EX5

roi hezkiyahu

13 5 2022

```
#imports
library(tidyverse)
library(MASS)
library(tidymodels)
library(glue)
```

Q1

Question 1.

Let $Y_1, \cdots, Y_n \sim f_\theta(y)$ with an unknown parameter θ.

1. Show that the sample α-quantile $y_{(\alpha)}$ is the M-estimator corresponding to $\rho(y,\theta) = \alpha(y-\theta)_+ + (1-\alpha)(\theta-y)_+$.
2. What is the asymptotic distribution of $y_{(\alpha)}$? Find an (asymptotic) $(1-\delta)100\%$ confidence interval for the α-quantile $F_\theta^{-1}(\alpha)$ of the distribution f_θ.

1

$$\text{we are looking for a } \hat{\theta} = argmin_\theta \sum_{i=1}^n \rho(y_i, \theta)$$

$$\text{denote } n_- = \sum_{i=1}^n I(y_i < \theta), \quad n_+ = n - n_-$$

$$\sum_{i=1}^n \rho(y,\theta) = n_+\alpha + (1-\alpha)n_-$$

$$\psi(y,\theta) = \begin{cases} -1, & y > \theta \\ 0, & y = \theta \\ 1, & y < \theta \end{cases}$$

$$\frac{\partial \sum_{i=1}^n \rho(y,\theta)}{\partial\theta} = -n_+\alpha + n_-(1-\alpha)$$

$$\text{if } \theta = \hat{\theta_{(\alpha)}} = y_{(\alpha)}:$$

$$n_+ = n(1-\alpha), \quad n_- = n\alpha \Rightarrow \frac{\partial \sum_{i=1}^n \rho(y,\theta)}{\partial\theta} = -n(1-\alpha)\alpha + (1-\alpha)n\alpha = 0 \Rightarrow \hat{\theta_{(\alpha)}} = argmin_\theta \sum_{i=1}^n \rho(y_i,\theta)$$

$$\text{notice that this is indeed a minimum point as the function is not bound from above w.l.o.g difine } \alpha > 0.5$$
$$\text{we can alwyas decreace } \theta \text{ and get a higher value for the function } \rho$$

2

$$\text{under regulatory conditions we saw in class that: } \hat{\theta} \dot{\sim} N(\theta^*, V^2)$$
$$\text{denote } p = P(y < \theta)$$
$$\theta^* = argmin(E(\rho(y,\theta))) = y_{(\alpha)}$$
$$E_{\theta_0}(\psi(y, y_{(\alpha)}))^2 = E(1) = 1$$
$$E_{\theta_0}\psi'(y, y_{(\alpha)}) = (E_{\theta_0}\psi(y, y_{(\alpha)}))' = (\int_{-\infty}^{\theta} f_{\theta_0}(y)dy - \int_{\theta}^{\infty} f_{\theta_0}(y)dy)' = (2F(\theta^*) - 1)' = 2f_{\theta_0}(y_{(\alpha)})$$
$$\text{thus we get: } \hat{\theta} \dot{\sim} N(y_{(\alpha)}, \frac{1}{4nf_{\theta_0}(y_{(\alpha)})^2})$$

Q2

Question 2.

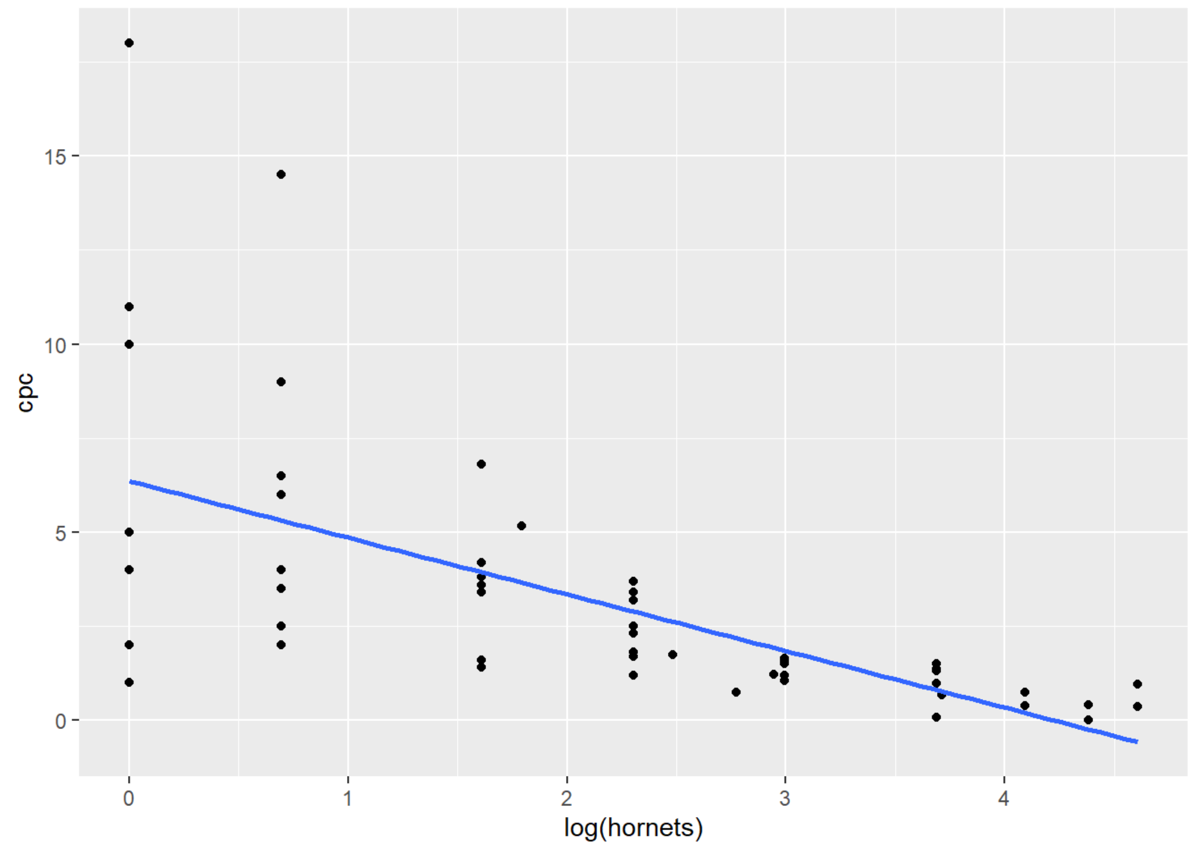
The file Hornets.dat (Hornets.dat) provides the results of the research on hornets' cells building. The file contains the numbers of hornets in the *i*-th box and the numbers of cells per capita, *CPC*, (hornet) that were built in the *i*-th box.

