Predictive Modeling

Series 1

Exercise 1.1

This exercise aims at carrying out a simple regression analysis for the data set constructed by Frank Anscombe. The data you need for this exercise is called **anscombe** and consists of four response variables y_i and four predictors x_i. Consider the four models $Y_i^k = \beta_0^k + \beta_1^k \cdot X_i^k + \epsilon_i$ for $k = 1, ..., 4$.

a) Determine for all four models the intercept and the slope of the least squares regression line and their standard errors. Determine as well $\hat{\sigma}$.

R-Hint:

```
reg <- lm(anscombe$y1 ~ anscombe$x1)
```

b) Plot the regression line for all four models in a scatter plot. Comment your observations of the results in a) and b).

R-Hint:

```
plot(anscombe$x1, anscombe$y1, ylab = "Y1", xlab = "X1")
abline(reg)   #add the regression line
```

Exercise 1.2

Prices of antique clocks: McClave and Benson collected data on the basis of auctions about age and price of antique clocks. You find them in the data file **antique_clocks.dat**.

a) Display the data as a scatter plot (price vs. age) and describe their functional dependence. R-Hints:

```
ant_clo <- read.table(file = "antique_clocks.dat", sep = ";",
    header = T)
plot(price ~ age, data = ant_clo)
```

b) Use a linear model to describe the relationship between **price** and **age** and determine the estimated coefficients. R-Hints:

```
fit <- lm(price ~ age, data = ant_clo)
coef(fit)   # exctracts estimated coefficient
summary(fit)   # extracts a summary of the fit
```

c) Draw a regression line in the scatter plot in a). Comment on the results. **R**-Hints:

```
abline(fit)
```

Exercise 1.3

An engineer intends to carry out an analysis of a windmill used for power generation. He collects data about the produced current (in Ampere) at different wind speeds (meter per second). You'll find the data in the file **windmill.dat**. (Source: Montgomery and Peck, *Introduction to Linear Regression Analysis*, Wiley.)

a) Generate a scatter plot (current (y-axis) vs. wind speed) and another scatter plot (current vs. $\frac{1}{\text{wind speed}}$). What do you observe? **R**-Hints:

```
windmill <- read.table("windmill.dat", header = T)
windmill$x <- 1/windmill$wind_speed
par(mfrow = c(1, 2))   # two plots within one graphics
plot(current ~ wind_speed, data = windmill)
plot(current ~ x, data = windmill)
```

b) Use the least squares method to fit the model

$$\text{current} \approx \beta_0 + \beta_1 x \qquad \text{with } x = \frac{1}{\text{wind speed}}$$

Determine the corresponding estimated coefficients and standard errors.

c) Determine a 99 % confidence interval for β_1. **R**-Hints:

```
confint(windmill_lm, parm = 2, level = 0.99)
```

d) Generate a scatter plot (current vs. wind speed). How do you interpret the coefficients β_0 and β_1 in these plots ?
Hint: Let the wind speed approach infinity to interpret β_0 and set the current to zero in order to interpret β_1. A sketch may be useful.

e) Determine the expected value, a 95 % confidence interval for the expected current and a 95 % prediction interval at wind speeds of $1\frac{\text{m}}{\text{s}}$ and $10\frac{\text{m}}{\text{s}}$. Comment on the results. **R**-Hints:

```
wm_new <- data.frame(x = c(1, 10))
```

Expected value along with confidence interval:

```
predict(windmill_lm, newdata = wm_new, interval = "confidence",
    level = 0.95)
```

Predicted value along with prediction interval:

```
predict(windmill_lm, newdata = wm_new, interval = "prediction",
    level = 0.95)
```

Exercise 1.4

In the middle of the 19th century, Scottish physicist James D. Forbes worked on a method to determine the altitude using the boiling point of water. It was known that the altitude can be determined by means of the air pressure. That is the reason why Forbes was interested in a relation between the boiling point of water and the air pressure. The data for this exercise originates from his work published in 1857. **`Forbes.dat`** contains the boiling point **`y`** (in Fahrenheit) and the air **`pressure`** (in inch of mercury) at 17 places in the Alps and in Scotland. (Source: S. Weisberg, *Applied Linear Regression*, Wiley (1985), p. 3)

a) Add the variable **`x`** $= 100 \cdot \log($**`pressure`**$)$ to the data frame **`Forbes`** and plot **`y`** versus **`pressure`** and **`y`** versus **`x`**. Comment on your observations with respect to the two plots.

b) Use a least squares fit to determine the regression line for **`y`** versus **`x`**. Have a look at the regression line in the scatter plot and describe your observations of the result.

c) Use a least squares fit to determine the regression line for **`y`** versus **`x`**, but now omit the 12th observation. Compare the values $\hat{\beta}_0, \hat{\beta}_1, \text{se}(\hat{\beta}_0), \text{se}(\hat{\beta}_1)$ and $\hat{\sigma}$ with the ones you have found in part b).

   R-Hints:

   ```
   fit2 <- lm(y ~ x, data=Forbes[-12,])
   % or
   fit2 <- lm(y ~ x, data=Forbes, subset=-12)
   ```

   In the following exercises, we keep the 12th observation omitted.

d) Test $H_0 : \beta_1 = 0$ versus $H_A : \beta_1 \neq 0$ in the model $Y_i = \beta_0 + x_i + \epsilon_i$ using the output of the regression analysis at the 5 %-level.

e) Determine a 95 %-confidence interval for the slope β_1.

f) Determine the expected value of Y given the predictor value $x_0 = 100 \cdot \log(26) = 325.81$. Determine a 95 % and a 99 % confidence interval for $E[Y|x_0]$.

Voluntary exercise: Plot a 99 % confidence band in the scatter plot.

g) Determine a 99 % prediction interval for the observed value of Y for $x_0 = 325.81$. Compare this interval with the confidence interval you have found in exercise f).

Exercise 1.5

We would like to simulate the distribution of the estimated coefficient values β_0 and β_1. Our model is $Y_i = 4 + 2x_i + \epsilon_i$ with the following x_i values:

i	1	2	3	4	5	6	7	8	9	10
x_i	0	3	4	8	10	11	13	16	17	20

The measurement errors ϵ_i are normally distributed with $\mu = 0$ and $\sigma^2 = 2$.

a) Simulate the 10 values of Y_i on the basis of the model $Y_i = 4 + 2x_i + \epsilon_i$ one hundred times and estimate the values of the regression coefficients β_0 and β_1.

R-Hint, Simulation:

```
x_sim <- c(0,3,4,8,10,11,13,16,17,20)
error_sim <- matrix(rnorm(10*?, mean=?, sd=?), ncol=??)
y_sim <- 4 + 2*x_sim + error_sim
coef <- matrix(0, ncol=2, nrow=100) # Initialisieren
for(i in 1:?) coef[i,] <- coef(lm(y_sim[,i] ~ x_sim))
```

b) Have a look at the distribution of the estimated regression coefficients by means of a histogram and a normal plot. Comment on your observations. Have a look at the joint distribution of the regression coefficients by means of a scatter plot.

c) Determine the mean value of the 100 estimations of β_0 and β_1. Determine as well their variance. Compare the results with the theoretical values.

Result Checker

E 1.4:

e) $[0.4813, 0.4930]$

f) $[205.099, 205.246]$ and $[205.070, 205.275]$

g) $[204.780, 205.566]$

Predictive Modeling

Solutions to Series 1

Solution 1.1

a) The coefficients entering the linear models are determined by means of the **R**-functions **lm()** and **summary()**.

