Solution to Test 1

1. a)

```
> RP.lm <- lm(AL ~ Hard + tTS, data = RP)
> summary(RP.lm)
Call:
lm(formula = AL ~ Hard + tTS, data = RP)

Residuals:
   Min     1Q Median     3Q    Max 
-71.00 -26.45  -0.09  17.73  73.56 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  979.221     77.658   12.61  7.9e-13 ***
Hard          -7.258      0.636  -11.41  7.8e-12 ***
tTS           -1.727      0.285   -6.06  1.8e-06 ***
---
Signif. codes: 
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 39 on 27 degrees of freedom
Multiple R-squared:  0.833,	Adjusted R-squared:  0.821 
F-statistic: 67.4 on 2 and 27 DF,  p-value: 3.16e-11
> library(robustbase)
> RP.rlm2 <- lmrob(AL ~ Hard + tTS, data = RP)
> summary(RP.rlm2)
Call:
lmrob(formula = AL ~ Hard + tTS, data = RP)
 \--> method = "MM"
Residuals:
    Min      1Q  Median      3Q     Max 
-144.32  -19.38   -0.55    8.04   40.57 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1035.053     36.157    28.6  < 2e-16 ***
Hard          -6.109      0.397   -15.4  6.9e-15 ***
tTS           -2.430      0.156   -15.6  4.9e-15 ***
---
Signif. codes: 
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Robust residual standard error: 24.5 
Multiple R-squared:  0.932,	Adjusted R-squared:  0.927 
Convergence in 8 IRWLS iterations

Robustness weights: 
 3 observations c(19,24,30)
	 are outliers with |weight| = 0 ( < 0.0033); 
 3 weights are ~= 1. The remaining 24 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.384   0.908   0.971   0.912   0.995   0.998 
Algorithmic parameters: 
       tuning.chi                bb        tuning.psi
```

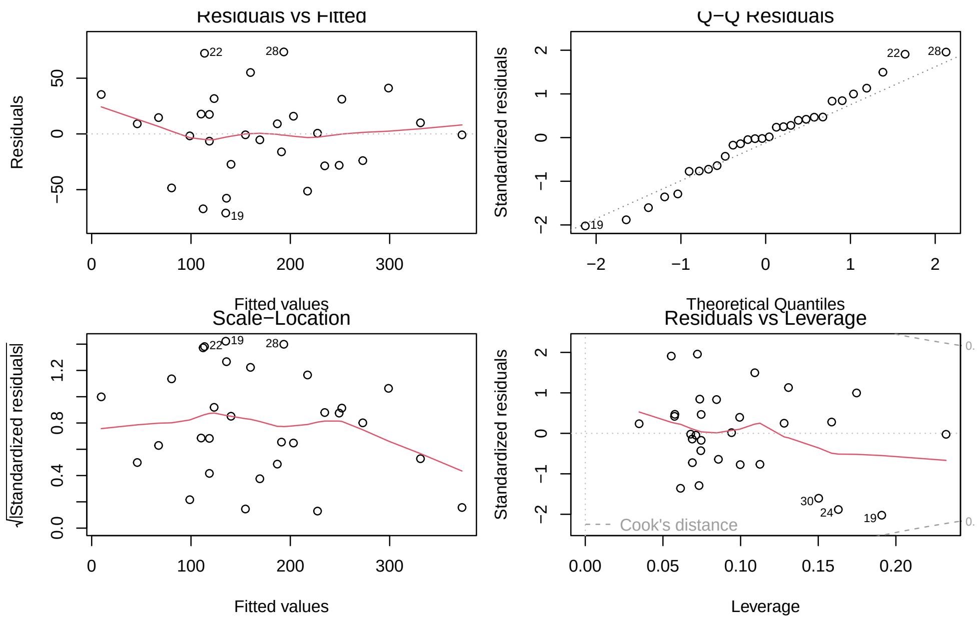
```
          1.55e+00           5.00e-01           4.69e+00
        refine.tol            rel.tol          scale.tol
          1.00e-07           1.00e-07           1.00e-10
         solve.tol           zero.tol        eps.outlier
          1.00e-07           1.00e-10           3.33e-03
             eps.x  warn.limit.reject warn.limit.meanrw
          3.64e-10           5.00e-01           5.00e-01
      nResample         max.it       best.r.s       k.fast.s
            500             50              2              1
          k.max    maxit.scale      trace.lev            mts
            200            200              0           1000
    compute.rd fast.s.large.n
             0           2000
                  psi           subsampling
            "bisquare"         "nonsingular"
                  cov compute.outlier.stats
         ".vcov.avar1"                  "SM"
seed : int(0)
```

```
> confint(RP.rlm2)
```

```
             2.5 %  97.5 %
(Intercept) 960.86 1109.24
Hard         -6.92   -5.29
tTS          -2.75   -2.11
```