1. Fit a linear model of *CPC* as a function of *log(Hornets)*. What can you say about the adequacy of the model? Try to find an appropriate transformation of the response and re-fit the model. Comment the results.
2. Fit robust regression using several M-estimators: Hampel, Huber, Tukey's bisquare, etc. Compare the results and compare them with the OLS fit from the previous paragraph.

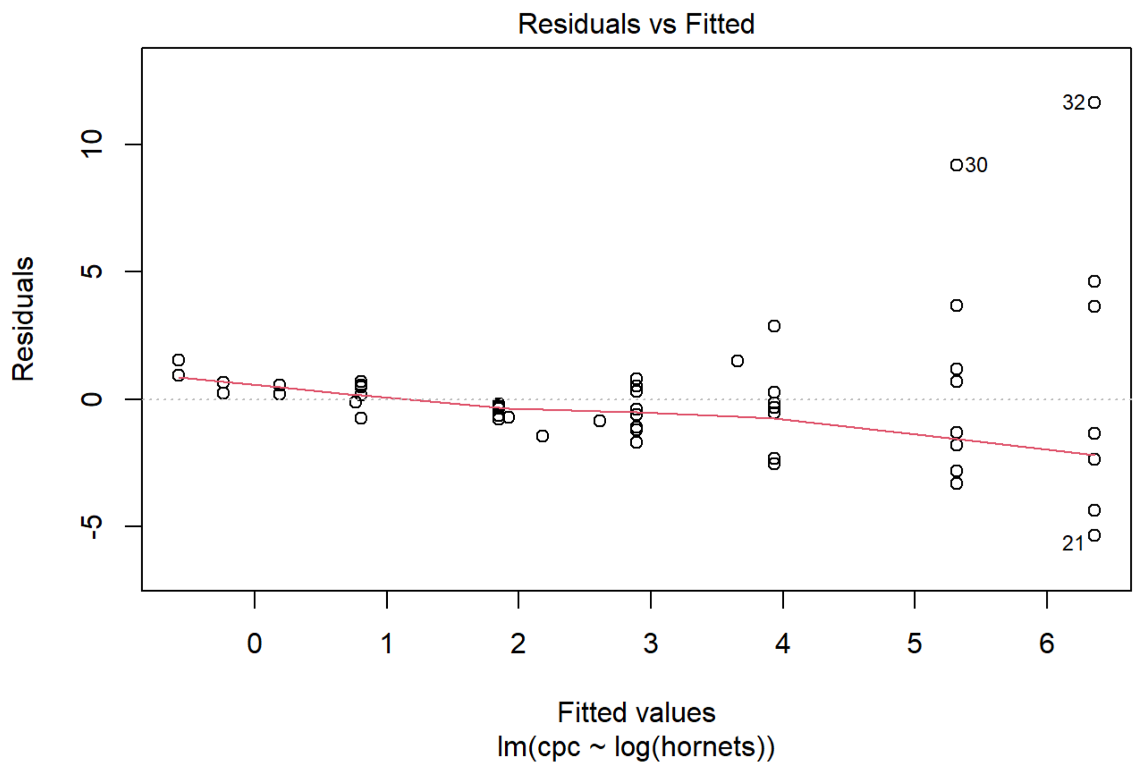
1

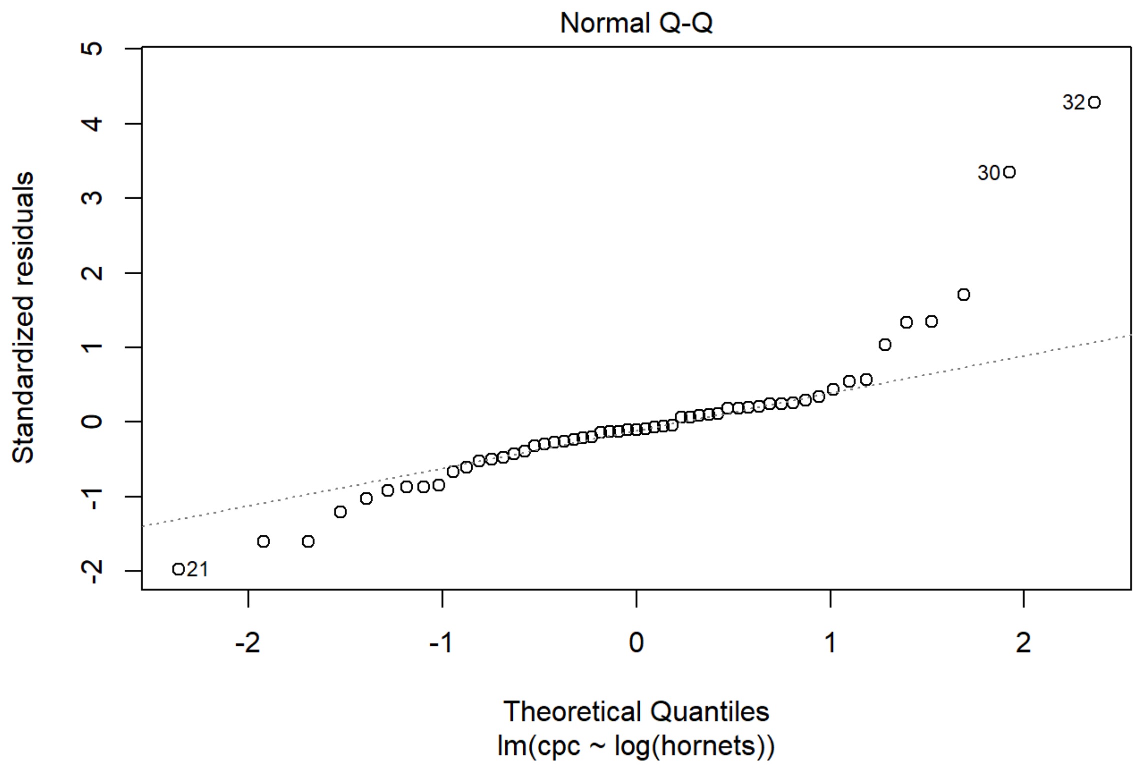
```
horn <- read.table("Hornets.dat",header = T)
horn %>%
  ggplot(aes(x = log(hornets),y = cpc)) +
  geom_point()+
  geom_smooth(se = F,method = lm)
```