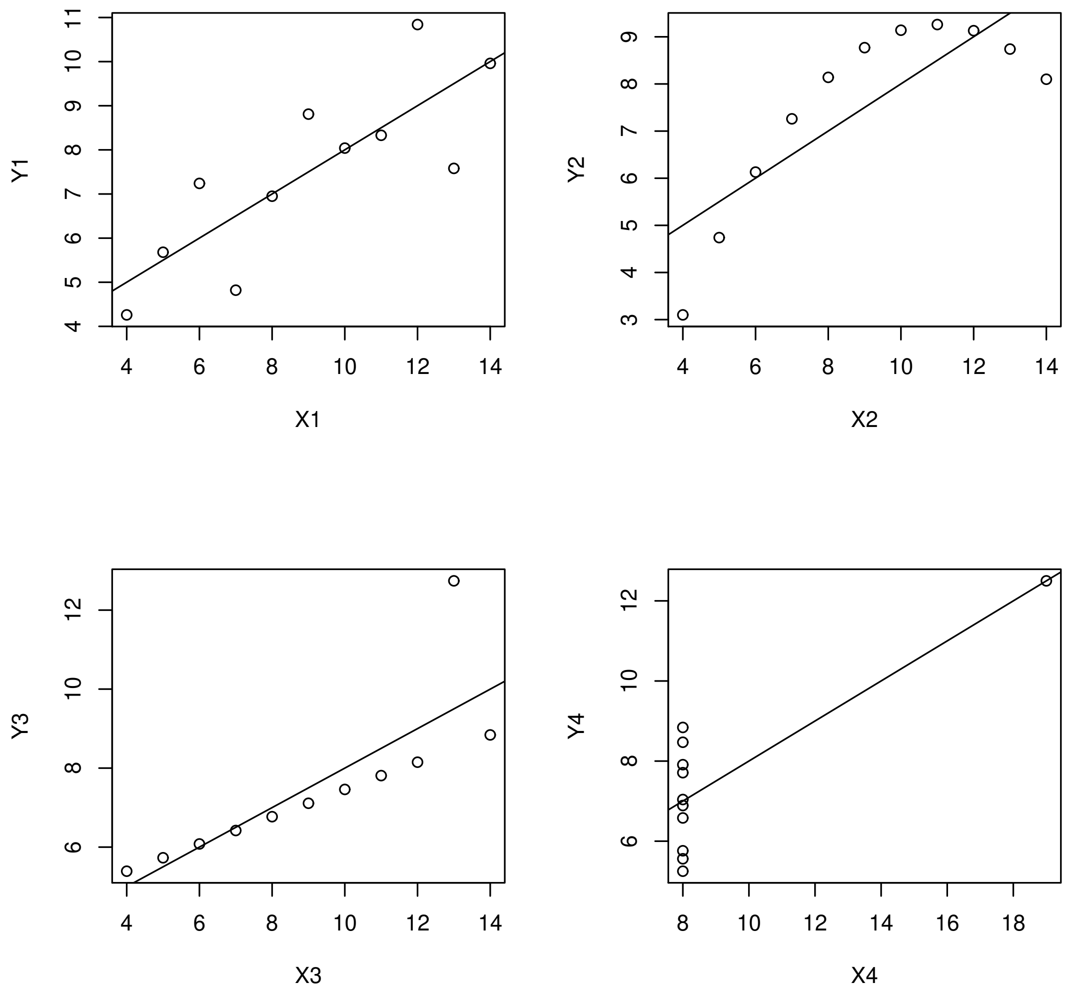
```
data(anscombe)
reg <- lm(anscombe$y1 ~ anscombe$x1)
reg2 <- lm(anscombe$y2 ~ anscombe$x2)
reg3 <- lm(anscombe$y3 ~ anscombe$x3)
reg4 <- lm(anscombe$y4 ~ anscombe$x4)

summary(reg)  #analogous for the other models

##
## Call:
## lm(formula = anscombe$y1 ~ anscombe$x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## anscombe$x1   0.5001     0.1179   4.241  0.00217 **
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665,Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

The intercept β_0 and the slope β_1 are almost identical in all four models (see table).

Even the standard errors and $\hat{\sigma}$ are very similar (1. model: standard error of $\beta_0 = 1.125$, standard error of $\beta_1 = 0.117$ and residual standard error: 1.237).

	model 1	model 2	model 3	model 4
intercept ($\hat{\beta}_0$)	3.000	3.001	3.002	3.002
slope ($\hat{\beta}_1$)	0.500	0.500	0.500	0.500

b)
```
par(mfrow = c(2, 2))
plot(anscombe$x1, anscombe$y1, ylab = "Y1", xlab = "X1")
abline(reg)
plot(anscombe$x2, anscombe$y2, ylab = "Y2", xlab = "X2")
abline(reg2)
plot(anscombe$x3, anscombe$y3, ylab = "Y3", xlab = "X3")
abline(reg3)
plot(anscombe$x4, anscombe$y4, ylab = "Y4", xlab = "X4")
abline(reg4)
```



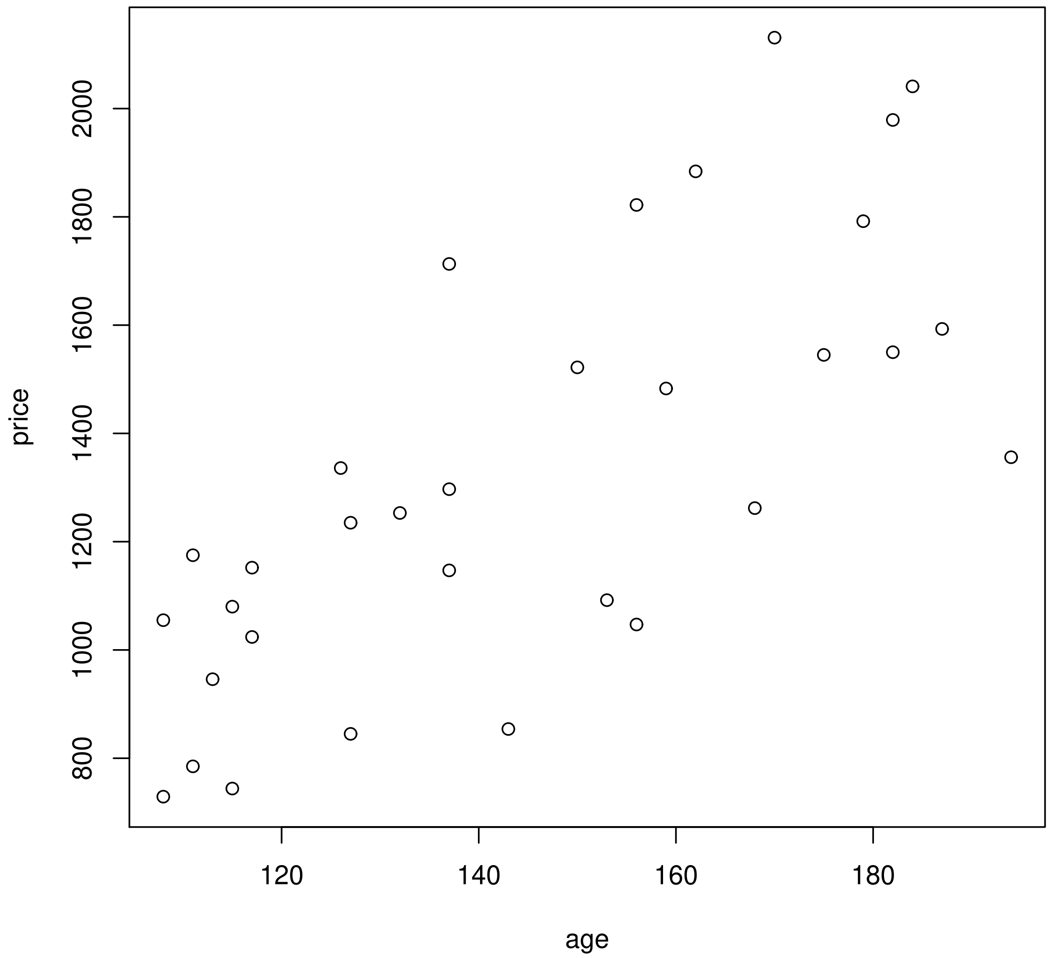
Conclusion: It is not sufficient to simply consider $\hat{\beta}_0$, $\hat{\beta}_1$ and their standard errors. These estimates are almost identical in every model, although the data are completely different. A (graphical) check is essential. Due to the similarity between the values of those coefficients, it is obvious that the regression lines are similar as well.

Solution 1.2

a)
```
d_dir <- "Daten/"
ant_clo <- read.table(file = paste(d_dir, "antique_clocks.dat",
    sep = ""), sep = ";", header = T)
summary(ant_clo)

