# Solution to Series 3

**1. a)** The required R commands are as follows:

```
> ## Load data
> load("conconi.rda")
> ## Scatter plot
> plot(puls ~ speed, data=conconi, pch=20, xlim=c(8.5,18.5), ylim=c(140,205))
> title("Conconi-Test: Puls vs. Speed")
> ## Regression
> fit <- lm(puls ~ speed, data=conconi)
> abline(fit, col="red")
> summary(fit)
Call:
lm(formula = puls ~ speed, data = conconi)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4947 -1.0123  0.5228  1.1825  3.6737

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  86.6105     2.5372   34.14   <2e-16 ***
speed         6.4070     0.1842   34.78   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.199 on 17 degrees of freedom
Multiple R-squared:  0.9861,        Adjusted R-squared:  0.9853
F-statistic:  1210 on 1 and 17 DF,  p-value: < 2.2e-16
```

- The multiple R-squared is 98.61%. This is the amount of the scatter in the pulse which can be explained by the increase in speed.
- As can be seen in the summary output $\hat{\beta}_1 = 6.4070$. This is the amount the pulse increases on average when the speed is increased by 1 km/h. The 95% confidence interval for the slope is computed as follows:

```
> ci <- confint(fit)
> ci
                2.5 %    97.5 %
(Intercept) 81.257578 91.963475
speed        6.018418  6.795617
```

  The confidence interval for the slope is shown in the last line. These values are also plausible for the increase in pulse when the speed is increased by 1 km/h.
- The estimate for the regression line at $x = 0$ is the intercept. It can be obtained from the summary output. The 95% confidence interval for this value is [81.26,91.96] (cf. R output belonging to the previous subquestion). Alternatively, one can make a prediction at the value $x = 0$ and compute the corresponding 95% confidence interval. This procedure will be discussed later in the class.

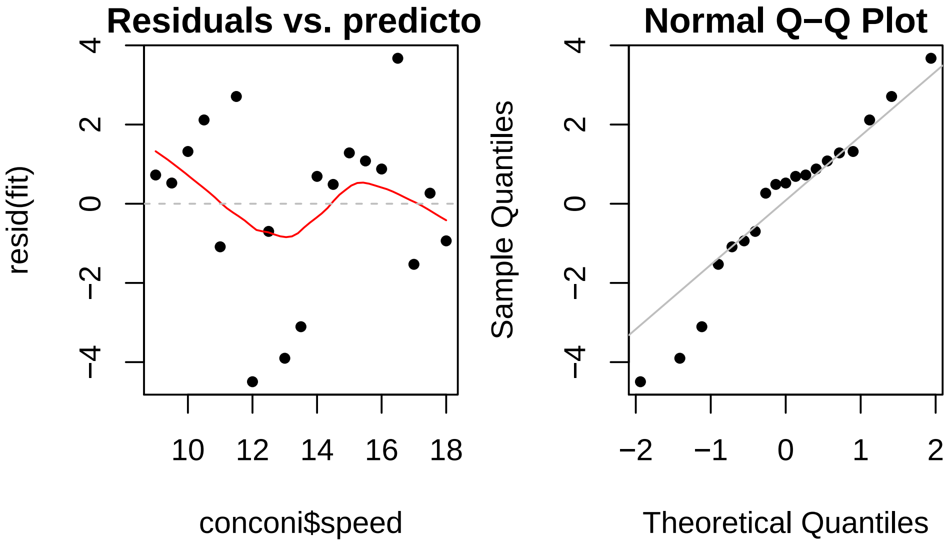
```
> new.x <- data.frame(speed = 0)
> predict(fit, new.x, interval="confidence")
       fit      lwr      upr
1 86.61053 81.25758 91.96347
```

The numerical values of these two approaches are identical. Please note that this prediction constitutes a strong extrapolation. Therefore, it is not reliable in general. In this case, however, both the predicted value as well as the interval seem to be realistic. Thus, it could be true that a linear relation also holds for slower speeds up until no movement. To verify this claim we would need to make additional measurements.

**b)** For Dani, the estimate is $\hat{\beta}_1 = 4.093$, i.e. it is smaller. We now need to establish whether the difference between the two increases is significant. To do this, we cannot take one of the two slopes as the ground truth and do a simple hypothesis test. Instead, we need to do a two-sample regression coefficient test. We have not seen such a test in class. An alternative is to test whether the two 95% confidence intervals (CI) for the slopes overlap. If this is not the case, then there is a significant difference. In Dani's case, the CI is approximately $[3.9; 4.3]$ (estimate +/- 2x standard error). So the two 95% confidence intervals do not overlap and there is a significant difference between the two increases of the pulse. However, we cannot deduce who is the better runner. The slower increase in pulse indicates that Dani might be the better runner but we do not know the intercept. Additionally, the maximally attainable pulse is different for each individual. Therefore, we cannot draw a conclusion without knowing these quantities.

**c)** The R Code is:

```
> ## residual plots
> par(mfrow=c(1,2))
> plot(conconi$speed, resid(fit), pch=20)
> lines(loess.smooth(conconi$speed, resid(fit)), col="red")
> title("Residuals vs. predictor")
> abline(h=0, col="grey", lty=2)
> qqnorm(resid(fit), pch=20)
> qqline(resid(fit), col="grey")
```



Alternatively, these plot can be obtained as follows:

```
> ## alternatively
> plot(fit, which=1, pch=20, id.n=5)
> plot(fit, which=2, pch=20)
```

