticity- and autocorrelation-consistent (HAC) standard errors

6. Models using qualitative variables

6.1 Models using qualitative regressors

6.2 Structural break test

6.3 Linear probability model

6.3 Logit model

1. Introduction
1.1 Preliminary remarks

The statistical software and programming environment R is developed as open source software by an international developer community and can be freely used and distributed by anyone under the GNU General Public License Version 2 (June 1991). The software is available for download from the website of the R Project (http://www.r-project.org). Source code and binary data for a range of operating systems are provided by CRAN (http://CRAN.R- project.org).

R is a highly flexible, interpreted programming language and environment for statistical and graphical data analysis. It is not a complete graphical user interface (GUI), but tools are available for developing one. A number of methods for analysing data can already be executed via menu selection in R Commander. When R functions are executed, all calculated values are not generally output. Instead, some results are first buffered in objects and can be retrieved subsequently. On the other hand, R does not store results if no object has been defined for them.

R can be used both interactively as well as in batch mode. Batch mode should be used when larger sequences of commands are used by executing them in a script file. The software is ready when the > sign appears. Commands can be separated from each other using a semicolon (;) or by beginning a new line. A plus (+) sign is displayed if a command is incomplete. Commands that have already been used can be retrieved using the arrow keys (↑and ↓). The # sign is used to prefix a comment. A decimal point (.) and not decimal comma (,) is used. The ESC key interrupts the calculation that is currently being executed.

All objects present in an R session can be displayed using the ls() command. An object obj can be deleted using the remove command rm(obj). Text like "Econometrics is fun" can be assigned to an object obj using quotation marks with an equal sign (=) or arrow (<_):

```
>obj = "Econometrics is fun" or   >obj <- "Econometrics is fun"
>obj
[1] "Econometrics is fun".
```

In this document, only the equal sign (=) is used for assignments. In this case, the object obj is a character type / string object.

Objects available in the current R environment can be displayed on the R console using the ls() command:

```
>ls()
 [1] "obj"
```

The current working directory can be checked using the getwd() command:

```
>getwd()
[1] "C:/Users/Kosfeld/Documents/Ökonometrie/Ökonometrie mit R/Daten"
```

This can be changed in the R menu via File – Change Dir… or using the setwd() command:

```
> setwd("C:/Users/Kosfeld/Documents/Ökonometrie/Ökonometrie mit R/Daten").
```

The R application can be exited with the quit command q() or by clicking on the x icon in the RGui program window (64 bit).

1.2 Calculating with numbers and functions

The R application can be used as a calculator by performing arithmetic operations like addition (+), subtraction (-), multiplication (*), division (/) and exponentiation (^) direct:

```
> 3+4
[1] 7

> 7-12
[1] -5

> 5*6
[1] 30

> 9/4
[1] 2.25
```

Exponentiation with any desired base is performed with the ^sign,

```
> 2^4
[1] 16,
```

whereas it is performed to base e (Euler's number: e = 2.718282) using the exp function:

```
>exp(3)
[1] 20.08554
```

The square root is calculated using the sqrt function:

```
>sqrt(64)
[1] 8
```

The logarithm to base e is formed with the log function,

```
> log(8)
[1] 2.079442
```

and the logarithm to base 10 with the log10 function:

```
> log10(1000)
```

```
[1] 3
```

Besides the arithmetic functions exp, sqrt and log, we should also mention here the trigonometric functions sin, cos and tan and their inverse functions asin, acos and atan. The angle as argument of the sinus function must be given as a radian, e.g. 2 (= 2*180/pi = 114.5916°):

```
> sin(2)
[1] 0.9092974
```

The constant π is stored in R under pi:

```
>pi
[1] 3.141593
```

The constant pi can be used with the arc sine function asin to convert the angle calculated in radians,

```
>asin(0.9)
[1] 1.119770,
```

into an angle expressed in degrees:

```
>asin(0.9)*180/pi
[1] 64.15807
```

The round function is used to round numbers to whole numbers,

```
>round(1.8786)
[1] 2
```

or to round them to the decimal places indicated by the argument digit=2 :

```
>round(1.8786, digit=2)
[1] 1.88
```

Exercises

1.2.1 Calculate the expression $4 \cdot (6\frac{1}{2} - \frac{1}{2})/2$!

1.2.2 Calculate the expression $(8\frac{1}{3} + 2\frac{1}{2})^2 / \sqrt{2}$ and round the result to three decimal places!

1.2.3 Calculate the expression $\log_{10} 1000 + \sqrt[3]{27}$!

1.2.4 Calculate the expression $x^2 = \log_2 16$!

1.2.5 Calculate the expression $\ln x = \sqrt{8}$!

1.3 Variables, vectors and matrices

In the previous examples, the result is lost after it is output on the screen. In order to make the results of an arithmetic operation available in a session, they must be stored under a variable name.

The result is not displayed with the assignment to the variable but only subsequently using an additional command referring to the variable name:

```
> a = log10(1000)
>a
[1] 3
```

Here, the object a is a special case of a numeric vector with just one element. In R, a vector is the most basic data structure or most simplest object type.

A vector can generally be defined by using the c():

```
> x = c(3, 6, 7, 1, 9, 4)
> x
[1] 3 6 7 1 9 4
```

A vector with a sequence of consecutive natural numbers is created using:

```
> y = 1:6
> y
[1] 1 2 3 4 5 6
```

A more general sequence of numbers can be assigned to a vector with the seq()command by specifying the increment, e.g.by = 0.5

```
> a = seq(1, 4, by = 0.5)
>a
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

or the length, e.g. length = 7

```
> b = seq(1, 4, length=7)
> b
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

Here, the first value specifies the initial value (from = 1), while the second value defines the final value (to = 4).

Operations can be executed on vector elements using the simple operators +, -, *, /and ^. In the following examples, the vectors x,

```
> x
[1] 3 6 7 1 9 4,
```

and y,

```
> y
[1] 1 2 3 4 5 6,
```

are used:

```
> d = x + 3
> d
[1]  6  9 10  4 12  7

> e = 2*x
> e
[1]  6 12 14  2 18  8

> f = x + y
> f
[1]  4  8 10  5 14 10

> g = x * y
> g
[1]  3 12 21  4 45 24
```

The%*% operator must be used to form the scalar product of the two vectors x and y:

```
> t(x)%*%y
    [,1]
[1,]  109
```

In this instance, row vector t(x)is multiplied with column vector y. Transposing a column vector to a row vector or a row vector to a column vector is effected using the t()function.

The outer product of the two vectors x and y results from multiplying column vector x with row vector t(y) using multiplication operator %*%:

```
> x%*%t(y)
   [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    3    6    9   12   15   18
[2,]    6   12   18   24   30   36
[3,]    7   14   21   28   35   42
[4,]    1    2    3    4    5    6
```

```
[5,]   9  18  27  36  45  54
[6,]   4   8  12  16  20  24
```

In this example, the result is a 6x6 matrix.

From a statistical perspective, a vector can be used to represent the values of an attribute (variable) acquired from n sampling units. However, the data of m attributes are generally acquired in sampling. These data can be neatly arranged in an n x m matrix.

A matrix can be generated e.g. from the values of a vector by specifying the number of rows and columns (nrow and ncol) with the matrix function:

```
> A = matrix(c(3,5,6,2,8,4), nrow=3, ncol=2)
> A
   [,1] [,2]
[1,]   3   2
[2,]   5   8
[3,]   6   4
```

Here, the first argument data is specified using the c() function. By default, the matrix is filled column by column. If the values are to be entered row by row, the argument byrow equals TRUE must be set:

```
> B = matrix(1:6, nrow=3, ncol=2, byrow=TRUE)
> b
   [,1] [,2]
[1,]   1   2
[2,]   3   4
[3,]   5   6
```

colnames() can used to assign names to the columns of a matrix:

```
>colnames(B) = c("b1", "b2")

> b
  b1 b2
[1,] 1 2
[2,] 3 4
[3,] 5 6
```

In the same way, rownames() can be used to assign names to the rows of a matrix.
Both operations can be accomplished simultaneously using the dimnames() command:

```
>dimnames(B) = list(c("B1", "B2", "B3"),c("b1", "b2"))
> b
 b1 b2
B1  1  2
```

```
B2  3  4
B3  5  6
```

The dim function is used to retrieve the dimensions of a matrix:

```
>dim(B)
[1] 3 2
```

The first value specifies the row number, while the second value specifies the column number.

For some calculations, it is expedient to define the rows and columns of a matrix as vectors. If the two columns of a matrix B contain the values of two variables, they can be extracted and e.g. assigned to variable vectors b1and b2:

```
> b1 = B[,1]
> b1
B1 B2 B3
 1  3  5
> b2 = B[,2]
> b2
B1 B2 B3
 2  4  6.
```

Conversely, functions rbind and cbind can be used to combine vectors into a matrix by joining them by rows or columns. The example below links vectors b1 and b2 row by row to form matrix B1:

```
> B1 = rbind(b1,b2)
> B1
   B1 B2 B3
b1  1  3  5
b2  2  4  6,
```

while the following example links them column by column to form matrix B2:

```
> B2 = cbind(b1,b2)
> B2
  b1 b2
B1  1  2
B2  3  4
B3  5  6.
```

Exercises

1.3.1 Assign the value $4^3 - 6 \cdot 3$ to the variable x and the expression $3 \cdot x + 5$ to the variable y. Calculate the expression $z = \left(\frac{1}{3}x^2 - x \cdot \sqrt{y}\right)$!

1.3.2 Calculate $s = \frac{\sqrt[3]{(y^3 - 3 \cdot x + 4)}}{y - 2}$ using the assignments defined in Exercise 1.3.1 for x and y!

1.3.3 Define a vector x from the values 3, 5, 8, 2, 4, 1!

1.3.4 Create a vector y with elements 4, 5, 6, 7, 8, 9 in three different ways!

1.3.5 Add and multiply the vectors x and y generated in Exercises 1.3.3 and 1.3.4 element by element and store the results in objects named s and m!

1.3.6 Form the scalar product of the two vectors x and y (see Exercises 1.3.3 and 1.3.4)!

1.3.7 Form the 2x3 matrices A and B from the two vectors x and y (see Exercises 1.3.3 and 1.3.4)!

1.3.8 Multiply matrix A with the transposed matrix B (see Exercise 1.3.7) and store the matrix product in an object named AB!

1.3.9 Populate a 3x3 matrix C row by row with the values 1 to 9!

1.3.10 Assign the names c1, c2 and c3 to the columns of matrix C generated in Exercise 1.3.9!

1.3.11 Combine vectors x and y created in exercises 1.3.3 and 1.3.4 to form a 6x2 matrix D!

1.4 Time series objects and charts

Econometrics makes use of time series data, cross-sectional data and panel data. While microeconomic analyses generally refer to a cross section or a panel of companies, households or employees, macroeconomic data such as gross domestic product, private consumption or investment are frequently available as time series. Econometric data are illustrated here primarily using time series data.

Time series can be defined with the function ts by specifying a vector or matrix with quantitative data, the start and end periods and the periodicity (frequency) as univariate – or general multivariate – time series. The example below uses the annual data of the consumer price index (CPI) for the period 2000 to 2011 as values in the time series cpi:

```
>cpi = ts(c(92.7, 94.5, 95.9, 96.9, 98.5, 100, 101.6, 103.9, 106.6, 107, 108.2, 110.7), start=2000,
end=2011,frequency=1)
```

Applying the str function to the cpi object displays its internal structure,

```
>str(cpi)
 Time-Series [1:12] from 2000 to 2011: 92.7 94.5 95.9 96.9 98.5 ...
```

while the class function can be used to specify the object class:

```
>class(cpi)
[1] "ts".
```

The definition and contents of the ts object cpi can be listed by entering the object name and confirming with the Return key:

```
>cpi
Time Series:
Start = 2000
End = 2011
Frequency = 1
 [1]  92.7  94.5  95.9  96.9  98.5 100.0 101.6 103.9 106.6 107.0 108.2 110.7
```

One of the two arguments end or frequency can be left out. The latter specifies the number of time series values per year:

- Yearly data: frequency=1
- Half-yearly data: frequency=2
- Quarterly data: frequency=4
- Monthly data: frequency=12

Instead of periodicity (frequency), it is also possible to define the equidistant difference (deltat) Δx_t between two consecutive time series values x_t and x_{t-1} as an argument of the ts function.

A description of a time series is obtained with the summary function:

```
>summary(cpi)
Min.1st Qu.Median       Mean 3rd Qu.Max.
92.70 96.65  100.80  101.40  106.70 110.70
```

It consists of the details of the smallest and largest values in the time series, the quartiles (2nd quartile = median) and the arithmetic mean.

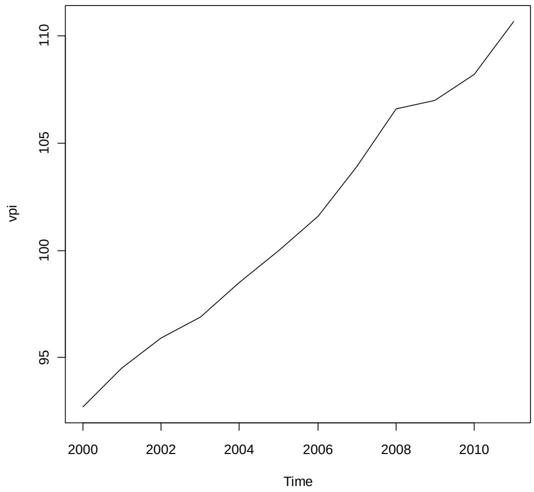
A time series chart can be created from a single time series using the plot function.[1] Here, time t is plotted on the abscissae while the ordinates describe the values of the time series x_t. If function plot is applied to a time series object, R is aware of the time, meaning that only the times series itself needs to be specified. The easiest way of obtaining a time series chart e.g. for the consumer price index is to specify just the name of the time series object (cpi) in the plot function:

```
>plot(cpi)
```

---

[1]The use of the ts.plot function is only of benefit when charting multiple time series.

Figure 1.1: Elementary time series chart of the consumer price index (CPI)



The elementary time series chart shown in Figure 1.1. can be enhanced by including optional arguments and low-level commands to modify the chart:

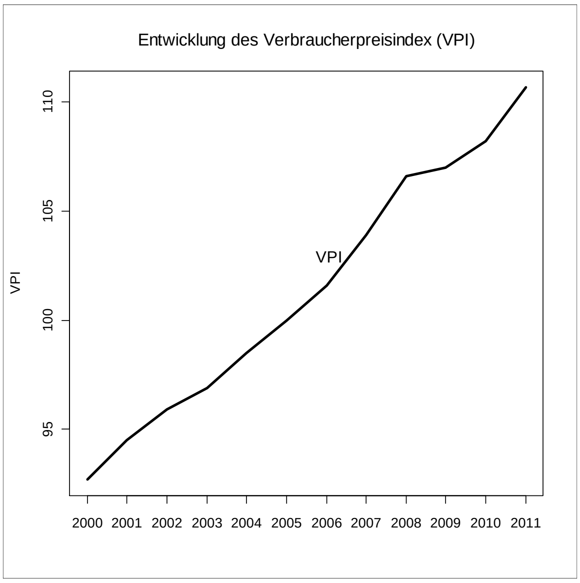
```
>plot(cpi, main="Development of the consumer price index (CPI)", xlab="", ylab="CPI",
lwd=3,cex.main=1.25, font.main=1)
>axis(side=1, at=c(2001,2003,2005,2007,2009,2011))
>text(2006, 103, "CPI", adj=0.4, cex=1.2)
>box(which="figure")
```

In the plot function (a high-level command), the argument main defines the heading for the time series chart. Ylab specifies labels for the ordinates, while xlab="" suppresses the "Time" label for the abscissae. The lwd parameter controls the line weight of the time series polygon (graph curve), and the cex.main parameter defines the size of the heading (default settings: lwd=1 and cex.main=1). The font parameter specifies the font type (1 = simple, 2 = bold 3 = italic 4 = bold italic).

As can be seen in Figure 1.1, the R software automatically creates the abscissae for the cpi time series indicating the even years. The abscissae with all the year numbers can be added to the active chart using the axis command with argument at. The side=1 argument specifies that the axis is to be appended below.

The text command labels the time series polygon with CPI. Text alignment is controlled with the parameter adj (0: left-justified, 0.5: centred, 1: right-justified). Font size is set to 1.2 with the cex parameter. The box command with the which="figure" or which="outer" argument draws a border around the chart.

Figure 1.2: Time series chart of the consumer price index (CPI)



When vectors and matrices contain time-related quantitative data, they can be converted into univariate or multivariate time series using the as.ts or ts commands. The vector cpi containing observation values for the consumer price index

```
>cpi = c(92.7, 94.5, 95.9, 96.9, 98.5, 100, 101.6, 103.9, 106.6, 107, 108.2, 110.7)
```

can accordingly be converted into a time series object with command

```
>cpi = as.ts(cpi)
```
or
```
>cpi = ts(cpi)
```

and whose values refer to time units 1, 2, …, 12:

```
>cpi
Time Series:
Start = 1
End = 12
Frequency = 1
 [1]  92.7  94.5  95.9  96.9  98.5 100.0 101.6 103.9 106.6 107.0 108.2 110.7
```

Specifying the arguments start, end and frequency in the as.ts command, as in the ts function, allows the time series to be assigned to the observation period 2001 – 2009. This date assignment can be performed subsequently with the ts command:

```
>cpi =ts(cpi, start=2000, frequency=1)
>cpi
Time Series:
Start = 2000
End = 2011
Frequency = 1
 [1]  92.7  94.5  95.9  96.9  98.5 100.0 101.6 103.9 106.6 107.0 108.2 110.7
```

An extract of a time series can be selected with the window function:

```
>cpi0308 = window(cpi, start=2003, end=2008)
> vpi0308
Time Series:
Start = 2003
End = 2008
Frequency = 1
[1]  96.9  98.5 100.0 101.6 103.9 106.6
```

In this example, the consumer price index for the time window 2003-2008 of the observation period is stored in ts object cpi0308.

Exercises

1.4.1 Create a vector GASCn from the values 1, 0.8, 0.7, 0.9, 1.2, 1.1, 1.4 for natural gas consumption!

1.4.2 Define the GASCn vector created in Exercise 1.4.1 as a time series and create an elementary time series chart!

1.4.3 Assign the period 2005 to 2011 to the GASCn time series created in Exercise 1.4.2!

1.4.4 Plot the GASCn time series that was assigned a date in Exercise 1.4.3 with a line weight of 3.5. Label the ordinates and the graph curve with GASCn. Define a heading for the time series chart with "Gas consumption 2005-2011" and a font size of 1.15.

1.5 Importing time series from data files

Short time series like the time series of the consumer price index cpi, which was used here to illustrate the plot function, can be entered easily in R via the keyboard. In contrast, long or multiple time series are usually imported from a data file. Data stored in .xls or .xlsx files can be converted with a spreadsheet application like Excel into .csv (comma separated values) text files. They can then be imported into R using the read.table or read.csv command.

The .csv file Consumfnc(data) contains multiple time series for estimating the macroeconomic consumption function for Germany in the period 1991 – 2011. The yearly data can be imported into R with either the read.table or the read.csv command and stored as an R object:

```
>Consumfnc.data = read.csv(file="Consumfnc(data).csv", dec=".", sep=";", header=TRUE)
>Consumfnc.data
  Year   PICPrivConsDispIncmPrivConsrDispIncmr
```

```
1 1991 79.42  882.55 1005.13  1111.24  1265.59
2 1992 82.84  949.79 1077.97  1146.54  1301.27
3 1993 85.86  989.11 1119.76  1152.00  1304.17
4 1994 88.03 1032.75 1159.39 1173.18 1317.04
5 1995 89.27 1066.47 1190.61 1194.66 1333.72
6 1996 90.11 1088.64 1212.71 1208.12 1345.81
7 1997 91.26 1110.82 1231.47 1217.20 1349.41
8 1998 91.75 1130.14 1251.21 1231.76 1363.72
9 1999 92.11 1161.86 1278.66 1261.38 1388.19
10 2000  92.84 1195.04 1300.70 1287.20 1401.01
11 2001  94.56 1233.43 1347.78 1304.39 1425.32
12 2002  95.68 1240.58 1363.26 1296.59 1424.81
13 2003  97.24 1264.51 1394.62 1300.40 1434.20
14 2004  98.36 1283.61 1419.00 1305.01 1442.66
15 2005 100.00 1306.98 1448.13 1306.98 1448.13
16 2006 100.99 1339.54 1480.52 1326.41 1466.01
17 2007 102.49 1356.73 1502.37 1323.77 1465.87
18 2008 104.19 1387.70 1542.58 1331.89 1480.55
19 2009 104.25 1387.43 1531.09 1330.87 1468.67
20 2010 106.28 1423.02 1575.85 1338.93 1482.73
21 2011 108.51 1474.42 1626.67 1358.79 1499.10
```

A number of optional parameters have been specified in the read.csv command beside the file name (file). The dec argument specifies that the decimal point in the data file Consumfnc(data).csv is to be used as the decimal character. The values themselves are separated by semicolons (sep=“;“) in the file – which is not shown here. The Boolean value TRUE for the header argument indicates that the first row of the Consumfnc(data).csv file contains the variable names.

The contents of the Consumfnc(data).csv file are assigned to the R object Consumfnc.data. As time specification, the first column contains the years of the observation values below the variable name Year. The second column contains the price index of private consumption (PIC). This is followed by the nominal values of private consumption (PrivCons) and disposable income (DispIncm) in columns 3 and 4. Finally, the real values of private consumption (PrivConsr) and disposable income (DispIncmr) are stored in columns 5 and 6.

Command

```
>class(Konsumfkt.data)
[1] "data.frame"
```

can be used to output the type of the Consumfnc.data object. Consumfnc.data is not yet a time series object; it is a data.frame-type object, i.e. it is a data set consisting of rows and columns like a matrix object. The rows contain the observation values of the statistical units, while the columns contain the variables. However, unlike a matrix object, a data.frame object can also contain non-numeric values.

The str command

```
>str(Consumfnc.data)
'data.frame':   21 obs. of  6 variables:
$ Year     : int  1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 ...
 $ PIC      : num  79.4 82.8 85.9 88 89.3 ...
 $ PrivCons : num  883 950 989 1033 1066 ...
 $ DispIncm : num  1005 1078 1120 1159 1191 ...
 $ PrivConsr: num  1111 1147 1152 1173 1195 ...
 $ DispIncmr: num  1266 1301 1304 1317 1334 ...
```

provides information about the structure of data.frame object Consumfnc.data.