##        age              price
##   Min.   :108.0   Min.   : 729
```

```
##   1st Qu.:117.0   1st Qu.:1053
##   Median :140.0   Median :1258
##   Mean   :144.9   Mean   :1327
##   3rd Qu.:168.5   3rd Qu.:1561
##   Max.   :194.0   Max.   :2131
```

```
plot(price ~ age, data = ant_clo)
```



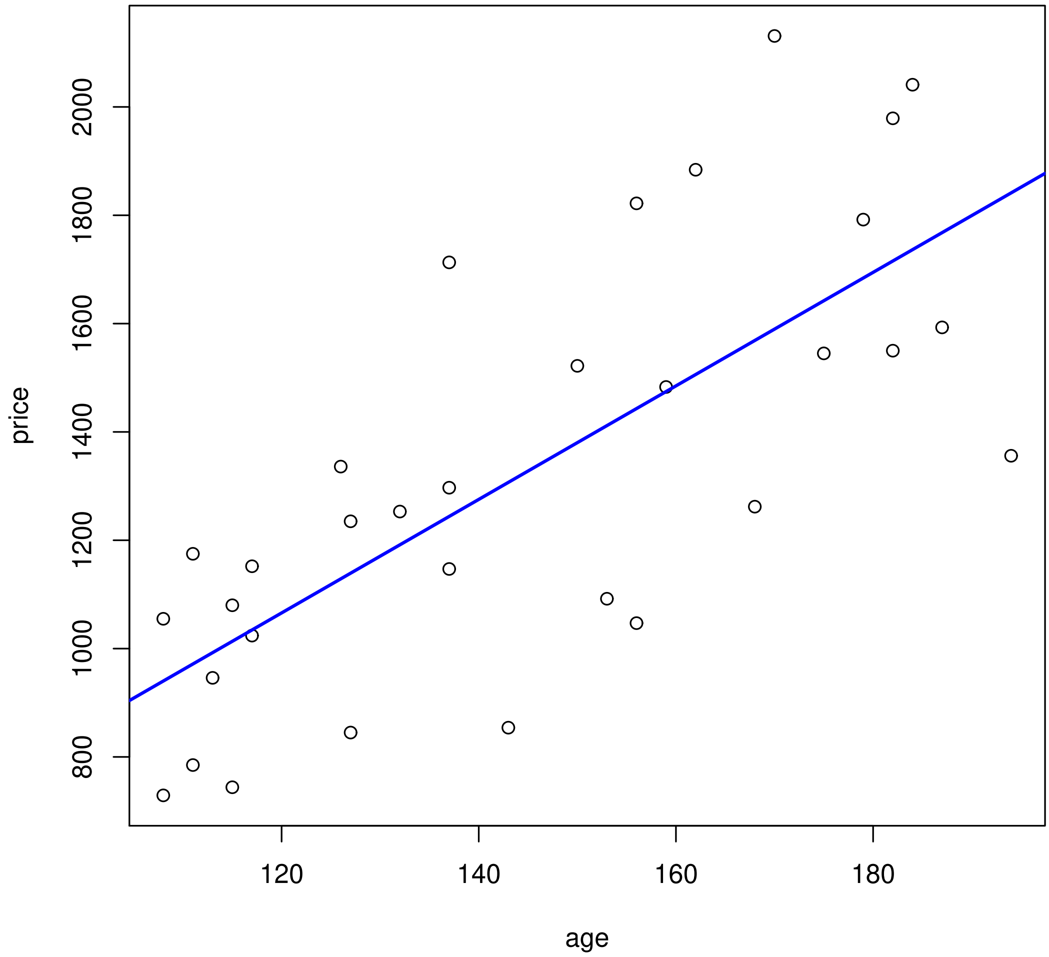
We observe a relatively strong variation in the data. However a linear trend can be observed: the older the clock, the more expensive it is.

b)
```
ant_clo_fit <- lm(price ~ age, data = ant_clo)
summary(ant_clo_fit)  # summary of the fit

##
```

```
## Call:
## lm(formula = price ~ age, data = ant_clo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -485.29 -192.66    30.75   157.21   541.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -191.66     263.89  -0.726    0.473
## age            10.48       1.79   5.854  2.1e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 273 on 30 degrees of freedom
## Multiple R-squared:  0.5332,Adjusted R-squared:  0.5177
## F-statistic: 34.27 on 1 and 30 DF,  p-value: 2.096e-06
```

The estimated coefficients are $\hat{\beta}_0 = -191.66$ and $\hat{\beta}_1 = 10.479$. The residual standard Error is $\hat{\sigma} = 273.028$

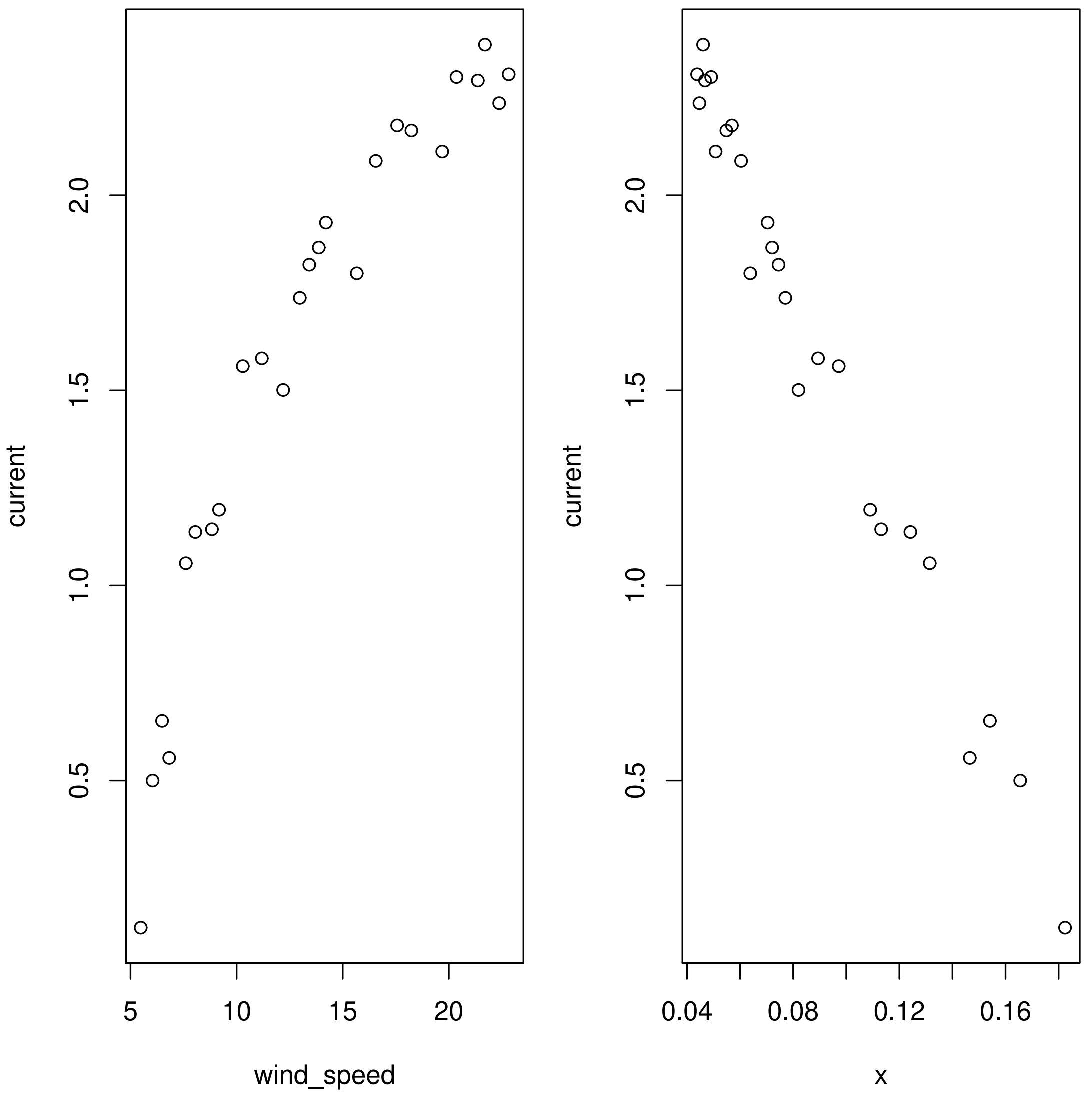
c)
```
plot(price ~ age, data = ant_clo)
abline(ant_clo_fit, col = "blue", lwd = 2)
```



The regression line fits the data quite well.

Solution 1.3

a)
```
d_dir <- "Daten/"
windmill <- read.table(paste(d_dir, "windmill.dat", sep = ""),
     header = T)
windmill$x <- 1/windmill$wind_speed
par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
plot(current ~ wind_speed, data = windmill)
plot(current ~ x, data = windmill)
```



The regression line in the second scatter plot (*current* vs. x) seems to describe data better as compared to the first one.

b)
```
windmill_lm <- lm(current ~ x, data = windmill)
summary(windmill_lm)

