Introduction to R

Exam Summer Term 2018

Student Name, Student ID, StudentName@uni-muenster.de

Contents

- Answer **7** out of **10** of the following exercises in either German or English.
- Hand in your solutions before Friday, 13 April 2018 at 10 am.
- It is advised to regularly check the learnweb and your emails in case of urgent updates.
- Please sent your solutions files to Willi Mutschler. We will confirm the receipt of your work also by email.
- The solution files should contain your executable and commented script file or preferably a R Notebook.
- You may use *any* available R package you find fit to solve the exercise.
- Please label your axes and title in your plots.
- **All students must work on their own.**

Linear Equation

Solve the linear equation $A \cdot x = b$ with

$$A = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 2 & 5 & 8 & 6 \\ 4 & 3 & 7 & 9 \\ 3 & 6 & 5 & 1 \end{pmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 4 \\ 3 \\ 5 \end{pmatrix}$$

Solution:

```
A = matrix(c(1,2,4,3,1,5,3,6,1,8,7,5,3,6,9,1),4,4)
b = matrix(c(1,4,3,5),4,1)
x=solve(A,b)
print(x)
```

```
##             [,1]
## [1,]  0.20512821
## [2,]  0.76923077
## [3,] -0.05128205
## [4,]  0.02564103
```

and compute the inverse as well as Eigenvalues of A.

Solution:

```
solve(A)
```

```
##            [,1]        [,2]        [,3]        [,4]
## [1,] -0.3589744 -0.43589744  0.38461538  0.23076923
## [2,]  0.6538462  0.11538462 -0.30769231  0.11538462
## [3,] -0.6602564  0.10897436  0.15384615 -0.05769231
## [4,]  0.4551282  0.07051282 -0.07692308 -0.09615385
```

```
eigen(A)$values
```

```
## [1] 17.543241+0.000000i -3.624256+0.000000i  0.040507+1.565859i
## [4]  0.040507-1.565859i
```

Functions

1. Write a function `psum(n,a)` that computes

$$s_{n,a} := \sum_{k=0}^{n} \frac{k^a}{k^a+1}$$

   for any natural number $n \in \{1, 2, ...\}$ and any $a > 0$.

Solution:

```
psum = function(n,a){
if (n==0) 0
else ((n^a)/((n^a)+1)+psum(n-1,a))}
psum(10,10)
```

```
## [1] 9.499006
```

2. Write a function `mymatrix(n)` that returns a $n \times n$ matrix such that: the first and last row as well as the first and last column contain only ones, whereas the remaining values are zero, e.g.

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Solution:

```
mymatrix = function(n){
  matrix(c(rep(1,n),rep(c(1,rep(0,n-2),1),n-2),rep(1,n)),n,n)
  }
mymatrix(4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    0    0    1
## [3,]    1    0    0    1
## [4,]    1    1    1    1
```

Passenger numbers

The file **apass.csv** contains monthly data on passenger numbers of US airlines from January 1949 to December 1959.

1. Read the data into a data frame.

Solution:

```
apass = read.csv2("../data/apass.csv",header=TRUE)
```

2. Create a vector that contains the corresponding dates. Add this vector to your data frame.

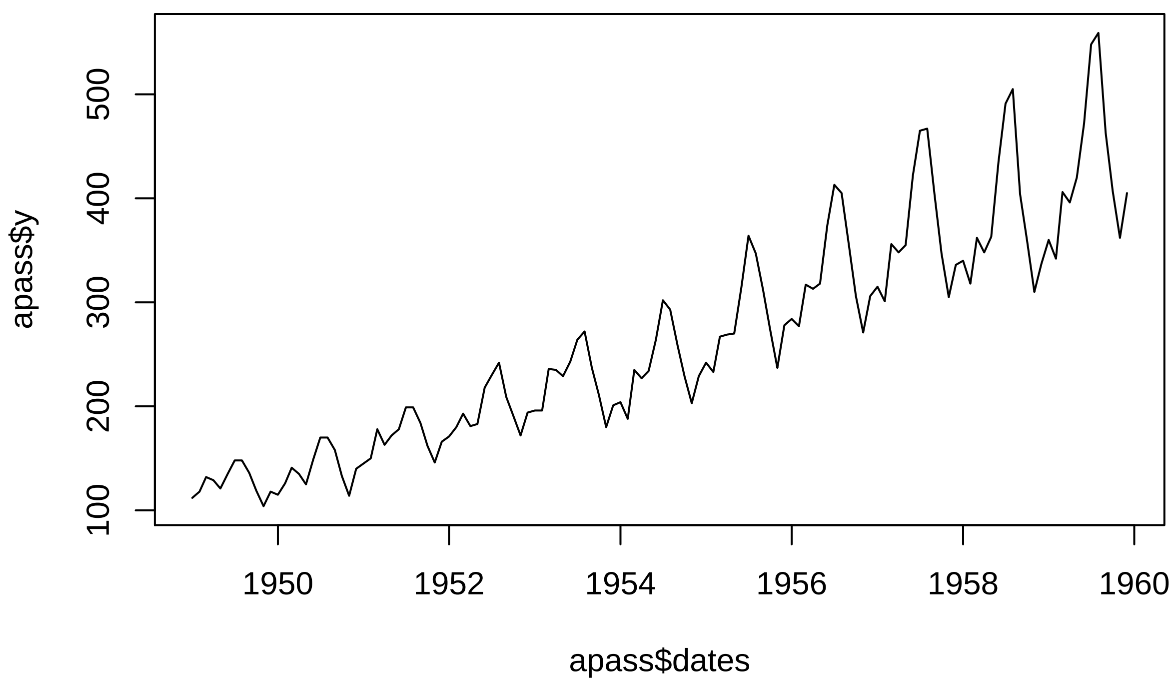
Solution:

```
dat1 <- strptime("1949-01-01","%Y-%m-%d")
dat2 <- strptime("1959-12-01","%Y-%m-%d")
dates <- seq(dat1, dat2, by = "month")
apass$dates <- dates
```