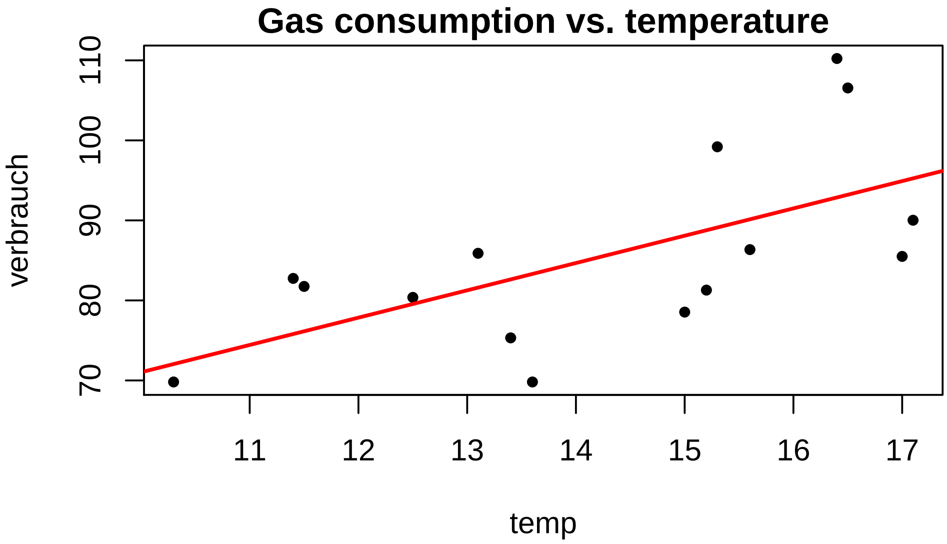
The smoother in the plot "residuals vs. predictor" deviates clearly from the horizontal line. We need to decide whether this deviation is random or systematic which is not easy to determine. The deviation is caused by three data points at a medium speed of 12-14km/h which are associated with strongly negative residuals. To come to a conclusion, some background knowledge is helpful: In theory, there is a linear relation between speed and pulse as long as the physical prerequisites (warm-up, fatigue) and the external conditions (weather, wind) are constant. Secondly, the lecturer received feedback regarding his speed only every 200m and his speed was not always at the value it should have taken. This explains the deviations from the linear relation and we can conclude that the deviation is not systematic. The fact that three data points are affected that are very close to each other seems to be "bad luck".

The variance of the errors seems to be constant, even though there are some residuals which are clearly larger than the rest. On the other hand, this does not seem to be a systematic deviation from the asumption of constant variance. Moreover, the observations seem to follow a Normal distribution except for the three data points discussed above.

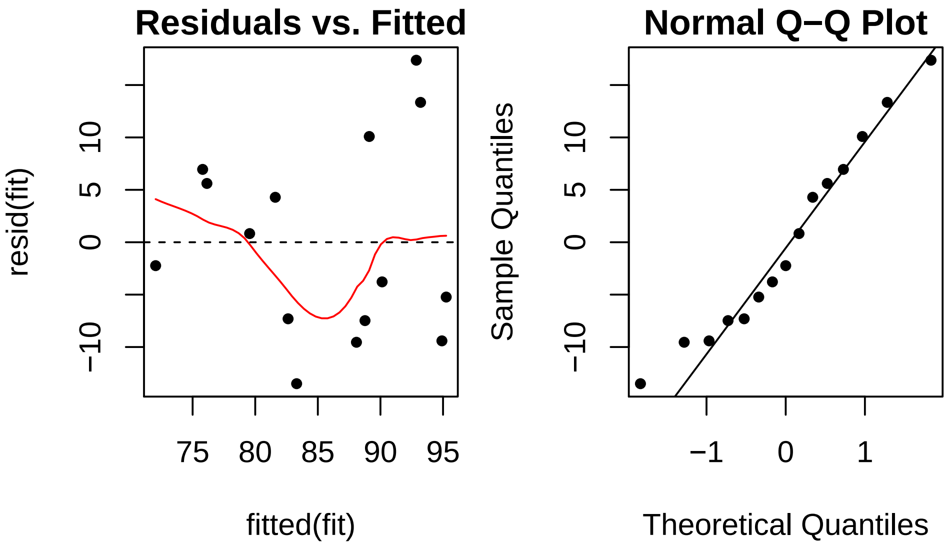
With respect to the correlation, we need to take into account that the measurements were made in a temporal sequence. Therefore, serial correlations could be present. In this case, their existence could be easily explained: In the test setup, it is difficult to achieve the required speed exactly. If one is too fast on a given 200m section and receives this feedback afterwards, one might run too slowly on the subsequent 200m section (which is associated with a recovery of the heart frequency), etc.

**2.** a)

```
> ## load data
> load("gas.rda")
> ## analysis of the gas consumption
> par(mfrow=c(1,1))
> plot(verbrauch ~ temp, data=gas, pch=20)
> title("Gas consumption vs. temperature")
> fit <- lm(verbrauch ~ temp, data=gas)
> abline(fit, col="red", lwd=2)
```

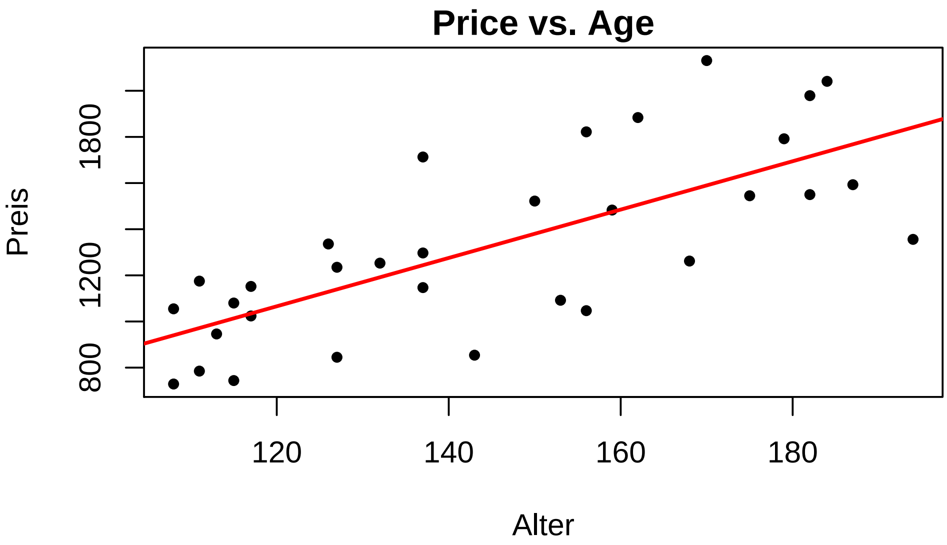


```
> par(mfrow=c(1,2))
> plot(fitted(fit), resid(fit), pch=20)
> lines(loess.smooth(fitted(fit), resid(fit)), col="red")
> title("Residuals vs. Fitted")
> abline(h=0, lty=2)
> qqnorm(resid(fit), pch=20); qqline(resid(fit))
```