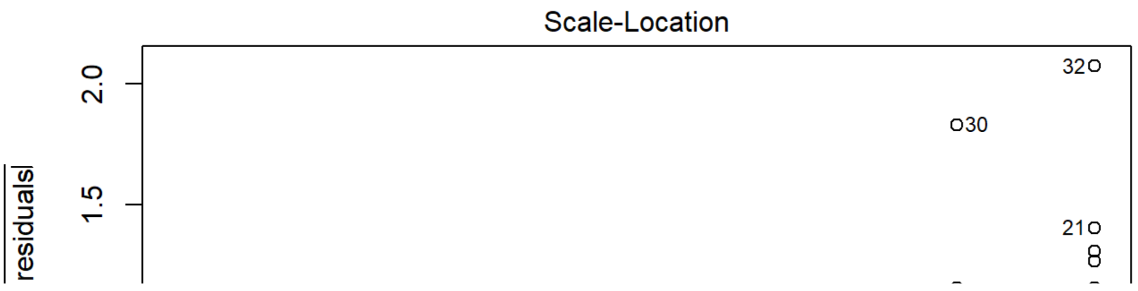
```
## `geom_smooth()` using formula 'y ~ x'
```

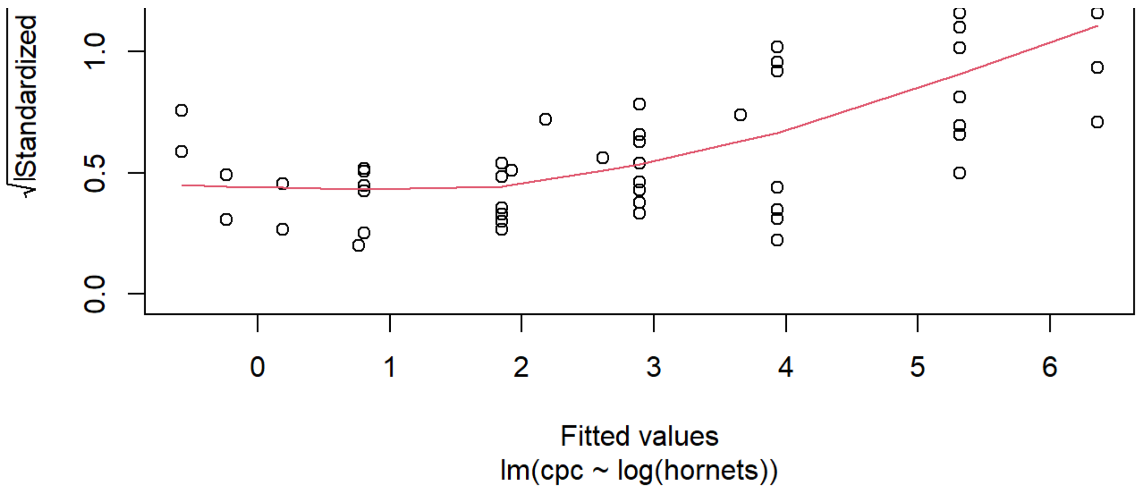


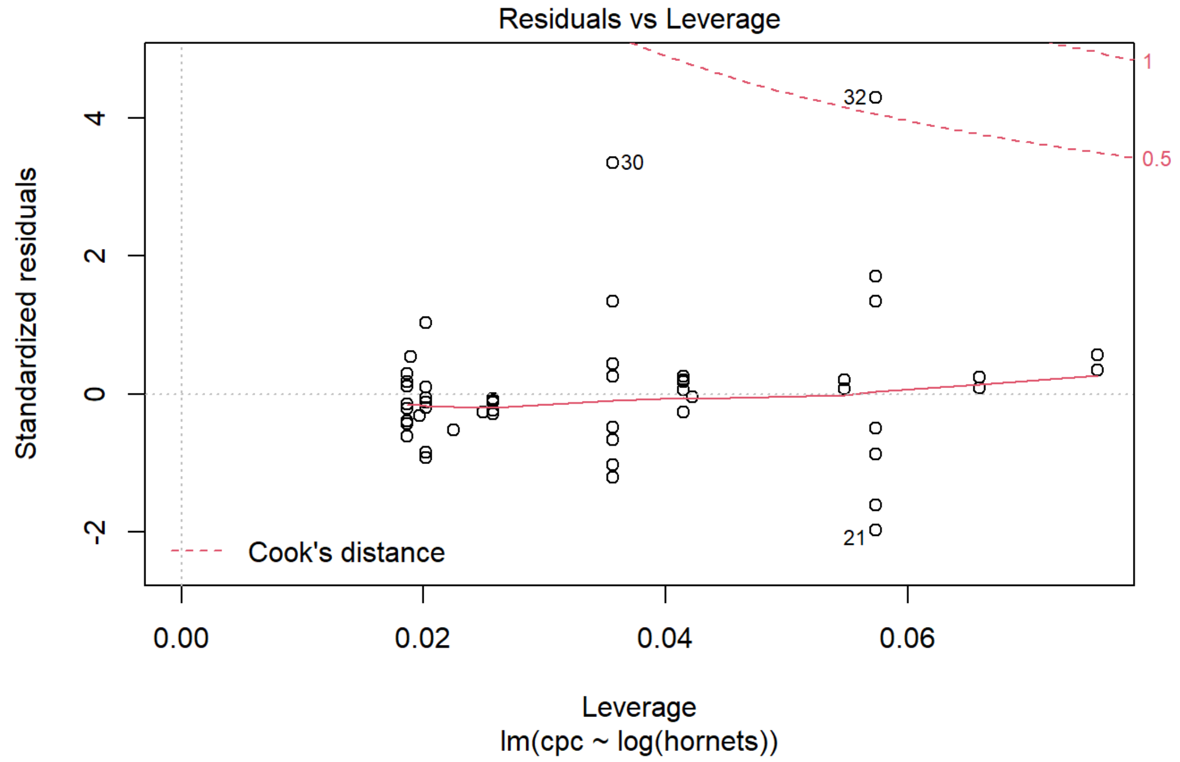
```
mod_1 <- lm(cpc~ log(hornets),data =horn)
plot(mod_1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cpc ~ log(hornets))
32
30
21

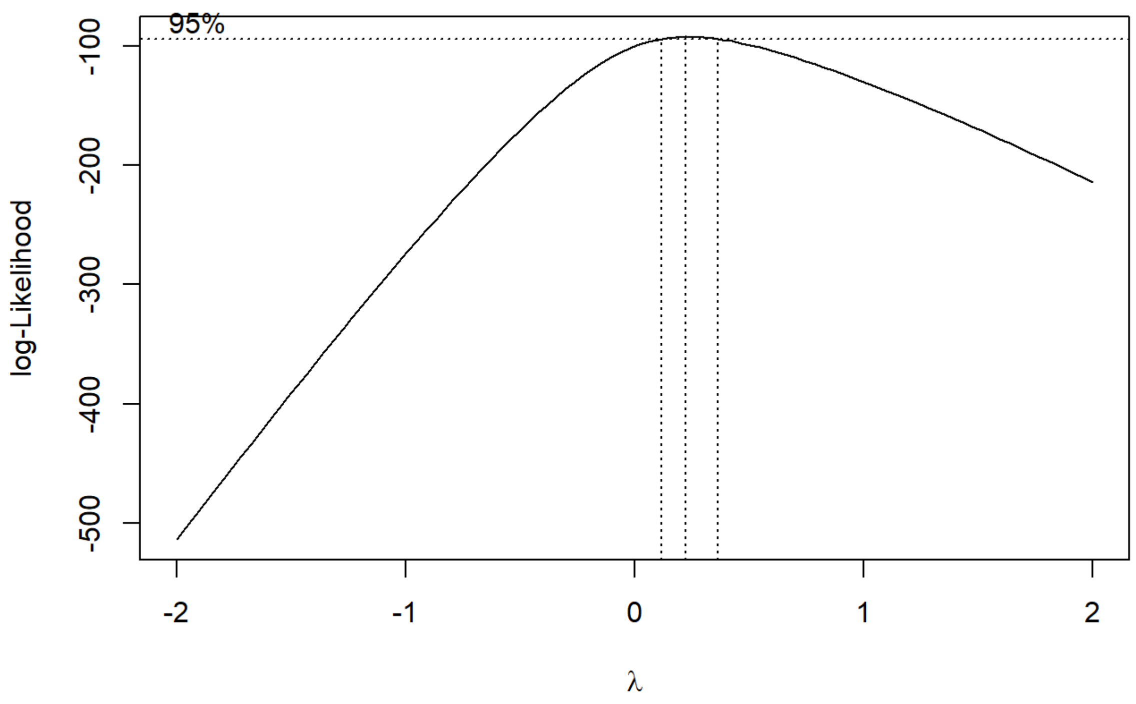

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(cpc ~ log(hornets))
32
30
21


Scale-Location
|residuals|
32
30
21



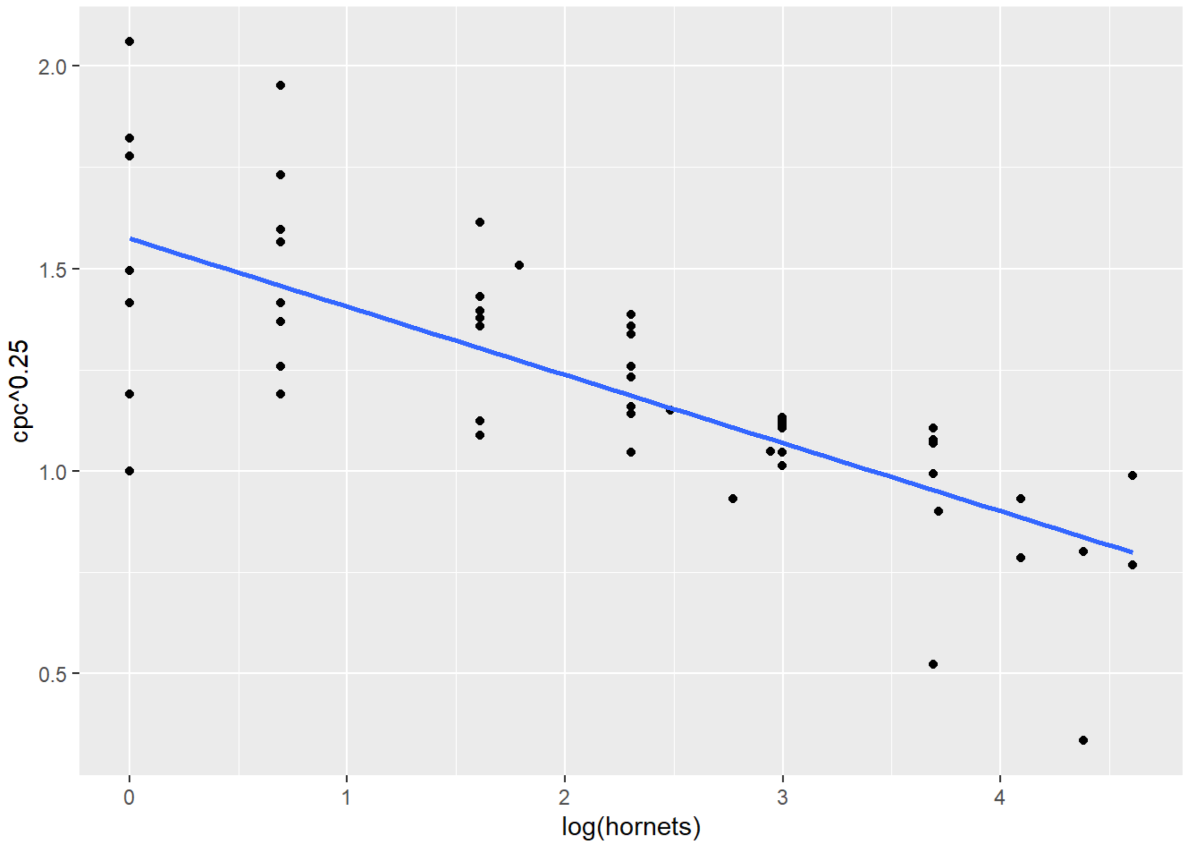