3. Plot the passenger numbers against the date vector.

Solution:

```
plot(apass$dates, apass$y, type = "l")
```

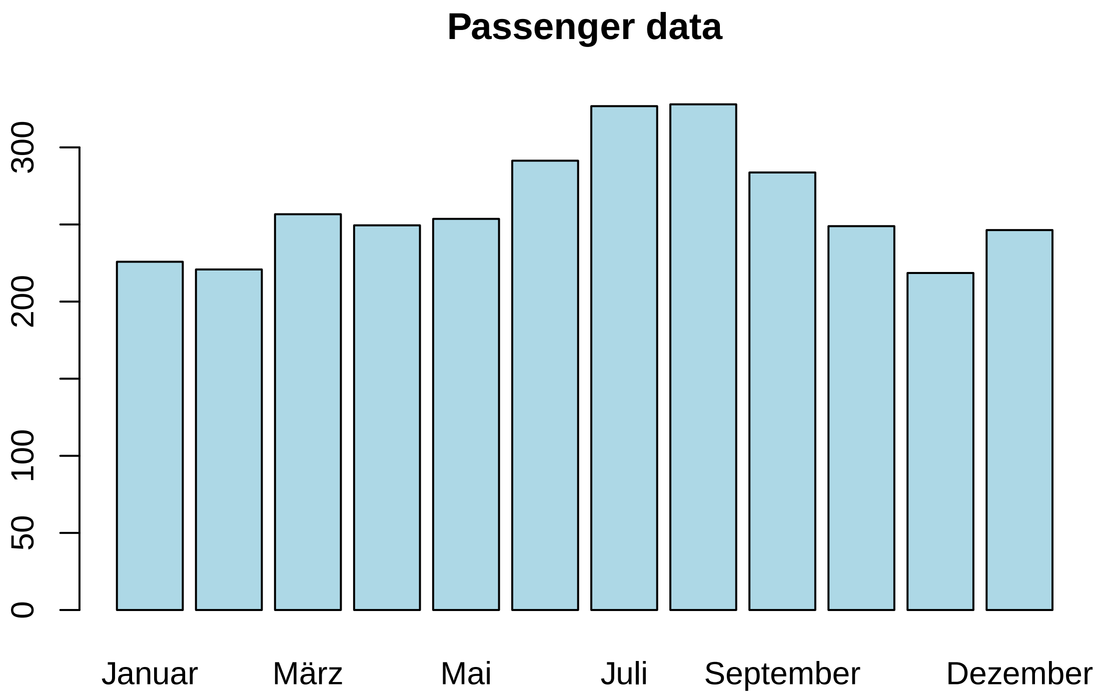


4. Calculate the mean passenger numbers for each month. Plot the means as a bar chart.

Solution:

```
strmon <- unique(months(apass$dates))
Z <- rep(NA,12)
for (i in 1:12) {
  Z[i] <- mean(apass$y[months(apass$dates) == strmon[i]])
```

```
}
barplot(Z, col = "lightblue", names = strmon, main = "Passenger data")
```



Porsche

The file **Porsche911.csv** contains data on Porsche 911 cars, `name` is an id for the owner, `loc` is the location, `age` is the age of the car, `TKM` is the mileage in thousands kilometers and `price` the current price listed on an internet plattform for used cars in thousand Euros.

1. Read in the data into a data frame and compute key descriptive statistics (mean, standard deviation, smallest and largest values, quartiles, covariance and correlation) for the variables `age`, `TKM` and `price`.

Solution:

```
Porsche911=read.csv2(file="../data/Porsche911.csv")[3:5]
summary(Porsche911)
```

```
##       age             TKM              price      
##  Min.   : 1.00   Min.   : 21.00   Min.   :11.00  
##  1st Qu.:20.25   1st Qu.: 51.95   1st Qu.:22.00  
##  Median :29.00   Median : 75.40   Median :29.50  
##  Mean   :25.27   Mean   : 70.04   Mean   :37.55  
##  3rd Qu.:34.00   3rd Qu.: 93.25   3rd Qu.:39.00  
##  Max.   :40.00   Max.   :100.00   Max.   :95.00  
```

```
cov(Porsche911)
```

```
##             age       TKM     price
## age    140.3583  194.4894 -251.5397
## TKM    194.4894  686.3065 -451.6083
## price -251.5397 -451.6083  587.8436
```