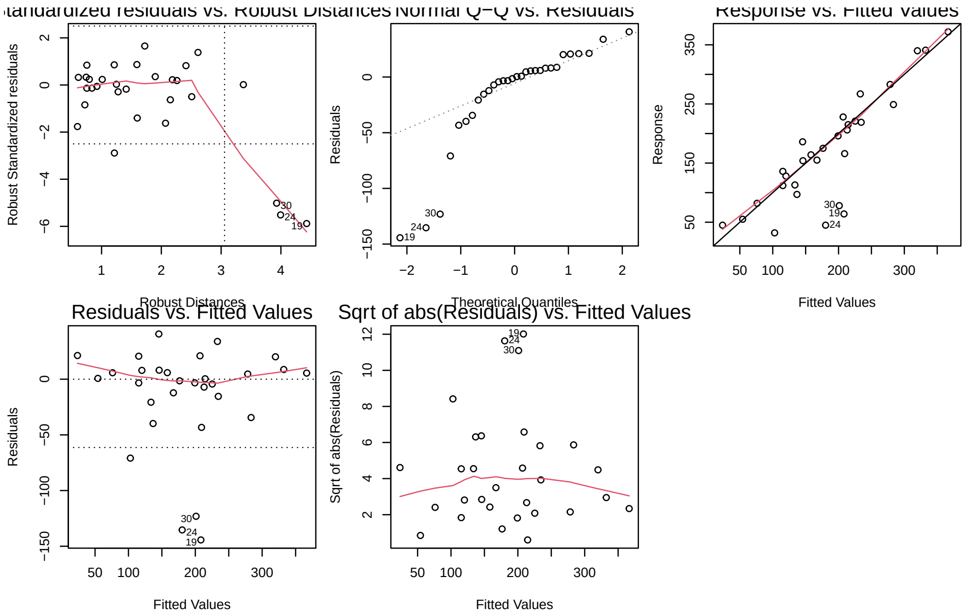
The classically estimated coefficients for `Hard` and `tTS` are not included in the 95% confidence intervals of the robustly estimated coefficients. Hence, the results differ significantly. The robustly estimated coefficient for `Hard` is smaller and that for `tTS` is larger than the corresponding classsically estimated coefficients. The estimations of the intercept do not differ so significantly.

b)
```
> par(mfrow = c(2, 2))
> plot(RP.lm)
```



There is hardly any evidence that any of the assumption is violated. There is also no outlier visible.

```
> par(mfrow = c(2, 3))
> plot(RP.rlm2)
```