```
boxcox(cpc~ log(hornets),data =horn)
```

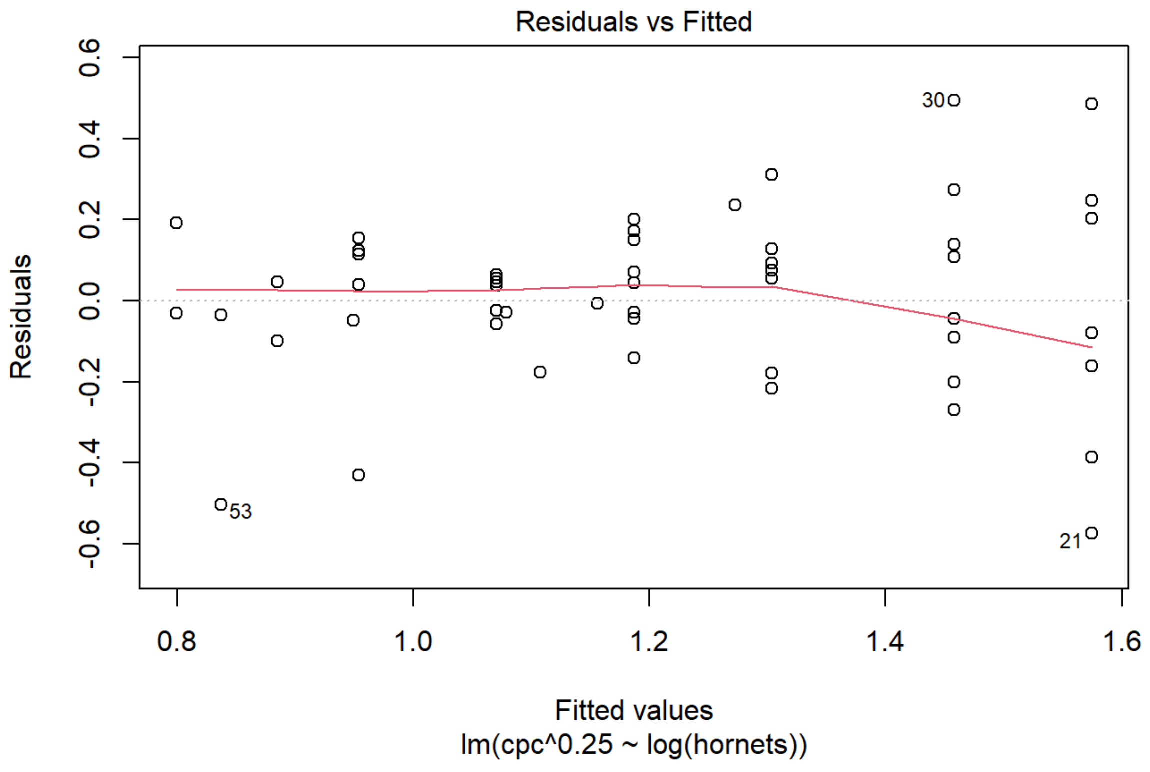


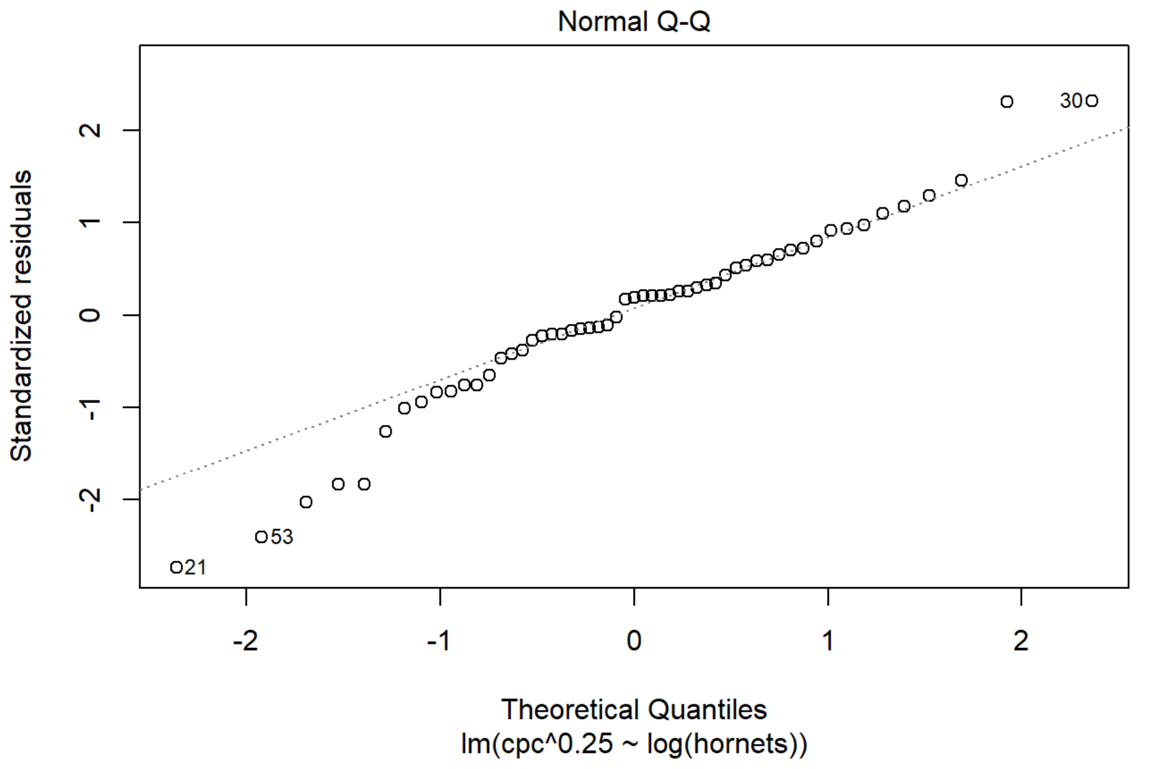
```
horn %>%
  ggplot(aes(x = log(hornets),y = cpc^0.25)) +
  geom_point()+
  geom_smooth(se = F,method = lm)
```

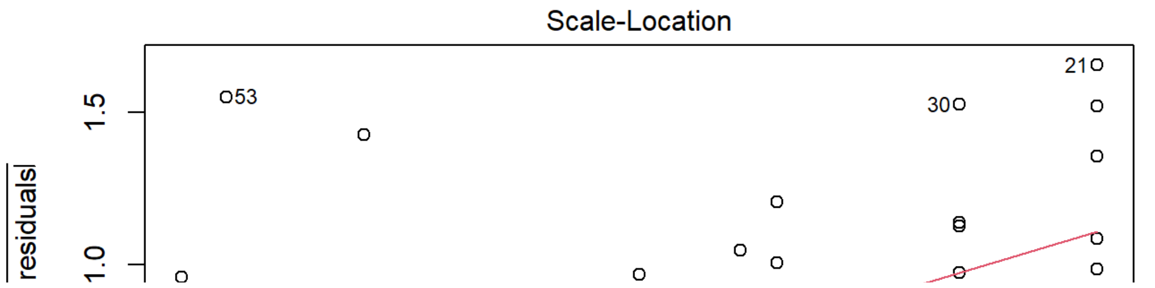
```
## `geom_smooth()` using formula 'y ~ x'
```