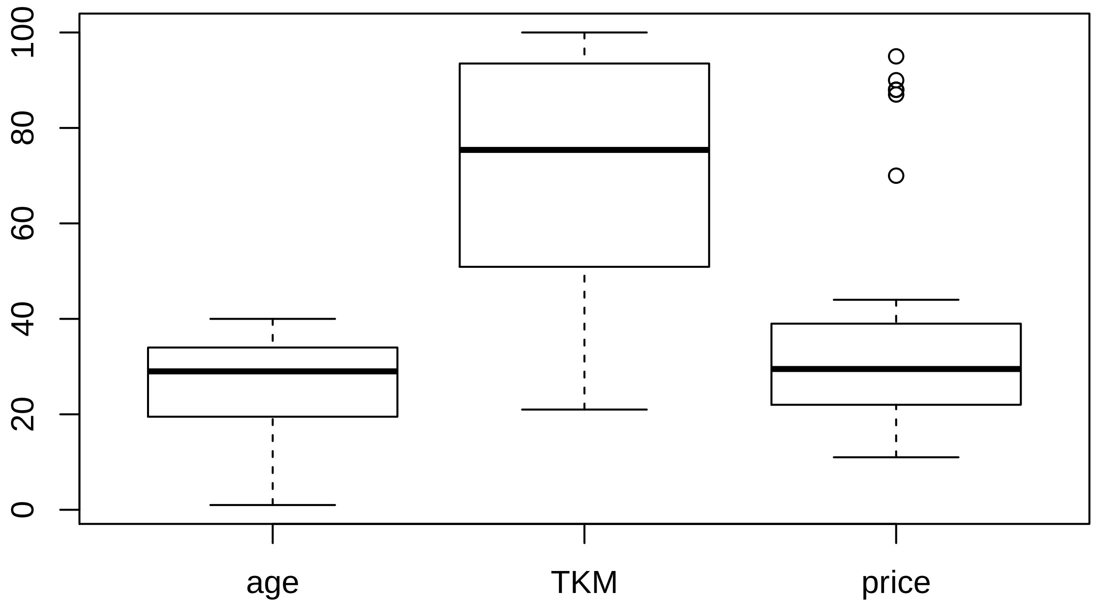
```
cor(Porsche911)
```

```
##              age        TKM      price
## age    1.0000000  0.6266396 -0.8757026
## TKM    0.6266396  1.0000000 -0.7110039
## price -0.8757026 -0.7110039  1.0000000
```

2. Plot boxplots for `age`, `TKM` and `price`.

Solution:

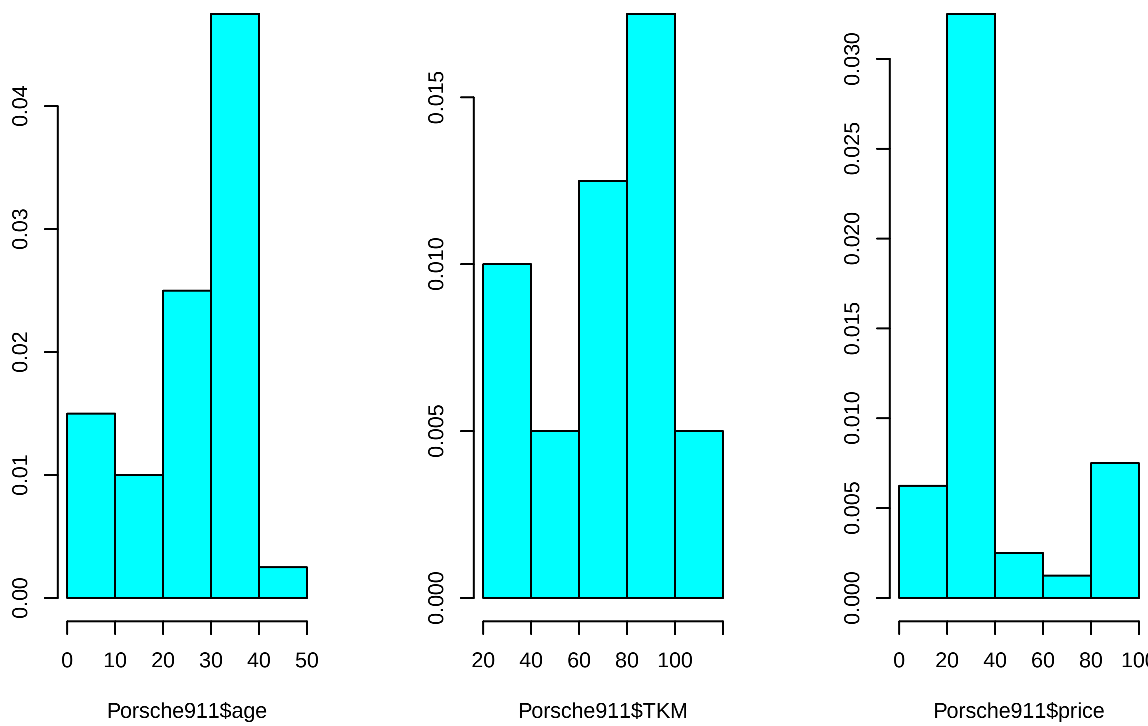
```
boxplot(Porsche911)
```



3. Generate the empirical cumulative distribution function as well as histograms for each of the variables `age`, `TKM` and `price`.