Since the macroeconomic consumption function is to be estimated in R using the real values of private consumption and disposable income with the help of regression analysis, we will now define a fifth column of the data.frame object Consumfnc.data as time series PCr,

```
>PCr = ts(Consumfnc.data$PrivConsr, start=1991, frequency=1)
>PCr
Time Series:
Start = 1991
End = 2011
Frequency = 1
 [1] 1111.24 1146.54 1152.00 1173.18 1194.66 1208.12 1217.20 1231.76 1261.38 1287.20 1304.39 1296.59
1300.40 1305.01 1306.98 1326.41 1323.77 1331.89 1330.87 1338.93 1358.79,
```

and a sixth column as time series DIr:

```
>DIr = ts(Consumfnc.data$DispIncmr, start=1991, frequency=1)
>DIr
Time Series:
Start = 1991
End = 2011
Frequency = 1
 [1] 1265.59 1301.27 1304.17 1317.04 1333.72 1345.81 1349.41 1363.72 1388.19 1401.01 1425.32 1424.81
1434.20 1442.66 1448.13 1466.01 1465.87 1480.55 1468.67 1482.73 1499.10.
```

The start argument of the ts function defines 1991 as the start year for the time series. The value 1 of the frequency argument specifies yearly data.

The commands

```
>class(PCr)
[1] "ts"
```

and

```
>class(DIr)
[1] "ts"
```

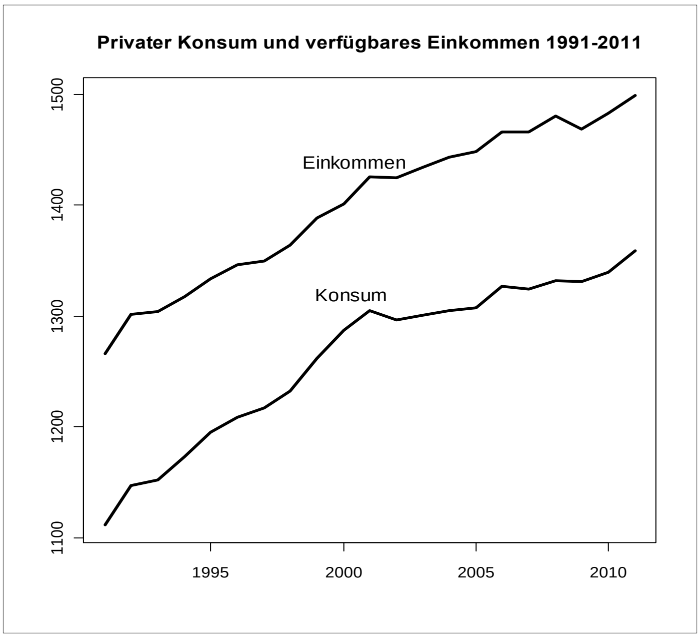
provide confirmation that PCr and DIr are ts-type objects, i.e. time series objects.

Multiple time series can best be plotted in R using the ts.plot function. The chart of the time series of real private consumption (PCr) and real disposable income (DIr) for the period 1991 – 2011 is shown in Figure 1.3:

```
>ts.plot(PCr, DIr, main="Private consumption and disposable income 1991-2011",gpars=list(xlab="",
lwd=3))
>text(2000, 1440, "Income", adj=0.4, cex=1.2)
>text(2000, 1320, "Consumption", adj=0.4, cex=1.2)
> box(which="figure")
```

The graphics parameters are listed in the gpars argument of the ts.plot function in order to be able to label the two time series.

Figure 1.3: Time series chart of private consumption and disposable income 1991 – 2011



The time series for private consumption shows a greater increase than the time series for disposable income up to 2001. However, real consumption falls more sharply than real disposable income in the first recession of the new millennium. At the same time, the consumption curve levels off noticeably. At the later edge, both aggregate values climb more strongly once more following the recession of 2009.

Exercises

1.5.1 Create a .csv-file Energy(long) from the data for energy demand models (Dregery/Kosfeld/Eckey, 2014, Ökonometrie, Table 7)!

1.5.2 Import the .csv-file Energy(long) into R as the data.frame object Energyl!

1.5.3 Define the variables GASPR and DHEATPR for data.frame object Energyl as time series!

1.5.4 Plot the GASPRtime series created in Exercise 1.5.3 using “Relative gas price 1980 – 1995” as title!

1.5.5 Create a time series chart using the time series GASPR and DHEATPR with “Relative gas and - district heat prices 1980-1995” as title!

1.5.6 Define the variables GASC and DISPIR for data.frame object Energy as time series! Plot both times series separately using “Natural gas consumption 1980–1995” and “Real disposable income 1980 –1995” as titles!

2. Regression model and least squares estimation

2.1 Multiple regression analysis and lm function

In the simple regression model, a dependent (endogenous) variable y is explained using an independent (exogenous) variable x. In the case of the linear relationship between the two variables, the regression model is

Index t can be a time or a cross-sectional index. The latter may refer to e.g. companies, consumers or countries. An expected value of zero, constant variance (homoscedasticity) and an absence of autocorrelation are the standard assumptions for the confounding variable u.

If the standard assumptions are valid, the regression coefficients β_1 and β_2 can be estimated with the ordinary least squares (OLS) method. β_1 is the absolute member and β_2 is the slope.

The lm function is available in R for the least squares estimation of a regression model. The example discussed here uses simple regression analysis based on the macroeconomic consumption function for Germany. The time series of private consumption (PCr) and disposable income (DIr) provided by the .csv file Consumfnc(data) imported in Section 1.5 are needed for the Keynesian consumption function. The Keynesian consumption function is estimated for the period 1997 – 2011 using the lm function and disregarding the phase of adjustment following German reunification:

```
> PCr9711 = ts(PCr[-c(1:6)], start=1997, freq=1)
```

or

```
> PCr9711 = window(PCr, start=1997, end=2011)
> PCr9711
Time Series:
Start = 1997
End = 2011
Frequency = 1
 [1] 1217.20 1231.76 1261.38 1287.20 1304.39 1296.59 1300.40 1305.01 1306.98 1326.41 1323.77 1331.89
1330.87 1338.93 1358.79

> DIr9711 = ts(DIr[-c(1:6)], start=1997, freq=1)
```

or

```
> DIr9711 = window(DIr, start=1997, end=2011)
> DIr9711
Time Series:
Start = 1997
End = 2011
Frequency = 1
 [1] 1349.41 1363.72 1388.19 1401.01 1425.32 1424.81 1434.20 1442.66 1448.13 1466.01 1465.87 1480.55
1468.67 1482.73 1499.10
>consfnc.lm = lm(PCr9711~DIr9711).
```

Model specification is performed inside the parentheses of the lm function. The response variable, PCr9711, is defined before the tilde (~), and the explanatory variable, DIr9711, is specified after the tilde. The absolute member is also estimated by default. The estimator result is stored in the lm object consfnc.lm:

```
>class(consfnc.lm)
[1] "lm"
```

A summary of the of the estimator result can be obtained by specifying the name of the lm object,

```
>konsfkt.lm

Call:
lm(formula = PKr9711 ~ VEr9711)
```

```
Coefficients:
(Intercept)      DIr9711
   57.9858       0.8659  ,
```

whereby only specification of the regression model and the OLS estimators of the regression coefficient are output. Autonomous consumption assumes the value 57.9858. A marginal propensity to consume of 0.8659 means that in the period under examination, 1997 – 2011, an average of 86.6% of additional income was spent on consumption.

Detailed output can be obtained using the summary function:

```
>summary(consfnc.lm)

Call:
lm(formula = PKr9711 ~ VEr9711)

Residuals:
   Min    1Q  Median      3Q      Max
-9.2379 -4.2249 -0.9917  2.0491 16.0817

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  57.98585   63.12937   0.919    0.375
DIr9711 0.86590 0.04394  19.706   4.58e-11 ***
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.31 on 13 degrees of freedom
Multiple R-squared: 0.9676,      Adjusted R-squared: 0.9651
F-statistic: 388.3 on 1 and 13 DF,  p-value: 4.579e-11.
```

It also provides a significance test for the estimated regression coefficients. On the basis if the p-values, it is apparent that the marginal propensity to consume is highly significant ($p<0.01$), while autonomous consumption, with a p-value of 0.375, does not deviate significantly from null.

The coefficient of determination is 0.9676, meaning that the OLS-estimated econometric model explains 96.76% of the variance of private consumption. With a p-value of less than 0.01, the F-test is highly significant. The specification of the macroeconomic consumption function undertaken can therefore be considered substantial.

The output also contains a number of residual figures. The median, the first and third quartiles and the minimum and maximum residual values are shown in a five-number representation. The Residual standard error of 7.31 reflects the unbiased standard deviation of the residuals. This figure is also called the standard error of regression (SER).

In addition to the estimated regression coefficients,

```
>consfnc.coeff = consfnc.lm$coeff
>konsfkt.coeff
 (Intercept)        DIr9711
  57.9858465    0.8658985
> class(konsfkt.coeff)
[1] "numeric"
```

the lm object consfnc.lm also contains various other components. The regression values available as a vector under fitted.values are required for regression diagnostics:

```
> PCr9711f = consfnc.lm$fitted.values
```

or

```
 PCr9711f = fitted(consfnc.lm)
> PCr9711f
1       2        3        4        5        6        7        8        9       10       11       12       13       14       15
1226.438 1238.829 1260.017 1271.118 1292.168 1291.727 1299.857 1307.183 1311.919 1327.402 1327.280
1339.992 1329.705 1341.879 1356.054
>class(PKr9711f)
[1] "numeric".
```

The vector PCr9711f is defined as a time series using the ts command:

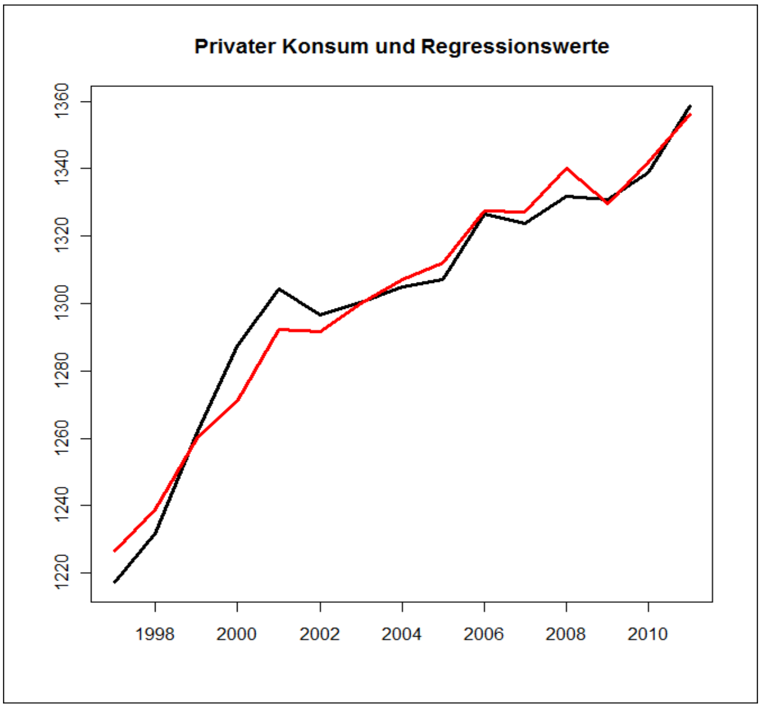
```
> PCr9711f = ts(PCr9711f, start=1997, freq=1)
> PCr9711f
Time Series:
Start = 1997
End = 2011
Frequency = 1
    1        2        3        4        5        6        7        8        9       10       11       12       13       14       15
1226.438 1238.829 1260.017 1271.118 1292.168 1291.727 1299.857 1307.183 1311.919 1327.402 1327.280
1339.992 1329.705 1341.879 1356.054.
```

The ts.plot function permits a graphical comparison of the observation values for private consumption and the regression values:

```
>ts.plot(PCr9711, PCr9711f, main="Private consumption and regression values",gpars=list(xlab="",
lwd=3, col=c("black","red")))
> box(which= "figure").
```

Figure 2.1 does in principle show the close connection between observed and adjusted time series. However, in the bull markets around the turn of the century and at the beginning of the subsequent crash, consumption was significantly above the values expected due to regression. Conversely, consumers were noticeably reluctant to spend at the beginning of the most recent financial crisis in 2007 – 2008.

Figure 2.1: Private consumption and regression values 1997 – 2011



The residuals of the consumption function available in the lm object consfnc.lm are defined as a time series for graphical diagnosis:

```
> consfnc.res = ts(consfnc.lm$residuals, start=1997, freq=1)
```

or

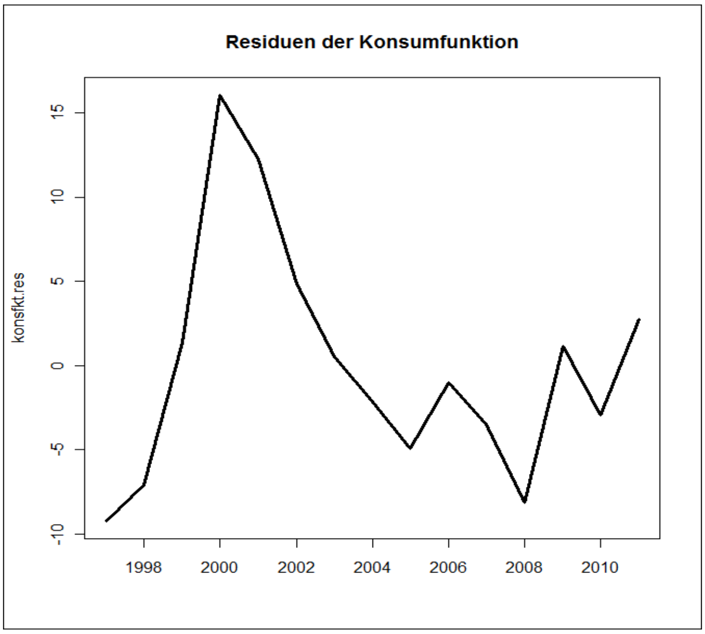
```
> consfnc.res = ts(resid(consfnc.lm), start=1997, freq=1)
> consfnc.res
Time Series:
Start = 1997
End = 2011
Frequency = 1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
-9.2379145 -7.0689218 1.3625424 16.0817239 12.2217319 4.8633401 0.5425534 -2.1729478 -4.9394125 -
0.9916773 -3.5104515 -8.1018412 1.1650328 -2.9494999 2.7357420.
```

The residuals plot is generated with the plot function:

```
>plot(consfnc.res, main="Residuals of the consumption function",xlab="", lwd=3)
> box(which= "figure").
```

Figure 2.2 clearly shows absolutely large residuals, particularly in the first half of the observation period. Here, the year 2000, where the residual with a value of 16.08 is outside the 2·SER range (SER=7.31), is identified as a potential outlier.

Figure 2.2: Residuals of the consumption function



Exercises

2.1.1 Regress the demand for natural gas (GASC) against the price for natural gas (GASPR) (refer to data.frame object Energyl for the data) using the lm function and store the result under the name energyfnc1.lm (=energy demand model 1)!

2.1.2 Output the summarised regression output of the lm object energyfnc1.lm to the R console and interpret it!

2.1.3 Output the detailed regression output of the lm object energyfnc1.lm to the R console and interpret it!

2.1.4 Define the regression values of energy demand model 1 as the time seriesGASCf1 and arrange it with the time series GASC in a time series chart with the title “Natural gas consumption and regression values 1980-1995”!

2.1.5 Define the time series of residuals energyfnc1.res for the natural gas demand model 1 and plot it under the title “Residuals of natural gas demand function 1”!

2.2 Multiple regression analysis and lm function

In the following example, the money demand function is used to illustrate multiple regression analysis. Data are imported from the .csv file Moneydemandfnc(data).csv and stored in data.frame object Moneydemandfnc.data:

```
>Moneydemandfnc.data = read.csv(file="Moneydemandfnc(data).csv", dec=".", sep=";", header=TRUE)
>class (Moneydemandfnc.data)
[1] "data.frame"
>Moneydemandfnc.data
  Year  M1EU  M1GER  M1GERr EURIBOR Rbond  CPI  Irate  GDP  PIGDP   GDPr
1 1997 1626.9  465.0  440.45    3.33  5.10  90.0 0.019 1912.6  94.72 1811.61
2 1998 1785.4  513.4  489.17    3.54  4.50  90.8 0.009 1959.7  95.28 1867.20
```

```
3  1999 1972.0  557.6  532.34    2.97  4.32  91.3 0.006 2000.2  95.47 1909.59
4  2000 2084.6  574.6  544.89    4.39  5.41  92.7 0.015 2047.5  94.83 1941.64
5  2001 2279.0  601.6  576.87    4.26  4.79  94.5 0.019 2101.9  95.89 2015.51
6  2002 2499.4  583.5  567.57    3.32  4.66  95.8 0.014 2132.2  97.27 2073.99
7  2003 2727.1  631.9  621.35    2.33  3.74  96.8 0.010 2147.5  98.33 2111.64
8  2004 2948.9  655.4  651.40    2.11  3.68  98.5 0.018 2195.7  99.39 2182.31
9  2005 3482.1  725.8  725.80    2.19  3.14 100.0 0.015 2224.4 100.00 2224.40
10 2006 3758.6  760.0  762.36    3.08  3.76 101.6 0.016 2313.9 100.31 2321.07
11 2007 3901.2  789.7  805.10    4.28  4.30 103.9 0.023 2428.5 101.95 2475.86
12 2008 4035.7  832.8  855.54    4.63  4.19 106.6 0.026 2473.8 102.73 2541.33
13 2009 4556.2 1015.9 1055.93    1.23  3.22 107.0 0.004 2374.5 103.94 2468.06
14 2010 4747.0 1110.2 1160.83    0.81  2.52 108.2 0.011 2476.8 104.56 2589.74
```

The variables are defined as follows:
M1EU: M1 money supply (nominal) in the European Union,
M1GER: M1 money supply (nominal) in Germany,
M1GERr: M1 money supply (real) in Germany,
EURIBOR: Euro Inter Bank Offered Rate (interest rate for euro-denominated term deposits in interbank transactions),
Rbond: Yield on bonds
CPI Consumer price index
Irate: Rate of inflation,
GDP: Gross domestic product (nominal) in Germany,
PIGDP: Gross domestic product price index in Germany
GDPr: Gross domestic product (real) in Germany,

The EURIBOR replaced the FIBOR (Frankfurt Interbank Offered Rate) as reference rate of interest for loans and investment products.

The econometric model of the money demand function is provided by a multiple regression model,

(2.2) $y_t = \beta_1 + \beta_2 \cdot x_{2t} + \beta_3 \cdot x_{3t} + u_t$, t=1.2….n,

where the logarithmised real money supply M1 (y) is established by the logarithms of real gross domestic product (x_1) and the reference interest rate EURIBOR (x_2). Standard assumptions are again required for the confounding variable u.

The logarithms of the variables are generated before the money demand function is estimated:

```
>attach(Moneydemandfnc.data)
> lnM1GERr = cbind(b1,M1GERr)
>class(lnM1GERr)
[1] "numeric"
> lnM1GERr
 [1] 6.087797 6.192710 6.277282 6.300584 6.357617 6.341364 6.431895 6.479124 6.587274 6.636419
6.690966 6.751733 6.962177 7.056891
>lnGDPr = log(GDPr)
```

```
>class(lnGDPr)
[1] "numeric"
>lnGDPr
 [1] 7.501971 7.532195 7.554644 7.571288 7.608628 7.637230 7.655220 7.688139 7.707242 7.749784
7.814343 7.840443 7.811188 7.859313
>lnEURIBOR = log(EURIBOR)
>class(EURIBOR)
[1] "numeric"
>lnEURIBOR
 [1] 1.2029723 1.2641267 1.0885620 1.4793292 1.4492692 1.1999648 0.8458683 0.7466879 0.7839015
1.1249296 1.4539530 1.5325569 0.2070142 -0.2107210
```

The ordinary least squares estimation is performed with the lm function, with the additive linking of the explanatory variables indicated by the plus sign (+):

```
>moneyfnc.lm = lm(lnM1GERr ~ lnGDPr+lnEURIBOR)
>summary(moneyfnc.lm)

Call:
lm(formula = lnM1GERr ~ lnGDPr + lnEURIBOR)

Residuals:
    Min  1Q   Median      3Q       Max
-0.085319 -0.034164  0.001443  0.025786  0.076636

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)       -8.66721          1.05730  -8.198   5.18e-06 ***
lnGDPr     1.99542    0.13571  14.704 1.41e-08 ***
lnEURIBOR   -0.14650    0.03223  -4.545 0.000837 ***
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05269  on 11  degrees  of freedom AIC:
Multiple R-squared: 0,971,    Adjusted R-squared: 0.9657
F-statistic:   184 on 2 and 11 DF,  p-value: 3.504e-09
```

Owing to the use of variables in logarithmised form, the estimated coefficients can be interpreted as elasticities. Accordingly, an increase in real GDP by 1% results in an increase in the real money demand of 2.00%. However, if the EURIBOR climbs by 1%, real money demand falls by 0.15%. The absolute member has no special economic significance; it is, instead, a technical constant that determines the location of the regression plane in the space mapped by the three variables.

All estimated regression coefficients are highly significant ($p<0.01$). Furthermore, the overall context is statistically valid ($F=182$, $p=0.0000$) with a coefficient of determination of 0.971. The standard error of regression (SER) is 0.053 on the logarithmic scale.