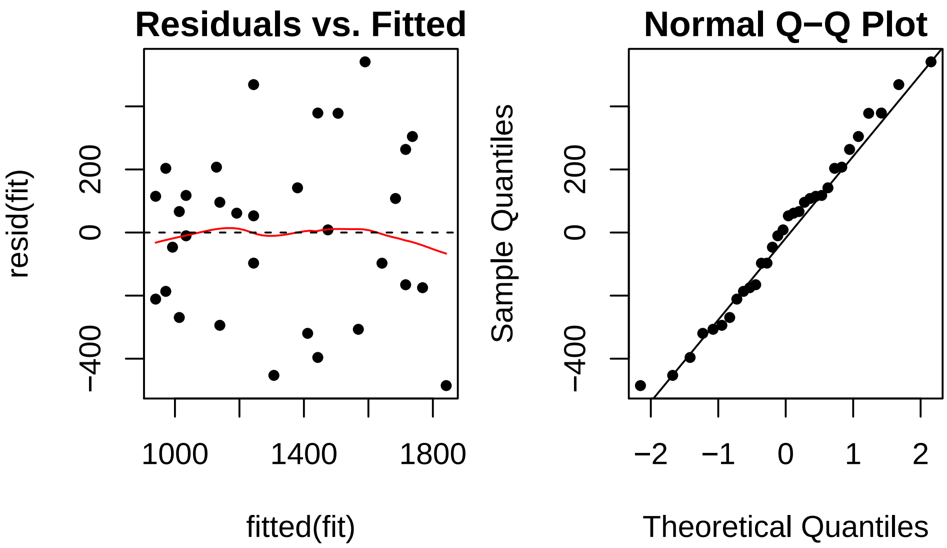


At first sight the scatter plot seems to suggest that the regression line fits well. However, the plot "Residuals vs. Fitted" gives another impression. The smoother shows a strong deviation towards the bottom which indicates the presence of a systematic error. Similarly, the variance seems to be larger for large fitted values. The Normal plot does not show any abnormalities. Whether the observations are correlated cannot be determined based on these two plots. We would have to know whether the observations were recorded in a temporal order and whether residuals of observations close in time show abnormalities. In summary, the two assumptions $\mathcal{E}(E_i) = 0$ and $\text{Var}(E_i) = \sigma_E^2$ might be violated. A log-transformation would yield a much better fit and should be applied.

**b)**
```
> ## load data
> load("antikeUhren.rda")
> ## analysis of the gas consumption
> par(mfrow=c(1,1))
> plot(Preis ~ Alter, data=antikeUhren, pch=20)
> title("Price vs. Age")
> fit <- lm(Preis ~ Alter, data=antikeUhren)
> abline(fit, col="red", lwd=2)
>
```



```
> par(mfrow=c(1,2))
> plot(fitted(fit), resid(fit), pch=20)
> lines(loess.smooth(fitted(fit), resid(fit)), col="red")
> title("Residuals vs. Fitted")
> abline(h=0, lty=2)
> qqnorm(resid(fit), pch=20); qqline(resid(fit))
```



This model shows a good fit. The smoother in the plot "Residuals vs. Fitted" is almost horizontal and does not show systematic deviations from the x-axis. The variance of the data points is approximately constant. It is only slightly smaller on the left which is not a significant problem. The Normal plot does not show any abnormalities. Whether the observations are correlated cannot be determined based on these two plots. These could occur if the clocks were sold at different auctions with systematically larger or smaller prices. This would cause a correlation of the corresponding residuals. In summary, however, we can speak of a well-fitting model.

**3.** **a)** False. Also a model with a large or very large $R^2$ can have a non-tolerable, systematic error.

**b)** False. In this case there is a relation between response and predictor, but this relation does not have to be causal.

**c)** True. The OLS estimator loses efficiency in the presence of long-tailed residuals but it remains unbiased. As long as the long tails are not too heavy, we can usually tolerate the effect.

**d)** False. A systematic deviation of the smoother in the "residuals vs. predictor" plot always means that the model yields wrong predictions which cannot be tolerated.

**e)** True. More precise insights are given by the 95% prediction interval, the computation of which is more complex than the above rule of thumb. However, if the number of observations is large enough, the deviation from this rule of thumb is small.

**4.** **a)** First we type in the data. The scatterplot of `runoff` versus `rainfall` suggests that a linear relationship holds.