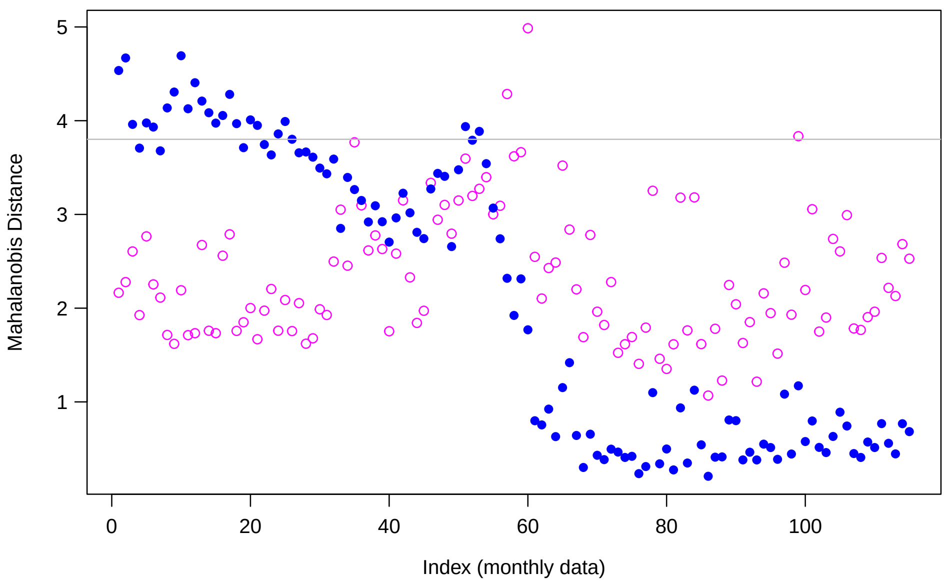
There are clearly three outliers which may be part of a distribution which is skewed to the left. Hence, there is clear evidence that the errors are not Gaussian distributed. Otherwise, there is little evidence that any other assumption is violated.
The conclusions from these two residual analyses do differ substantially with respect to the assessment of how adequate the assumption of Gaussian errors is.

2. **a)** (i) Stahel-Donoho estimator

(ii)
```
> library(rrcov)
> Fish2A <- Fish2[, 1:6]
> set.seed(1)
> Fish2A.rCov <- CovRobust(Fish2A)
> Fish2A.cov <- cov(Fish2A)
```

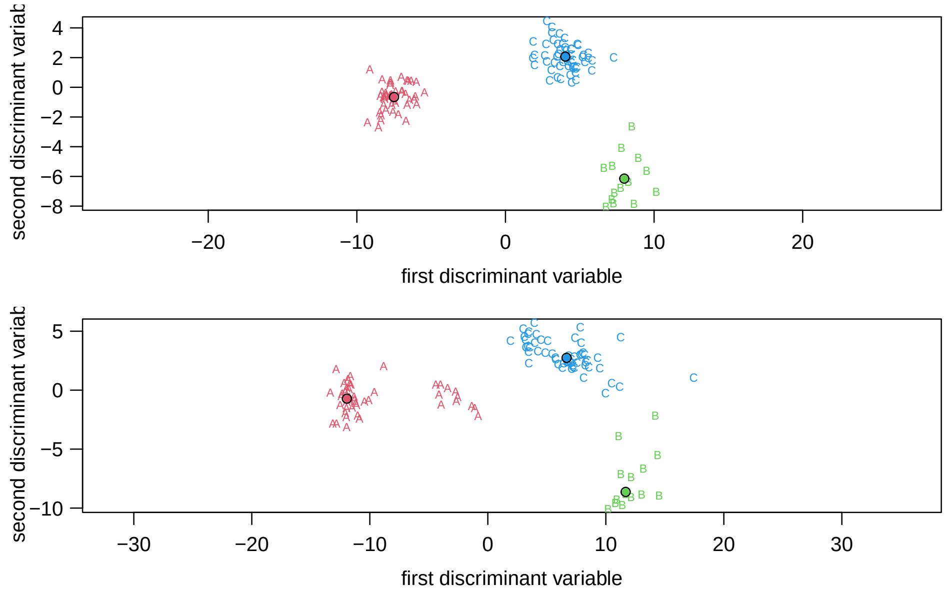
classically: 0.75, 0.31, 0.27 robustly: 6.02, 2.12, 2.23 The values are clearly different. The estimated variance using the robust method is at least 6 times larger than the classically estimated one.