The real M1 money supply is to be compared with the adjusted values on the original level instead of on the logarithmic scale. For this purpose, the column vector M1GERr of the data.frame object Moneydemandfnc.data is first defined as a time series:

```
> M1GERr = ts(M1GERr, start=1997, freq=1)
> M1GERr
Time Series:
Start = 1997
End = 2010
Frequency = 1
[1]  440.45  489.17  532.34  544.89  576.87  567.57  621.35  651.40  725.80  762.36  805.10  855.54 1055.93
1160.83
```

Furthermore, the logarithmised regression values moneyfnc.lm$fitted.values contained in the lm object moneyfnc.lm are de-logarithmised using the exponential function exp and, in the process, defined as a time series:

```
> M1GERrf = ts(exp(moneyfnc.lm$fitted.values), start=1997, freq=1)
```

or

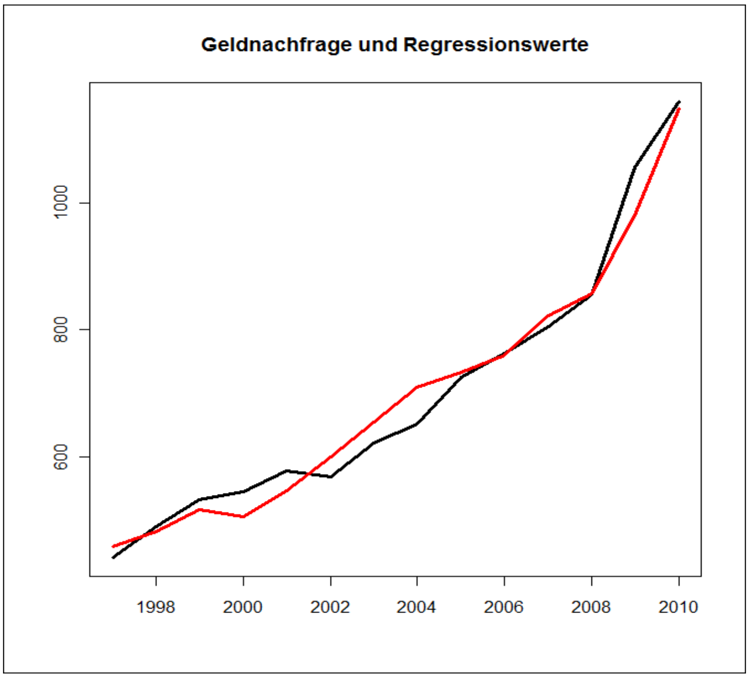
```
> M1GERrf = ts(M1GERr, start=1997, freq=1)
> M1GERf
Time Series:
Start = 1997
End = 2010
Frequency = 1
       1        2        3        4        5        6        7        8        9       10       11       12       13       14
 457.6557  481.7705  516.9682  504.6920  546.1313  599.7149  654.7324  709.4168  732.9743  759.0281
822.7575  856.8187  981.4547 1148.5570
```

Again, the plot of the multiple time series is effected with the ts.plot function:

```
>ts.plot(M1GERr,M1GERrf, main="Money demand and regression values", gpars=list(xlab="", lwd=3,
col=c("black","red")))
>box(which="figure")
```

Figure 2.3 shows an underestimation of the money demand in the first and last thirds and an overestimation of the money demand in the middle third of the observation period.

Figure 2.3: Money demand and regression values 1997 – 2010



Using the time series of the logarithmised residuals,

```
> moneyfnc.res = ts(moneyfnc.lm$residuals, start=1997, freq=1)
```

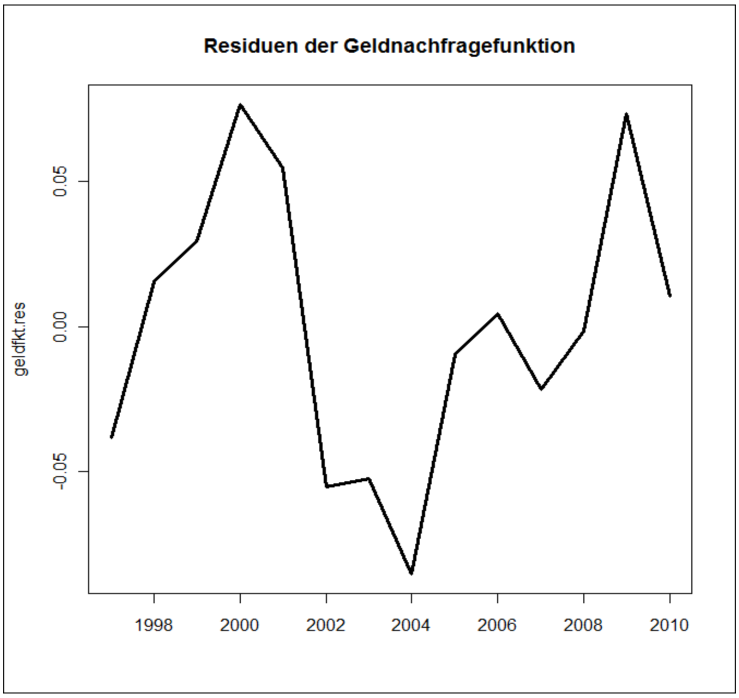
or

```
> moneyfnc.res = ts(resid(moneyfnc.lm), start=1997, freq=1)
> moneyfnc.res
Time Series:
Start = 1997
End = 2010
Frequency = 1
1 2 3 4 5 6 7 8 9 10 11 12 13
14
-0.038320129 0.015242321 0.029300955 0.076635526 0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130 0.004380105 -0.021695053 -0.001493488 0.073141298 0.010628878,
```

it is possible to generate the residuals plot:

```
>plot(moneyfnc.res, main="Residuals of the money demand function",xlab="", lwd=3)
> box(which= "figure").
```

Figure 2.4: Residuals of the money demand function



Despite the distinct cyclical trend, no residuals can be identified outside the 2·SER range (SER=0.053).

Exercises

2.2.1 Regress the demand for natural gas (GASC) against the price for natural gas (GASPR) and the price for district heat DHEATPR (refer to data.frame object EnergyI for the data) using the lm function and store the result under the name energyfnc2.lm (=energy demand model 2)!

2.2.2 Output the summarised regression output of the lm object energyfnc2.lm to the R console and interpret it!

2.2.3 Output the detailed regression output of the lm object energyfnc2.lm to the R console and interpret it!

2.2.4 Define the regression values of energy demand model 2 as the time series GASCf2 and arrange it with the time series GASC in a time series chart with the title “Natural gas consumption and regression values 1980-1995”!

2.2.5 Define the time series of residuals energyfnc2.res for the natural gas demand model 2 and plot it under the title “Residuals of natural gas demand function 2”!

2.3 Multiple regression analysis and matrix computation

In order to determine the ordinary least squares estimator (OLS estimator)

(2.3) $$\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

for the vector of regression coefficients $\boldsymbol{\beta}' = (\beta_1\ \beta_2\ \beta_3)$ of the money demand function with the help of matrix computation, the 14x3 observation matrix **X** and the 14x1 vector **y** of the dependent variables are first defined:

```
> n = length(M1GERr)
>n
[1] 14

>one = rep(1,n)
>one
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1

> X = cbind(one, lnGDPr, lnEURIBOR)
> X
one   lnGDPrlnEURIBOR
 [1,]  1 7.501971  1.2029723
 [2,]  1 7.532195  1.2641267
 [3,]  1 7.554644  1.0885620
 [4,]  1 7.571288  1.4793292
 [5,]  1 7.608628  1.4492692
 [6,]  1 7.637230  1.1999648
 [7,]  1 7.655220  0.8458683
 [8,]  1 7.688139  0.7466879
 [9,]  1 7.707242  0.7839015
[10,]  1 7.749784  1.1249296
[11,]  1 7.814343  1.4539530
[12,]  1 7.840443  1.5325569
[13,]  1 7.811188  0.2070142
[14,]   1 7.859313 -0.2107210

> y = lnM1GERr
>y
[1]  6.087797  6.192710  6.277282  6.300584  6.357617  6.341364  6.431895  6.479124  6.587274  6.636419
6.690966 6.751733 6.962177 7.056891
```

The transposed observation matrix **X‘**,

```
>Xt = t(X)


>Xt
```

```
        [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]     [,10]    [,11]    [,12]
        [,13]    [,14]
one       1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.0000000 1.000000 1.0000000 1.000000
lnGDPr   7.501971 7.532195 7.554644 7.571288 7.608628 7.637230 7.6552202 7.688139 7.7072425 7.749784
7.814343 7.840443 7.8111877  7.8593128
lnEURIBOR  1.202972  1.264127  1.088562  1.479329  1.449269  1.199965  0.8458683  0.746688  0.7839015
1.124930 1.453953 1.532557 0.2070142 -0.2107210,
```

Can be used with the operator %*% of the matrix multiplication to form the matrix product **X'X**:

```
>XtX = Xt %*% X
>XtX
one  lnGDPrlnEURIBOR
one       14.00000 107.5316  14.16841
lnGDPr   107.53163 826.1196 108.47594
lnEURIBOR  14.16841 108.4759  17.66181.
```

It is quicker to calculate the matrix product **X'X** using the crossprod function:

```
>XtX = crossprod(X)
>XtX
one  lnGDPrlnEURIBOR
one       14.00000 107.5316  14.16841
lnGDPr   107.53163 826.1196 108.47594
lnEURIBOR  14.16841 108.4759  17.66181
```

The inverse matrix (**X'X**)$^{-1}$ is formed using the solve function:

```
>XtXi = solve(XtX)
>XtXi
one       lnGDPrlnEURIBOR
one       402.705276 -51.6650087 -5.7346850
lnGDPr   -51.665009   6.6346088  0.6973148
lnEURIBOR  -5.734685   0.6973148  0.3742266.
```

After multiplying the inverse (**X'X**)$^{-1}$ from the right with vector **X'y**,

```
>Xty = Xt %*% y
>Xty
[,1]
one       91.15383
lnGDPr   700.56231
lnEURIBOR  91.06668
```

or

```
>Xty = crossprod(X,y)
>Xty
```

```
[,1]
one       91.15383
lnGDPr   700.56231
lnEURIBOR 91.06668,
```

the desired least squares estimator $\hat{\boldsymbol{\beta}}$ is returned :

```
> b = XtXi %*% Xty
> b
[,1]
one       -8.6672107
lnGDPr    1.9954165
lnEURIBOR  -0.1464953.
```

The estimated regression coefficients, which are output here with a greater level of accuracy, correspond to those of the lm function.

The main diagonal of the estimated variance-covariance matrix of the regression coefficients,

(2.4)
$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} \widehat{V}(\hat{\beta}_1) & \widehat{\mathrm{Cov}}(\hat{\beta}_1,\hat{\beta}_2) & \widehat{\mathrm{Cov}}(\hat{\beta}_1,\hat{\beta}_3) \\ \widehat{\mathrm{Cov}}(\hat{\beta}_2,\hat{\beta}_1) & \widehat{V}(\hat{\beta}_2) & \widehat{\mathrm{Cov}}(\hat{\beta}_2,\hat{\beta}_3) \\ \widehat{\mathrm{Cov}}(\hat{\beta}_3,\hat{\beta}_1) & \widehat{\mathrm{Cov}}(\hat{\beta}_3,\hat{\beta}_2) & \widehat{V}(\hat{\beta}_3) \end{bmatrix} = \hat{\sigma}^2 \cdot (\mathbf{X}'\mathbf{X})^{-1},$$

contains the estimated variances required for confidence intervals and significance tests. $\hat{\sigma}^2$ is the unbiased estimator of the variance of the confounding variable, σ^2,

(2.5)
$$\hat{\sigma}^2 = \frac{1}{n-k}\sum_{t=1}^{n}\hat{u}_t^2 \quad ,$$

defined by dividing the sum of the squared residual squares $\hat{u}_t^2$ by the number of degrees of freedom, n-k:

```
>dim(X)
[1,] 14  3
> k = dim(X)[2]
> k
[1] 3
> s2 = sum(moneyfnc.res^2)/(n-k)
> s2
[1] 0.002775921.
```

Multiplying the elements of the inverse $(\mathbf{X'X})^{-1}$ with residual variance $\hat{\sigma}^2$ produces the estimated variance-covariance matrix:

```
>covb = s2*XtXi
>covb
                  one          lnGDPr        lnEURIBOR
one          1.11787798   -0.143417975   -0.015919032
lnGDPr      -0.14341798    0.018417149    0.001935691
lnEURIBOR   -0.01591903    0.001935691    0.001038823
```

$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}})$ can be determined directly in R with the vcov function:

```
> covb = vcov(moneyfnc.lm)
> covb
              (Intercept)       lnGDPr      lnEURIBOR
(Intercept)    1.11787798   -0.143417975   -0.015919032
lnGDPr        -0.14341798    0.018417149    0.001935691
lnEURIBOR     -0.01591903    0.001935691    0.001038823
```

After extracting the estimated variance of the regression coefficients from the variance-covariance matrix $\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}})$,

```
>vb = diag(covb)
>vb
    One   lnGDPr        lnEURIBOR
1.117877977 0.018417149 0.001038823,
```

it is possible to produce the standard errors of the regression coefficients using the root function sqrt:

```
>seb = sqrt(vb)
>seb
1.05729749   0.13570980   0.03223078.
```

Dividing the regression coefficients (b) by their standard errors (seb) returns the t-values

(2.6) H_0: $ß_j = 0$, j=1,2,…,k

required for the significance tests on the null hypotheses:

```
>tb = b/seb
>tb
              [,1]
one      -8.197514
lnGDPr   14.703556
```

```
lnEURIBOR   -4.545198.
```

They are to be compared to the critical value determined from the t-distribution with n-k=14- -3=11 degrees of freedom. With a significance level α of 5%, the critical value is defined by the 0.975 quantile $t_{11;0.975}$ in a two-sided test; this can be determined in R with the quantile function qt:

```
> tkrit5 = qt(0.975, 11)
> tkrit5
[1] 2.200985
```

Since all t values of the estimated regression coefficients are absolutely larger than the critical value $t_{11;0,975}$, the null hypotheses (2.6) at the significance level of 5% must be rejected. This shows that real GDP and the EURIBOR have a significant influence on real money demand. Moreover, the technical constant is significant.

Alternatively, the significance of the independent variables on the base of the p-values van be assessed; R provides the distribution function pt for this. The p-values

```
> p.tb = (1 - pt(abs(tb), 11))*2
> p.tb
                [,1]
one       5.176329e-06
lnGDPr  1.405882e-08
lnEURIBOR   8.366998e-04
```

return the actual significance levels of the estimated regression coefficients of the money demand function.

Finally, the F-test is to be performed over the general context specified with the money demand function using the determination coefficient R^2. To this end, the determination coefficient is first calculated as the ratio of the variances of the logarithmised regression and observation values of the real money supply using the var function:

```
> lnM1GERrf = moneyfnc.lm$fitted.values
> lnM1GERrf
    1       2       3       4       5       6       7       8       9      10      11      12      13      14
6.126117         6.177468         6.247981         6.223948         6.302860         6.396454
          6.484227         6.564443         6.597111         6.632039         6.712662         6.753226
          6.889036         7.046262
> R2 = var(lnM1GERrf)/var(lnM1GERr)
> R2
[1] 0.9709833.
```

Using the R^2 value, the test statistic of the F-test,

(2.6) $$F = \frac{R^2/(k-1)}{(1-R^2)/(n-k)},$$

assumes the value

```
> F = (n-k)*R2/((1-R2)*(k-1))
> F
[1] 184.0461.
```

With a significance level of 5%, the critical value equals the 95% quantile of the F-distribution with k-1=2 and n-k=11 degrees of freedom:

```
> Fkrit5 = qf(0.95,k-1,n-k)
> Fkrit5
[1] 3.982298.
```

As F>Fkrit5, the overall context with a significance level of 5% is statistically valid.

This conclusion can also be reached on the basis of the p value,

```
>p.F = 1 - pf(F,k-1,n-k)
> p.F
[1] 3.503993e-09,
```

as it is smaller than the specified significance level α of 5%.

Exercises

2.3.1 Define observation matrix XE2 and vector ye2 for the dependent variables of the energy demand model 2 (refer to data.frame object Energyl for the data)!

2.3.2 Determine the least squares estimator be2 for the vector of the regression coefficient of energy demand model 2 using matrix computation!

2.3.3 Calculate unbiased estimator s2e3 for confounding variance σ^2 of energy demand model 2 using the sum function!

2.3.4 Define the estimated variance-covariance matrix covbe2 of least squares estimator be2!

2.3.5 Test the estimated regression coefficients of energy demand model 2 using the main diagonal elements of variance-covariance matrix covbe2 and using

- the critical value
- the p-value

For significance (α=0.05)!

2.3.6 Check the significance (α=0.05) of the overall context postulated by energy demand model 2 using

- the critical value
- the p-value

3. Multicollinearity
3.1 Regression using three explanatory variables

In order to illustrate the methods for revealing multicollinearity, we will use an additional example of the logarithmised inflation rate as explanatory variable in the money demand function:

```
>lnIrate = log(Moneydemandfnc.data$Irate)
>lnIrate
 [1] -3.963316 -4.710531 -5.115996 -4.199705 -3.963316 -4.268698 -4.605170 -4.017384 -4.199705 -4.135167
-3.772261 -3.649659 -5.521461 -4.509860
```

Real money demand is now explained by real GDP, the EURIBOR reference interest rate and the inflation rate, with all variables being measured on the logarithmic scale. The ordinary least squares estimate of the extended money demand function yields the following result:

```
> moneyfnc1.lm = lm(lnM1GERr ~ lnGDPr+lnEURIBOR+lnIrate)
>summary(moneyfnc1.lm)

Call:
lm(formula = lnM1GERr ~ lnGDPr + lnEURIBOR + lnIrate)

Residuals:
     Min  1Q   Median       3Q        Max
-0.059256 -0.019728 -0.001689  0.015969  0.074864

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)        -9.90849           1.12439  -8.812   5.00e-06 ***
lnGDPr      2.11293    0.13391  15.779 2.15e-08 ***
lnEURIBOR   -0.09649    0.03798  -2.541   0.0293 *
lnIrate     -0.06651    0.03327  -1.999   0.0735 .
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04671  on 10  degrees  of freedom:
Multiple R-squared: 0.9793,         Adjusted R-squared: 0.973
F-statistic: 157.4 on 3 and 10 DF,  p-value: 1.028e-08.
```

With an unchanged direction of influence of explanatory variables lnGDPr and lnEURIBOR, an increase in the inflation rate (lnIrate) has a negative effect on real money demand. This effect, which is significant at a 10% level, can, as with an interest rate, be interpreted as an economisation of holding money with an increasing rate of inflation.

While the overall effect of the independent variables are not affected by multicollinearity, the individual effects in the case of distinct multicollinearity are subject to uncertainty since attribution to individual regressors is more difficult.

3.2 Simple and advanced correlation analysis

Various diagnostic methods can be used to check whether multicollinearity needs to be classified as severe. The correlations between the exogenous variables are determined in simple correlation analysis. Thanks especially to the low number of regressors, this can be performed in R by calculating correlation coefficients in pairs using the cor function:

```
>r23 = cor(lnGDPr,lnEURIBOR)
> r23
[1] -0.4425414
>r24 = cor(lnGDPr,lnIrate)
> r24
[1] 0.08459758
>r34 = cor(lnEURIBOR,lnIrate)
> r34
[1] 0.5511654.
```

In the case of a larger number of response variables, a clear presentation can be achieved by outputting the correlation matrix. To this end, the original observation matrix X is extended to include the logarithmised lnIrate inflation rate:

```
> X1 = cbind(X, lnIrate)
> X1
     One  lnGDPr  lnEURIBOR     lnIrate
[1,]  1 7.501971 1.2029723 -3.963316
[2,]  1 7.532195 1.2641267 -4.710531
 [3,]  1 7.554644 1.0885620 -5.115996
 [4,]  1 7.571288 1.4793292 -4.199705
 [5,]  1 7.608628 1.4492692 -3.963316
 [6,]  1 7.637230 1.1999648 -4.268698
 [7,]  1 7.655220 0.8458683 -4.605170
 [8,]  1 7.688139 0.7466879 -4.017384
 [9,]  1 7.707242 0.7839015 -4.199705
[10,]  17.749784  1.1249296  -4.135167
[11,]  1 7.814343 1.4539530 -3.772261
[12,]  1 7.840443 1.5325569 -3.649659
[13,]  1 7.811188 0.2070142 -5.521461
[14,]  1 7.859313 -0.2107210 -4.509860
```

The correlation matrix of the exogenous variables RX1 can be generated using the cor function with the X1 without the dummy variables contained in the first column:

```
> RX1 = cor(X1[,-1])
> RX1
```

```
              lnGDPr   lnEURIBOR      lnIrate
lnGDPr   1.00000000  -0.4425414  0.08459758
lnEURIBOR -0.44254137  1.0000000 0.55116538
lnIrate   0.08459758  0.5511654   1.00000000
```

After standardising the variables with the scale function,

```
> Z1 = scale(X1[,-1])     # Standardised observation matrix (without dummy variables)
> Z1
          lnGDPr  lnEURIBOR      lnIrate
 [1,]  -1.48957145  0.3776700  0.7077189
 [2,]  -1.23786047  0.4986288 -0.7310173
 [3,]  -1.05090481  0.1513751 -1.5117267
 [4,]  -0.91228721  0.9242828  0.2525603
 [5,]  -0.60131912  0.8648263  0.7077189
 [6,]  -0.36311649  0.3717214  0.1197168
 [7,] -0.21328757 -0.3286544 -0.5281491
 [8,]  0.06086803 -0.5248255  0.6036143
 [9,]  0.21996336 -0.4512198  0.2525603
[10,]  0.57425280  0.2233075  0.3768270
[11,]  1.11191575  0.8740906  1.0755893
[12,]  1.32927906  1.0295630  1.3116560
[13,]  1.08563709 -1.5922588 -2.2924361
[14,]  1.48643103 -2.4185071 -0.3446326
attr(,"scaled:center")
  lnGDPr lnEURIBOR      lnIrate
7.680831  1.012030 -4.330873
attr(,"scaled:scale")
  lnGDPr lnEURIBOR      lnIrate
0.1200743 0.5055808 0.5193547
>class(Z1)
[1] "matrix",
```

correlation matrix RX1 can be calculated through matrix multiplication using the %*% operator,

```
> RX1 = t(Z1)%*%Z1/(n-1)
> RX1
              lnGDPr    lnEURIBOR      lnIrate
lnGDPr   1.00000000 -0.4425414 0.08459758
lnEURIBOR -0.44254137  1.0000000 0.55116538
lnIrate      0.08459758  0.5511654 1.00000000
```

or with the help of the crossprod function,

```
> RX1 = crossprod(Z1)/(n-1)
> RX1
             lnGDPr    lnEURIBOR      lnIrate
```

```
lnGDPr   1.00000000 -0.4425414 0.08459758
lnEURIBOR -0.44254137  1.0000000 0.55116538
lnIrate      0.08459758  0.5511654 1.00000000
```

According to the rule of thumb of simple correlation analysis, multicollinearity is indicated by correlation coefficients that are absolutely larger than 0.8. If all correlation coefficients are absolutely smaller than 0.8, multicollinearity in the money demand function is not severe.

In contrast, the rule of thumb of correlation analysis rests on the multiple correlation coefficients $R_{yx2x3x4}$. The determination coefficient of 0.9793 generated in the regression output yields $\sqrt{0,9793} = 0,9896$ meaning that no multicollinearity is revealed here, either.

3.3 Determinant and condition number

With exact multicollinearity, the product matrix **X'X** is singular and thus not invertible. The same applies to the RX1 correlation matrix of exogenous variables. The determinant of both matrices is in this case equal to zero. Clear multicollinearity is therefore identified by determinants of both matrices being close to zero.