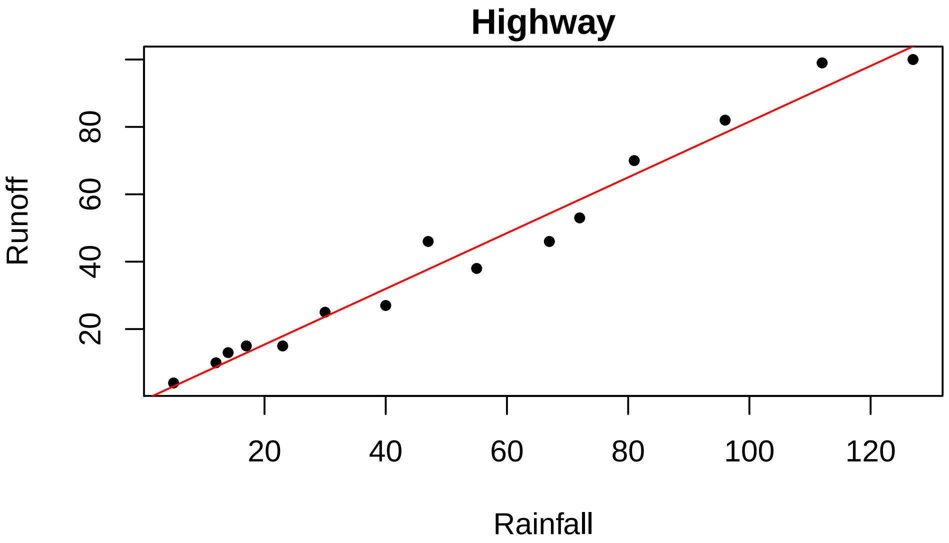
```
> rainfall <- c(5, 12, 14, 17, 23, 30, 40, 47, 55, 67, 72, 81, 96, 112, 127)
> runoff <- c(4, 10, 13, 15, 15, 25, 27, 46, 38, 46, 53, 70, 82,  99, 100)
> hwy.runoff <- data.frame(rainfall=rainfall, runoff=runoff)
> plot(hwy.runoff$runoff ~ hwy.runoff$rainfall, pch=20, xlab="Rainfall", ylab="Runoff",
      main="Highway")
> ## fit a simple linear regression
> fit <- lm(runoff ~ rainfall, data=hwy.runoff)
> abline(fit, col="red")
> ## Summary
> summary(fit)
```

```
Call:
lm(formula = runoff ~ rainfall, data = hwy.runoff)

Residuals:
   Min     1Q Median     3Q    Max 
-8.279 -4.424  1.205  3.145  8.261 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.12830    2.36778  -0.477    0.642    
rainfall     0.82697    0.03652  22.642  7.9e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.24 on 13 degrees of freedom
Multiple R-squared:  0.9753,	Adjusted R-squared:  0.9734 
F-statistic: 512.7 on 1 and 13 DF,  p-value: 7.896e-12
```



**b)** An $R^2$ of 0.98 is extremely high, i.e. a huge part of the variation in the data can be attributed to the linear association between runoff and rainfall volume.

**c)** There is a significant linear association between runoff and rainfall volume, since the null hypothesis $\beta_1 = 0$ is clearly rejected.

As estimates we obtain $\hat{\beta}_0 = -1.12$, i.e. when it does not rain, the runoff is negative. Clearly, this is not possible. Also note that we do not have observations at $x = 0$. Thus, interpreting the intercept is an extrapolation. On the other hand, this also shows that this model might not be the best. Lastly, we note that the estimate of the intercept is not significant.

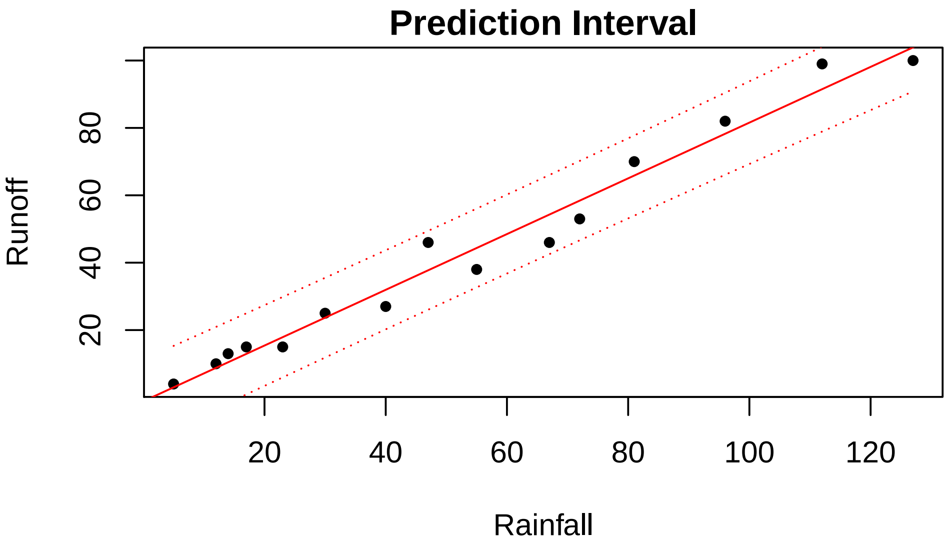
For the slope we obtain $\hat{\beta}_1 = 0.827$. This value is smaller than 1 (even significantly smaller as the 95% confidence interval for the slope indicates). Thus, it is statistically shown that not all the rain runs off via the canalization. It is plausible that part of the rain evaporates or trickles away.

**d)**
```
> pred <- predict(fit, newdata=data.frame(rainfall=50), interval="prediction")
```

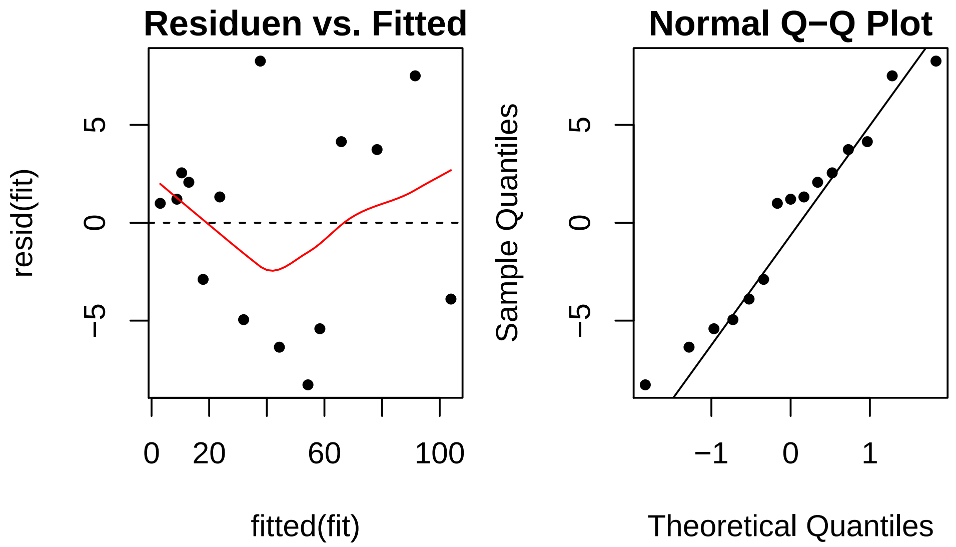
If the rainfall volume takes a value of 50 we find a runoff volume of 40.22 with a 95% prediction interval of [28.53,51.92].