##
## Call:
## lm(formula = current ~ x, data = windmill)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20547 -0.04940  0.01100  0.08352  0.12204
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.9789     0.0449   66.34   <2e-16 ***
## x            -15.5155     0.4619  -33.59   <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09417 on 23 degrees of freedom
## Multiple R-squared:   0.98,Adjusted R-squared:  0.9792
## F-statistic:  1128 on 1 and 23 DF,  p-value: < 2.2e-16
```

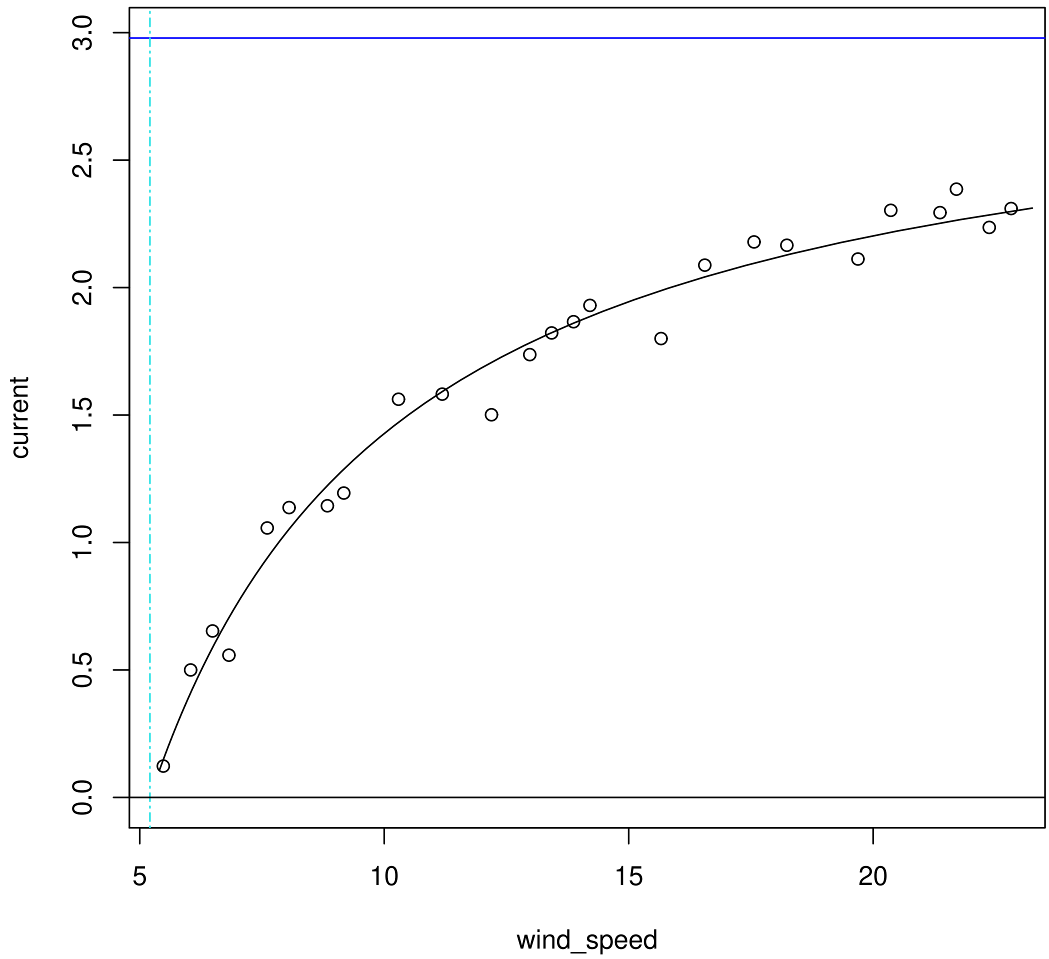
The coefficients are $\hat{\beta}_0 = 2.979$ and $\hat{\beta}_1 = -15.515$. The standard errors are $\text{se}(\hat{\beta}_0) = 0.0449$ and $\text{se}(\hat{\beta}_1) = 0.462$.

c)
```
confint(windmill_lm, parm = 2, level = 0.99)

##       0.5 %    99.5 %
## x -16.8121 -14.21881
```

```
## Explicit calculation in R:
qt(0.995, 23)  ##  99.5%-quantile of t-distribution

## [1] 2.807336

# with 23 degrees of freedom
-15.5155 + c(-1, 1) * qt(0.995, 23) * 0.4619

## [1] -16.81221 -14.21879
```

d)
```
plot(current ~ wind_speed, data = windmill, ylim = range(windmill$cur
    coef(windmill_lm)[1], 0))
## max. power production
abline(h = coef(windmill_lm)[1], col = "blue")
## expected power production:
range(windmill$x)  ## 0.04381808 0.18242629

## [1] 0.04381808 0.18242629

## [1] 0.04381808 0.18242629
wm_new_2 <- data.frame(x = seq(0.043, 0.185, length = 50))
lines(1/wm_new_2$x, predict(windmill_lm, newdata = wm_new_2))
## How much wind is required in order to produce power
## at all
abline(v = -coef(windmill_lm)[2]/coef(windmill_lm)[1], col = 5,
```

```
     lty = 6)
abline(h = 0)
```



The model is

$$\text{current} \approx \beta_0 + \beta_1 \frac{1}{\text{wind speed}}$$

As the wind speed approaches infinity, β_0 becomes the maximally accessible current production (horizontal blue line).

The coefficient β_1 is harder to interpret. It refers to the wind speed, at which the windmill starts to produce an electrical current at all:

$$0 = \beta_0 + \beta_1 \frac{1}{\text{windspeed}_0}$$

$$\text{windspeed}_0 = -\frac{\beta_1}{\beta_0}$$

This means, the larger the absolute value of β_1 the larger the wind speed has to be, in order to have a windmill producing power.

e)
```
wm_new <- data.frame(x = c(1, 1/10))   #
predict(windmill_lm, newdata = wm_new, interval = "confidence",
    level = 0.95)