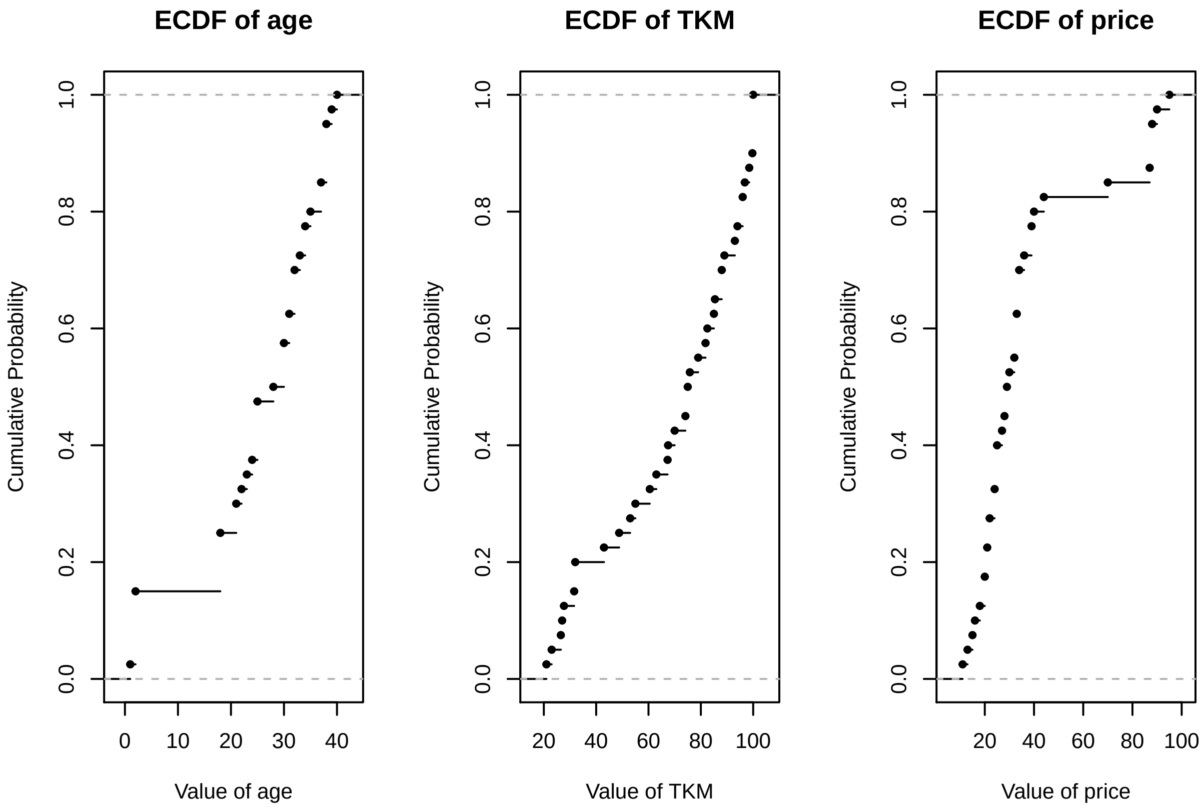
Solution:

```
library(MASS)
par(mfrow=c(1,3))
truehist(Porsche911$age)
truehist(Porsche911$TKM)
truehist(Porsche911$price)
```




```
plot(ecdf(Porsche911$age), main = "ECDF of age", xlab = "Value of age", ylab = "Cumulative Probability")
plot(ecdf(Porsche911$TKM), main = "ECDF of TKM", xlab = "Value of TKM", ylab = "Cumulative Probability")
```

```
plot(ecdf(Porsche911$price), main = "ECDF of price", xlab = "Value of price", ylab = "Cumulative Probabi
```



Law of large numbers

Let $X_1, X_2, ...$ be a sequence $X_1, X_2, ...$ from an $AR(1)$ process:

$$(X_i - \mu) = \rho (X_{i-1} - \mu) + \varepsilon_i$$

where ε_i is uniformly distributed on the interval $[-1, 1]$ and $|\rho| < 1$. Define the sequence of random variables

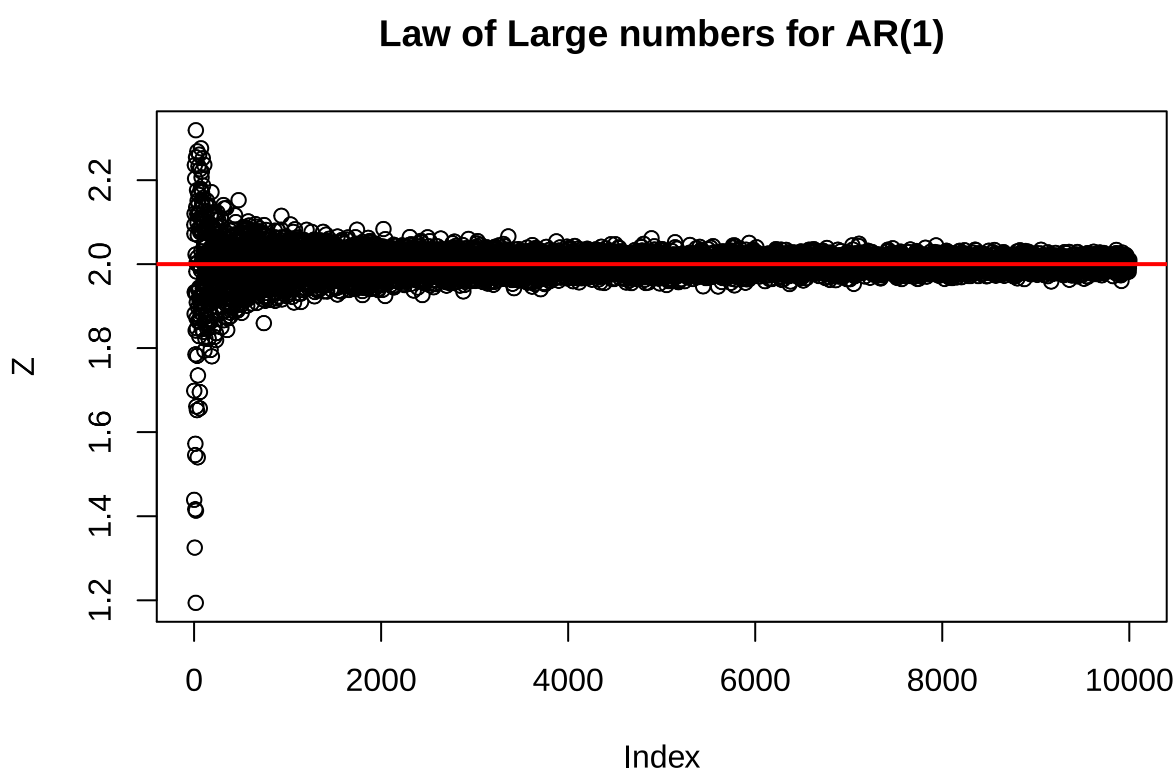
$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

The weak law of large numbers states that the sample average $\bar{X}_n$ converges in probability towards the expected value μ when n tends to infinity.