Product matrix **X'X**, X1tX1, can be formed with the operator %*% for matrix multiplication,

```
> X1tX1 = t(X1) %*% X1


> X1tX1
                 one        lnGDPr   lnEURIBOR       lnIrate
one       14.00000  107.5316   14.16841   -60.63223
lnGDPr       107.53163  826.1196 108.47594 -465.63729
lnEURIBOR   14.16841  108.4759   17.66181   -59.48022
lnIrate  -60.63223 -465.6373  -59.48022  266.09699,
```

or with the help of the crossprod function,

```
> X1tX1 = crossprod(X1)
> X1tX1
                One  lnGDPr   lnEURIBOR    lnIrate
one       14.00000  107.5316   14.16841   -60.63223
lnGDPr       107.53163  826.1196 108.47594 -465.63729
lnEURIBOR   14.16841  108.4759   17.66181   -59.48022
lnIrate      -60.63223 -465.6373 -59.48022  266.09699
```

Since the value of their determinant,

```
> X1tX1.det = det(X1tX1)
> X1tX1.det
[1] 13.81835,
```

is considerably larger than zero, it does not indicate a severe multicollinearity problem.

The determinant of correlation matrix **R** also equals zero for perfect multicollinearity and equals the maximum value of one for uncorrelated variables. Using the correlation matrix RX1 calculated in Section 3.2 yields

```
> RX1.det = det(RX1)
> RX1.det
[1] 0.4519481.
```

The value of the determinant again clearly deviates from zero, meaning that no distinct multicollinearity is indicated.

The condition number of the correlation matrix RX1 requires its eigenvalues:

```
> ev.RX1 = eigen(RX1)$values
> ev.RX1
[1] 1.6670172 1.0825430 0.2504398.
```

It is the result of the square root of the ratio of the largest and smallest eigenvalues:

```
> cond.RX1 = sqrt(ev.RX1[1]/ev.RX1[3])
> cond.RX1
[1] 2.579992.
```

The rule of thumb says that severe multicollinearity exists when a condition is larger than 30, which is not the case here.

3.4 Method of auxiliary regressions

With the method of auxiliary regressions, each of the explanatory variables , x_j, j=2,3,…,k, is regressed against the other k-2 “genuine” exogenous variables and one dummy variable. The F-test is then used to check whether there is a significant statistical relationship between x_j and the other variables. The test statistics of the F-test are calculated here using the coefficient of determination of the auxiliary regressions.

The F-tests on the money demand function extended to include the logarithmised inflation for multicollinearity require the number of observations n and the number of regressors k:

```
> dim(X1)
[1] 144
```

```
> n = dim(X1)[1]
> n
[1] 14
> k = dim(X1)[2]
> k
[1] 4.
```

Three auxiliary regressions must be performed for the extended money demand function:

```
>reg2.lm = lm(lnGDPr~lnEURIBOR+lnIrate)
>lnGDPrf = reg2.lm$fitted.values
> R2.2 = var(lnGDPrf)/var(lnGDPr)
> R2.2
[1] 0.3508514
> F.2 = (n-k+1)*R2.2/((1-R2.2)*(k-2))
> F.2
[1] 2.972636

> reg3.lm = lm(lnEURIBOR~lnGDPr+lnIrate)
>lnEURIBORf = reg3.lm$fitted.values
> R2.3 = var(lnEURIBORf)/var(lnEURIBOR)
> R2.3
[1] 0.5447941
> F.3 = (n-k+1)*R2.3/((1-R2.3)*(k-2))
> F.3
[1] 6.582444

>reg4.lm = lm(lnIrate~lnGDPr+lnEURIBOR)
>lnIratef = reg4.lm$fitted.values
> R2.4 = var(lnIratef)/var(lnIrate)
> R2.4
[1] 0.4379853
> F.4 = (n-k+1)*R2.4/((1-R2.4)*(k-2))
> F.4
[1] 4.286221.
```

As can be seen, with a significance level of 5%, only the value of the test statistic F.2 is not significant,

```
> Fkrit5 = qf(0.95,k-2,n-k+1)
> Fkrit5
[1] 3.982298,
```

while at the 1% level no significant F-values occur:

```
> Fkrit1 = qf(0.99,k-2,n-k+1)
> Fkrit1
[1] 7.205713.
```

Under the strict assumptions of the F-test, a potential multicollinearity of the extended money demand function is therefore only insignificant if a higher error of the 2nd type is acceptable.

In order to avoid the strict assumptions, it is however often necessary to draw on tolerance coefficients and variance inflation factors as exploratory instruments. Since the tolerance coefficients tol2, tol3 and tol4 do not fall below the threshold value of 0.10,

```
> tol2 = 1 - R2.2
> tol2
[1] 0.6491486

> tol3 = 1 - R2.3
> tol3
[1] 0.4552059

> tol4 = 1 - R2.4
> tol4
[1] 0.562014,
```

severe multicollinearity of the extended money demand function cannot be inferred. However, the fact that multicollinearity is not completely absent is indicated by the variance inflation factors VIF2, VIF3 and VIF4. Compared to a situation of uncorrelated independent variables, the variances of the estimated regression coefficients are 1.5 to 2.2 times higher:

```
> VIF2 = 1/tol2

> VIF2
[1] 1.540479

> VIF3 = 1/tol3
> VIF3
[1] 2.196808

> VIF4 = 1/tol4
> VIF4
[1] 1.779313.
```

Exercises

3.1 Regress the demand for natural gas (GASC) against the price for natural gas (GASPR), the price for district heat DHEATPR and disposable income (DISPR) refer to (data.frame object EnergyI for the data) using the lm function and store the result under the name energyfnc3.lm (=energy demand model 3)!

3.2 Output the detailed regression output of the lm object energyfnc3.lm to the R console and interpret it!

3.3 Assess the extent of multicollinearity in the energy demand model 3 based on a simple and an extended correlation analysis!

3.4 Can severe multicollinearity be inferred in the energy demand model 3 based on determinant criterion applied to product matrix **X'X** and correlation matrix **R**?

3.5 Do you need to revise the assessment you made in Exercise 3.4 owing to the condition number of correlation matrix **R**?

3.6 Assess the extent of multicollinearity in the energy demand model 3 using the auxiliary regression method on the basis of

- The F-test (α=0.05),
- The tolerance coefficients

3.7 Determine the variance inflation coefficients for the energy demand model 3 and interpret them

4. Heteroscedasticity

4.1 Testing for heteroscedasticity

Constancy of the confounding variable u (homoscedasticity) was assumed in the least squares estimation of the econometric one-equation model. Heteroscedasticity is present if this condition does not apply. The regression coefficients estimated using OLS are no longer efficient in this case. Exaggerated variance estimators involve a loss of precision in the estimated regression coefficients. As a result, the significance tests of the regression coefficients cease to be valid. In addition, the confidence intervals they are used to form are biased.

4.1.1 Goldfeld-Quandt test

Tests for heteroscedasticity can be used to check whether the assumption of homoscedasticity is violated in an econometric model. The Goldfeld-Quandt test examines whether heteroscedasticity is present in groups:

H_0: $\sigma_1^2 = \sigma_2^2$ and H_1: $\sigma_1^2 \neq \sigma_2^2$.

In order to test this for the money demand function, the observations of variables are arranged according to the values of the exogenous variables lnGDPr in ascending order. To this end, the observation matrix **X** is first extended to include vector **y** of the independent variables,

```
>yX = cbind(y,X)
>yX
          one      lnGDPr   lnEURIBOR
[1,] 6.087797   1 7.501971  1.2029723
[2,] 6.192710   1 7.532195  1.2641267
[3,] 6.277282   1 7.554644  1.0885620
 [4,] 6.300584   1 7.571288  1.4793292
 [5,] 6.357617  1 7.608628  1.4492692
 [6,] 6.341364   1 7.637230  1.1999648
 [7,] 6.431895   1 7.655220  0.8458683
 [8,] 6.479124   1 7.688139  0.7466879
 [9,] 6.587274   1 7.707242  0.7839015
[10,] 6.636419   1 7.749784  1.1249296
[11,] 6.690966   1 7.814343  1.4539530
[12,] 6.751733   1 7.840443  1.5325569
[13,] 6.962177   1 7.811188  0.2070142
[14,] 7.056891   1 7.859313 -0.2107210,
```

and the general name y for the independent variable is replaced by the variable name lnM1GERr:

```
>colnames(yX)
[1] "y"  "one"   "lnGDPr"        "lnEURIBOR"
>colnames(yX)[[1]] = "lnM1GERr"
>colnames(yX)
[1] "lnM1GERr"  "one"    "lnGDPr"         "lnEURIBOR".
```

The order function is used to sort the variables contained in the matrix yX:

```
>yX.order = yX[order(lnGDPr),]
>yX.order
     lnM1GERr one  lnGDPr          lnEURIBOR
 [1,] 6.087797   1 7.501971  1.2029723
 [2,] 6.192710   1 7.532195  1.2641267
 [3,] 6.277282   1 7.554644  1.0885620
 [4,] 6.300584   1 7.571288  1.4793292
```

```
 [5,] 6.357617  1 7.608628  1.4492692
 [6,] 6.341364  1 7.637230  1.1999648
 [7,] 6.431895  1 7.655220  0.8458683
 [8,] 6.479124  1 7.688139  0.7466879
 [9,] 6.587274  1 7.707242  0.7839015
[10,] 6.636419  1 7.749784  1.1249296
[11,] 6.962177  1 7.811188  0.2070142
[12,] 6.690966  1 7.814343  1.4539530
[13,] 6.751733 1 7.840443  1.5325569
[14,] 7.056891  1 7.859313  -0.2107210
```

The two samples with the smallest and largest lnGDPr values are defined by extracting the two median observations with sequence numbers 7 and 8 (c=2):

```
> y1 = yX.order[1:6,1]
> y1
[1] 6.087797 6.192710 6.277282 6.300584 6.357617 6.341364
> X1 = yX.order[1:6,3:4]
> X1
     lnGDPr    lnEURIBOR
[1,] 7.501971  1.202972
[2,] 7.532195  1.264127
[3,] 7.554644  1.088562
[4,] 7.571288  1.479329
[5,] 7.608628  1.449269
[6,] 7.637230  1.199965

> y2 = yX.order[9:14,1]
> y2
[1] 6.587274 6.636419 6.962177 6.690966 6.751733 7.056891
> X2 = yX.order[9:14,3:4]
> X2
     lnGDPr    lnEURIBOR
[1,] 7.707242  0.7839015
[2,] 7.749784  1.1249296
[3,] 7.811188  0.2070142
[4,] 7.814343  1.4539530
[5,] 7.840443  1.5325569
[6,] 7.859313 -0.2107210
```

The vector y1 (y2) contains the values of the endogenous variables of the smallest (largest) lnGDPr values (1st sample and 2nd sample). Accordingly, the matrix X1 (X2) contains the values of the exogenous variables (without absolute member) of the smallest (largest) lnGDPr values (1st sample and 2nd sample).

The estimator results for the 1st sample are

```
> moneyfnc1.lm = lm(y1 ~ X1)
>summary(moneyfnc1.lm)

Call:
lm(formula = y1 ~ X1)

Residuals:
       1          2          3          4          5          6
-0.0450976 -0.0007379  0.0586455  0.0166274  0.0083077 -0.0377451

Coefficients:
              Estimate  Std. Error t value Pr(>|t|)
(Intercept)   -7.64581    3.43841  -2.224   0.1127
X1lnGDPr       1.82232    0.46046   3.958   0.0288 *
X1lnEURIBOR    0.08954    0.14874   0.602   0.5897
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04914  on 3  degrees  of freedom:
Multiple R-squared: 0.8614,        Adjusted R-squared: 0.7691
F-statistic: 9,325 on 2 and 3 DF,  p-value: 0.05158
```

and for the 2nd sample

```
> moneyfnc2.lm = lm(y2 ~ X2)
>summary(moneyfnc.lm)

Call:
lm(formula = y2 ~ X2)

Residuals:
       1         2         3         4         5         6
-0.015639  0.005879  0.046680 -0.014262  0.006853 -0.029509


Coefficients:
              Estimate  Std. Error t value Pr(>|t|)
```

```
(Intercept)        -9.00394          2.14449  -4.199  0.02465 *
X2lnGDPr            2.04264          0.27453   7.441  0.00502 **
X2lnEURIBOR -0.17380     0.02261  -7.688  0.00457 **
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03454  on 3  degrees  of freedom:
Multiple R-squared: 0.9797,         Adjusted R-squared: 0.9662
F-statistic:  72.4 on 2 and 3 DF,  p-value: 0.002892.
```

Using the residual square sums of the two samples,

```
> sq1 = sum(moneyfnc1.lm$resid^2)
> sq1
[1] 0.007243815
> sq2 = sum(moneyfnc2.lm$resid^2)
> sq2
[1] 0.003579361,
```

it is possible to calculate the values for the test statistic GQ of the Goldfeld-Quandt test

```
> GQ = sq1/sq2
> GQ
[1] 2.023773.
```

Using the values n, k and c,

```
> n = dim(yX)[1]
> n
[1] 14
> k = dim(yX)[2]-1
> k
[1] 3
> c = 2
> c
[1] 2
```

it is possible to calculate the number of degrees of freedom (n-c)/2-k required for the critical value of the F-distribution. A significance level of 5% results in the critical F-value

```
> Fkrit5 = qf(0.95,(n-c)/2-k,(n-c)/2-k)
> Fkrit5
[1] 9.276628.
```

Because GQ < Fkrit5, the null hypothesis of homoscedasticity at the 5% significance level cannot be dismissed using the Goldfeld-Quandt test. Alternatively, the test decision can be reached by a comparison of the p-value,

```
> p.GQ = 1 - pf(GQ,(n-c)/2-k,(n-c)/2-k)
> p.GQ
[1] 0.2886486,
```

with the nominal significance level α=0.05 . p.GQ >α means that the null hypothesis of homoscedasticity can be maintained for the money demand function.

The Goldfeld-Quandt test shown here in detail can be performed direct with the R function gqtest after calling program library lmtest in the zoo package,

```
>library(lmtest),
```

and this must be specified as follows:

```
> moneyfnc.gq = gqtest(moneyfnc.lm, fraction=2, alternative = "less", order.by=~ lnGDPr).
```

The test result is stored in the htest object moneyfnc.gq,

```
>class(moneyfnc.DW)
[1] "htest",
```

which has the following structure:

```
> str(moneyfnc.gq)
List of 5
 $ statistic: Named num 0.494
  ..- attr(*, "names")= chr "GQ"
 $ parameter: Named num [1:2] 3 3
  ..- attr(*, "names")= chr [1:2] "df1" "df2"
 $ method   : chr "Goldfeld-Quandt test"
 $ p.value  :num 0.289
 $ data.name: chr "geldfkt.lm"
 - attr(*, "class")= chr "htest".
```

This already yields the value of the test statistic and the p-value. Calling the htest object moneyfnc.gq returns the entire test result with an indication of the number of degrees of freedom:

```
>moneyfnc.gq

	Goldfeld-Quandt test

data:  moneyfnc.lm
GQ = 0.4941, df1 = 3, df2 = 3, p-value = 0.2886.
```

The reported p-value of 0.2886 with 3 degrees of freedom in the numerator and the denominator matches that of the detailed calculation. The GQ test statistic of 0.4941 was calculated as the ratio between the residual variances of the second and first samples, sq2/sq1. It therefore exactly equals the reciprocal of the rounded test statistic yielded by the detailed calculation: 2.0238 (=1/0.4941).[2]

4.1.2 Breusch-Pagan test

The Breusch-Pagan test is used to test whether confounding variance is a function of observable variables n $z_2, z_3, \ldots, z_p$:

$$(4.1)\quad \sigma_t^2 = h\left(\alpha_1 + \alpha_2 \cdot z_{2t} + \ldots + \alpha_p \cdot z_{pt}\right).$$

The null hypothesis of homoscedasticity,

$$(4.2)\quad H_0: \alpha_2 = \alpha_3 = \ldots = \alpha_p = 0,$$

is dismissed if at least one α_j is unequal 0:

H_1: at least one $\alpha_j \neq 0$, j=2,3,…,p.

In the money demand function, the confounding variance is considered under the alternative hypothesis as a linear function of logarithmised real GDP, lnGDP:

$$(4.3)\quad \sigma_t^2 = \alpha_1 + \alpha_2 \cdot \mathrm{lnBIPr}_t.$$

Using the maximum likelihood estimator (ML estimator)

[2] In econometric textbooks, the test statistic of the Goldfeld-Quandt test is generally defined as the ratio of the larger and smaller residual variances of both test statistics. Cf. Dreger/Kosfeld/Eckey (2014, p. 87f.).

$$\hat{\sigma}^2_{ML} = \frac{1}{n}\sum_{t=1}^{n}\hat{u}^2_t$$

for the confounding variance σ^2 of the money demand function

```
> s2ML = sum(moneyfnc.resl^2)/n
> s2ML
[1] 0.002181081,
```

it is possible to operationalise the dependent variable $\hat{u}^2_t/\hat{\sigma}^2_{ML}$ of the auxiliary:

```
>  u2s2ML = moneyfnc.res^2/s2ML
> u2s2ML
Time Series:
Start = 1997
End = 2010
Frequency = 1
       1 2          3          4          5          6          7          8          9          10         11         12         13
          14
0.673259049     0.106519840     0.393633287     2.692703651     1.374720572     1.391484020
          1.255636746     3.337517679     0.044358498     0.008796244     0.215799134     0.001022661
          2.452751793     0.051796826.
```

The auxiliary regression of the dependent variables $\hat{u}^2_t/\hat{\sigma}^2_{ML}$, u2s2ML against the independent variable lnGDPr,

```
>auxreg.bp = lm(u2s2ML ~ lnGDPr),
```

yields the estimated result

```
>summary(auxreg.bp) Call:

Call:
lm(formula = u2s2ML ~ lnGDPr)

Residuals:
   Min    1Q  Median       3Q        Max
-1.1225 -0.7889 -0.5904  0.3091  2.3488
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   12,837   20,510   0.626         0.543
lnGDPr   -1.541    2.670  -0.577      0.574

Residual standard error: 1,156  on 12  degrees  of freedom:
Multiple R-squared: 0.02701,    Adjusted R-squared: -0.05407
F-statistic: 0.3331 on 1 and 12 DF,  p-value: 0.5745.
```

The difference between the total variation square sum SST of the dependent variables u2s2ML of auxiliary regression (4.3),

```
>SST.bp = sum((u2s2ML-mean(u2s2ML))^2)
>SST.bp
[1] 16.47925,
```

and the residual variation square sum SSR,

```
>SSR.bp = sum(auxreg.bp$res^2)
>SSR.bp
[1] 16.03410,
```

results in the variation square sum SSE:

```
>SSE.bp = SST.bp - SSR.bp
>SSE.bp
[1] 0.4451416.
```

The test statistic of the Breusch-Pagan test is therefore

(4.4) $$BP_1 = \frac{1}{2} \cdot SSE,$$

```
> BP1 = SSE.bp/2
> BP1
[1] 0.2225708,
```

which is χ^2- distributed asymptotically with 1 (=p-1) degree of freedom. Because

```
> chikrit5 = qchisq(0.95, 1)

> chikrit5
```

```
[1] 3.841459
```

$\chi^2_{1;\,0.95}$ $BP_1 < \chi^2_{0.95}$ is valid, so that the null hypothesis (4.2) at the 5% significance level cannot be dismissed. This test outcome can similarly be achieved with the p-value,

```
> p.BP1 = 1 - pchisq(BP1, 1)
> p.BP1
[1] 0.637088,
```

because p=0.637>α=0.05. Thus, as with the Goldfeld-Quandt test, the Breusch-Pagan test does not reveal any heteroscedasticity, either.

The Breusch-Pagan test is also directly available in the lmtest program library of the zoo. Specifying

```
>moneyfnc.bp = bptest(moneyfnc.lm, varformula= ~ lnGDPr, studentize=FALSE)
```

causes the test result to be output to the R console by entering the htest object name moneyfnc.bp and confirmation with the Return key,

```
>moneyfnc.bp

Breusch-Pagan test

data:  moneyfnc.lm
BP = 0.2226, df = 1, p-value = 0.6371,
```

Which, after allowing for rounding, matches the result of the detailed calculation.

Exercises

4.1.1 Use a detailed calculation to compute the test statistic of the Goldfeld-Quandt test for the demand for natural gas (GASC) depending on the price of natural gas (GASPR) and disposable income (DISPIR) (=energy demand model 2a)! Test its significance (α=0,05) using

- the critical value,
- the p-value!

4.1.2 Perform the Goldfeld-Quandt test on the energy demand model 2a using the functions available in program library lmtest and interpret the result!

4.1.3 Use a detailed calculation to compute the test statistic of the Breusch-Pagan test for the demand for natural gas (GASC) depending on the price of natural gas (GASPR) and disposable income (DISPIR) (=energy demand model 2a)! Test its significance (α=0,05) using

- the critical value,
- the p-value!

4.1.4 Perform the Breusch-Pagan test on the energy demand model 2a using the functions available in program library lmtest and interpret the result!

4.2 Weighted least squares (WLS) estimation

In the event heteroscedasticity, parameters of the multiple regression model can be estimated using the weighted least squares (WLS). To this end, an estimate of the dependence of variance σ_t^2 of the squared values of an exogenous variable z is selected:

(4.4) $$\sigma_t^2 = \sigma^2 \cdot z_t^2, \quad t = 1,2,...,n.$$

With the WLS method, the variables of a regression model are multiplied with the inverse observation values $1/z_t$ before the transformed OLS model is estimated. If, for example, heteroscedasticity of the money demand function can be ascribed to the real gross domestic product x_2, instead of the transformed model

(4.5a) $$y_t/x_{2t} = \beta_1 \cdot 1/x_{2t} + \beta_2 \cdot x_{2t}/x_{2t} + \beta_3 \cdot x_{3t}/x_{2t} + u_t/x_{2t}$$

or

(4.5b) $$y_t^* = \beta_1 \cdot x_{1t}^* + \beta_2 + \beta_3 \cdot x_{3t}^* + u_t^*$$

with $y_t^* = y_t/x_{2t}$, $x_{1t}^* = 1/x_{2t}$, $x_{3t}^* = x_{3t}/x_{2t}$, $u_t^* = u_t/x_{2t}$.

for the estimation.