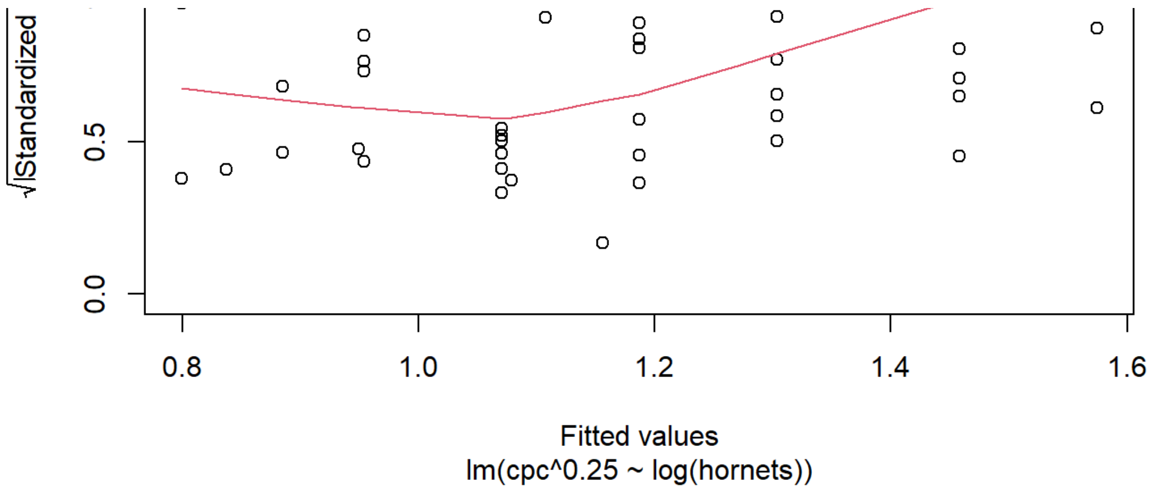


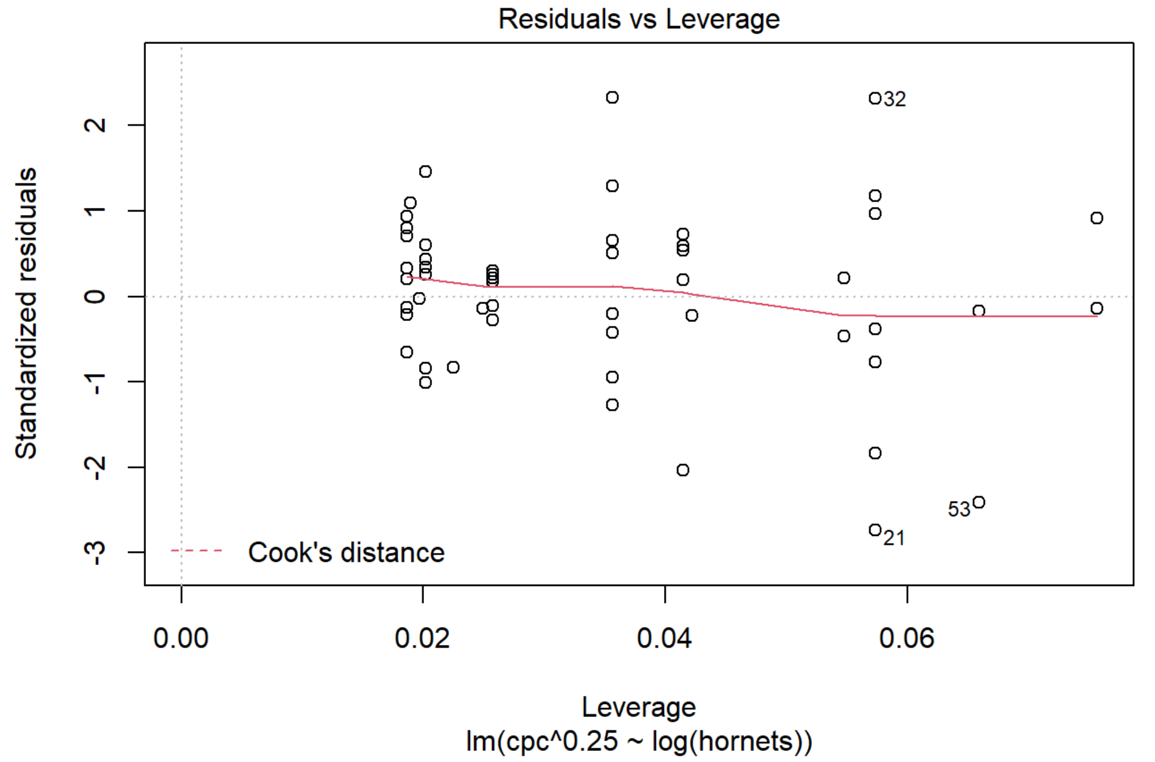
```
mod_2 <- lm(cpc^0.25~ log(hornets),data =horn)
plot(mod_2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cpc^0.25 ~ log(hornets))
30
53
21
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0.8
1.0
1.2
1.4
1.6