We can also draw the 95% prediction interval to the data.

```
> plot(hwy.runoff$runoff ~ hwy.runoff$rainfall, pch=20, xlab="Rainfall", ylab="Runoff",
       main="Prediction Interval")
> abline(fit, col="red")
> interval <- predict(fit, interval="prediction")
> lines(hwy.runoff$rainfall, interval[,2], lty=3, col="red")
> lines(hwy.runoff$rainfall, interval[,3], lty=3, col="red")
```



**e)**
```
> par(mfrow=c(1,2))
> plot(fitted(fit), resid(fit), pch=20, main="Residuen vs. Fitted")
> abline(h=0, lty=2)
> lines(loess.smooth(fitted(fit), resid(fit), span=0.9), col="red")
> qqnorm(resid(fit), pch=20); qqline(resid(fit))
```

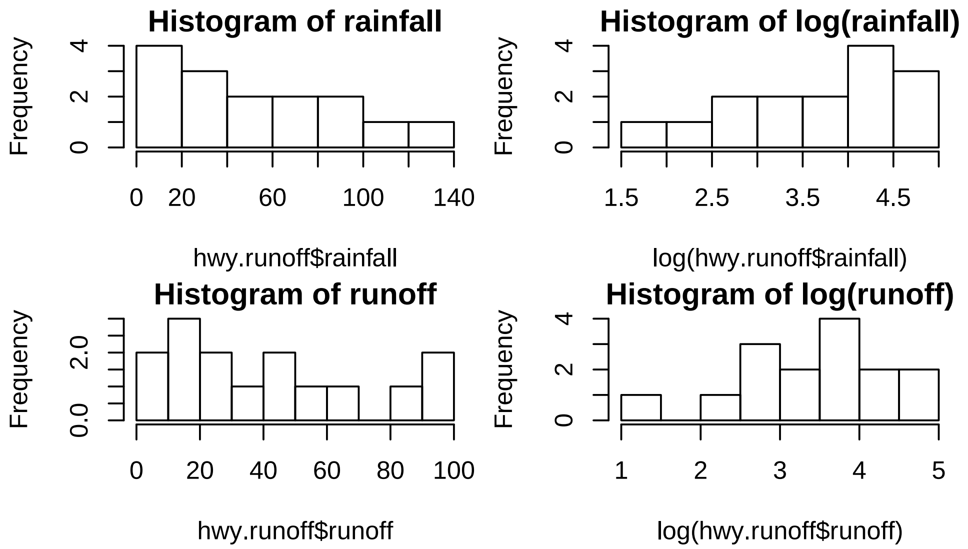


We check the model assumptions ((i) expectation of the errors is zero, (ii) constant error variance, (iii) approximate Normal distribution) with the Tukey-Anscombe plot (residuals vs. fitted values) (assumptions (i) and (ii)) and the normal plot (assumption (iii)). For the assumption of uncorrelated errors there is no suitable plot in this case. However, this is no reason to assume that the errors are uncorrelated.

While the Normal distribution of the errors seems to be satisfied, the Tukey-Anscombe plot (residuals vs. fitted values) raises some doubts. Even though there is a large multiple R-squared and also the test for the slope is highly significant, the expected value of the errors does not seem to be zero. The smoother deviates systematically from the horizontal line. Similarly, it seems like the variance of the residuals is larger at large rainfall values. Thus, the assumption of constant error variance might be violated. As we shall see, we can describe the relation between runoff and rainfall more accurately with a different simple linear regression model.

**f)** For each variable we create a histogram:

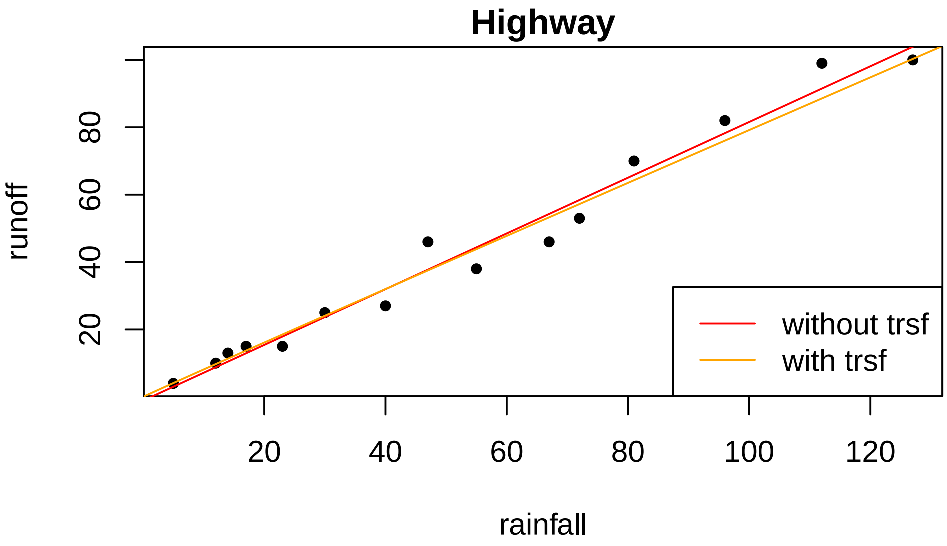
```
> par(mfrow=c(2,2))
> hist(hwy.runoff$rainfall, 8, main="Histogram of rainfall")
> hist(log(hwy.runoff$rainfall), 8, main="Histogram of log(rainfall)")
> hist(hwy.runoff$runoff, 8, main="Histogram of runoff")
> hist(log(hwy.runoff$runoff), 8, main="Histogram of log(runoff)")
```