The transformed variables are identified by the suffix s:

```
> lnM1GERrs = lnM1GERr/lnGDPr
>Intercepts = 1/lnGDPr
>lnEURIBORs = lnEURIBOR/lnGDPr.
```

The WLS estimate of the money demand results from its specification using the transformed variables lnM1GERrs, Intercepts and lnEURIBORs in the lm function:

```
>moneyfnc.wls = lm(lnM1GERrs ~ Intercepts+lnEURIBORs)
>summary(moneyfnc.wls)
```

```
Call:
lm(formula = lnM1GERrs ~ Intercepts + lnEURIBORs)

Residuals:
      Min 1Q   Median       3Q        Max
-0.0110534 -0.0045686  0.0001382  0.0033927  0.0100421

Coefficients:
              Estimate  Std. Error t value Pr(>|t|)
(Intercept)        1.99529 0.13652  14.615   1.50e-08 ***
Intercepts        -8.66753          1.06358   -8.149   5.48e-06 ***
lnEURIBORs  -0.14523      0.03285   -4.422   0.00103 **
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.006879  on 11  degrees  of freedom:
Multiple R-squared: 0.9334,        Adjusted R-squared: 0.9213
F-statistic: 77.06 on 2 and 11 DF,  p-value: 3.387e-07.
```

The WLS estimate differs only marginally from the OLS estimate as the Goldfeld-Quandt test and the Breusch-Pagan test revealed no heteroscedasticity in the money demand function, so that its use here was only for illustration purposes.

When identifying estimated regression coefficients, it is important to note that the absolute member (Intercept) of 1.99529 of the WLS model indicates income elasticity, as the model variables were multiplied with the reciprocals of the logarithmised real GDP. In contrast, the estimated regression coefficient of the variable lnGDPRs-8.66753, describes the new absolute member. On the other hand, identifying interest rate elasticity of -0.14523 as estimated regression coefficient of the variable lnEURIBORs is unproblematic.

Exercises

4.2.1 Transform the variables of the regression of the natural gas demand (GASC) against the natural gas price (GASPR) and disposable income (DISPIR) (=energy demand model 2a, for data refer to data.frame object Energyl) using approach (4.4) with z_t=$DISPIR_t$!

4.2.2 Estimate energy demand model 2a using approach (4.4) with z_t=$DISPIR_t$ with the weighted least squares (WLS) method and store the result under energyfnc2a.wls!

4.2.3 Compare the OLS and WLS estimators for energy demand model 2a! How can the result be interpreted?

4.2.4 Calculate the determination coefficients for the quality of the WLS estimate of energy model 2a for the original model variables!

4.3 Heteroscedasticity-consistent (HC) standard errors

Since, with heteroscedasticity, it is not the OLS estimators of the regressions coefficients that are distorted, but their standard errors, this would be the very point to repair the defect in the model. For this purpose, White (1980) proposed substituting the variance-covariance matrix of the estimated coefficients (2.4) with

(4.6) $$\hat{C}\mathrm{ov}(\hat{\boldsymbol{\beta}}) = (\mathbf{X'X})^{-1} \cdot \mathbf{X'} \cdot \mathrm{diag}(\hat{u}_1^2, \hat{u}_2^2, ..., \hat{u}_n^2) \cdot \mathbf{X} \cdot (\mathbf{X'X})^{-1}$$

where, instead of the residual variance $\hat{\sigma}^2$ the squared $\hat{u}_1^2, \hat{u}_2^2, ..., \hat{u}_n^2$ are used. In this was, the varying scatter of the confounding variable u for various cross-section time units t are taken into account. The roots of the diagonal elements of the estimated variance-covariance matrix (4.6) are called heteroscedasticity-consistent (HC) standard errors.

White's original HC estimator of the covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ for the estimated regression coefficients of the money demand function using matrix calculus if the time series of the squared residuals,

```
> u2 = moneyfnc.res^2
> u2
Time Series:
Start = 1997
End = 2010
Frequency = 1
           1            2            3            4            5            6
1.468432e-03  2.323284e-04  8.585460e-04  5.873004e-03  2.998376e-03  3.034939e-03
           7  8  9 10  11 12
2.738645e-03  7.279395e-03  9.674946e-05  1.918532e-05  4.706753e-04  2.230506e-06
          13  14
5.349650e-03  1.129731e-04,
```

is transformed into a diagonal matrix:

```
> U2 = diag(u2):

> covb.White = XtXi%*%Xt%*%U2%*%X%*%XtXi
> covb.White
                      one       lnGDPr    lnEURIBOR
one        0.295919018   -0.0382850964    0.0012679068
lnGDPr    -0.038285096    0.0049663856  -0.0002368461
lnEURIBOR  0.001267907   -0.0002368461   0.0004711442.
```

The main diagonal elements of the matrix covb.White contain White's original HC estimators of the variances $\widehat{\mathrm{Var}}(\hat{\beta}_j)$, j=1, 2, 3, which can be used with existing heteroscedasticity instead of the variance estimator determined under the assumption of homoscedasticity for significance tests and confidence intervals.

However, the car and sandwich packages in R contain functions for calculating various HC estimators of the variance-covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$. In the car package,

```
>library(car)
Load required package: MASS
Load required package: nnet
Load required package: survival
Load required package: splines,
```

the function is called hccm. The original White estimator (4.6) can be generated with this b y specifying type="hc0":

```
> covb.hc0 = hccm(moneyfnc.lm, type="hc0")
> covb.hc0
              One          lnGDPr          lnEURIBOR
(Intercept)     0.295919018   2.77990  -0.0382850964   0.0012679068
lnGDPr   -0.038285096   0.0049663856  -0.0002368461
lnEURIBOR   0.001267907    -0.0002368461  0.0004711442.
```

White's original HC estimator (HC0 estimator) possesses the asymptotic property of consistency. Nevertheless, these properties can be improved in a modified form for small samples. The "hc1"-type estimator is adjusted to the finite sample size by multiplying the right side of (4.6) with the factor n/(n-k):

```
> covb.hc1 = hccm(moneyfnc.lm, type="hc1")

> covb.hc1
             (Intercept)          lnGDPr          lnEURIBOR
(Intercept)  0.37662420  -0.0487264863    0.0016136996
lnGDPr   -0.04872649   0.0063208544  -0.0003014405
lnEURIBOR  0.00161370   -0.0003014405  0.0005996381.
```

A presetting is made using the hccm function of the "hc3"-type HC selector in such a way that the commands

```
> covb.hc3 = hccm(moneyfnc.lm)
```

and

```
> covb.hc3 = hccm(moneyfnc.lm, type="hc3")
```

yield the same result:

```
> covb.hc3
               (Intercept)          lnGDPr  lnEURIBOR
(Intercept)       0.50161377   2.77990  -0.0651046342   0.0021542700
lnGDPr   -0.06510463   0.0084731370  -0.0004095327
lnEURIBOR    0.00215427     -0.0004095327  0.0008405830.
```

The HC3 estimator results from (4.6) by substituting the residual squares $\hat{u}_t^2$ with the values $\hat{u}_t^2/(1-h_{tt})^2$. Here, the h_{tt} values are the main diagonal elements of the so-called hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ which transforms the vector of observed values of the dependent variables, **y**, into the vector of the regression values $\hat{\mathbf{y}}$. The diagonal elements h_{tt} indicate the influence of the t-th observation of the independent variables (leverage) on the relevant regression value.

Alternatively, heteroscedasticity-consistent (HC) estimators of the variance-covariance matrix of the OLS-estimated regression coefficients can be generated with the vcovHC function, which is available in R after the sandwich package is loaded:

```
> library(sandwich)
Load required package: zoo.
```

When the lm object moneyfnc.lm is specified, the vcovHC function can be used to calculate various HC estimators specifying the type argument, e.g. type="HC0". The HC3 estimator is preset once more. The uncorrected estimator is requested by specifying type="const".

For the significance tests of the regression coefficients of the money demand function, taking account of potentially existing heteroscedasticity, use is made here of the HC3 estimator of the variance-covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$. For this purpose, the main diagonal elements (=estimated variances of $\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_k$) are extracted from the HC estimator covb.hc3:

```
>vb.hc3 = diag(covb.hc3)
>vb.hc3
  (Intercept)      lnGDPr  lnEURIBOR
```

0.501613769 0.008473137 0.000840583.

The resultant heteroscedasticity-consistent (HC) standard errors of the regression coefficients,

```
>seb.hc3 = sqrt(vb.hc3)
>seb.hc3
 (Intercept)      lnGDPr  lnEURIBOR
0.70824697  0.09204964  0.02899281
```

differ from the values calculated under the assumption of homoscedasticity (= confounding variance for all observations:

```
>seb = sqrt(vb)
>seb
 (Intercept)      lnGDPr  lnEURIBOR
1.05729749  0.13570980  0.03223078.
```

While the estimated standard error of the regression coefficients of interest rate variables decreases by a factor of 1.11, the HC3 estimator of the standard error falls by a factor of 1.47 compared with the uncorrected estimators for the income variables.

A comparison of the t-values calculated with the HC standard errors

```
> tb.hc3 = b/seb.hc3
> tb.hc3
                 [,1]
one        -12.237554
lnGDPr   21.677613
lnEURIBOR  -5.052814
```

with the 0.975 quantile of the t-distribution with 11 (=n-k=14-3) degrees of freedom,

```
> tkrit5 = qt(0.975, 11)
> tkrit5
[1] 2.200985
```

shows that all regression coefficients are significant at the 5% level, similar to the use of the standard errors of the OLS model. Their actual significance is revealed by the p-values:

```
>  p.tbhc = (1 - pt(abs(tb.hc3), 11))*2
>  p.tbhc
                 [,1]
```

```
one       9.507521e-08
lnGDPr  2.248521e-10
lnEURIBOR   3.704356e-04..
```

However, stepwise calculation is not necessary in order to perform the significance test via the regression coefficients with a HC estimator of the variance-covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$. Using the coeftest function included in the lmtest R package,

```
> library(lmtest),
```

it is possible to generate test output similar to the original OLS estimate. Since the HC estimator on which the significance test is based is selected and calculated via the vcov argument with function vcovHC, the R package sandwich must also be loaded, unless this has already been done:

```
> library(sandwich)
Load required package: zoo.
```

The argument df is used to define the number of degrees of freedom when requesting t-tests of the estimated regression coefficients. However, if df=Inf is specified, the significance tests are performed with normal approximation (z-tests).

Significance tests in the form of t-tests of the regression coefficients of the money demand function are performed using the type-HC3 standard error by specifying coeftest function (n=14, k=3)

```
> coeftest.HC3 = coeftest(moneyfnc.lm, df=n-k, vcov=vcovHC(moneyfnc.lm, type="HC3"))
```

or based on the presetting of the HC3 estimators in function vcovHC

```
> coeftest.HC3 = coeftest(moneyfnc.lm, vcov=vcovHC(moneyfnc.lm))
```

```
> coeftest.HC3

t test of coefficients:

              Estimate    Std. Error    t value    Pr(>|t|)
(Intercept)       -8.667211        0.708247  -12.2376   9.508e-08 ***
lnGDPr        1.995417    0.092050   21.6776   2.249e-10 ***
lnEURIBOR   -0.146495    0.028993  -5.0528 0.0003704 ***
```

```
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

The test results match those of the stepwise tests using the HC3 estimator of the variance-covariance matrix of the estimated regression coefficients.

Exercises

4.3.1 Calculate the heteroscedasticity-consistent variance-covariance matrix of the regression coefficient of energy demand model 2a using formula (4.6). Check the result using the `hccm` function.

4.3.2 Calculate the HC1 estimator for $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ of the energy demand model 2a using the `hccm` and `vcovHC` function!

4.3.3 Compare the HC3 standard error of the regression coefficient of energy demand model 2a with the OLS standard errors. Characterise the differences!

4.3.4 Test the significance of the regression coefficients of energy demand model 2a (α=0.05)

- step by step with function `vcovHC`,
- with function `coeftest`

using the HC3 standard error!

5. Autocorrelation

5.1 Autocorrelation coefficient and correlogram

Autocorrelation means the correlation of values in a time series (x_t). If the sequence of the times series values is consecutive, we speak of first-order autocorrelation. In this case, the correlation between n-1 value pairs (x_2, x_1), (x_3, x_2), …, (x_n, x_{n-1}) is measured similarly to the Pearson correlation coefficient:

$$r_1 = \frac{\sum_{t=2}^{n}(x_t - \bar{x})(x_{t-1} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \quad (5.1)$$

The covariance of the consecutive values relates to the variance of the time series, with the factor 1/n in equation (5.1) being neglected. In addition, the mean value $\bar{x}$ of the entire time series (x_t) is used in (5.1) instead of the mean values of the two partial series in order to calculate the first-order autocorrelation coefficient, r_1.

Econometrics examines whether the residuals $\hat{u}_t$, t=1,2,…,n, of a regression autocorrelate. The empirical coefficient of first-order autocorrelation of residuals, $\hat{\rho}_1$, can be calculated from the n-1 value pairs $(\hat{u}_2, \hat{u}_1), (\hat{u}_3, \hat{u}_2), ..., (\hat{u}_n, \hat{u}_{n-1})$. Because $\bar{\hat{u}} = 0$ it can be defined by

(5.2)
$$\hat{\rho}_1 = \frac{\sum_{t=2}^{n} \hat{u}_t \cdot \hat{u}_{t-1}}{\sum_{t=1}^{n} \hat{u}_t^2}$$

. The designation $\hat{\rho}_1$ identifies an estimate of the unknown theoretical coefficient of first-order autocorrelation, ρ_1, of the confounding variable u_t.

In order to determine the empirical coefficient of second-order correlation of the residuals, $\hat{\rho}_2$,n-2 ordered pairs $(\hat{u}_3, \hat{u}_1), (\hat{u}_4, \hat{u}_2), ..., (\hat{u}_n, \hat{u}_{n-2})$ are compared with each other:

(5.3)
$$\hat{\rho}_2 = \frac{\sum_{t=3}^{n} \hat{u}_t \cdot \hat{u}_{t-2}}{\sum_{t=1}^{n} \hat{u}_t^2} \quad .$$

In general the coefficient of the k-th order of the residuals, $\hat{\rho}_k$, the correlation between the n-k value pairs $(\hat{u}_{k+1}, \hat{u}_1), (\hat{u}_{k+2}, \hat{u}_2), ..., (\hat{u}_n, \hat{u}_{n-k})$ is defined by

(5.4)
$$\hat{\rho}_k = \frac{\sum_{t=k+1}^{n} \hat{u}_t \cdot \hat{u}_{t-k}}{\sum_{t=1}^{n} \hat{u}_t^2}$$

.

In the following example, the first- to fourth-order autocorrelation coefficients of the residuals of the money demand function are to be calculated. For this purpose, the time series of residuals u is first defined:

```
>u =ts(moneyfnc.lm$residuals, start=1997, freq=1)
Time Series:
Start = 1997
End = 2010
Frequency = 1
1    2    3    4    5    6    7    8    9    10    11    12
```

-0.038320129 0.015242321 0.029300955 0.076635526 0.054757433 -0.055090280 -0.052332065 -0.085319372 -0.009836130 0.004380105 -0.021695053 -0.001493488
13 14
0.073141298 0.010628878.

The time series of residuals lagged by one period, u_1, is defined by

```
> u_1 = lag(u, -1)
> u_1
Time Series:
Start = 1998
End = 2011
Frequency = 1
       1          2          3          4          5          6          7          8          9         10         11         12
13         14
-0.038320129 0.015242321 0.029300955 0.076635526 0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130 0.004380105 -0.021695053 -0.001493488 0.073141298 0.010628878
```

. According to equation (5.2), the result for the numerator of the first-order autocorrelation coefficient is

> r1.Z = sum(u*u_1, na.rm=TRUE)   or   r1.Z = sum(u*u_1)
> r1.Z
[1] 0.01203739

and for the denominator

```
> r.N = sum(u^2)
> r.N
[1] 0.03053513.
```

$\hat{\rho}_1$ thus assumes the value

```
> r1 = r1.Z/r.N
> r1
[1] 0.3942144
```

[3].

Using the residual series u_2, which is lagged by two periods,

```
> u_2 = lag(u, -2)
> u_2
Time Series:
Start = 1999
```

[3] Calculating autocorrelation coefficients of residuals direct with the cor function returns slightly different results, as in this case the standard mean value of zero for the entire residual series is not used for both time series; instead, the calculation uses the mean values of the relevant lag series.

```
End = 2012
Frequency = 1
        1          2          3          4          5          6          7          8          9         10         11         12
13         14
-0.038320129  0.015242321  0.029300955  0.076635526  0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130  0.004380105 -0.021695053 -0.001493488  0.073141298  0.010628878,
```

returns the second-order autocorrelation coefficient $\hat{\rho}_2$:

```
> r2.Z = sum(u*u_2, na.rm=TRUE)    or    r1.Z = sum(u*u_2)
> r2.Z
[1] -0.001992230
> r2 = r2.Z/r.N
> r2
[1] -0.06524385.
```

In the same way, the third-order autocorrelation coefficient $\hat{\rho}_3$ can be calculated:

```
> u_3 = lag(u, -3)
> u_3
Time Series:
Start = 2000
End = 2013
Frequency = 1
1          2          3          4          5          6          7          8          9         10         11         12         13
14
-0.038320129  0.015242321  0.029300955  0.076635526  0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130  0.004380105 -0.021695053 -0.001493488  0.073141298  0.010628878
> r3.Z = sum(u*u_3, na.rm=TRUE)    or    r1.Z = sum(u*u_3)
> r3.Z
[1] -0.01013049
> r3 = r3.Z/r.N
> r3
[1] -0.3317651,
```

as can the fourth-order autocorrelation coefficient: $\hat{\rho}_4$,

```
> u_4 = lag(u, -4)
> u_4
Time Series:
Start = 2001
End = 2014
Frequency = 1
        1          2          3          4          5          6          7          8          9         10         11         12
13         14
-0.038320129  0.015242321  0.029300955  0.076635526  0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130  0.004380105 -0.021695053 -0.001493488  0.073141298  0.010628878
```

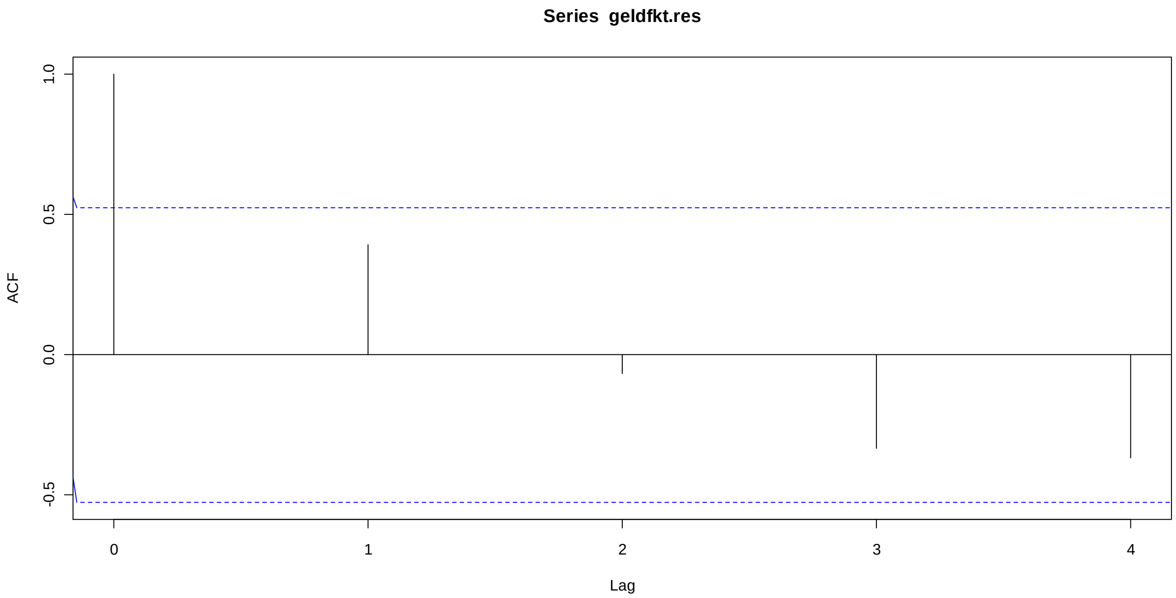
```
> r4.Z = sum(u*u_4, na.rm=TRUE) )    or    r1.Z = sum(u*u_4)
> r4.Z
[1] -0.01119989
> r4 = r4.Z/r.N
> r4
[1] -0.3667872
```

The acf function can be used to plot the autocorrelation coefficients of the money demand function moneyfnc.lm up to a specified lag:

```
>res.acf = acf(moneyfnc.res, lag.max=4).
```

The plot of the autocorrelation coefficients shown in figure 5.1 is called a correlogram.

Figure 5.1: Correlogram of the residuals of the money demand function



The two blue dotted lines indicate the boundaries of the 95% confidence interval for the autocorrelation coefficients. Apart from the autocorrelation of the residual series to lag 0 ($\hat{\rho}_0 = 1$), no autocorrelation coefficient is outside the confidence ranges, meaning that there is no significance at the 5% level for lags 1 to 4.

The acf object res.acf,

```
>class(res.acf)
[1] "acf",
```

contains the attribute acf with the calculated autocorrelation coefficients:

```
>str(res.acf)
```

```
List of 6
 $ acf   : num [1:5, 1, 1] 1 0.3942 -0.0652 -0.3318 -0.3668
 $ type  :chr "correlation"
 $ n.used: int 14
$ lag   : num [1:5, 1, 1] 0 1 2 3 4
$ series: chr "geldfkt.res"
$ snames: NULL
- attr(*, "class")= chr "acf".
```

The autocorrelation coefficients can be stored in vector r,

```
> r = res.acf$acf[-1]
>r
[1]  0.39421443 -0.06524385 -0.33176508 -0.36678718
```

They are also based on the general formula (5.4), so that they are identical to the autocorrelation coefficients calculated in detail.

Exercises

5.1.1 Use detailed calculation to determine the autocorrelation coefficients of the residuals of energy demand model 1 (energyfnc1.lm) for a maximum lag of 4!

5.1.1 Create the correlogram of the residuals of energy demand model 1 (energyfnc1.lm) for a maximum lag of 6, and interpret it!

5.1.2 Store the first- to sixth-order autocorrelations determined in Exercise 5.1.2 in a vector r and plot them using the barplot function!

5.2 Testing for autocorrelation

5.2.1 Durbin-Watson test

The stochastic dependency of residuals with an interval of one, two or more periods can be measured with the autocorrelation coefficient. In econometrics, the significance of first-order autocorrelation can be verified using the Durbin-Watson test or the Breusch-Godfrey test. The latter is often used to check whether the confounding variable of an econometric model follows a higher-order autoregressive process.

Although the Durbin-Watson statistic DW,

$$DW = \frac{\sum_{t=2}^{n}\left(\hat{u}_t - \hat{u}_{t-1}\right)^2}{\sum_{t=1}^{n}\hat{u}_t^2} \tag{5.5}$$

can be directly calculated in R using the lag and sum functions:

```
> DW.Z = sum((u-u_1)^2, na.rm=TRUE)
> DW.Z
[1] 0.03541408
> DW.N = sum(u^2)
> DW.N
[1] 0.03053513
> DW = DW.Z/DW.N
> DW
[1] 1.159781,
```

the critical values for this are not immediately available.

However, the Durbin-Watson test (DW test) can be performed after the car package is loaded,

```
>library(car)
Load required package: MASS
Load required package: nnet
Load required package: survival
Load required package: splines,
```

using the durbinWatsonTest function:

```
>moneyfnc.DW = durbinWatsonTest(moneyfnc.lm)
>moneyfnc.DW
lag Autocorrelation D-W  Statistic p-value 1          0.2184553        1.544104          0.14
1          0.3942144          1.159781          0,026
 Alternative hypothesis: rho != 0.
```

Given the p-value of 0.026, the null hypothesis of missing correlation for a two-sided test (presetting: alternative="two.sided") at a significance level of 5% is rejected. The class and structure of the moneyfnc.DW object is displayed by calling the R functions class and str:

```
>class(moneyfnc.DW)
[1] "durbinWatsonTest"
>str(geldfkt.DW)
List of 4
$ r        : num 0.394
 $ dw      : num 1.16
$ p        : num 0.026
$ alternative: chr "two.sided"
- attr(*, "class")= chr "durbinWatsonTest".
```

Exercises

5.2.1.1 Calculate the Durbin-Watson statistic from the residuals of energy demand model 1 (energyfnc1.lm) and provide a descriptive interpretation!

5.2.1.2 Test the residuals of energy demand model 1 (energyfnc1.lm) with the Durbin-Watson test for autocorrelation (α=0.05)!

5.2.1.3 What is the result of the Durbin-Watson test for the residuals of energy demand model 3 (energyfnc3.lm) with a significance level of 5%?

5.2.2 Breusch-Godfrey test

The Breusch-Godfrey test (BP test) is a test for autocorrelation of any order. In order to check the residuals $\hat{u}_t$ of the money demand function,

```
> u
       1           2           3           4           5           6           7           8           9          10          11          12
-0.038320129  0.015242321  0.029300955  0.076635526  0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130  0.004380105 -0.021695053 -0.001493488
      13          14
 0.073141298  0.010628878,
```

for first-order autocorrelation, the auxiliary regression

(5.6) $\hat{u}_t = \alpha_0 + \alpha_1 \cdot \hat{u}_{t-1} + \mathbf{x}_t'\boldsymbol{\delta} + w_t$

with $\mathbf{x}_t^{'} = (x_{2t}\ x_{3t} \ldots x_{kt})$ and $\boldsymbol{\delta}_t^{'} = (\delta_2\ \delta_3 \ldots \delta_k)$ must be performed. If the null hypothesis of mission first-order autocorrelation is valid,

(5.7) H_0: $\alpha_1 = 0$,

it can be expected that the estimated regression coefficient for $\alpha 1$ is not significantly unequal to zero. The null hypothesis (5.7) is rejected in particular if the test statistic

(5.8) $BG = n \cdot R^2$,

exceeds the (1-α) quantile of χ^2 distribution by a degree of freedom, or the p-value is smaller than the prescribed level of significance α. R^2 is the coefficient of determination of the auxiliary regression.

When executing the auxiliary regression (5.6), care must be taken to ensure that $\hat{u}_{t-1}$ is set to zero ($\hat{u}_1 = 0$):

```
> u0 = 0
> u0_1 = ts(c(u0, u_1), start=1997, end=2010, freq=1)
> u0_1
Time Series:
Start = 1997
End = 2010
Frequency = 1
             1           2           3           4           5           6           7           8           9          10          11
```

```
 0.000000000  -0.038320129   0.015242321   0.029300955   0.076635526   0.054757433  -0.055090280  -
0.052332065 -0.085319372 -0.009836130  0.004380105 -0.021695053
        12          13
-0.001493488  0.073141298.
```

With the explanatory variables lnGDPr (= x_2) and lnEURIBOR (= x_3), the OLS estimate of the auxiliary regression (5.6) for the money demand function returns the following result:

```
> auxreg.bg1 = lm(u ~ u0_1 + lnGDPr + lnEURIBOR)
>summary(auxreg.bg1)

Call:
lm(formula = u ~ u0_1 + lnGDPr + lnEURIBOR)

Residuals:
    Min   1Q   Median       3Q        Max
-0.07723 -0.02903  0.00710  0.02382  0.07412

Coefficients:
                 Estimate      Std. Error  t value Pr(>|t|)
(Intercept)  -1.309e-03  1.019e+00  -0.001    0.999
u0_1          3.960e-01  2.912e-01    1.360    0.204
lnGDPr       9.516e-05  1.308e-01    0.001    0.999
lnEURIBOR    8.678e-04  3.106e-02    0.028    0.978

Residual standard error: 0.05076  on 10  degrees  of freedom:
Multiple R-squared: 0.1561,         Adjusted R-squared: -0.09707
F-statistic: 0.6166 on 3 and 10 DF,  p-value: 0.6198.
```

The test statistic BG according to equation (5.8) assumes the value

```
> R2 = summary(auxreg.bg1)$r.squared
> R2
[1] 0.1561036
> n
[1] 14
> BG1 = n*R2
> BG1
[1] 2.18545
```

which falls below the 95% quantile of the χ^2-distribution with one degree of freedom:

```
> chikrit5 = qchisq(0.95, 1)
> chikrit5
[1] 3.841459.
```

Accordingly, the p-value is

```
> p.BG1 = 1 - pchisq(BG1, 1)
> p.BG1
[1] 0.1393203
```

larger than the specified significance level α of 5%, so that, unlike the Durbin-Watson test, the null hypothesis of the missing autocorrelation cannot be dismissed with the BG(1) test.

This result can be obtained direct using the bgtest function, which is available in the lmtest:

```
>library(lmtest)
Load required package: zoo.
```

When testing for first-order autocorrelation, the bgtest function only needs the original regression model to be specified in R notation or as an lm object. Lag order 1 and the use of χ^2 statistics are pre-set:

```
> moneyfnc.BG1 = bgtest(moneyfnc.lm)
> moneyfnc.BG1

Breusch-Godfrey test for serial correlation of order 1 data:

data:  moneyfnc.lm
LM test = 2.1855, df = 1, p-value = 0.1393.
```

The R object moneyfnc.BG belongs to class htest,

```
>class(geldfkt.BG1)
[1] "htest",
```

and has the following structure:

```
>str(moneyfnc.BG1)
List of 5
 $ statistic: Named num 2.19
  ..- attr(*, "names")= chr "LM test"
 $ parameter: Named int 1
  ..- attr(*, "names")= chr "df"
 $ method   : chr "Breusch-Godfrey test for serial correlation of order 1"
$ p.value  :num 0.139
 $ data.name: chr "geldfkt.lm"
 - attr(*, "class")= chr "htest".
```

BG(2) and BG(4) tests can be used to reveal higher-order autocorrelation of residuals of the money demand function:

```
> moneyfnc.BG2 = bgtest(moneyfnc.lm, order=2)
```

```
> moneyfnc.BG2

Breusch-Godfrey test for serial correlation of order 2 data:

data:  moneyfnc.lm
LM test = 3.2049, df = 2, p-value = 0.2014

> geldfkt.BG4 = bgtest(geldfkt.lm, order=4)
> geldfkt.BG4

Breusch-Godfrey test for serial correlation of order 4 data:

data:  moneyfnc.lm
LM test = 6.1846, df = 4, p-value = 0.1858.
```

Based on the results of the Breusch-Godfrey tests there is no indication of second- or fourth-order autocorrelation of the confounding variable of the money demand function.

Exercises

5.2.2.1 Perform the auxiliary regression of the Breusch-Godfrey test for energy demand model 1 (energyfnc1.lm) as the basis for a

- Test for first-order autocorrelation,
- Test for second-order autocorrelation

5.2.2.2 Use the estimate results of the auxiliary regressions performed in Exercise 5.2.2.1 as the basis for calculating the test statistic of the

- BG(1) test,
- BG(2) test

Test the BG(1) statistic (BG(2) statistic) using the critical value (p-value) (α=0.05)

5.2.2.3 Compare the test results determined in Exercise 5.2.2.2 with the results of the bgtest function!

5.3 Cochran-Orcutt method

Ordinary least square (OLS) estimators of regression coefficients lose their efficiency characteristics with autocorrelated confounding variables. This means that their estimated standard errors are distorted so that the significance tests can no longer be validly performed. If the generalised least squares (GLS) method is used, it is again possible to validly test the estimated regression coefficients.

The example below shows a GLS estimate in the form of the Cochran-Orcutt method for the money demand function using the logarithmised gross domestic product lnGDPr and the logarithmised rate of interest lnEURIBOR as explanatory variables.

A GLS estimate according to the Cochran-Orcutt method corresponds to an OLS estimate of the transformed regression model

(5.9) $y_t^* = \mathbf{x}_t^{*'} \boldsymbol{\beta} + v_t$

with

(5.10) $$\begin{aligned} y_t^* &= y_t - \hat{\phi} \cdot y_{t-1}, \quad t = 2,3,\ldots,n \\ \mathbf{x}_t^{*'} &= \mathbf{x}_t^{'} - \hat{\phi} \cdot \mathbf{x}_{t-1}^{'} \end{aligned}$$

The estimated autoregressive parameter $\hat{\phi}$ obtained from the Markov process

(5.11) $u_t = \phi \cdot u_{t-1} + v_t$

is required in order to calculate the transformed variable values. To this end, use is made of the time series of residuals u andu_1 defined for Durbin-Watson test (Section 5.2.1):

```
>u
Time Series:
Start = 1997
End = 2010
Frequency = 1
         1          2          3          4          5          6          7          8          9         10         11         12
-0.038320129  0.015242321  0.029300955  0.076635526  0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130  0.004380105 -0.021695053 -0.001493488
        13         14
0.073141298  0.010628878
> u_1
Time Series:
Start = 1998
End = 2011
Frequency = 1
         1          2          3          4          5          6          7          8          9         10         11         12
-0.038320129  0.015242321  0.029300955  0.076635526  0.054757433 -0.055090280 -0.052332065 -
0.085319372 -0.009836130  0.004380105 -0.021695053 -0.001493488
        13         14
 0.073141298  0.010628878.
```

The OLS estimate of the Markov process (5.11),

```
> u.AR1 = lm(u[-1] ~ 0+u_1[-14])
> u.AR1

Call:
lm(formula = u[-1] ~ 0 + u_1[-14])
```

```
Coefficients:
u.lag1[-14]
    0.3957,
```

returns $\hat{\phi} = 0.3957$. Without having to take explicit account of the different dating of the time series objects u and u_1, it is possible to perform the least squares estimate of the first order autoregressive process after it has been linked to a matrix object:

```
> U = cbind(u, u_1)
> U
Time Series:
Start = 1997
End = 2011
Frequency = 1
               u           u_1
1997 -0.038320129          NA
1998  0.015242321 -0.038320129
1999  0.029300955  0.015242321
2000  0.076635526  0.029300955
2001  0.054757433  0.076635526
2002 -0.055090280  0.054757433
2003 -0.052332065 -0.055090280
2004 -0.085319372 -0.052332065
2005 -0.009836130 -0.085319372
2006  0.004380105 -0.009836130
2007 -0.021695053  0.004380105
2008 -0.001493488 -0.021695053
2009  0.073141298 -0.001493488
2010  0.010628878  0.073141298
2011       NA       0.010628878
>u.AR1 = lm(U[,1] ~ 0+U[,2])
> u.AR1

Call:
lm(formula = u[1] ~ 0 + u_1[2])

Coefficients:
U[, 2]
0.3957  .
```

The transformations (5.10) for the model variables lnM1GERr, lnGDPr and lnEURIBOR can be performed using the estimated value of 0.3957 for the autoregressive parameter ϕ:

```
>phi = u.AR1$coeff
```

```
>phi
u.lag1[-14]
  0.3956783

> lnM1GERr = ts(lnM1GERr, start=1997, freq=1)
> lnM1GERr
Time Series:
Start = 1997
End = 2010
Frequency = 1
 [1]  6.087797  6.192710  6.277282  6.300584  6.357617  6.341364  6.431895  6.479124  6.587274  6.636419
6.690966 6.751733 6.962177 7.056891
> lnM1GERrs98_10 = lnM1GERr - phi*lag(lnM1GERr, k=-1)
> lnM1GERrs98_10
Time Series:
Start = 1998
End = 2010
Frequency = 1
 [1]  3.783901  3.826961  3.816799  3.864612  3.825793  3.922754  3.934162  4.023625  4.029977  4.065079
4.104262 4.290663 4.302108

>lnGDPr = ts(lnGDPr, start=1997, freq=1)
>lnGDPr

Time Series:
Start = 1997
End = 2010
Frequency = 1
 [1]  7.501971  7.532195  7.554644  7.571288  7.608628  7.637230  7.655220  7.688139  7.707242  7.749784
7.814343 7.840443 7.811188 7.859313
> lnGDPPrs98_10 = lnGDPr - phi*lag(lnGDPr, k=-1)
> lnBIPrs98_10
Time Series:
Start = 1998
End = 2010
Frequency = 1
 [1]  4.563828  4.574317  4.582079  4.612833  4.626660  4.633334  4.659134  4.665212  4.700195  4.747922
4.748476 4.708894 4.768595

>lnEURIBOR = ts(lnEURIBOR, start=1997, freq=1)
>lnEURIBOR
Time Series:
Start = 1997
End = 2010
Frequency = 1
 [1]  1.2029723  1.2641267  1.0885620  1.4793292  1.4492692  1.1999648  0.8458683  0.7466879  0.7839015
1.1249296  1.4539530  1.5325569 0.2070142
```

```
[14] -0.2107210
> lnEURIBORs98_10 = lnEURIBOR - phi*lag(lnEURIBOR, k=-1)
> lnEURIBORs98_10
Time Series:
Start = 1998
End = 2010
Frequency = 1
 [1] 0.7881366 0.5883744 1.0486088 0.8639306 0.6265204 0.3710682 0.4119962 0.4884533 0.8147567
1.0088427 0.9572591 -0.3993854 -0.2926321.
```

The smallest squares estimate of the money demand function with the transformed values (Cochran-Orcutt method) returns the following results:

```
> moneyfnc.co = lm(lnM1GERrs98_10 ~ lnGDPrs98_10+lnEURIBORs98_10)
>summary(moneyfnc.co)

Call:
lm(formula = lnM1GERrs98_10 ~ lnGDPrs98_10 + lnEURIBORs98_10)

Residuals:
   Min   1Q   Median      3Q       Max
-0.08430 -0.01421  0.01305  0.02013  0.06827

Coefficients:
                  Estimate  Std. Error t value  Pr(>|t|)
(Intercept)       -4.56129    0.99328  -4.592   0.000992 ***
lnGDPrs98_10       1.85211    0.21176   8.746   5.35e-06 ***
lnEURIBORs98_10 -0.15595    0.03213  -4.853   0.000668 ***
---
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04806  on 10  degrees  of freedom:
Multiple R-squared: 0.9355,       Adjusted R-squared: 0.9226
F-statistic: 72.53 on 2 and 10 DF,  p-value: 1.115e-06
```

While estimated income elasticity only decreased slightly from around 2 to 1.85, the influence of the rate of interest on money demand, with a value of -0.156 instead of -0.147, is estimated to be slightly higher. The absolute member in relation to the original regression equation of the logarithmised variables is, at -7.548 (=-4.56129/(1-0.39568)), somewhat lower than the OLS estimate (-8.667). All estimated regression coefficients remain statistically valid at the 1% level.

As the Durbin-Watson test shows,

```
>library(car)
Load required package: MASS
Load required package: nnet
Load required package: survival
```

```
Load required package: splines.
>moneyfnc.DWco
lag Autocorrelation D-W  Statistic p-value
1         0.2184553         1.544104          0.14
 Alternative hypothesis: rho != 0,
```

autocorrelation is no longer statistically detectable for the residuals of the transformed money demand function.

Exercises

5.3.1 Reassess the time series GASC and GASPR of energy demand model 1 for autocorrelation by applying the transformation of the Cochran-Orcutt method!

5..3.2 Determine the GLS estimator of the regression coefficients of the money demand model 1 using the Cochran-Orcutt method. Compare the GLS estimators with the ordinary least square estimators!

5.3.3 What is the percentage variance in natural gas consumption (GASC) in the GLS estimate performed in Exercise 5.3.2 that is explained by the price of natural gas (GASPR)?

5.3.4 Are the GLS residuals of energy demand model 1 (see Exercise 5.3.2) based on the Durbin-Watson tests free from autocorrelation (α=0.05)?

5.4 Heteroscedasticity- and autocorrelation-consistent (HAC) standard error

As in the case of heteroscedasticity, the OLS estimators of the regression coefficients of the econometric standard model for autocorrelated confounding variables remain unbiased, while their standard errors are distorted. It is therefore appropriate in this case – as an alternative to a GLS estimate e.g. using the Cochran- Orcutt method – to adjust the common standard errors in order to repair the defect in the model. Newey and West (1987) developed a heteroscedasticity- and autocorrelation-consistent (HAC) estimator of the variance-covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ which, without considering autocorrelation (maximum lag L=0), coincides with White's heteroscedasticity-consistent (HC) estimator (4.6).

R provides the NeweyWest function for calculating the Newey/West HAC estimator for $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ in the sandwich package:

```
> library(sandwich)
Load required package: zoo.
```

If a potentially existent autocorrelation in the money demand function is to be directly included in the estimate of the variance-covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ without prior deactivation, the argument prewhite must be set to FALSE when function NeweyWest is called.

Using the automatic selection procedure of the maximum lag length p of Newey and West (1994) results in the HAC estimator

```
> covb.nw = NeweyWest(moneyfnc.lm, prewhite=FALSE)
> covb.nw
            (Intercept)        lnGDPr   lnEURIBOR
(Intercept) 0.358638890 -0.0463005546   0.0020541166
lnGDPr               -0.046300555  0.0059945391 -0.0003583977
lnEURIBOR  0.002054117  -0.0003583977 0.0005830118.
```

which possesses the property of consistency relevant for large samples. As can be seen, the HAC estimator **covb.nw** is based on a maximum lag length of 1:

```
> NeweyWest(moneyfnc.lm, lag=1, prewhite=FALSE)
            (Intercept)        lnGDPr   lnEURIBOR
(Intercept) 0.358638890 -0.0463005546   0.0020541166
lnGDPr  -0.046300555  0.0059945391 -0.0003583977
lnEURIBOR  0.002054117  -0.0003583977 0.0005830118.
```

Adjusting the Newey/West HAC estimator for $\text{Cov}(\hat{\boldsymbol{\beta}})$ to the low volume of the current sample through multiplication with the factor n/(n-k) (adjust =TRUE) results in the adjusted HAC estimator

```
> covb.nwa = NeweyWest(moneyfnc.lm, prewhite=FALSE, adjust=TRUE)
> covb.nwa
           (Intercept)        lnGDPr   lnEURIBOR
(Intercept) 0.45644950 -0.0589279786   0.0026143302
lnGDPr  -0.05892798  0.0076294134 -0.0004561425
lnEURIBOR  0.00261433  -0.0004561425 0.0007420150.
```

for which the automatic selection process also determined a maximum lag length of 1.

As in the case of heteroscedasticity, the significance of the regression coefficients could be robustly tested after extraction of the variance estimator of the estimated regression coefficients. However, at this point, we make direct use of the coeftest in the R package lmtest. After loading the package,

```
> library(lmtest),
```

the coeftest function, which can access the Newey/West procedure, is available. Neglecting the adjustment factor n/(n-k) for small samples and using normal approximation (z-test) returns the test results

```
> coeftest.NWz = coeftest(moneyfnc.lm, df=Inf, vcov=NeweyWest(money.lm, prewhite=FALSE))
> coeftest.NWz

z test of coefficients:

              Estimate    Std. Error    z value    Pr(>|z|)
(Intercept)      -8.667211        0.598865  -14.4727   2.2e-16 ***
lnGDPr        1.995417     0.077424    25.7724    < 2.2e-16 ***
lnEURIBOR   -0.146495     0.024146    -6.0672     1.302e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

In contrast, including the adjustment factor n/(n-k) the following results are returned by the t-test when the number of degrees of freedom, n-k=14-3=11, is specified:

```
> coeftest.NWt = coeftest(moneyfnc.lm, df=n-k, vcov=NeweyWest(moneyfnc.lm, prewhite=FALSE, adjust= TRUE))
> coeftest.NWt

t test of coefficients:

              Estimate     Std. Error    t value    Pr(>|t|)
(Intercept)      -8.667211        0.675610  -12.829   5.839e-08 ***
lnGDPr        1.995417    0.087347    22.845   1.278e-10 ***
lnEURIBOR   -0.146495    0.027240  -5.378 0.0002241 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

The highly test results obtained with validity of the standard assumptions remain unchanged when heteroscedasticity- and autocorrelation-consistent standard errors are used independently of adjustment to the volume of the sample. Nevertheless, the HAC standard error of the estimated regression coefficient of income decreases by a factor of 1.55 when the sample adjustment is included compared with the uncorrected standard deviation. In the case of the rate of interest, there is a reduction by a factor of 1.18.

Exercises

5.4.1 Specify the heteroscedasticity-and autocorrelation-consistent (HAC) estimator of the variance-covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ according to Newey and West for the OLS-estimated energy demand model 1 (energiefkt1.lm) with and without adjustment for finite samples using the automatic selection procedure for maximum lag length!

5.4.2. What was the maximum lag length established by the automatic selection procedure according to Newey/West in Exercise 5.4.1?

5.4.3 Use the HAC estimator of the variance-covariance matrix $\text{Cov}(\hat{\boldsymbol{\beta}})$ calculated in Exercise 5.4.1 and adapted to finite samples to determine the standard errors of the estimated regression coefficients! Compare them with the uncorrected standard errors of the OLS estimate

5.4.4 Examine the significance of the estimated regression coefficients of energy demand model 1 using the HAC standard errors calculated in exercise 5.4.3 with the help of the t-test (α=0.05)!

5.4.5 Test the significance of the regression coefficients of energy demand model 1 (α=0.05) using the coeftest function with the help of the

- z-test without adjustment for finite samples,
- t-test with adjustment for finite samples!

6. Models using qualitative variables

6.1 Models using qualitative regressors

Standard models of regression analysis investigate relationships between metrically scaled variables. There are however applications where qualitative variables are also important. For example, in the case of some products different demand behaviour can be observed between men and women. Furthermore, in certain studies it may prove necessary to consider different consumption or savings behaviour of different social classes in the population. Geographical effects generally need to be taken into account in regional studies, and temporal effects in time series analyses.

All these examples involve qualitative, independent variables that need to be integrated into a regression model in an appropriate manner. In the simplest case, the qualitative variables consist of two states such as “male” and “female” for gender or “employed” or “unemployed” for employment. Under these circumstances, qualitative regressors can be included in an econometric model in a simple way using a dummy variable (0.1 variable). It is unimportant which value – “0” or “1” – is used for a category e.g. “male” or “female”. Nevertheless, a category of particular interest, e.g. “purchase of product A” will be coded with “1 and the less important category “non-purchase of product A” with “0”. The regression coefficient of the dummy variable measures the markup (+) or markdown to be made in the event of the presence of a property of interest (“1”) when determining the regression values of the statistical units concerned.

As an example of the inclusion of a qualitative regressor in an econometric model, let us consider demand q_t for a type of goods dependent on the total expenditure x_t of individual consumers t. If demand is gender-specific, the binary variable gender can be included in the form of a dummy variable,

$$(6.1)\quad g_t = \begin{cases} 1, & \text{if woman} \\ 0, & \text{if man} \end{cases},$$

in the stochastic demand function

$$(6.2)\quad q_t = \beta_1 + \beta_2 \cdot x_t + \beta_3 \cdot g_t + u_t$$

In order to estimate the demand model (6.2), data of eight purchasers are used with regard to their demand for a DIY article (Demand), their total expenditure (TotExp) and their gender (Gender). These data are present together with details on their social status (SocSt): 1 low, 2 middle, 3 high) in the file Purchasers.csv.

After importing the data from the .csv file Purchasers into the data-frame object Purchasers.df,