b) (i)
```
> Fish2A.D2 <- mahalanobis(x = Fish2A, center = apply(Fish2A, 2, mean),
                           cov = Fish2A.cov)
> Fish2A.rD2 <- mahalanobis(x = Fish2A, center = Fish2A.rCov@center,
                            cov = Fish2A.rCov@cov)
> xAchse <- 1:nrow(Fish2A)
> plot(xAchse, sqrt(Fish2A.D2), type = "n", las = 1, mgp = c(2.5, 0.8, 0),
       xlab = "Index (monthly data)", ylab = "Mahalanobis Distance",
              ylim = range(sqrt(c(Fish2A.rD2, Fish2A.D2))))
> points(xAchse, sqrt(Fish2A.D2), col = "magenta")
> points(xAchse, sqrt(Fish2A.rD2), col = "blue", pch = 16)
> abline(h = sqrt(qchisq(0.975, df = ncol(Fish2A))), col = 'gray', las = 2)
```



(ii) classically: 3; robustly: 22

c) (i)
```
> library(MASS)
> source("https://stat.ethz.ch/Teaching/WBL/Source-WBL-8/03.RCodes/rg2-fkt-v2.R")
> Fish2.lda <- lda(Species ~ ., data = Fish2)
> Fish2.Rlda <- rlda(x = data.matrix(Fish2[, 1:6]), grouping = Fish2$Species)
> par(mfrow = c(2, 1), mar = c(4, 4, 0.5, 0.5), mgp = c(2.2, 0.8, 0))
> p.ldv(Fish2.lda, data = Fish2[, 1:6], group = Fish2$Species)
> p.ldv(Fish2.Rlda, data = Fish2[, 1:6], group = Fish2$Species)
```



The classical LDA shows three nicely separated groups. However, the group shapes are not a disk as the LDA methods indents to achieve. Using the robustified version, the three classes do separate clearly. However, it seems that class `A` consists of two classes. There is an outlier in class `C` and there are two outliers in class `B`.

(ii) The covariance matrix W (within class covariance matrix) and the centers of the classes are estimated robustly but not the covariance matrix of the centers.

3. a)
```
> H.nlr1 <- nls(Y ~ a * S / (1 + S / k), data = d.hake, start = list(a = 5, k = 40))
> summary(H.nlr1)
```

```
Formula: Y ~ a * S/(1 + S/k)
```

```
Parameters:
  Estimate Std. Error t value Pr(>|t|)
a     5.75       1.03    5.57  9.1e-05 ***
k    33.16      11.41    2.91    0.012 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15 on 13 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 8.33e-07
```

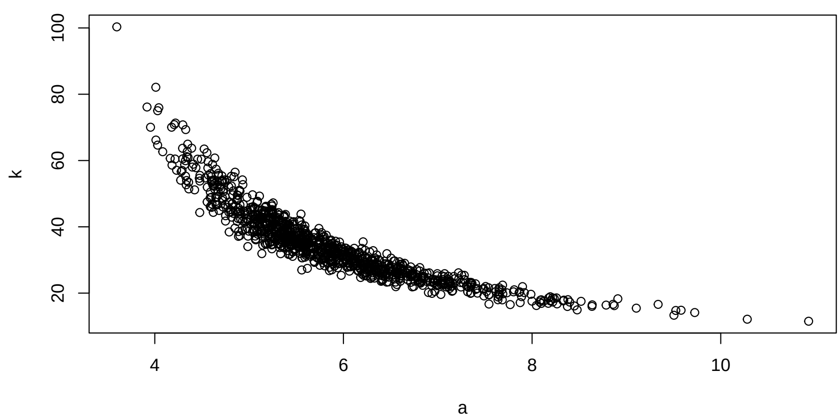
b)

```
> ## Wald-type
> H.coef <- coef(H.nlr1)
> h <- qt(0.975, nrow(d.hake) - 2) * coef(summary(H.nlr1))[, 2]
> cbind(H.coef - h, H.coef + h)
  [,1] [,2]
a  3.5    8
k  8.5   58
> ## Profile likelihood
> confint(H.nlr1)
  2.5% 97.5%
a  4.1   8.9
k 15.7  70.6
> ## Bootstrap
> library(nlstools)
> set.seed(1)
> H.boot <- nlsBoot(H.nlr1)
> summary(H.boot)
------
Bootstrap statistics
  Estimate Std. error
a      5.9       0.99
k     34.8      10.98

------
Median of bootstrap estimates and percentile confidence intervals
  Median 2.5% 97.5%
a    5.7  4.3   8.2
k   33.7 17.3  60.5
```

The profile likelihood and the bootstrap solution coincide more or less whereas the Walt-type solution is clearly shifted to the left.