##           fit        lwr        upr
## 1 -12.536597 -13.408613 -11.664581
## 2   1.427314   1.386768   1.467861
```

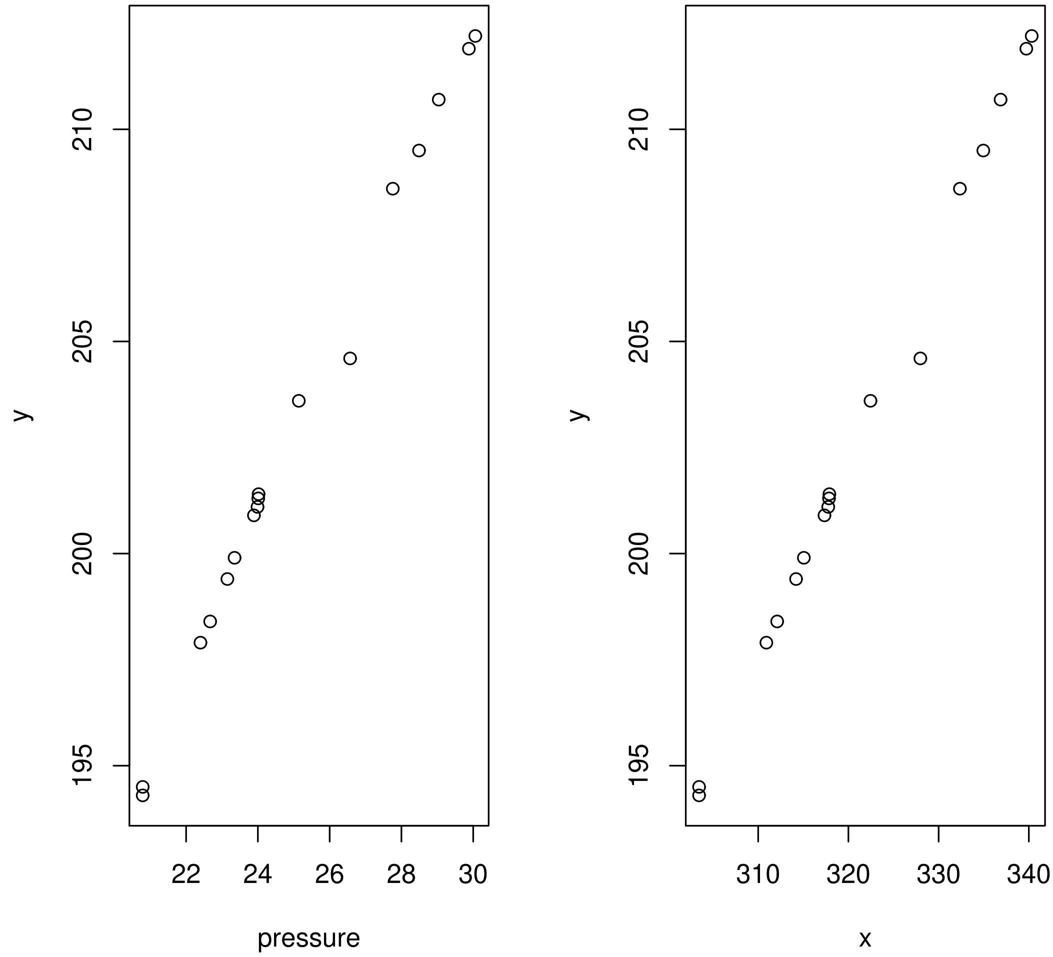
Expected value with prediction interval:

```
predict(windmill_lm, newdata = wm_new, interval = "prediction",
    level = 0.95)

##           fit        lwr        upr
## 1 -12.536597 -13.430108 -11.643086
## 2   1.427314   1.228331   1.626298
```

For the speed of $10\frac{\text{m}}{\text{s}}$ we obtain a value of 1.43 A. As expected, the prediction intervals are (slightly) larger than the confidence intervals. The results for a wind speed of one meter per second do not make sense, because the windmill does not yet rotate (see exercise before). This problem arises because of the extrapolation of the model, which in this case is obviously non-sense.

Solution 1.4

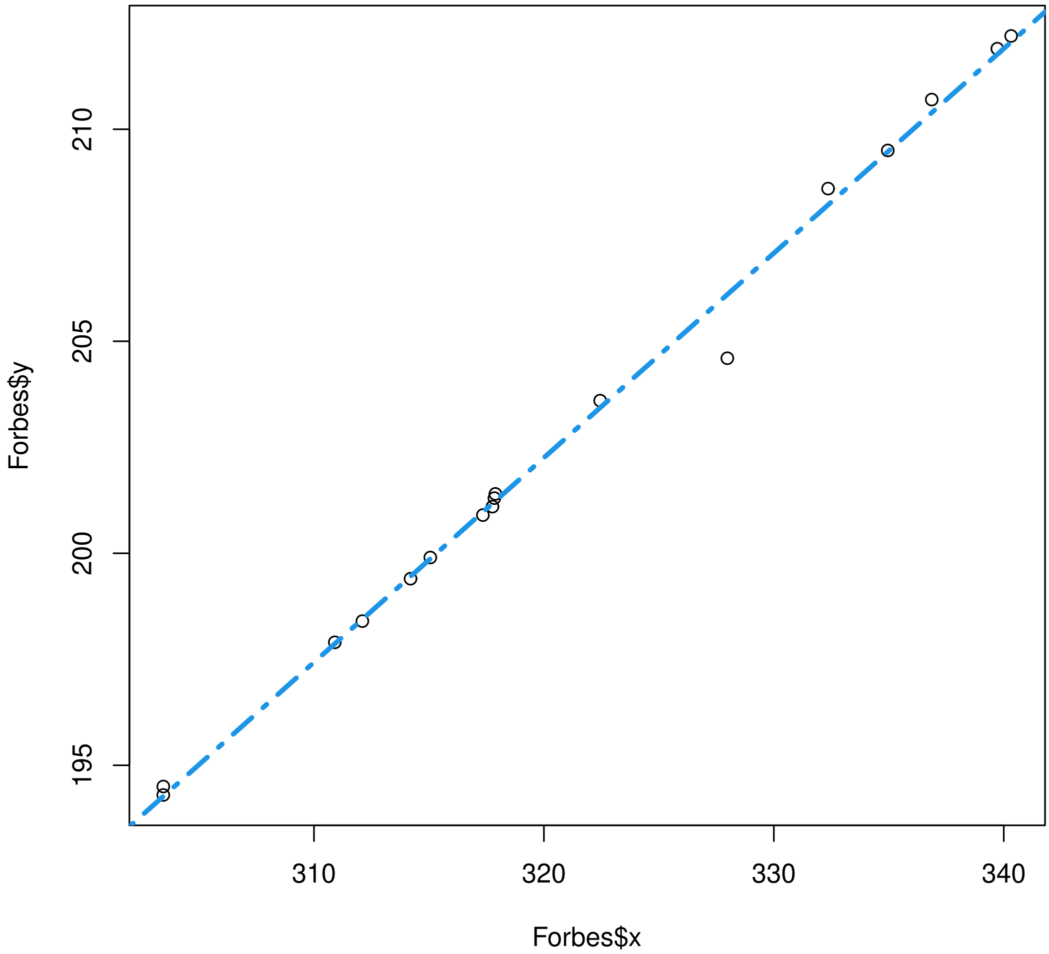
a)
```
d_dir <- "Daten/"
Forbes <- read.table(paste(d_dir, "Forbes.dat", sep = ""),
    header = T)
Forbes$x <- 100 * log(Forbes$pressure)
par(mfrow = c(1, 2))
plot(y ~ pressure, data = Forbes)
plot(y ~ x, data = Forbes)
```



If we have a thorough look at the plot, we observe that data points in the first scatter plot lie on a slightly curved line. In the second scatter plot we observe that data points scatter almost perfectly around a straight line.

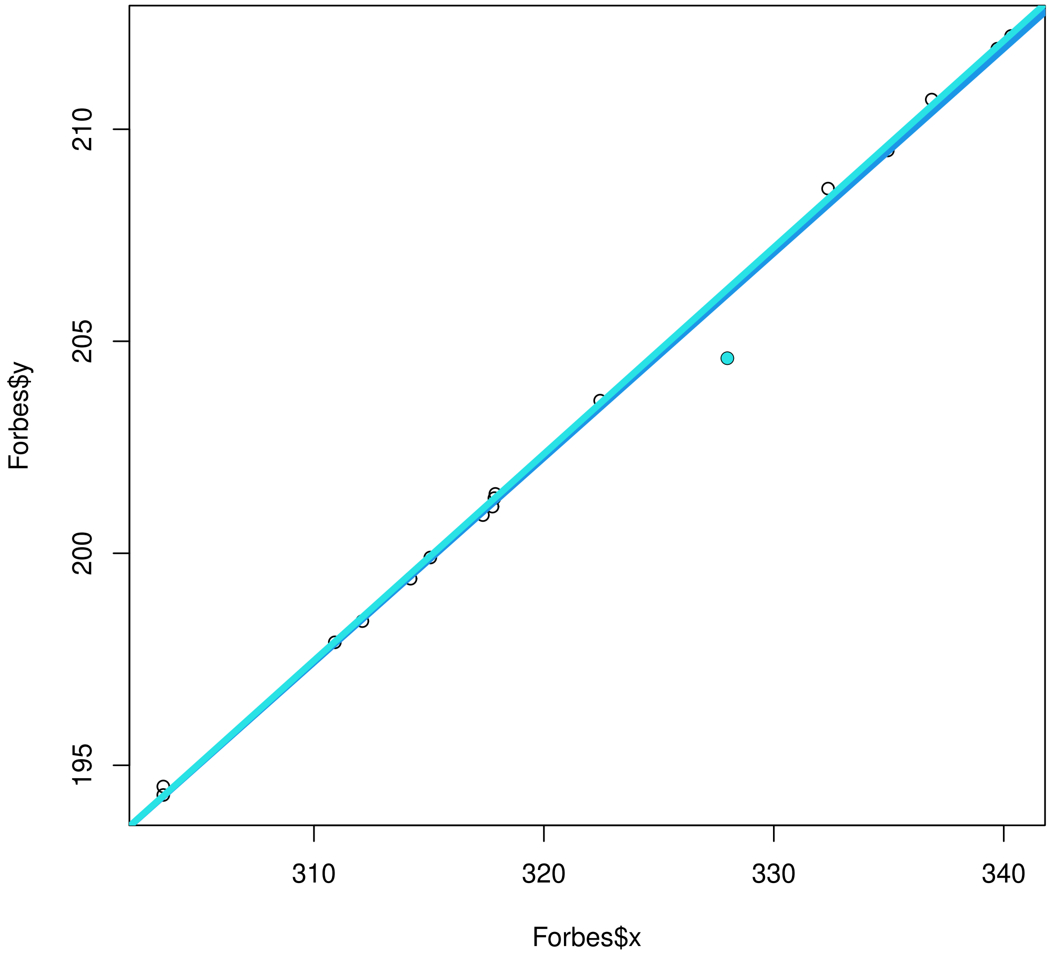
b)
```
Forbes_lm <- lm(y ~ x, data = Forbes)
summary(Forbes_lm)