Runoff and rainfall volume are both variables which can only take positive values. Both are slightly skewed to the right. In addition, the variance of the residuals is larger at large rainfall values. Thus, a log-transformation is suitable for both variables.

g) After fitting the new regression model, we need to exponentiate the fitted values in order to add them to the original plot.

```
> par(mfrow=c(1,1))
> fit.loglog <- lm(log(runoff) ~ log(rainfall), data=hwy.runoff)
> plot(runoff ~ rainfall, data=hwy.runoff, pch=20, main="Highway")
> abline(fit, col="red")
> xx <- data.frame(rainfall=0:150)
> yy <- predict(fit.loglog, newdata=xx)
> lines(xx$rainfall, exp(yy), col="orange")
> legend("bottomright", lty=1, col=c("red", "orange"), legend=c("without trsf", "with trsf"))
```



h)
```
> summary(fit.loglog)

Call:
lm(formula = log(runoff) ~ log(rainfall), data = hwy.runoff)

Residuals:
     Min       1Q   Median       3Q      Max
-0.20980 -0.10952  0.02828  0.08727  0.20388

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.18369    0.13803  -1.331    0.206
log(rainfall)  0.98917    0.03676  26.908 8.75e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1293 on 13 degrees of freedom
Multiple R-squared:  0.9824,	Adjusted R-squared:  0.981
F-statistic:   724 on 1 and 13 DF,  p-value: 8.748e-13
```

The relation was already strong without the transformation with a p-value of $10^{-12}$ corresponding to the null hypothesis $\beta_1 = 0$ and an $R^2$ of $0.975$. After the transformation these values are $10^{-13}$ and $0.982$.

Additionally, this model is preferable from a practical point of view as it cannot yield negative runoff values. Similarly, the coefficient $\beta_1$ is easier to interpret. Without transformation, $\hat{\beta}_1 = 0.827$ means that for an additional unit of rain, there are 0.827 additional units of runoff. With the transformation, the interpretation is that for 1% of additional amount of rain, there is 0.989% of additional runoff. In other words, 98.9% of the rain runs off via the canalization, the rest evaporates or trickles away.

i)
```
> ## prediction
> new.x  <- data.frame(rainfall=50)
> pred.y <- predict(fit.loglog, new.x, interval="prediction")
> exp(pred.y)
```

```
       fit      lwr      upr
1 39.88372 29.86744 53.25903
```

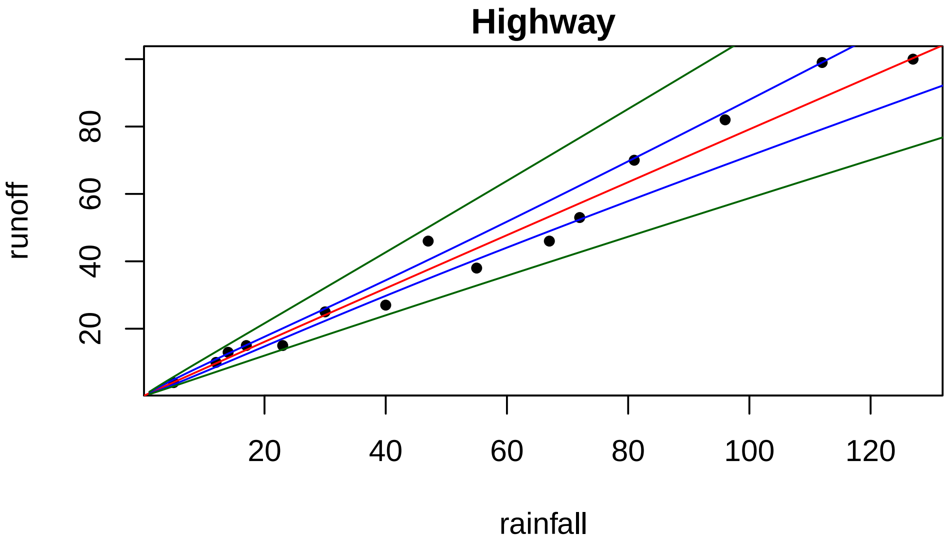
Note that the point prediction is not the expected value of the target but its median. If we want to compute the former, we need to adjust as follows (cf. script p. 33):

```
> exp(pred.y[1] + (summary(fit.loglog)$sigma^2)/2)
```

```
[1] 40.21832
```

The predicted value is very close to the one from the model without log-transformations. This is not true in general. In some cases, this difference can be large. Additionally, please note that the 95% prediction interval is no longer symmetric:

```
> ## prediction and confidence interval
> plot(runoff ~ rainfall, data=hwy.runoff, pch=20, main="Highway")
> xx <- data.frame(rainfall=0:150)
> yy <- predict(fit.loglog, newdata=xx, interval="confidence")
> lines(xx$rainfall, exp(yy[,1]), col="red")
> lines(xx$rainfall, exp(yy[,2]), col="blue")
> lines(xx$rainfall, exp(yy[,3]), col="blue")
> yy <- predict(fit.loglog, newdata=xx, interval="prediction")
> lines(xx$rainfall, exp(yy[,2]), col="darkgreen")
> lines(xx$rainfall, exp(yy[,3]), col="darkgreen")
```