Show by simulation that the law of large numbers holds despite the intertemporal dependence in X. In particular, show the convergence by means of an appropriate plot. Hint: You may set the parameters to e.g. $\rho = 0.5$ and $\mu = 2$ (or any other value you find fit.)

Solution:

```
n <- 10000
Z <- rep(NA,n)
rho=0.5
mu=2
for (i in 1:n) {
  Z[i] <- mean(filter((1-rho)*mu+runif(i,min=-1,max=1),rho,method="recursive",init=(1-rho)*mu))
}
plot(Z, main="Law of Large numbers for AR(1)");
abline(h=mu,lwd=2,col="red")
```



Limits of maxima

Let $X_1, X_2, \ldots$ be an i.i.d. sequence of random variables uniformly distributed on the interval $[0, 1]$. Define the random variables

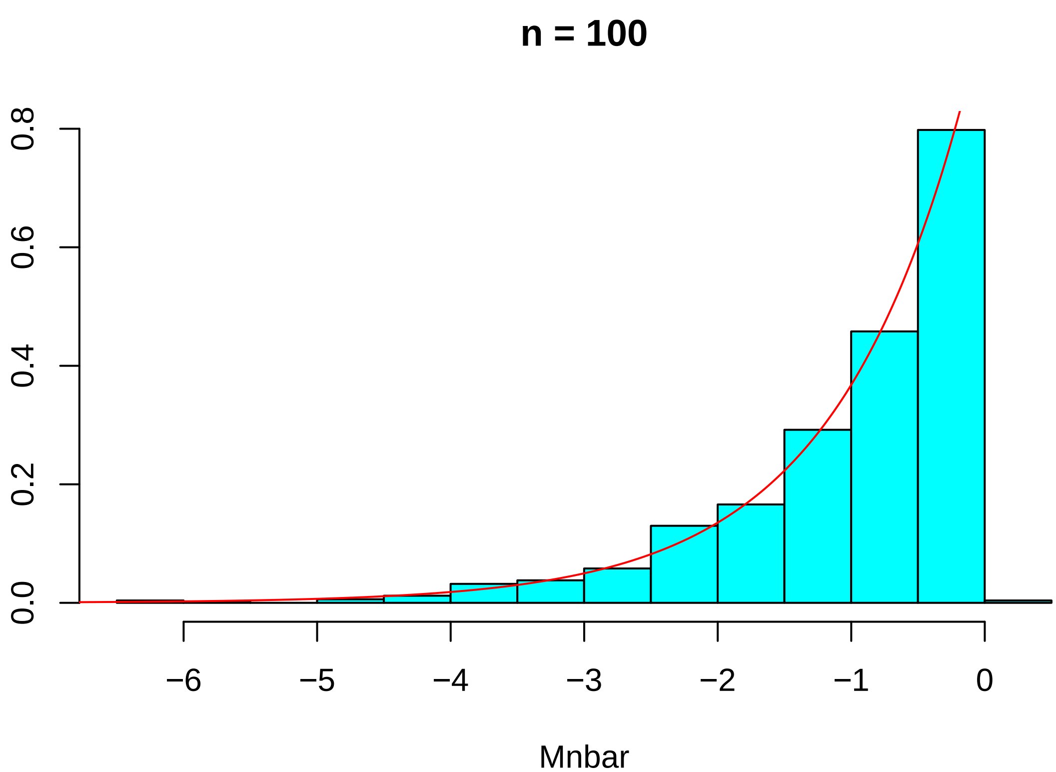
$$M_n = \max_{i=1,\ldots,n} X_i$$

and its normalized version $\overline{M}_n = (M_n - 1) \cdot n$. One can show that the limit distribution of $\overline{M}_n$ is the Weibull distribution with density $\exp(x)$. Write an R program to illustrate that $\overline{M}_n$ converges in distribution. To this end, set $n = 100$ and $R = 1000$ and consider the $R \times n$ matrix with uniformly distributed random variables X_i. Show the convergence by means of an appropriate plot.

Note: Try to avoid using loops (use e.g. apply instead). You will not get full points if you use a loop.

Solution:

```
library(MASS)
n <- 100
R <- 1000 # how many times to draw individual X_i's, note that i = 1,2,..,n
X <- matrix(runif(n*R), R, n) #matrix of N rows and n columns filled with random numbers
Mn <- apply(X, 1, max) # get max of each row
Mnbar <- (Mn-1)*n #standardization
#display the sequence of random variables
truehist(Mnbar, main = paste("n =", toString(n),sep =" "))
coord <- par("usr")
x <- seq(coord[1], coord[2], length.out = 500)
lines(x, exp(x), col = "red")
```



Student teacher ratio

Load the dataset **caschool.csv** into the object `caschool`. This dataset is discussed in great detail in the textbook of Stock and Watson.

1. Make the following variables accessible:

- test score `testscr`
- student-teacher ratio `str`
- percentage of English language learners `el_pct`
- expenditures per student `expn_stu`

Solution:

```
caschool <- read.csv("../data/caschool.csv")
testscr <- caschool$testscr
str <- caschool$str
el_pct <- caschool$el_pct
expn_stu <- caschool$expn_stu
```

2. Regress `testscr` on a constant and `str`. Assign the residuals of the regression into the variable `r1`. Now regress `testscr` on an intercept, `str`, `el_pct` and `expn_stu`. Put the residuals into the variable `r2`. Compute the sum of squared residuals for both regressions. Display `r1` and `r2` in one plot, where the points of `r2` are marked red.