```
> Purchasers.df = read.csv("Purchasers.csv")
> Purchasers.df
  Purchasers Demand TotExp Gender SocSt
1     1        9      48     0      2
2     2        1      30     1      1
3     3        4      23     0      1
4     4        8      56     0      3
5     5       14      72     0      3
6     6        4      32     0      2
7     7       12      42     0      1
8     8        2      43     1      3,
```

the demand function can be estimated in R using the lm function with the ordinary least squares method:

```
> Purchasers.lm1 = lm(Demand~TotExp+Gender, data=Purchasers.df).
```

The regression output

```
>summary(Purchasers.lm1)

Call:
lm(formula = Demand ~ TotExp + Gender, data = Purchasers.df)

Residuals:
```

```
      1        2        3        4        5        6        7        8
0.03554  0.70761 -0.31983 -2.45075  0.57668 -1.99190  4.15025 -0.70761

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.04676   2.88903  0.016   0.9877
TotExp     0.18579   0.05975   3.109   0.0266 *
Gender     -5.32793   2.02666  -2.629   0.0466 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2,393  on 5  degrees  of freedom:
Multiple R-squared:  0.8182,   Adjusted R-squared:  0.7455
F-statistic: 11.25 on 2 and 5 DF,  p-value: 0.0141
```

shows that gender has a significant influence on demand for DIY articles. The effect of the qualitative regressor Gender, with a p-value of 0.0466 at the 5% level is significant. The same applies to total expenditure (TotExp), where the p-value is 0.0266. The two regressors explain 81.82% of the variance in demand for DIY articles (Demand). With an F-value of 11.25, the overall context provided by the demand function is statistically valid (p = 0.0141).

While the estimated regression coefficient of variable TotExp quantifies the expected increase in demand for DIY articles (Demand) for an increase in individual expenditure by 1 unit with 0.186 units, the value of the regression coefficient of the dummy variable Gender reflects the different level of demand between men and women. On average, demand among women for DIY articles is 5.328 units lower than among men:

Demand function for men:

$$\hat{y}_{male} = 0{,}047 + 0{,}186 \cdot x$$

Demand function for women:

$$\hat{y}_{female} = 0{,}047 + 0{,}186 \cdot x - 5{,}328 \cdot g\,.$$

However, a rotation of the demand function may also be relevant instead of a parallel shift. In this case, the assumption of the same marginal consumption trend β_2 among male and female purchasers must be abandoned. The binary regressor gender (g_t) must then be incorporated into demand model in the form of an interaction with total expenditure (x_t):

(6.3) $$q_t = \beta_1 + \beta_2 \cdot x_t + \beta_3 \cdot x_t \cdot g_t + u_t\,.$$

The stochastic demand function (6.3) can then be estimated using the lm function by including the interaction term made up of total expenditure and gender in the form TotExp:Gender:

```
> Purchasers.lm2 = lm(Demand~TotExp+TotExp:Gender, data=Purchasers.df).
>summary(Purchasers.lm2)

Call:
lm(formula = Demand ~ TotExp + TotExp:Gender, data = Purchasers.df)

Residuals:
      1        2        3        4        5        6        7        8
0.02597 -0.20056 -0.13266 -2.52327  0.37825 -1.87555  4.18790  0.13992

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.32141    2.77990  -0.116   0.9125
TotExp         0.19366    0.05797   3.341   0.0205 *
TotExp:Gender -0.14292    0.05276  -2.709   0.0423 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2,351  on 5  degrees  of freedom:
Multiple R-squared:  0.8245,    Adjusted R-squared:  0.7543
F-statistic: 11.74 on 2 and 5 DF,  p-value: 0.01291.
```

This demonstrates a significant influence of total expenditure (TotExpg) and the interaction of total expenditure and gender (TotExp:Gender) on demand for DIY articles (Demand) at the 5% level. The estimated regression coefficient of the interaction term TotExp:Gender shows that the marginal propensity to consume DIY articles among women is 0.143 lower than among men:

Demand function for male purchasers (marginal propensity to consume: 0.194):

$$\hat{y}_{male} = 0{,}047 + 0{,}194 \cdot x$$

Demand function for female purchasers (marginal propensity to consume: 0,194-0,143=0,051):

$$\hat{y}_{female} = 0{,}047 + 0{,}051 \cdot x \,.$$

Two dummy categories must be formed for quality regressors with three categories in order to quantify their influence on the dependent variables in an econometric model. If, for example, the qualitative variable “gender” in the demand model is replaced by the qualitative variable “social status” with the categories low, middle and high, it would be possible to form the following dummy variables for the last two categories:

$$s_{2t} = \begin{cases} 1, & \text{if middle social status} \\ 0, & \text{or else} \end{cases}$$

and

$$s_{3t} = \begin{cases} 1, & \text{if high social status} \\ 0, & \text{or else} \end{cases}$$

The “social status” regressor in the multiple regression model is replaced by the two dummy variables s_2 and s_3, which are declared as “middle social status” and “high social status”:

(6.4) $q_t = \beta_1 + \beta_2 \cdot x_t + \beta_3 \cdot s_{2t} + \beta_4 \cdot s_{3t} + u_t$.

The category of low social status then serves as reference category. If both dummy variables for a purchaser or household assume the value 0, low social status is present.

However, dummy variables do not need to be explicitly formed in R in the case of a polytomous regressor. Instead, this occurs automatically if specified in the lm function as an object of type factor. Accordingly, the demand model (6.4) for purchaser data will be estimated with dummy variable s_2 and s_3 for middle and high social status without their prior definition if the polytomous variable SocSt is included as factor factor(SocSt) in the lm function:

```
> Purchasers.lm3 = lm(Demand~TotExp+factor(SocSt), data=Purchasers.df)
> summary(Purchasers.lm3)

Call:
lm(formula = Demand ~ TotExp + factor(SocSt), data = Purchasers.df)

Residuals:
      1       2       3       4       5       6       7       8 
-0.7849 -3.9823  1.8920  0.4106 -0.1593  0.7849  2.0903 -0.2514 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    -7.33621    3.27726  -2.239   0.0888 .
TotExp          0.41062    0.09278   4.426   0.0115 *
factor(SocSt)2 -2.58848    2.42241  -1.069   0.3455
factor(SocSt)3 -8.06897    3.12104  -2.585   0.0610
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2,515  on 4  degrees  of freedom:
Multiple R-squared:  0.8394,    Adjusted R-squared:  0.7189
F-statistic: 6,968 on 3 and 4 DF,  p-value: 0.0457.
```

Total expenditure (TotExp) is again significant at the 5% level. The two negative regression coefficients of the dummy variables "middle social status" (factor(SocSt)2) and "high social status" (factor(SocSt)3) indicate that DIY articles tend to be more in demand among households with a lower social status. Since, in addition, $|\hat{\beta}_3| < |\hat{\beta}_4|$ is true, it is safe to conclude that demand tends to decline among those with a higher social status.

Accordingly, demand for DIY articles from households with a middle social status sinks vis-à-vis households with a low social status by 2.588 units. Demand from households with a high social status sinks vis-à-vis households with a medium social status by 5.481 (=8.069-2.588) units, and vis-à-vis households with a low social status by 8.069 units.

The status-specific demand functions are:

for low social status: $\hat{y}_{s1} = -7{,}336 + 0{,}411 \cdot x_t + u_t$

for middle social status: $\hat{y}_{s2} = -9{,}924 + 0{,}411 \cdot x_t + u_t$

for high social status: $\hat{y}_{s1} = -17{,}994 + 0{,}411 \cdot x_t + u_t$.

However, since $\hat{\beta}_3$ is not significant (p=0.346), at any common level, the decrease in demand among households with a middle social status compared with households with a low social status is not statistically confirmed. However, with a p-value of 0.061, the influence of high social status on demand can be said to be slightly significant.

Exercises

6.1.1 Details of six employed persons include their income (Income), their age (Age) and their professional status (Prof: 1 self-employed, 0 not self-employed):

```
> Income = c(2400, 2700, 2800, 3400, 3200, 3500)
> Age = c(28, 43, 33, 52, 38, 58)
> Prof = c(0, 0, 1, 0, 1, 0).
```

Determine the arithmetic mean of the variables Income and Age and the table of absolute and relative frequencies for professional status with the help of available R functions!

6.2.2 Regress income against age and professional status (refer to Exercise 6.1.1 for the data)! Specify the regression functions for the two groups of self-employed and non self-employed and interpret them!

6.2.3 Regress income against age and the interaction between age and professional status (refer to Exercise 6.1.1 for data) and interpret the generated regression output using the `summary` function!

6.2.4 Replace the binary variable “professional status” in 6.1.1 with the polytomous variable “profession” with observed category types 2, 1, 3, 1, 3, 2! Regress income against age and profession variables and interpret the regression function estimated with the OLS method!

6.2 Structural break test

Events sometimes occur in an economy that permanently change its structure in a certain respect.

In this case, econometricians speak of a structural break. Prior to the structural break in period n_1+1, the influences of exogenous variables x_{jt}, j=1,2,…,k, on the endogenous variable y_t are defined by the regression coefficients β_j^1, j=1,2,…,k,

(6.5) $$y_t = \sum_{j=1}^{k} \beta_j^1 \cdot x_{jt} + u_t, \quad t = 1,2,...,n_1,$$

while dependency after the structural break is identified by the regression coefficients β_j^2:

(6.6) $$y_t = \sum_{j=1}^{k} \beta_j^2 \cdot x_{jt} + u_t, \quad t = n_1+1, n_1+2,...,n.$$

Using dummy variables d_1 and d_2,

$$d_{1t} = \begin{cases} 1 & \text{für } t = 1,2,...,n_1 \\ 0 & \text{für } t = n_1+1, n_1+2,...,n \end{cases}$$

and

$$d_{2t} = \begin{cases} 0 & \text{für } t = 1,2,...,n_1 \\ 1 & \text{für } t = n_1+1, n_1+2,...,n \end{cases}$$

the econometric model for the entire observation period can be formulated in order to represent the structural break:

(6.7) $$y_t = \sum_{j=1}^{k} \beta_j^1 \cdot x_{jt} \cdot d_{1t} + \sum_{j=1}^{k} \beta_j^2 \cdot x_{jt} \cdot d_{2t} + u_t, \quad t = 1,2,...,n.$$

This takes account of the fact that all regression coefficients can change when a structural break occurs.

The Chow test can be used to examine whether an event has actually resulted in a structural break or not. A structural break is present when one or more regression coefficients β_j, j=1,2,…,k, have changed in period n_1+1, n_1+2, …, n compared with period 1,2,…,n_1 . In this case, constancy of the structural parameters of the econometric model over the entire observation period can no longer be assumed.

The Chow test compares the residual squared sums Q_1 and Q_2 of the regressions before and after the structural break (6.5) and (6.6), with the residual squared sum Q of the regression

$$y_t = \sum_{j=1}^{k} \beta_j \cdot x_{jt} + u_t, \quad t = 1,2,...,n \tag{6.8}$$

over the entire observation period. The null hypothesis

$$H_0 : \beta_j^1 = \beta_j^2 \quad \text{für alle } j = 1,2,...,k$$

is rejected at significance level αif the test statistic

$$F = \frac{[Q - (Q_1 + Q_2)] \ / \ k}{(Q_1 + Q_2) \ / \ (n - 2k)} \tag{6.9}$$

exceeds the (1-α) quantile of F-distribution with k and n-2k degrees of freedom.

An example of the Chow test is to examine whether the introduction of the euro as cash on 1 January 2002 was accompanied by a structural break in the money demand function. The residuals **moneyfnc$residuals** of the money demand function in the period 1997-2010 (n=14) that were estimated in Section 2.2 provide the squared sum

```
> Q = sum(moneyfnc.lm$residuals^2)
> Q
[1] 0.03053513.
```

The residual squared sums Q_1 and Q_2 before and after the potential structural break can be calculated from the regressions for the partial periods 1997-2001 and 2002-2010:

Partial period 1997-2001 (n_1=5)	Partial period 2002-2010 (n_2=9)
> moneyfnc97_01.lm = lm(log(M1GERr)~log(GDPr)+ log(EURIBOR), data=Moneydemandfnc.data[1:5,]) >summary(moneyfnc97_01.lm) Call: lm(formula = log(M1GERr) ~ log(GDPr) + log(EURIBOR), data = Moneydemandfnc.data[1:5, ,])	> moneyfnc02_10.lm = lm(log(M1GERr)~log(GDPr)+ log(EURIBOR), data=Moneydemandfnc.data[6:14,]) >summary(moneyfnc02_10.lm) Call: lm(formula = log(M1GERr) ~ log(GDPr) + log(EURIBOR), data = Moneydemandfnc.data[6:14, ,])

Residuals: 1 2 3 4 5 -0.019795 0.006544 0.012275 0.025402 -0.024425 Coefficients: Estimate Std. Error t value Pr(>|t|) (Intercept) -14.69078 3.58389 -4.099 0.0547 . log(GDPr) 2.78725 0.48710 5.722 0.0292 * log(EURIBOR) -0.09268 0.11820 -0.784 0.5151 --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 0.03023 on 2 degrees of freedom: Multiple R-squared: 0.9587, Adjusted R-squared: 0.9174 F-statistic: 23.21 on 2 and 2 DF, p-value: 0.0413	Residuals: Min 1Q Median 3Q Max -0.049799 -0.016867 -0.003266 0.019674 0.054725 Coefficients: Estimate Std. Error t value Pr(>|t|) (Intercept) -11.39945 1.23635 -9.220 9.18e-05 *** log(GDPr) 2.34808 0.15890 14.777 6.04e-06 *** log(EURIBOR) -0.16603 0.02326 -7.137 0.000381 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 0.03643 on 6 degrees of freedom: Multiple R-squared: 0.9823, Adjusted R-squared: 0.9764 F-statistic: 166.6 on 2 and 6 DF, p-value: 5.532e-06
> Q1 = sum(moneyfnc97_01.lm$residuals^2) > Q1 [1] 0.001827181	> Q2 = sum(moneyfnc02_10.lm$residuals^2) > Q2 [1] 0.007964608

The test statistic (6.9) of the Chow test thus assumes the value

```
> F.Chow = ((Q-(Q1+Q2))/3)/((Q1+Q2)/(14-2*3))
> F.Chow
[1] 5.649179
```

which is to be compared at a significance level of 5% with the 95%-quantile of an F-distribution using k=3 and n-2·k=8 degrees of freedom

```
> Fkrit5 = qf(0.95, 3, 8)

> Fkrit5
[1] 4.066181
```

Significance owing to F.Chow=5.649179> Fkrit5=4.066181 is similarly indicated by the p-value

```
> p.FChow = 1-pf(F.Chow, 3, 8)
> p.FChow
[1] 0.02242426
```

so that the suspected structural break is statistically corroborated.

The Chow test can be performed in R in a compressed form using the chow.test function implemented in the gap package:

```
> library(gap)
```

Before the chow.test function is called, the vectors of dependent variables y1 and y2

```
> y1 = lnM1GERr[1:5]
> y1
[1] 6.087797 6.192710 6.277282 6.300584 6.357617

> y2 = lnM1GERr[6:14]
> y2
[1] 6.341364 6.431895 6.479124 6.587274 6.636419 6.690966 6.751733 6.962177
[9] 7.056891,
```

and the matrices of the independent variables,

```
> X1 = cbind(lnGDPr[1:5], lnEURIBOR[1:5])
> X1
        [,1]       [,2]
[1,] 7.501971 1.202972
[2,] 7.532195 1.264127
[3,] 7.554644 1.088562
[4,] 7.571288 1.479329
[5,] 7.608628 1.449269

> X2 = cbind(lnGDPr[6:14], lnEURIBOR[6:14])

> X2
         [,1]       [,2]
 [1,] 7.637230  1.1999648
 [2,] 7.655220  0.8458683
 [3,] 7.688139  0.7466879
 [4,] 7.707242  0.7839015
 [5,] 7.749784  1.1249296
 [6,] 7.814343  1.4539530
 [7,] 7.840443  1.5325569
 [8,] 7.811188  0.2070142,
 [9,] 7.859313 -0.2107210,
```

need to be defined for the two partial periods before and after the potential structural break:

```
> moneyfnc.chow = chow.test(y1,X1,y2,X2)
> moneyfnc.chow
   F value       d.f.1        d.f.2     P value
5.64917937 3.00000000 8.00000000 0.02242426.
```

The test statistic of the Chow test `F value` is output with a precision of eight decimal places. In line with the Chow test previously performed step by step, the p-value of 0.022 exhibits significance at the 5% level and thus confirms the structural break in the overall German money supply function following the introduction of the euro as cash.

Exercises

6.2.1 Estimate the Keynesian consumption function for the partial periods 1997–2001 and 2002–2011 (refer to Section 1.5: `Consumfnc.data` for the data) and determine the residual squared sums Q, Q_1 and Q_2 for the entire period (residuals: `consfnc.lm$res`) and for the two partial periods!

6.2.2. Calculate the test statistic of the Chow test using the values output in Exercise 6.2.1 and check using the

- Critical value,
- p-value,

whether the introduction of the euro in 2002 was accompanied by a structural break in the consumption function!

6.2.3 Test for the presence of a structural break in the consumption function (Section 2.1: `consfnc.lm`) using the R function `chow.test`!

6.3 Linear probability model

A linear probability model is used to specify a relationship between a qualitative dependent variable and a range of explanatory variables. It represents the simplest form of a microeconomic model to explain economic choices. For example, an examination can be made to see what independent variables determine the purchase of a product or how the choice of a means of transportation is explained.

As an example of the use of the linear probability model, let us consider the purchase of a product (y_t) dependent on the income (x_{2t}) and gender (x_{3t}) of a consumer t . The attribute profiles of 25 consumers from a cross-sectional survey are present in the data file `Consumers.csv`:

```
> Consumers.df = read.csv("Consumers.csv")
> Consumers.df
  Consumer Purchase Income Gender
```

1	1	0	2500	1
2	2	1	3000	0
3	3	1	3200	0
4	4	1	3500	1
5	5	0	2800	1
6	6	0	3000	1
7	7	1	4000	0
8	8	0	2200	1
9	9	1	3800	0
10	10	0	2900	1
11	11	1	2600	1
12	12	0	2400	0
13	13	1	4200	0
14	14	1	3700	0
15	15	0	3300	1
16	16	1	4100	1
17	17	0	2500	1
18	18	1	3900	0
19	19	1	4400	0
20	20	0	2000	1
21	21	0	3100	1
22	22	1	3400	0
23	23	1	3500	1
24	24	1	3800	0
25	25	0	2200	1.

The dichotomous dependent variable Purchase assumes the value of 1 if the consumer has chosen to purchase the product. In the event of non-purchase, this variable assumes the value 0. Income is a metrically scaled variable (in euros) used to explain the probability of the purchase of the product. The dummy variable Gender, set to 1 if the consumer is female and 0 if male, is also used.

With the linear probability (LP) model, as the simplest choice model, the probability of a purchase changes in a linear way to changes in the independent variables:

(6.10) $p_t = \boldsymbol{\beta}'\mathbf{x}_t$

with

$$\mathbf{x}_t = (x_{1t} = 1 \quad x_{2t} \quad \ldots \quad x_{kt})'$$

and

$$\boldsymbol{\beta} = (\beta_1 \quad \beta_2 \quad \ldots \quad \beta_k)'.$$

In the LP model, the random variable y_t consists of the probability given by p_t (6.10), as its systematic part, and the confounding variable u_t:

(6.11) $y_t = p_t + u_t = \boldsymbol{\beta}'\mathbf{x}_t + u_t$.

Applying the lm function to the data set Consumers.df returns the OLS estimate of LP model (6.11):

```
> Consumers.lm = lm(Purchase~Income+Gender, data=Consumers.df)
> summary(Consumers.lm)

Call:
lm(formula = Purchase ~ Income + Gender, data = Consumers.df)

Residuals:
    Min   1Q   Median     3Q      Max
-0.47181 -0.24857  0.00583  0.18082  0.83214