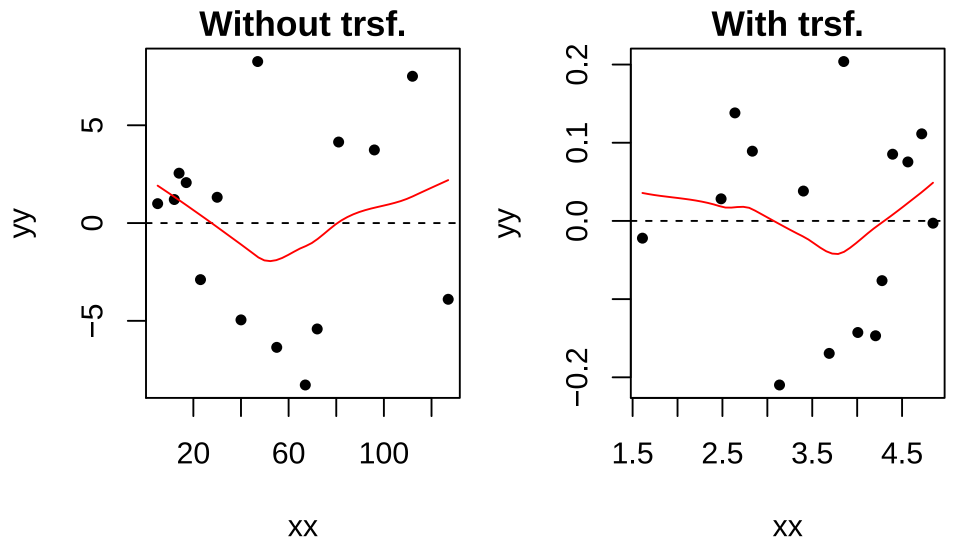


The asymmetry is not very strong in this example. In other cases, it can be much stronger.

j) The residual plot with transformation looks better than the one without transformation. The improvement is not very large, but the model with transformation is preferable nonetheless. Another reason for this is the fact that the model with transformation cannot yield negative values – neither as fitted values nor in the prediction interval. The transformed model predicts a runoff of 0 for a rainfall of 0 – which is another desirable property.

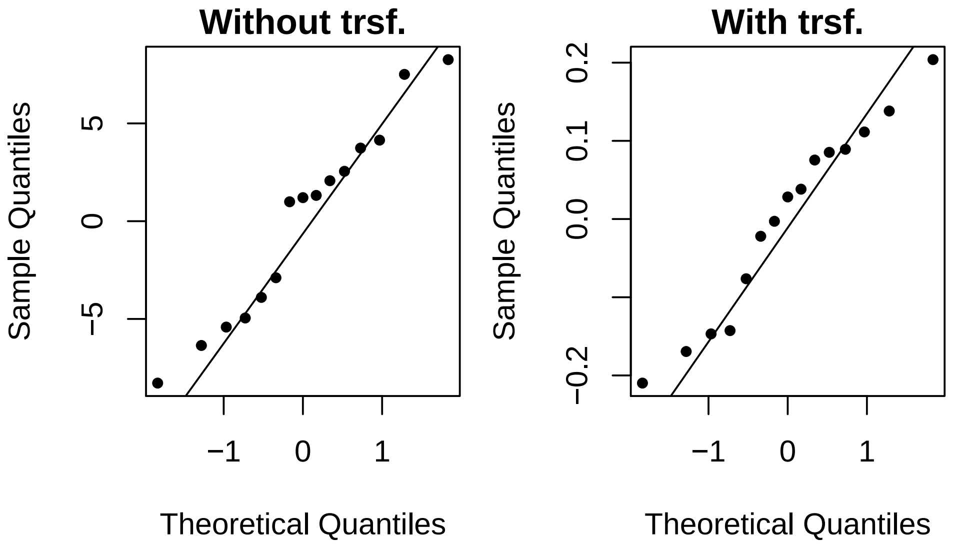
In summary, there are only small differences between the two models but the transformed model is more appropriate.

```
> ## residuals plots
> par(mfrow=c(1,2))
> xx <- hwy.runoff$rainfall; yy <- resid(fit)
> plot(xx, yy, pch=20, main="Without trsf.")
> abline(h=0, lty=2)
> lines(loess.smooth(xx, yy, span=0.9, family="gaussian"), col="red")
> xx <- log(hwy.runoff$rainfall); yy <- resid(fit.loglog)
> plot(xx, yy, pch=20, main="With trsf.")
> abline(h=0, lty=2)
> lines(loess.smooth(xx, yy, span=0.9, family="gaussian"), col="red")
```



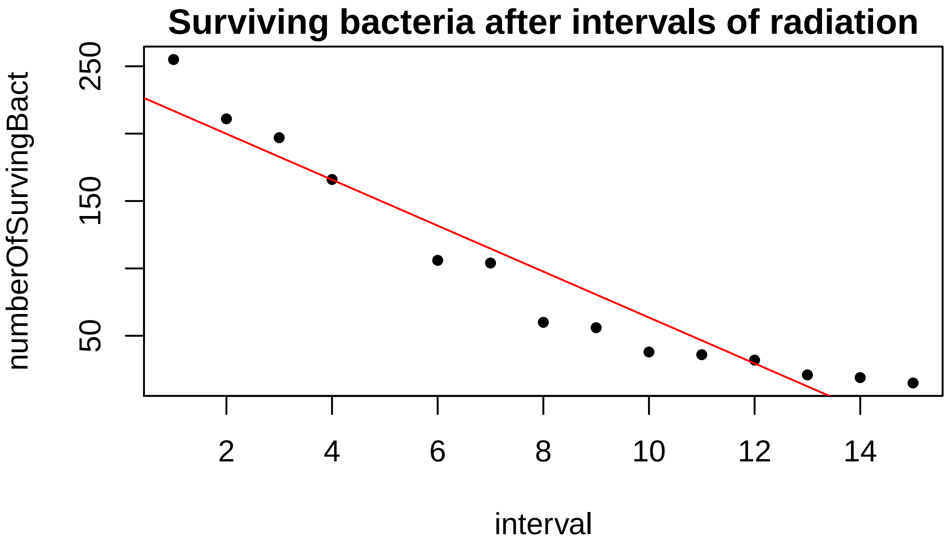
Both of the Normal plots do not show any abnormalities:

```
> ## Normal plots
> par(mfrow=c(1,2))
> qqnorm(resid(fit), pch=20, main="Without trsf."); qqline(resid(fit))
> qqnorm(resid(fit.loglog), pch=20, main="With trsf."); qqline(resid(fit.loglog))
```