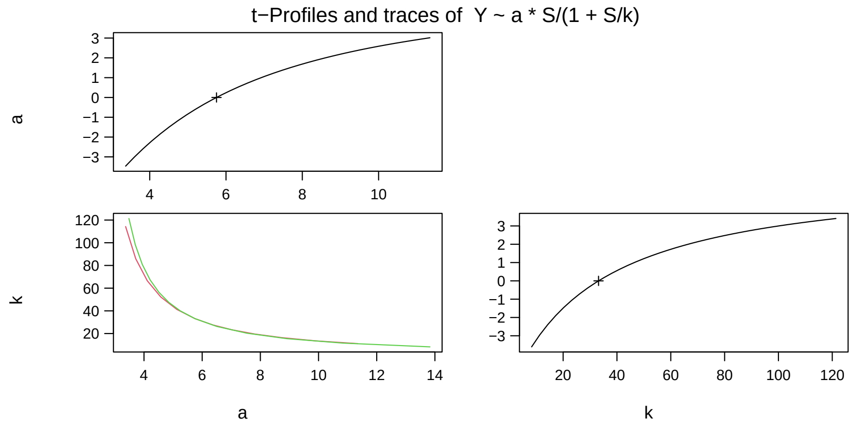
c)

```
> plot(H.boot$coefboot)
```



The bootstrap simulation shows that the points scatter within a “crescent” telling that the two coefficients are highly, but nonlinearly negative correlated.

d)
```
> library(sfsmisc)
> H.prof1 <- profile(H.nlr1)
> p.profileTraces(H.prof1)
```

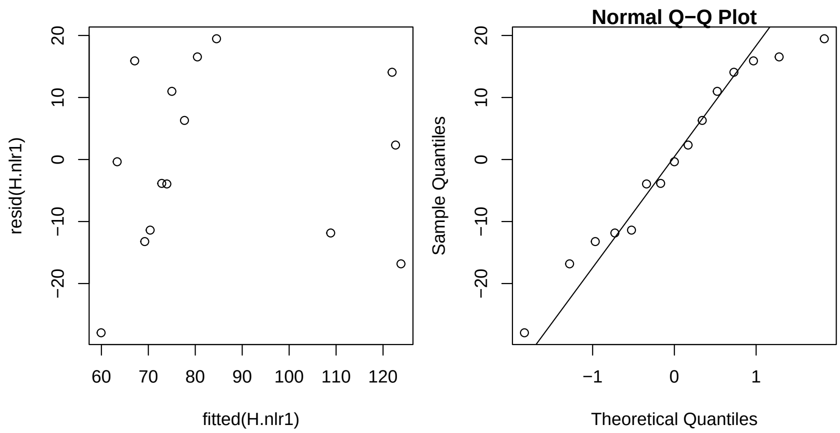


The profile traces are highly curved and the estimated coefficients a and k are nonlinearly correlated, t-profiles are highly curved –> therefore the differences between Wald-type and profile confidence intervals.

Conclusion: The confidence-intervals for Wald-type are less reliable. The interpretation of the estimated coefficients is difficult, because they are highly negatively correlated.

```
> summary(H.nlr1, cor = TRUE)

Formula: Y ~ a * S/(1 + S/k)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a     5.75       1.03    5.57  9.1e-05 ***
k    33.16      11.41    2.91    0.012 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15 on 13 degrees of freedom

Correlation of Parameter Estimates:
  a
k -0.97

Number of iterations to convergence: 5
Achieved convergence tolerance: 8.33e-07
```

e)
```
> par(mfrow = c(1, 2))
> plot(fitted(H.nlr1), resid(H.nlr1))
> qqnorm(resid(H.nlr1))
> qqline(resid(H.nlr1))
```