Coefficients:
                Estimate  Std. Error  t value Pr(>|t|)
(Intercept)     -0.6619899  0.4251094  -1.557 0.133688
Income      0.0004342  0.0001144   3.796 0.000991 ***
Gender  -0.2991227  0.1552109  -1.927 0.066966 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3216  on 22  degrees  of freedom:
Multiple R-squared:  0.6306,    Adjusted R-squared:  0,597
F-statistic: 18.77 on 2 and 22 DF,  p-value: 1.75e-05.
```

While income is highly significant, gender only exhibits slight significance. The regression coefficient of income of 0.00043 means that the purchase probability for a product increases by 4.3% when a consumer's income climbs by 100 units. The negative regression coefficient of the gender attribute of -0.29912 indicates a 29.9% lower probability of purchase by a women compared with a man. Overall, the two variables of income and gender explain 63.1% of the purchase of a product.

For female consumer 1, who did not buy the product (y_1=0) with attribute $\mathbf{x}_1' = (0 \quad 2500 \quad 1)$ there is a calculated purchase probability of

$$\hat{p}_1 = -0{,}6619899 + 0{,}0004342 \cdot 2500 - 0{,}2991227 \cdot 1 = 0{,}12439.$$

In contrast, the purchase probability of male consumer 2 with attribute profile $\mathbf{x}_2' = (1 \quad 3000 \quad 0)$ who bought the product (y_1=1) is

$$\hat{p}_2 = -0{,}6619899 + 0{,}0004342 \cdot 3000 = 0{,}64161.$$

Both purchase probabilities are between 0 and 1 and are easily interpretable. Owing to his higher income and gender, consumer 2 is significantly more likely to purchase than consumer 1. In reality, consumer 2 chose to purchase the product (y_2=1), while consumer 1 chose not to (y_2=0)

However, there is no guarantee with the linear probability model that the calculated probability values as regression values will be within the range of 0 and 1. Our example also returns negative purchase probabilities for consumers 8, 20 and 25 owing to their attribute profiles. In contrast, the calculated probabilities for consumers 7, 13, 18 and 19 exceed the value 1. Furthermore, the confounding variable of the LP model (6.11) is not homoscedastic, but instead varies with the values of the explanatory variables. Finally, the applicability of the LP model is restricted by the fact that the strength of the effects of the x-variables on the probability of an event under consideration is assumed to be independent of its level.

Exercises

6.3.1 Estimate the linear probability (LP) model to explain professional status depending on income and age using the OLS method (refer to Exercise 6.1.1 for data) and interpret the regression coefficients

6.3.2. Calculate the regression values of the LP model estimated in Exercise 6.3.1 and interpret them taking into account the required properties

6.3.3 For which employed persons do the probabilities determined in Exercise 6.3.1 result in wrong classification?

6.4 Logit model

For these reasons, empirical economic research uses the probit and logit models to explain economic choices instead of the linear probability model. While in the probit model e.g. purchase probability is provided by standard normal distribution, the logit model is based on the distribution function of logistic distribution. As the form of the probability functions of the probit and logit models are similar, we will consider the logit model as it is more manageable in terms of estimation.

The logit model is provided by the specification of the distribution function F in the form of logistic distribution, (6.12) $$p_t = \frac{e^{\boldsymbol{\beta}'\mathbf{x}_t}}{1+e^{\boldsymbol{\beta}'\mathbf{x}_t}} = \frac{1}{1+e^{-\boldsymbol{\beta}'\mathbf{x}_t}} = \frac{1}{1+e^{-(\beta_1+\beta_2 \cdot x_{2t}+\ldots+\beta_k \cdot x_{kt})}},$$

which after logarithmisation gives the specification

(6.13) $$\ln\frac{p_t}{1-p_t} = \boldsymbol{\beta}'\mathbf{x}_t$$

is required in order to calculate the transformed variable values. The values $\ln[p_t/(1-p_t)]$ are the logarithmised odds $p_t/(1-p_t)$, for which the name logits is commonly used. In the logit model (6.13), the regression coefficients β_j measure the impulses of the exogenous variables x_{jt} on the log odds.

The odds

(6.14) $$\frac{p_t}{1-p_t} = e^{\boldsymbol{\beta}'\mathbf{x}_t}$$

describe the relationship of probability of an event occurring y_t=1 relative to the probability of the non-event y_t=0. Finally, the effect coefficients e^{β_j} indicate how the odds change when the exogenous variable x_{jt} is increased by one unit.

The logit model (6.13) can be estimated using the R function glm with the maximum likelihood (ML) method, which can be used to adjust generalised linear models. Like the R function lm, it is included in the stats package, but has a number of additional arguments. The family argument must be set to binomial(logit) in order to estimate a logit model with a binary dependent variable.

Otherwise, the specification used in the LP model of our consumer example can be transferred to the glm function:

```
> Consumers.logit = glm(Purchase~Income+Gender, family=binomial(logit), data=Consumers.df)
> summary(Consumers.logit)

Call:
glm(formula = Purchase~Income+Gender, family=binomial(logit),
   data=Consumers.df))

Deviance Residuals:
    Min    1Q   Median     3Q      Max
-1.25668  -0.25625  0.04079  0.21363  2.45062

Coefficients:
              Estimate  Std. Error  z value Pr(>|z|)
```

```
(Intercept)   -11.726436   5.960102  -1.967   0.0491 *
Income     0.004480   0.002038   2.198   0.0280 *
Gender     -2.874458   1.836447  -1.565   0.1175
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

0.1 ' ' 1 (Dispersion parameter for binomial family taken to  be 1)

   Null deviance: 34.296  on 24  degrees  of freedom:
Residual deviance: 12.144  on 22  degrees  of freedom
AIC: 18.144

Number of Fisher Scoring iterations: 7).
```

The regression output reveals that income in the logit model has a positive influence on the log odds of product purchase and is significant at the 5% level. If income increases by 1 euro, the log odds increase by 0.004480. As in the LP model, the regression coefficient of gender In concrete terms, the log odds for a product purchase by a female consumer are 2.874458 units lower than by a male consumer. However, the estimated regression coefficient of gender is not significant.

When interpreting the model estimate, it should be noted that the attribute fittted.valuesdoes not refer to the estimated log odds $\ln[p_t/(1-p_t)]$, but instead to the estimated purchase probabilities $\hat{p}_t$. They result from using the estimated regression coefficients $\hat{\beta}_j$ in equation (6.12):

(6.15) $$\hat{p}_t = \frac{1}{1+e^{11,72644-0,00448 \cdot x_{2t}+2,87446 \cdot x_{3t}}}$$

A comparison with the values of the dependent variables shows how the purchase probabilities are reflected in actual choices for or against a product purchase:

```
> Consumers.Purchase_Prob = cbind(Consumers.df$Purchase, Consumers.logit$fitted.values)
> Consumers.Purchase_Prob
  [,1]  [,2]
1   0   0.032299316
2   1   0.847455848
3   1   0.931557636
4   1   0.746594318
5   0   0.113472195
6   0   0.238726661
7   1   0.997964938
```

8	0	0.008628669
9	1	0.995028799
10	0	0.166906242
11	1	0.049649805
12	0	0.274191476
13	1	0.999168355
14	1	0.992240669
15	0	0.545981924
16	1	0.977440353
17	0	0.032299316
18	1	0.996818294
19	1	0.999660384
20	0	0.003540007
21	0	0.329238206
22	1	0.970884801
23	1	0.746594318
24	1	0.995028799
25	0	0.008628669.

If not just the observed values for income that are used in (6.15), but instead variable income over the relevant value range of 0 to 5000,

```
> Income.x = 0:5000,
```

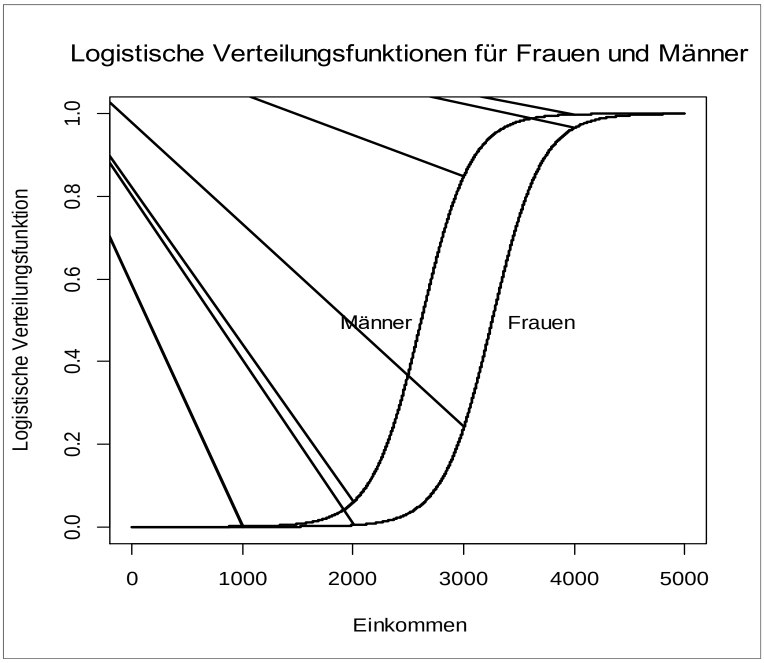
the function values of the distribution function of the logistic distribution for men and women resulting from the model estimate can be calculated:

```
> pt.male = 1/(1+exp(11.72644-0.00448*Income.x))
> pt.female = 1/(1+exp(11.72644-0.00448*Income.x+2.87446)).
```

They contain the probabilities for product purchase when a certain income level is reached. In order to represent both distribution functions in a chart, the lines function is used for the latter instead of the plot function:

```
> plot(Income.x,pt.male, main="Logistic distribution functions for women and men", xlab= "Income", ylab="Logistic distribution function", type="l", lwd=2, font.main=1, cex.main=1.25)
> lines(Income.x,pt.female, main="Logistic distribution functions for women and men", xlab= "Income", ylab="Logistic distribution function", type="l", lwd=2, font.main=1, cex.main=1.25)
> text(2200,0.5, labels = "Men")
> text(3700,0.5, labels = "Women")
> box(which="figure")
```

Figure 6.1: Logistic distribution functions for men and women



It is clear from the chart that the regression coefficients cannot be interpreted in the same way as with linear regression. Increasing income in the mid-range around € 3,000 by one unit causes purchase probability to increase far more strongly than the same s´ change at the periphery e.g. around € 1,000 or around € 5,000.

The estimated effect coefficients $e^{\hat{\beta}_j}$ and the marginal effects of the variables on purchase probability are provided by the **glm** functions. However, they can be calculated using the **logitor** and **logitmfx** functions of the R package **mfx**,

```
> library(mfx).
```

The **logitor** function also outputs the effect coefficients with their standard errors, as well as z- and p-values:

```
> Consumers.effcoef = logitor(Purchase~Income+Gender, data=Consumers.df)
> Consumers.effcoef
Call:
logitor(formula = Purchase ~ Income + Gender, data = Consumers.df)

Odds Ratio:
          OddsRatio  Std. Err.       z  P>|z|
Income  1.0044905  0.0020476   2.1979  0.02796 *
Gender   0.0564467  0.1036614 -1.5652  0.11753
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

The effect coefficient of income of 1.0044905 means that the odds for a product purchase increase by 0.45% when a consumer's income climbs by 1 euro. Similarly, the estimated effect coefficient of gender of 0.0564467 indicates that the chance ratio for product purchase compared with non-purchase sinks when changing from the group of male consumers ($x_3 = 0$) to the group of female consumers ($x_3 = 1$) by 94.36%.

The marginal effect of an independent variable x_j on estimated purchase probability $\hat{p}_t$ depends not only on their estimated regression coefficient $\hat{\beta}_j$, but also x_{jt} of the variables:

$$\frac{\partial \hat{p}_t}{\partial x_{jt}} = \frac{e^{\hat{\boldsymbol{\beta}}'\mathbf{x}_t}}{\left(1+e^{\hat{\boldsymbol{\beta}}'\mathbf{x}_t}\right)^2} \cdot \hat{\beta}_j \tag{6.16}$$

However, when marginal effects are specified, reference is frequently made to the mean values of the variables, which is why the argument atmean must be set to TRUE (presetting) thelogitfmx function:

```
> Consumers.margeff = logitmfx(Purchase~Income+Gender, atmean=TRUE, data=Consumers.df)
> Consumers.margeff
Call:
logitor(formula = Purchase ~ Income + Gender, data = Consumers.df,
   atmean = TRUE)

Marginal Effects:
                  dF/dx         Std. Err.        z      P>|z|
Income   0.00088043   0.00040834   2.1561   0.03108 *
Gender  -0.49707668      0.22825481  -2.1777   0.02943 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

0.1 ' ' 1 dF/dx is for discrete change for the  following variables:

[1] "Gender".
```

The marginal effect of income of 0.00088 results at an arithmetic mean of €3,200,

```
> Income.mean = mean(Consumers.df$Income)
> Income.mean
[1] 3200,
```

on the condition that the probability of a female (male) consumer is present through the relative frequency of sample (relative frequency of a female consumer: 14/25=0.56):

```
> n = length(Consumers.df$Gender)
> n
[1] 25
> n1 = sum(Consumers.df$Gender)
> n1
[1] 14
> h1  = n1/n
> h1
[1] 0.56
> b1 = Consumers.logit$coeff[1]
> b1
(Intercept)
  -11.72644
> b2 = Consumers.logit$coeff[2]
> b2
  Income
0.004480407
> b3 = Consumers.logit$coeff[3]
> b3
Sex
 -2.874458
> Income.margeff = as.numeric((exp(b1+b2*3200+b3*h1)/(1+exp(b1+b2*3200+b3*h1))^2)*b2)
> Income.margeff
[1] 0.0008804252.
```

The probability of the purchase of a product accordingly increases by approximately 8.8 percentage points when mean income rises by € 100. For a male consumer with average income, there is a marginal effect of

```
> > Income.margeffm = as.numeric((exp(b1+b2*3200+b3*0)/(1+exp(b1+b2*3200+b3*0))^2)*b2)
> Income.margeffm
[1] 0.0002856618
```

and the corresponding marginal effect for a female consumer is given by

```
> > Income.margefff = as.numeric((exp(b1+b2*3200+b3*0)/(1+exp(b1+b2*3200+b3*0))^2)*b2)
> Income.margeffw
[1] 0.001100869
```

.

By contrast, the marginal effect for the qualitative variable gender is not calculated based on the formula (6.16), but instead using (6.12) as the difference between the purchase probabilities of a female and male consumer with average income:

```
> pt.female.mean = as.numeric(1/(1+exp(-b1-b2*3200-b3)))
> pt.female.mean
[1] 0.434481
> pt.male.mean = as.numeric(1/(1+exp(-b1-b2*3200)))

> pt.male.mean
[1] 0.9315576
> Gender.margeff = pt.female.mean - pt.male.mean
> Geschlecht.margeff
[1] -0.4970767.
```

Here, the marginal effect of -0.4970767, indicates that the probability of a product purchase falls by 49.7 percentage points when a switch is made from male to female consumer.

The quality of the model adjustment can be checked in various ways. For a detailed examination, various types of residuals can be analysed that are available when the residuals function is applied to the glm object Consumers.logit. Specifying the type argumentas “response“,

```
> resid.response = residuals(Consumers.logit, type="response")
> resid.response
          1            2            3            4            5
-0.0322993160  0.1525441520  0.0684423640  0.2534056822 -0.1134721948
          6            7            8            9           10
-0.2387266614  0.0020350623 -0.0086286691  0.0049712006 -0.1669062422
         11           12           13           14           15
 0.9503501946 -0.2741914763  0.0008316448  0.0077593308 -0.5459819239
         16           17           18           19           20
 0.0225596471 -0.0322993160  0.0031817059  0.0003396163 -0.0035400070
         21           22           23           24           25
-0.3292382064  0.0291151990  0.2534056822  0.0049712006 -0.0086286691,
```

causes the residuals to be formed as the difference between the dichotomous dependent variables (response variable) Purchase and the estimated purchase probabilities $\hat{p}_t$:

```
> Consumers.df$Purchase-Consumers.logit$fitted.values
           1            2            3            4            5
-0.0322993160  0.1525441520  0.0684423640  0.2534056822 -0.1134721948
           6            7            8            9           10
-0.2387266614  0.0020350623 -0.0086286691  0.0049712006 -0.1669062422
          11           12           13           14           15
 0.9503501946 -0.2741914763  0.0008316448  0.0077593308 -0.5459819239
          16           17           18           19           20
 0.0225596471 -0.0322993160  0.0031817059  0.0003396163 -0.0035400070
          21           22           23           24           25
-0.3292382064  0.0291151990  0.2534056822  0.0049712006 -0.0086286691.
```

A chart containing the response residuals can be created with the plot command. In order to show the residuals as vertical lines, the type argument is set to “h”:

```
> plot(resid.response, main=”Response residuals”, xlab=“Consumer”, ylab= “Response residuals”, type=“h”, lwd=2,cex.main=1.25, font.main=1)
> box(which=“figure”)
```

Figure 6.2: Chart of the response residuals of the logit model

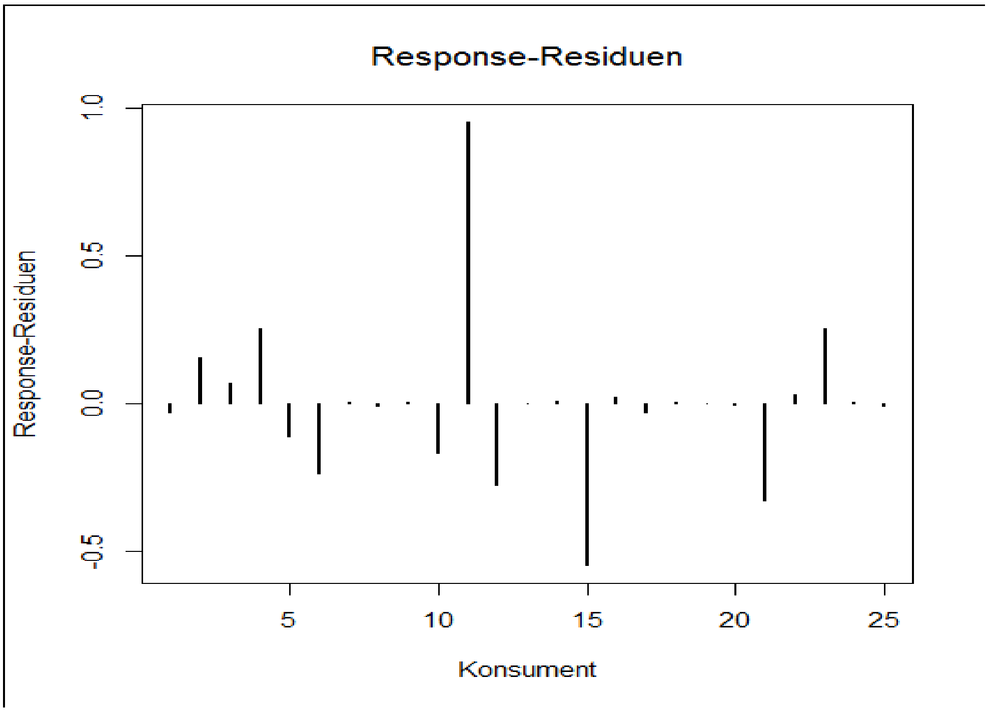


Figure 6.1 shows the differences between the purchase decisions of consumers and the estimated purchase probability. Here there are noticeably large differences for consumers 11 and 15 in particular.

In global terms, the quality of the model adjustment can be checked with the help of the likelihood ratio test and using a pseudo R^2 comparable to the coefficient of determination. For the validity of the null hypothesis

$$H_0: \beta_2 = \beta_3 = \ldots = \beta_p = 0$$

the logarithmised likelihood function (restricted likelihood function) is provided by

(6.17) $$L_0^* = \ln L_0 = n_1 \cdot \ln\left(\frac{n_1}{n}\right) + (n - n_1) \cdot \ln\left(\frac{n - n_1}{n}\right)$$

. In contrast, the unrestricted likelihood function L* is based on the adjustment of the logit model:

(6.18) $$L^* = \ln L = \sum_{t=1}^{n} y_t \cdot \ln \hat{p}_t + (1 - y_t) \cdot \ln(1 - \hat{p}_t) \, .$$

Under the null hypothesis (6.17) the likelihood ratio statistic

(6.19) $$l = -2\left(L_0^* - L^*\right)$$

asymptotically describes a χ^2 distribution with k-1 degrees of freedom.

In the consumer example, the null hypothesis H_0: $\beta_2 = \beta_3 = 0$ indicates that the two variables income and gender have no influence on the probability of product purchase. The likelihood ratio statistic (6.19) can be calculated using equations (6.18) and (6.19):

```
> L0s = n1*log(n1/n)+(n-n1)*log((n-n1)/n)
> L0s
[1] -17.14825
> Ls = Consumers.df$Purchase%*%log(Consumers.logit$fitted.values)  + (1-
Consumers.df$Purchase)%*% log(1-Consumers.logit$fitted.values)
> Ls
       [,1]
[1,] -6.071946
> l = -2*(L0s - Ls)
> l
      [,1]
[1,] 22.1526.
```

The 99% quantile of the χ^2 distribution with 2 degrees of freedom assumes the value of

```
> chi2krit1 = qchisq(0.99, 2)
> chi2krit1
[1] 9.21034
```

As the value of the test statistic of 22.1526 exceeds the critical value of 9.21034, the estimated logit model (6.15) is statistically valid at the 1% level. The high significance is also confirmed by the p-value

```
> p.chi = 1 - pchisq(l, 2)
> p.chi
        [,1]
[1,] 1.547477e-05
```

which is clearly over the nominal level of significance α of 0.01.

The pseudo coefficient of significance of McFadden, PR^2, is based on the ratio L_0^*/L^* of the logarithmised likelihood function:

$$PR^2 = 1 - \frac{L^*}{L_0^*} \quad (6.20)$$

PR^2 reflects the degree of improvement in the explanation of the logits that results from the inclusion of variables x_2, x_3, …, x_k compared to a model that only contains the constant. Although the pseudo R^2 is between 0 and 1, values between 0.2 and 0.4 indicate a good adjustment.

In the consumer example, the pseudo R^2 of 0.656

```
> PR2 = 1 - Ls/L0s
> PR2
        [,1]
[1,] 0.645914,
```

indicates good model adjustment with the two variables income and gender.

Exercises

6.3.1 Estimate the logit model to explain professional status depending on income and age using the maximum likelihood method(refer to 6.1.1 for data) and interpret the regression coefficients!

6.3.2 Plot the logistic distribution functions for self-employed and non-self-employed based on the regression coefficients estimated in Exercise 6.3.1!

6.3.3 Use the logit model estimated in Exercise 6.3.1 to determine the

- Effect coefficients
- Marginal Effects:

and interpret them!

6.3.4 Represent the estimated probabilities and response residuals in scatter diagram (model estimate: Exercise 6.3.1)!

6.3.5 Check null hypothesis H_0: $\beta_2 = \beta_3$ using the likelihood ratio test for the Logit model estimated in Exercise 6.3.1 (α=0.05)!

6.3.6 Calculate McFadden's Pseudo R^2 for the logit model estimated in Exercise 6.3.1 and interpret it!