Solution:

```
simple <- lm(testscr~str)
multiple <- lm(testscr~str + el_pct + expn_stu)
r1 <- residuals(simple)
r2 <- residuals(multiple)
sum(r1^2)
```

```
## [1] 144315.5
```

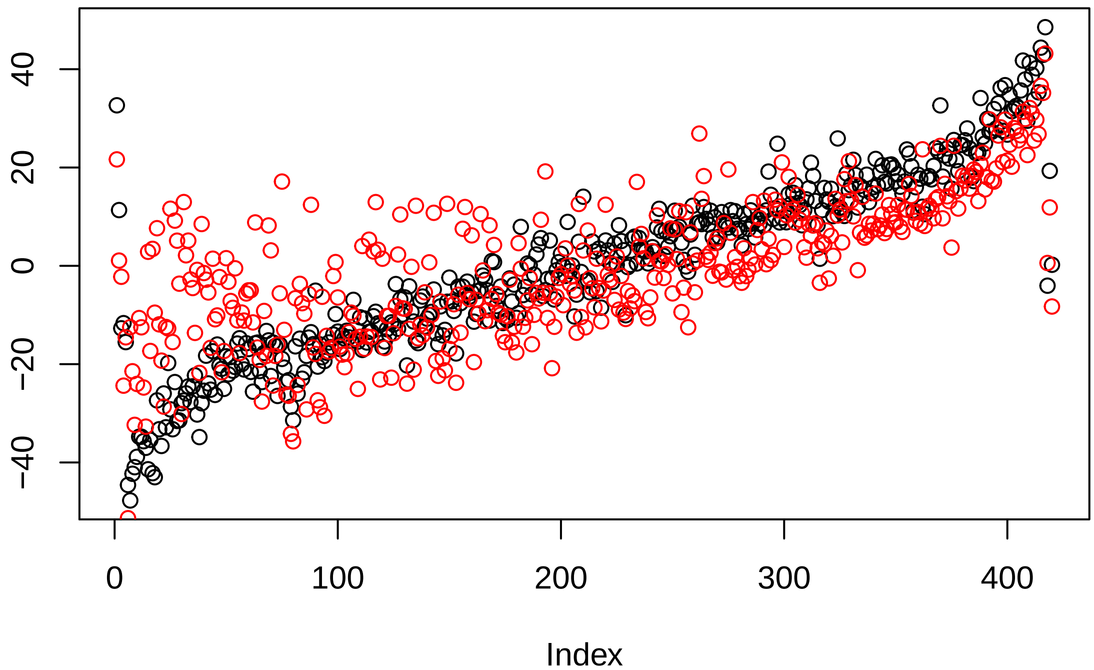
```
sum(r2^2)
```

```
## [1] 85699.69
```

```
sum(r1^2) > sum(r2^2)
```

```
## [1] TRUE
```

```
plot(r1,ylab="")
points(r2,col="red")
```



3. Consider the regression of `testscr` on a constant, `str`, `elp_ct` and `expn_stu`. Predict the value of `testscr` for a school district with an average class size (`str`) of 25 students, a percentage of English learners (`el_pct`) of 60% and an average expenditures per student (`expn_stu`) of 4000$.

Solution:

```
predict(multiple, newdata=data.frame(str=25, el_pct=60, expn_stu=4000))
```

```
##        1
## 618.5282
```

4. Reconsider the regression of `testscr` on a constant, `str`, `el_pct` and `expn_stu`. Compute the heteroscedastic robust standard errors.

Solution:

```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(sandwich)
```

```
regr <- lm(testscr~str + el_pct + expn_stu)
summary(regr)
```

```
##
## Call:
## lm(formula = testscr ~ str + el_pct + expn_stu)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -51.340 -10.111   0.293  10.318  43.181
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 649.577960  15.205716  42.719  < 2e-16 ***
## str          -0.286400   0.480523  -0.596  0.55149
## el_pct       -0.656023   0.039106 -16.776  < 2e-16 ***
## expn_stu      0.003868   0.001412   2.739  0.00643 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.35 on 416 degrees of freedom
## Multiple R-squared:  0.4366, Adjusted R-squared:  0.4325
## F-statistic: 107.5 on 3 and 416 DF,  p-value: < 2.2e-16
```

```r
coeftest(regr, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error  t value Pr(>|t|)
## (Intercept) 649.5779596  15.6686203  41.4573  < 2e-16 ***
## str          -0.2863998   0.4875129  -0.5875  0.55721
## el_pct       -0.6560228   0.0321143 -20.4278  < 2e-16 ***
## expn_stu      0.0038679   0.0016074   2.4063  0.01655 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5. Test the null hypothesis that the coefficients on `str` and `expn_stu` both equal 0 and the coefficient on `el_pct` equals -0.7. Hint: Use the linearHypothesis function of the `car` package.

Solution:

```r
library(car)
linearHypothesis(regr,c("str=0","expn_stu=0","el_pct=-.7"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## str = 0
## expn_stu = 0
## el_pct = - 0.7
##
## Model 1: restricted model
## Model 2: testscr ~ str + el_pct + expn_stu
##
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    419 89117
## 2    416 85700  3    3416.9 5.5287 0.0009919 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Asymptotic normality

Consider the multiple linear regression model $y = X\beta + u$. In R, generate the matrix X by executing the following commands:

```
library(MASS)
X <- cbind(1,mvrnorm(n=100,c(5,10),matrix(c(1,0.9,0.9,1),2,2)))
```

Assume that the true coefficient vector is

$$\beta = \begin{pmatrix} 3 \\ 2 \\ -1 \end{pmatrix}$$

and the error terms are i.i.d. uniformly distributed on the interval $[-1, 1]$. Hence, the assumption of normally distributed error terms is violated.

1. Write an R program that generates $R = 10000$ random samples of size $n = 100$ each. Generate an empty vector `Z <- rep(NA,R)`. For each sample $i = 1, \ldots, R$, compute the OLS estimate $\hat{\beta}$ of β and store the second component of $\hat{\beta}$ in the i-th element of the vector `Z`.