##
## Call:
## lm(formula = y ~ x, data = Forbes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50249 -0.04380  0.03427  0.16540  0.38366
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.863838   2.851689   16.78 3.93e-11 ***
## x            0.482467   0.008866   54.42  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4223 on 15 degrees of freedom
## Multiple R-squared:  0.995,Adjusted R-squared:  0.9946
## F-statistic:  2962 on 1 and 15 DF,  p-value: < 2.2e-16

par(mfrow = c(1, 1))
plot(Forbes$x, Forbes$y)
abline(Forbes_lm, col = 4, lty = 6, lwd = 3)
```



The above regression line fits data rather well, although an outlier is clearly visible - we can identify this point by means of the **R**-function **identify()** : it is the 12th observation.

c)
```
plot(Forbes$x, Forbes$y)
points(Forbes$x[12], Forbes$y[12], col = 5, pch = 16)
abline(Forbes_lm, col = 4, lwd = 4)
ForbesR_lm <- lm(y ~ x, data = Forbes[-12, ])
# or
ForbesR_lm <- lm(y ~ x, data = Forbes, subset = -12)
abline(ForbesR_lm, col = 5, lty = 1, lwd = 4)
```



The **residual standard error** and the **standard errors** are reduced by a factor of 3.

In the following exercises we keep the 12th observation omitted.

d) Because the p-value of β_1 is smaller than 0.05 (=significance level), the null-hypothesis $\beta_1 = 0$ has to be rejected; i.e. β_1 is significantly different from 0 at the 5 % level.

e)
```
confint(ForbesR_lm, parm = 2, level = 0.95)

##       2.5 %     97.5 %
## x 0.4813568 0.4929508
```

A 95 %-confidence interval for the slope β_1 is given by $[0.481, 0.493]$.

f)
```
x0 <- data.frame(x = 325.81)
predict(ForbesR_lm, newdata = x0)

##        1
## 205.1726

predict(ForbesR_lm, newdata = x0, interval = "confidence",
    level = 0.95)

##        fit      lwr      upr
## 1 205.1726 205.0989 205.2463

predict(ForbesR_lm, newdata = x0, interval = "confidence",
    level = 0.99)

##        fit      lwr      upr
## 1 205.1726 205.0703 205.2749
```

The expected value is 205.17. The confidence intervals are [205.099, 205.246] and [205.070, 205.275]. As expected, the 99 % confidence interval is larger than the 95 % interval.

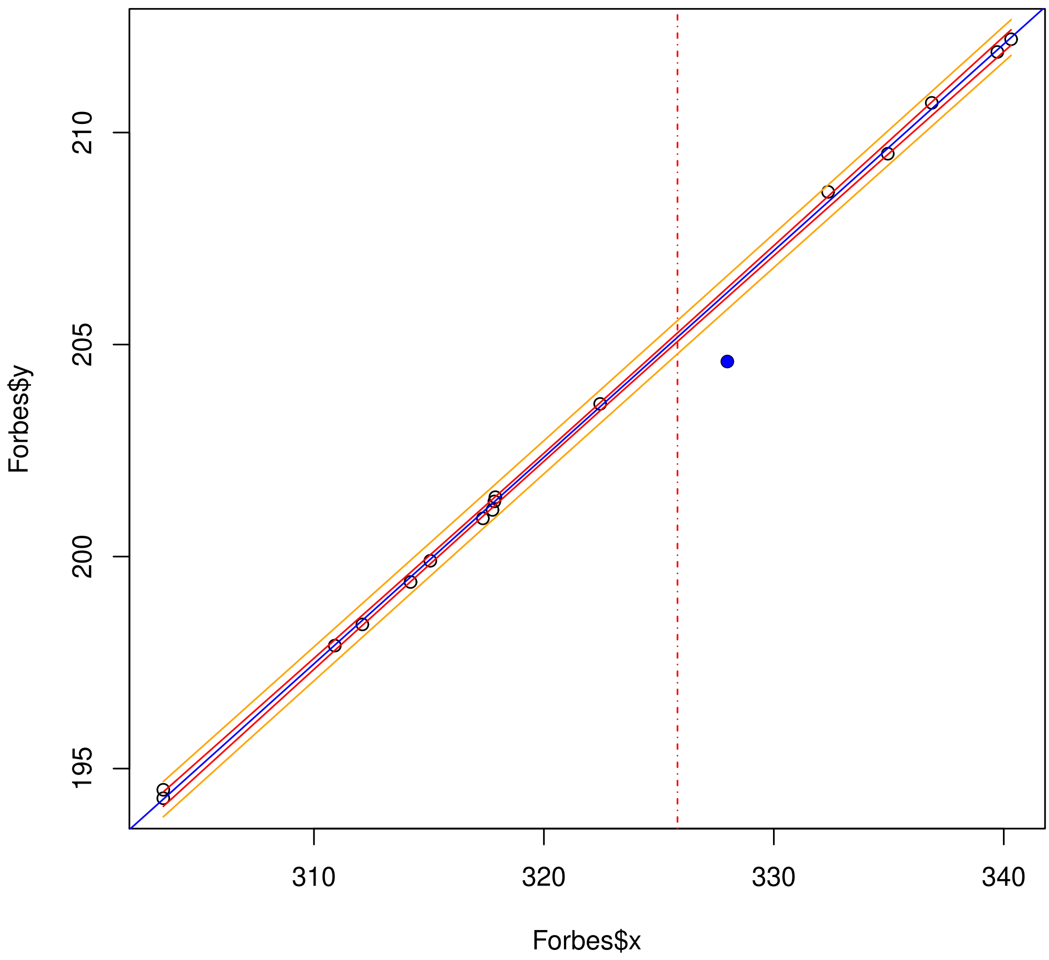
g)
```
predict(ForbesR_lm, newdata = x0, interval = "prediction",
    level = 0.99)

##        fit      lwr      upr
## 1 205.1726 204.7797 205.5656
```

A 99 %-prediction interval is [204.780, 205.566]. As expected, this interval is larger than the corresponding 99 %-confidence interval.

Voluntary Exercise:

```
plot(Forbes$x, Forbes$y)
points(Forbes$x[12], Forbes$y[12], col = "blue", pch = 16)
abline(ForbesR_lm, col = "blue", lty = 1)
abline(v = 325.81, lty = 4, col = "red")
x0 <- data.frame(x = seq(min(Forbes$x), max(Forbes$x), length = 50))
ForbesR.cia <- predict(ForbesR_lm, newdata = x0, interval = "confiden
    level = 0.99)
ForbesR.pia <- predict(ForbesR_lm, newdata = x0, interval = "predicti
    level = 0.99)
lines(x0$x, ForbesR.cia[, "upr"], col = "red")
lines(x0$x, ForbesR.cia[, "lwr"], col = "red")
lines(x0$x, ForbesR.pia[, "upr"], col = "orange")
```

```r
lines(x0$x, ForbesR.pia[, "lwr"], col = "orange")
```



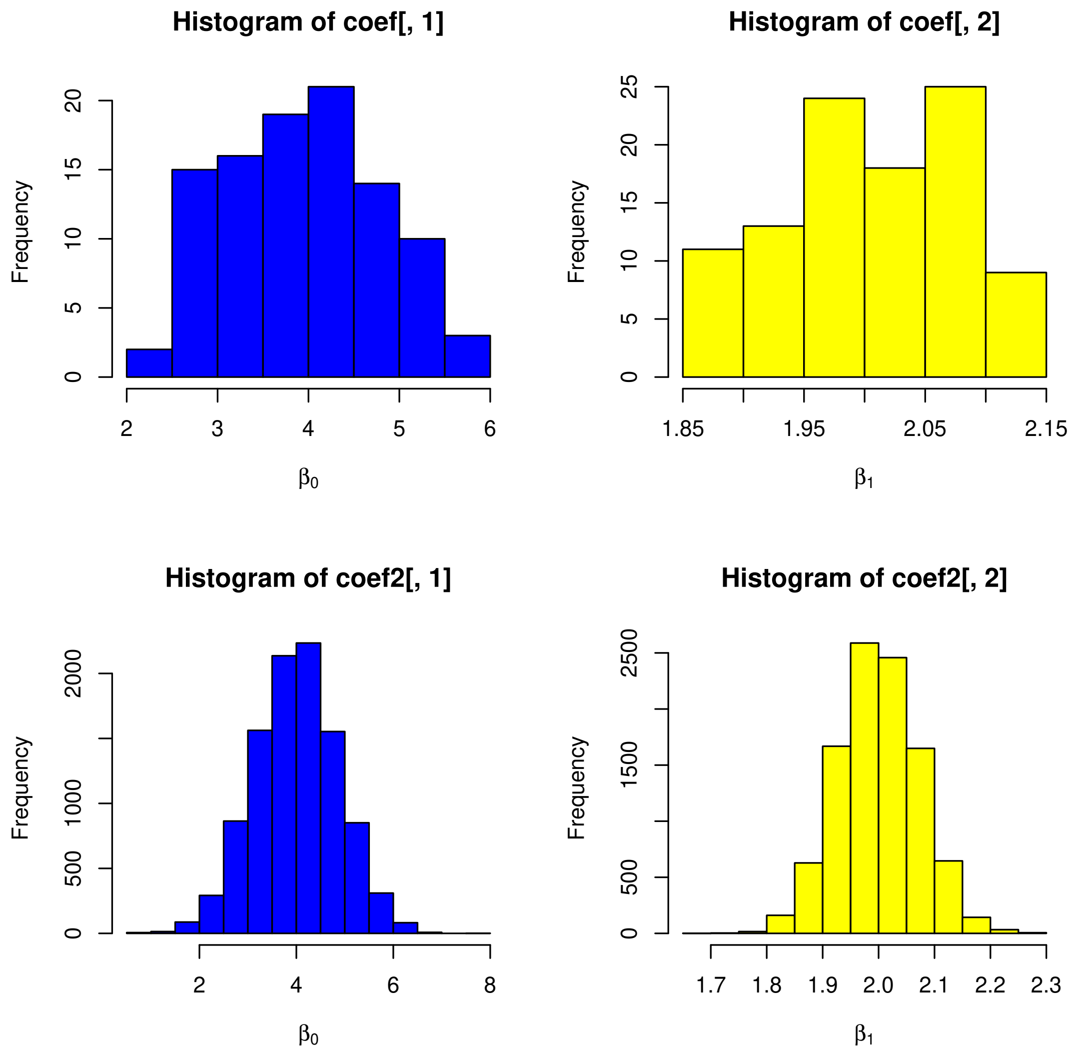
Solution 1.5

a)
```r
x_sim <- c(0, 3, 4, 8, 10, 11, 13, 16, 17, 20)
set.seed(4711)  ## always the same random numbers
error_sim <- matrix(rnorm(10 * 100, mean = 0, sd = sqrt(2)),
     ncol = 100)
y_sim <- 4 + 2 * x_sim + error_sim
coef <- matrix(0, ncol = 2, nrow = 100)
for (i in 1:100) coef[i, ] <- coef(lm(y_sim[, i] ~ x_sim))
error_sim2 <- matrix(rnorm(10 * 10000, mean = 0, sd = sqrt(2)),
     ncol = 10000)
y_sim2 <- 4 + 2 * x_sim + error_sim2
```

```
coef2 <- matrix(0, ncol = 2, nrow = 10000)
## Attention, this may take a while!
for (i in 1:10000) coef2[i, ] <- coef(lm(y_sim2[, i] ~ x_sim))
```

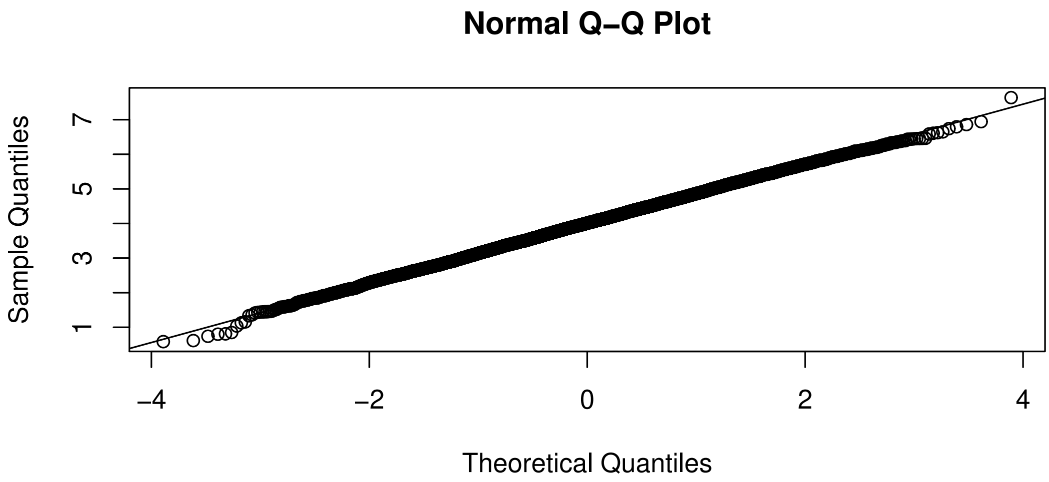
b)

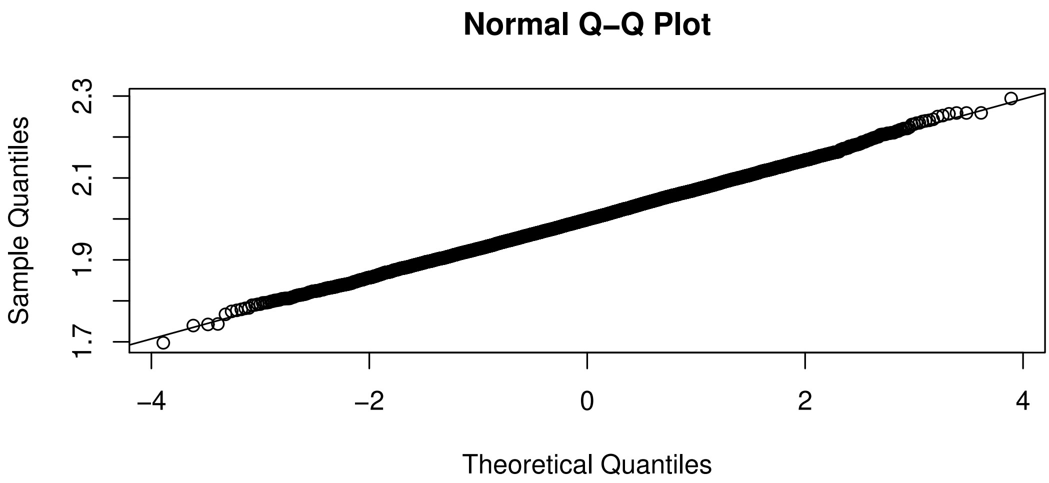
```
par(mfrow = c(2, 2))
hist(coef[, 1], xlab = expression(beta[0]), col = "blue")
hist(coef[, 2], xlab = expression(beta[1]), col = "yellow")
##
hist(coef2[, 1], xlab = expression(beta[0]), col = "blue")
hist(coef2[, 2], xlab = expression(beta[1]), col = "yellow")
```