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(cpc^0.25 ~ log(hornets))
30
53
21
2
1
0
-1
-2


Scale-Location
residuals|
21
30
53
1.5
1.0



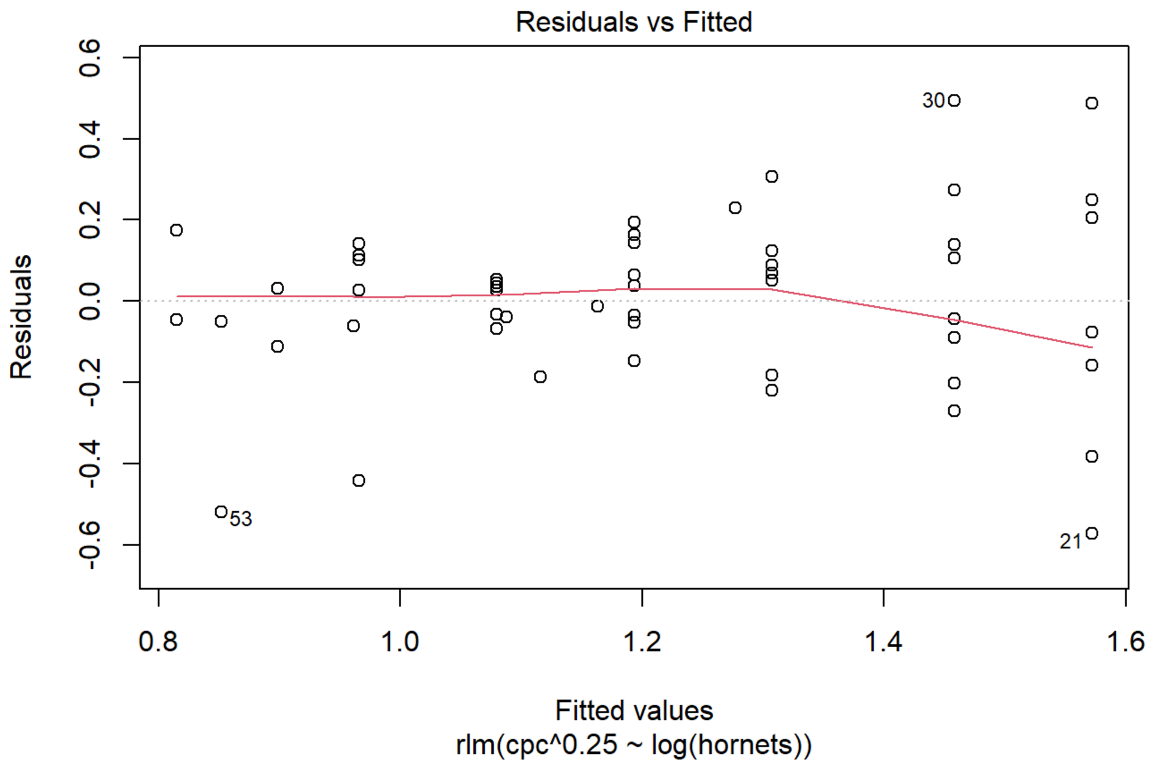


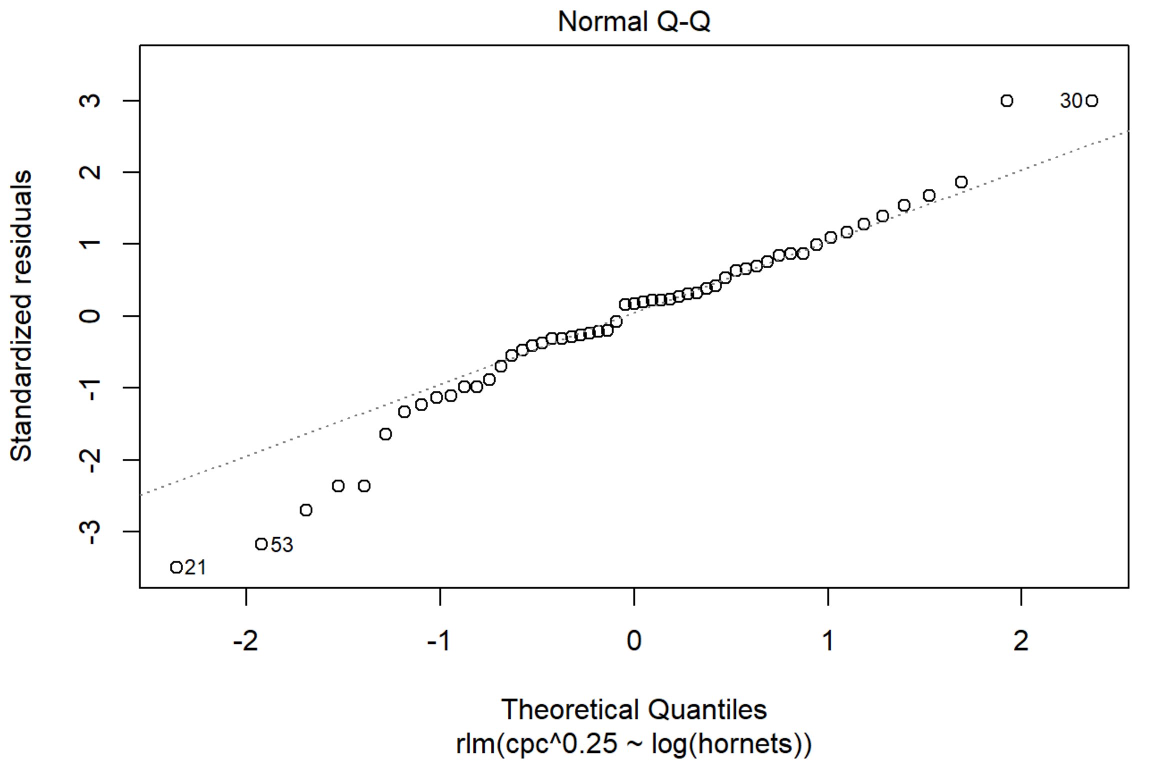
from the residuals of model 1 we can see that the linear model tested does not adequately fit, after applying a ^0.25 transformation to CPC we can see a better fit.

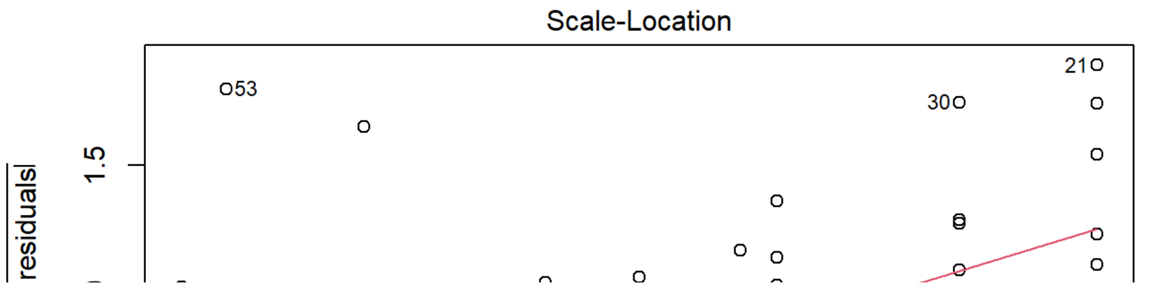
also it looks like the variance is not equal for all x values.