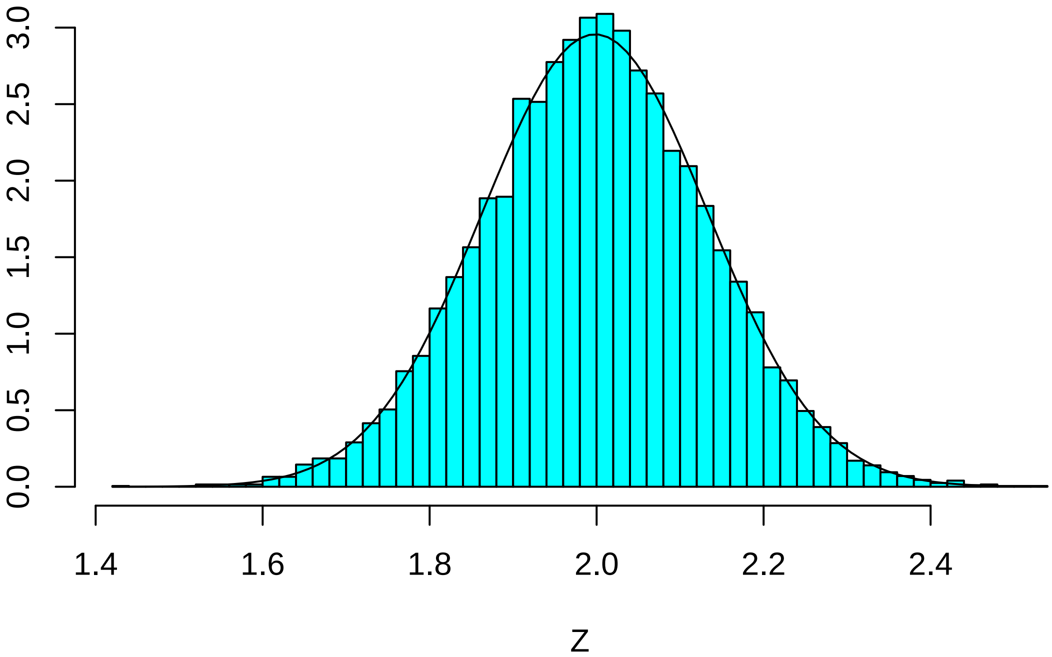
Solution:

```
beta_true <- c(3,2,-1)
n <- 100
R <- 10000
Z <- rep(NA,R)
for (i in 1:R) {
  u <- runif(n,-1,1)
  X <- cbind(1,mvrnorm(n=n,c(5,10),matrix(c(1,0.9,0.9,1),2,2)))
  y <- X%*%beta_true + u
  beta_hat <- solve(t(X)%*%X)%*%t(X)%*%y
  Z[i] <- beta_hat[2]
}
```

2. Plot the histogram of `Z`. Compute the mean m and standard deviation s of `Z` and add the density of $N(m, s)$ to the plot.

Solution:

```
library(MASS)
truehist(Z)
m1 <- mean(Z)
s1 <- sd(Z)
curve(dnorm(x,mean=m1,sd=s1),add=T)
```


0.0
0.5
1.0
1.5
2.0
2.5
3.0
1.4
1.6
1.8
2.0
2.2
2.4
Z

Stochastic frontier analysis

Consider the Cobb-Douglas production function

$$y = Ax_1^{\alpha}x_2^{\beta}$$

By definition, the production function returns the maximal output for given inputs, and actual production cannot be larger than y. Due to inefficiencies, actual production could be modeled (in logs) as

$$\ln y = \ln A + \alpha \ln x_1 + \beta \ln x_2 - u$$

where u is a **non-negative** random variable. Since other disturbances (e.g. measurement errors) can enter the production function, it is more common to add another, **symmetrically distributed**, disturbance term v,

$$\ln y = \ln A + \alpha \ln x_1 + \beta \ln x_2 - u + v$$

Assume that u is exponentially ($u \sim Exp(\lambda)$) and v normally ($v \sim N(0, \sigma^2)$) distributed. One can show that if u and v are independent then the density function of $\varepsilon = v - u$ is given by

$$f_\varepsilon(\varepsilon) = \lambda \exp\left(\lambda\varepsilon + \frac{1}{2}\lambda^2\sigma^2\right)\Phi\left(\frac{-\varepsilon}{\sigma} - \lambda\sigma\right)$$

where Φ is the distribution function (`pnorm`) of $N(0,1)$ and exp the exponential function (`exp`).

1. Load the dataset **sfa.csv**. This dataset is an abbreviated version of table F7.2 of Greene, 2008. The original data appeared in Zellner and Revankar, *Generalized Production Functions*, Review of Economic Studies, 36 (1969), 241-250.

Solution:

```
sfa <- read.csv("../data/sfa.csv")
head(sfa)
```

```
##         State ValueAdd Capital   Labor
## 1     Alabama  126.148   3.804  31.551
## 2  California 3201.486 185.446 452.844
## 3 Connecticut  690.670  39.712 124.074
## 4     Florida   56.296   6.547  19.181
## 5     Georgia  304.531  11.530  45.534
## 6    Illinois  723.028  58.987  88.391
```