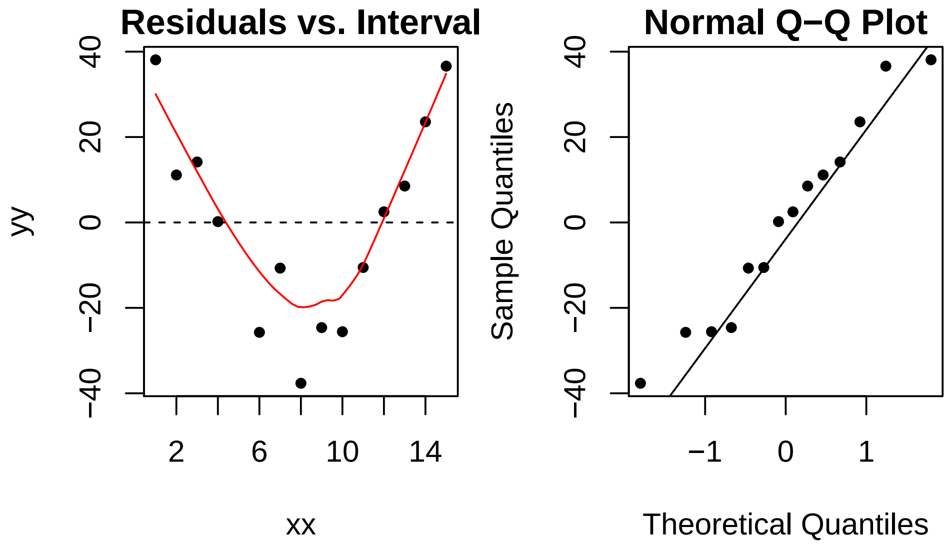
**5.** **a)** The scatter plot shows that a OLS regression line does not describe the relation between the two quantities appropriately. We need to apply a transformation.

```
> ## load data
> interval <- 1:15
> numberOfSurvingBact <- c(255, 211, 197, 166, NA, 106, 104, 60, 56, 38, 36, 32, 21, 19, 15)
> ## scatter plot
> par(mfrow=c(1,1))
> plot(interval, numberOfSurvingBact, pch=20)
> title("Surviving bacteria after intervals of radiation")
> ## regression
> fit.orig <- lm(numberOfSurvingBact ˜ interval)
> ## add regressions line to plot
> abline(fit.orig, col="red")
```



**b)** As the “Residuals vs. Predictor” plot shows, the expected value of the errors is not equal to zero. Instead there is a systematic deviation. The picture is typical for situations where a transformation should be applied. The Normal plot does not show any problems. Please note, however, that it does not make much sense to study the distribution of the residuals carefully since they belong to an incorrect model. The residuals corresponding to the correct model will be different and may have a different distribution.

```
> ## diagnostics plots
> par(mfrow=c(1,2))
> xx <- interval[-5]; yy <- resid(fit.orig)
> plot(xx, yy, pch=20, main="Residuals vs. Interval")
> abline(h=0, lty=2)
> lines(loess.smooth(xx, yy), col="red")
> qqnorm(resid(fit.orig), pch=20); qqline(resid(fit.orig))
```



**c)** A log-transformation should be applied for the target value “number of surviving bacteria”. This variable can only take positive values and the histogram shows a positive skew. For the predictor we do not need to apply a transformation. The scale of the response is arbitrary, negative values could also be used. In addition, the log-response model fulfills the hint given in the problem formulation: per radiation interval the proportion of bacteria that is killed remains constant. From the estimate of the coefficient $\hat{\beta}_1$ we can read off this value:

```
> fit.log <- lm(log(numberOfSurvingBact) ~ interval)
> exp(coef(fit.log)[2])
```

```
 interval
0.8116327
```

After each interval, 81.16% of the bacteria remain alive. In other words, on average 18.84% of the bacteria are killed per interval.

**d)** The prediction on the original scale for the interval 5 can be obtained as follows:

```
> ## predictions and intervals
> new.x <- data.frame(interval=c(5))
> predi <- predict(fit.log, newdata=new.x, interval="prediction")
> ## prediction for the median of the conditional distribution when interval=5
> exp(predi)
```

```
       fit      lwr      upr
1 124.0672 96.29711 159.8456
```

This value is the median of the conditional distribution. The expected value is slightly larger, namely

```
> ##  prediction for the expected value of the conditional distribution when interval=5
> exp(predi + (summary(fit.log)$sigma^2)/2)[1]
```

```
[1] 124.8256
```

The estimate for the relative decrease in the number of surviving bacteria was already discussed above:

```
> ## proportional change after one interval
> exp(-0.208707)
```

```
[1] 0.811633
```

The 95% confidence interval is given by:

```
> ## confidence interval
> exp(confint(fit.log, "interval"))
```

```
             2.5 %    97.5 %
interval 0.7998456 0.8235935
```

To estimate the expected number of bacteria at the beginning, we predict for the interval 0, together with the confidence interval:

```
> ## starting value (i.e. interval=0), including confidence interval
> new.x <- data.frame(interval=c(0))
> predi <- predict(fit.log, newdata=new.x, interval="confidence")
> # prediction for the median of the conditional distribution when interval=0
> exp(predi)
```

```
       fit      lwr      upr
1 352.2568 307.3775 403.6888
```

```
> # prediction for the expected value of the conditional distribution when interval=0
> exp(predi + (summary(fit.log)$sigma^2)/2)[1]
```

```
[1] 354.41
```