In the Tukey-Anscombe plot there is no evidence that the expected errors are not 0 or that the variance of the errors is not constant. The point in the normal plot scatter nicely around the line. Hence there is no evidence that the error is not normally distributed.
From this two graphics, there is no evidence that the assumptions are violated and hence we can rely on the inferential results.

f)

$$h\langle S_0, a, k\rangle \pm \sqrt{\widehat{\sigma}^2 + \widehat{\sigma}^2_{x_0}} \cdot q^{t_{n-p}}_{1-\alpha/2} = [\ 64.86\ ,\ 130.98\ ]$$

with

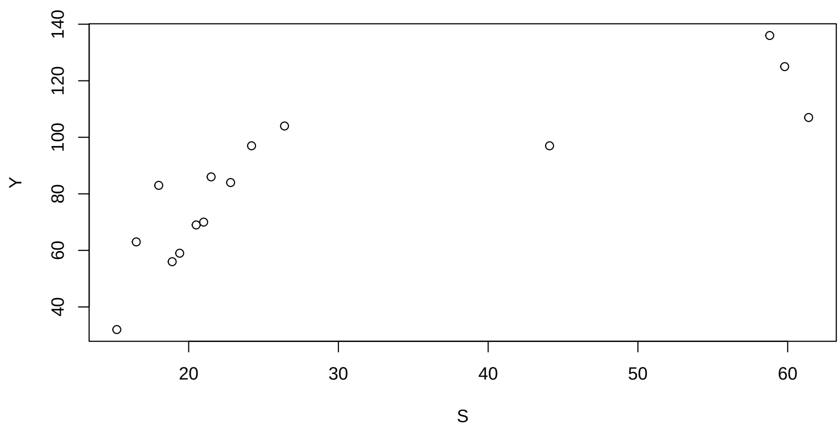
$$h\langle S_0, a, k\rangle = 97.92172, \qquad \widehat{\sigma}^2 = 216.077, \qquad \widehat{\sigma}^2_{x_0} = 18.0727, \qquad q^{t_{n-p}}_{1-\alpha/2} = 2.160369.$$

R-Code:

```
> ##sigma^2:
> t.s2 <- summary(H.nlr1)$sigma^2
> t.s2

[1] 216

> ## se(eta_0):
> f.a0 <- deriv(Y ~ a * S / (1 + S / k), c("a", "k"), function(a, k, S){})
> t.ak <- coef(H.nlr1)
> t.a0 <- f.a0(t.ak[1], t.ak[2], 35)
> t.a0 <- as.matrix(attr(t.a0, "gradient"))
> t.cov <- summary(H.nlr1)$cov.unscaled
> t.s2.x0 <- t.s2 * (t.a0 %*% t.cov %*% t(t.a0))
> t.s2.x0

     [,1]
[1,]   18

> ## Quantile of the t-distribution
> t.qt <- qt(0.975, summary(H.nlr1)$df[2])
> t.qt

[1] 2.2

> ## confidence interval
> predict(H.nlr1, newdata = list(S = 35))

[1] 98

> predict(H.nlr1, newdata = list(S = 35)) + sqrt(c(t.s2 + t.s2.x0)) * t.qt * c(-1, 1)

[1]  65 131
```

g)

```
> plot(Y  ~ S, data = d.hake)
```



In the plot we can see that for $S < 30$ there is a linear behavior. We also know that for $S = 0$ holds $h\langle S, a, k\rangle = 0$. Therefore we fit a linear regression model without intercept for all points $S < 30$. The result of the linear regression is the wanted slope at $S = 0$.

```
> coef(lm(Y ~ S - 1, data = d.hake, subset = S < 30))
```

```
  S
3.6
```

For large S the Beverton-Holt equation converges to $h\langle S, a, k\rangle = a \cdot k$ and so $k = \frac{h\langle S,a,k\rangle}{a}$. We can see in the plot that for large S the function $h\langle S, a, k\rangle$ converges to a high value of Y. Therefore we take the maximum of Y as approximation for $h\langle S, a, k\rangle$.

```
> max(d.hake$Y) / 3.6
[1] 38
```