2. Write an R program to estimate the parameters A, α, β, λ and σ by maximum likelihood on this dataset.

Solution:

```
## negative log-Likelihood function
neg_log_likeli <- function(thet, dat){
  A <- thet[1]
  alpha <- thet[2]
  beta <- thet[3]
  lambda <- thet[4]
  sigma <- thet[5]
  x1 <- dat[,3]
  x2 <- dat[,4]
  y <- dat[,2]
```

```r
  eps <- log(y) - log(A) - alpha*log(x1) - beta*log(x2)
  log_likeli <- sum(log(lambda*exp(lambda*eps+lambda^2/2*sigma^2)*pnorm(-eps/sigma-lambda*sigma,mean=0,
  return(-log_likeli)
}

# Estimate parameters
opt <- optim(c(7, 0.3, 0.7, 10, .1), neg_log_likeli, dat=sfa, hessian=T,control=c(maxit=5000))
opt
```

```
## $par
## [1] 7.9194127 0.2625018 0.7703623 7.3942188 0.1713962
##
## $value
## [1] -2.860488
##
## $counts
## function gradient
##      512       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##             [,1]       [,2]        [,3]      [,4]       [,5]
## [1,]   9.240673  220.88401   299.17773  1.006997  12.672484
## [2,] 220.884014 6413.44070  8054.50541 22.567215 173.961216
## [3,] 299.177727 8054.50541 10490.92791 31.047041 277.984249
## [4,]   1.006997   22.56722    31.04704  0.222978  -2.947495
## [5,]  12.672484  173.96122   277.98425 -2.947495 914.229088
```

3. Compute the asymptotic standard errors.

Solution:

```r
sqrt(diag(solve(opt$hessian)))
```

```
## [1] 1.86524162 0.09200213 0.11094965 3.42546237 0.03842999
```

Variance estimation in GARCH

When one considers an iid sample $X_1, \dots, X_n$ from $X \sim N(\mu, \sigma^2)$ then one usually estimates the variance σ^2 using

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

The distribution of the normalized estimator for the variance is given by:

$$\frac{(n-1)\,\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-1}$$

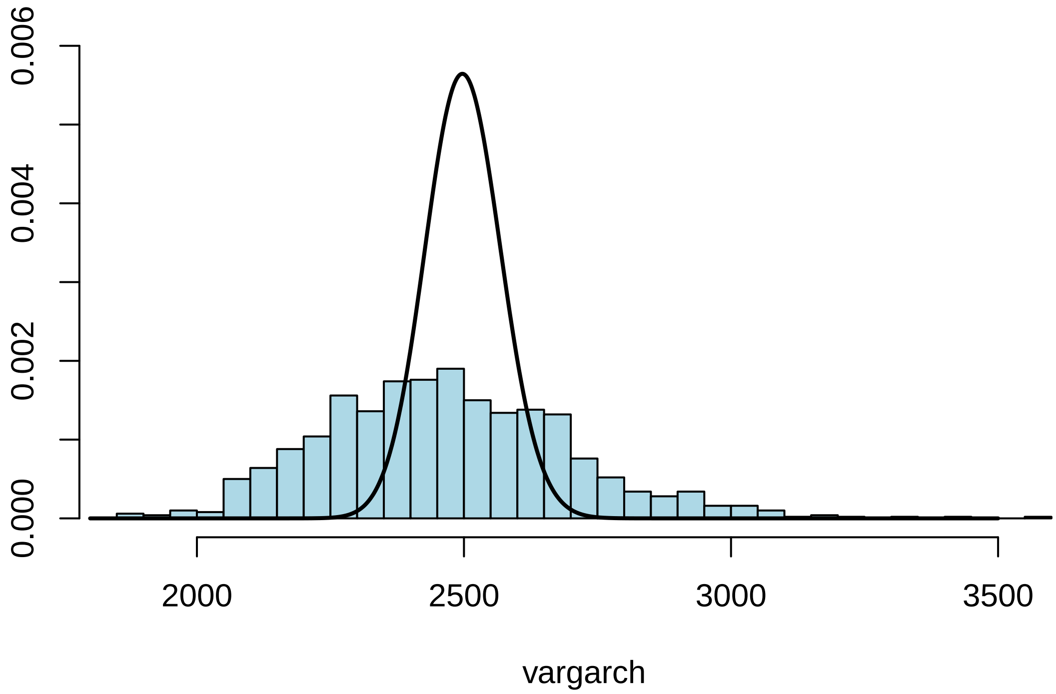
where σ^2 is the true variance. Consider the case when the observations are not iid but are stochastically dependent over time. To this end, assume that $X_1, \dots, X_n$ is a time series generated by a $GARCH(1,1)$ process

$$\begin{aligned} X_i &\sim N(0, \sigma_i^2) \\ \sigma_i^2 &= \omega + \alpha X_{i-1}^2 + \beta \sigma_{i-1}^2 \end{aligned}$$

with $\omega = 0.1$, $\alpha = 0.1$, $\beta = 0.85$ and sample size equal to $n = 2500$. Show by simulations that the distribution of the normalized estimator for the variance is not χ^2_{n-1}-distributed. Hint: The true unconditional variance of this GARCH process is $\sigma^2 = 2$.

Solution:

```r
omega <- 0.1
alpha <- 0.1
beta <- 0.85
n <- 2500
R <- 1000

sigma2 <- rep(NA,n)
sigma2[1] <- 0

X <- rep(NA,n)
X[1] <- 0

vargarch <- rep(NA,R)

for (r in 1:R){
  for (i in 2:n){
    sigma2[i] <- omega+alpha*X[i-1]^2+beta*sigma2[i-1]
    X[i] <- rnorm(n=1,mean=0,sd=sqrt(sigma2[i]))
  }
  varest <- 1/(n-1)*sum((X-mean(X))^2)
  vargarch[r] <- (n-1)*varest/2
}

truehist(vargarch,col="lightblue",ylim=c(0,0.006),nbins=30)
g <- seq(1800,3500,length=1000)
lines(g,dchisq(g,df=n-1),lwd=2)
```


0.000
0.002
0.004
0.006
2000
2500
3000
3500
vargarch