```
par(mfrow = c(2, 1))
qqnorm(coef2[, 1])
qqline(coef2[, 1])
```

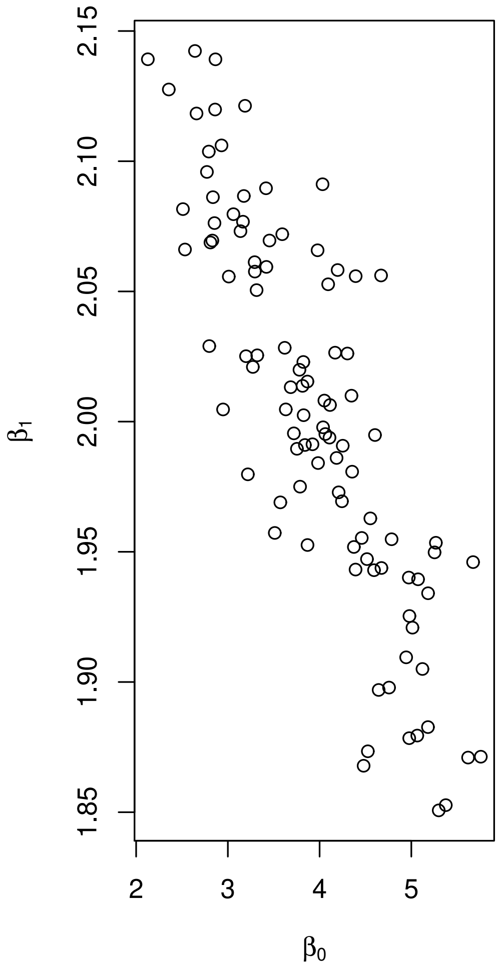
```
##
qqnorm(coef2[, 2])
qqline(coef2[, 2])
```

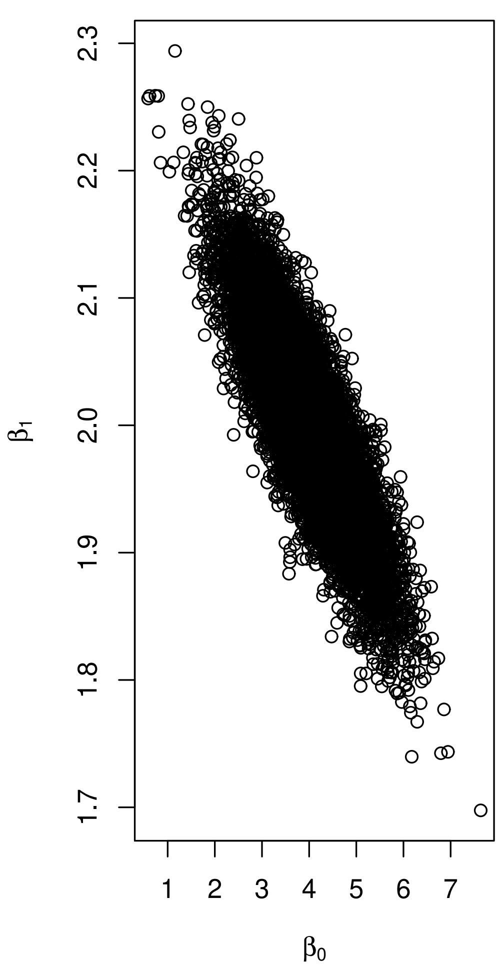




The following scatter plot shows the correlation between the estimated coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$. On the left with 100 simulated straight lines, on the right with 10 000.

```
par(mfrow = c(1, 2))
plot(coef[, 1], coef[, 2], xlab = expression(beta[0]), ylab = express
plot(coef2[, 1], coef2[, 2], xlab = expression(beta[0]),
    ylab = expression(beta[1]))
```





c) The following results depend on the concrete simulation, unless you fix the randomized values with **set.seed()** :

```
## beta_0
cat("arithmetic mean of beta_0 (100 simulations): ", mean(coef[,
    1]))

## arithmetic mean of beta_0 (100 simulations):  3.935716

cat("standard deviation of beta_0 (100 simulations): ",
    sd(coef[, 1]))

## standard deviation of beta_0 (100 simulations):  0.8481557

cat("variation of beta_0 (100 simulations): ", var(coef[,
```

```
    1]))

## variation of beta_0 (100 simulations):  0.7193682

cat("arithmetic mean of beta_0 (10000 simulations): ", mean(coef2[,
    1]))

## arithmetic mean of beta_0 (10000 simulations):  4.001601

cat("standard deviation of beta_0 (10000 simulations): ",
    sd(coef2[, 1]))

## standard deviation of beta_0 (10000 simulations):  0.8627985

cat("variation of beta_0 (10000 simulations): ", var(coef2[,
    1]))

## variation of beta_0 (10000 simulations):  0.7444213

## beta_1
cat("arithmetic mean of beta_1 (100 simulations): ", mean(coef[,
    2]))

## arithmetic mean of beta_1 (100 simulations):  2.00391

cat("standard deviation of beta_1 (100 simulations): ",
    sd(coef[, 2]))

## standard deviation of beta_1 (100 simulations):  0.07282633

cat("variation of beta_1 (100 simulations): ", var(coef[,
    2]))

## variation of beta_1 (100 simulations):  0.005303674

cat("arithmetic mean of beta_1 (10000 simulations): ", mean(coef2[,
    2]))

## arithmetic mean of beta_1 (10000 simulations):  1.999616

cat("standard deviation of beta_1 (10000 simulations): ",
    sd(coef2[, 2]))

## standard deviation of beta_1 (10000 simulations):  0.07211866

cat("variation of beta_1 (10000 simulations): ", var(coef2[,
    2]))

## variation of beta_1 (10000 simulations):  0.005201102
```

According to theory the estimates should scatter around $\beta_0 = 4$ and $\beta_1 = 2$ (due to the standard deviation of the error term $\sigma = \sqrt{2}$). For the (estimated) variance of $\hat{\beta}_0$ and $\hat{\beta}_1$ the following formulae need to be calculated:

```
## se(beta_0) :
SSx <- sum((x_sim - mean(x_sim))^2)

sqrt(2 * (1/10 + mean(x_sim)^2/SSx))

## [1] 0.8616496

## se(beta_1) :

sqrt(2/SSx)

## [1] 0.0722064
```

The theoretical standard errors $\text{se}(\hat{\beta}_0) = 0.862$ and $\text{se}(\hat{\beta}_1) = 0.0722$ that we computed above correspond to the average deviation of the parameter estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ around the true parameter values β_0 and β_1.

The more simulations we run, the closer the emprirical standard deviations of the set of estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ get to the *true* standard deviations $\text{se}(\hat{\beta}_0)$ and $\text{se}(\hat{\beta}_1)$. On the basis of 10 000 simulations we find $\text{se}(\hat{\beta}_0) \approx 0.8616496$ and $\text{se}(\hat{\beta}_1) \approx 0.0721$. We conclude that the theoretical standard errors $\text{se}(\hat{\beta}_0)$ and $\text{se}(\hat{\beta}_1)$ and the (approximately) true values of $\text{se}(\hat{\beta}_0)$ and $\text{se}(\hat{\beta}_1)$ agree to a high degree.