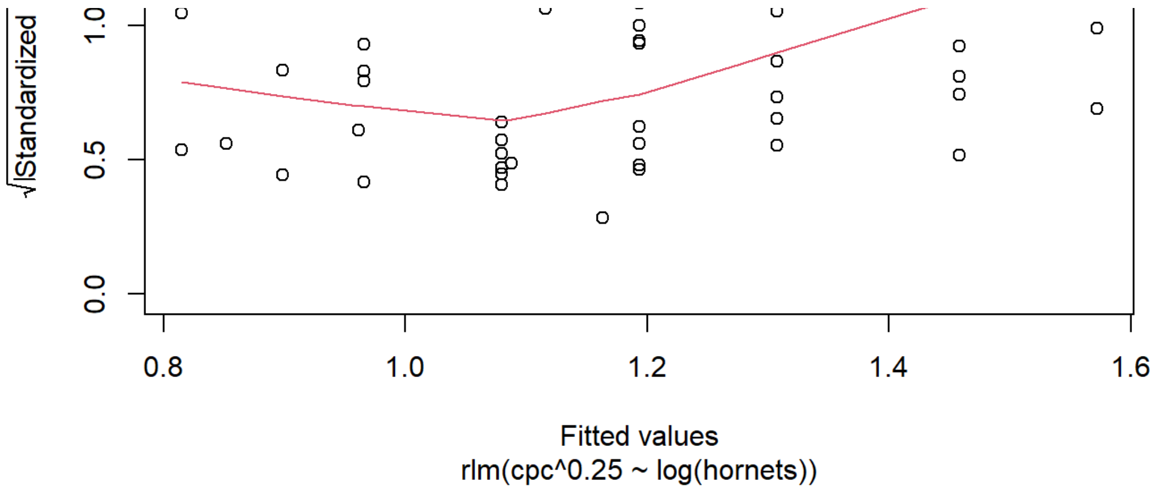
2

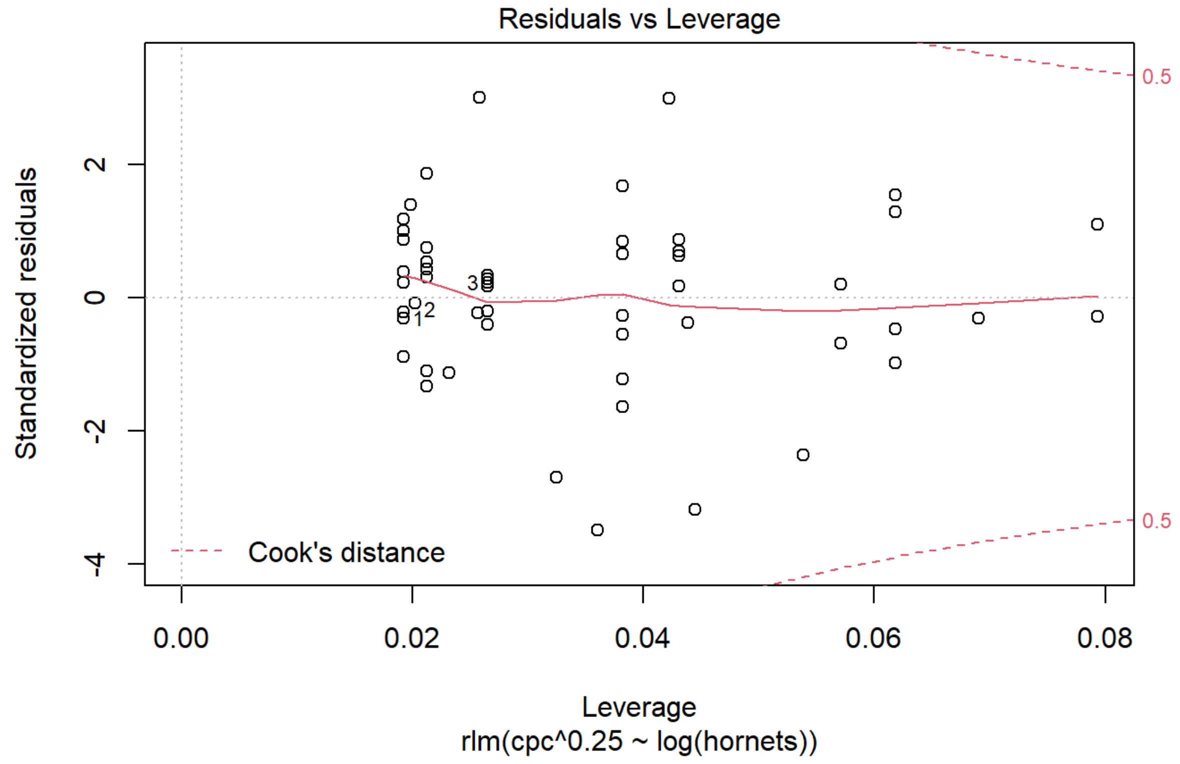
```
hampel <- rlm(cpc^0.25~ log(hornets),data =horn,psi = psi.hampel)
huber <- rlm(cpc^0.25~ log(hornets),data =horn,psi = psi.huber)
tukey <- rlm(cpc^0.25~ log(hornets),data =horn,psi = psi.bisquare)
plot(hampel)
```


Residuals vs Fitted
Residuals
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
30
53
21
0.8
1.0
1.2
1.4
1.6
Fitted values
rlm(cpc^0.25 ~ log(hornets))

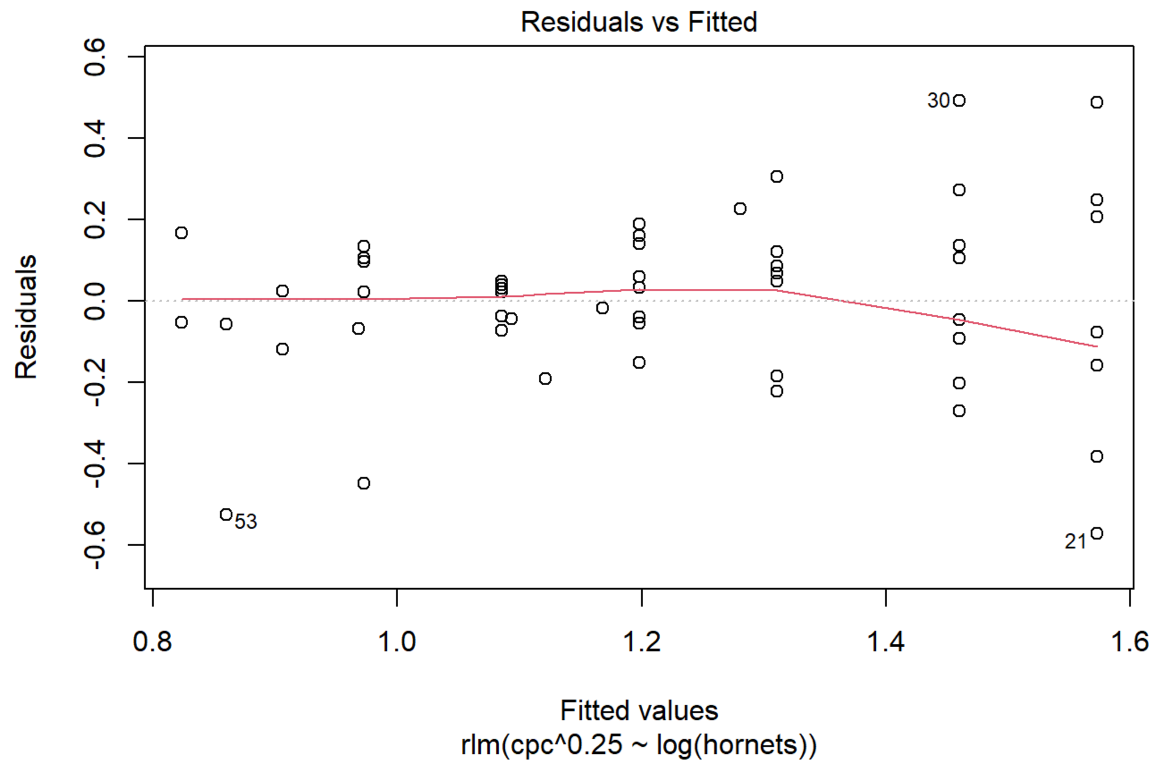

Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-3
30
53
21
-2
-1
0
1
2
Theoretical Quantiles
rlm(cpc^0.25 ~ log(hornets))

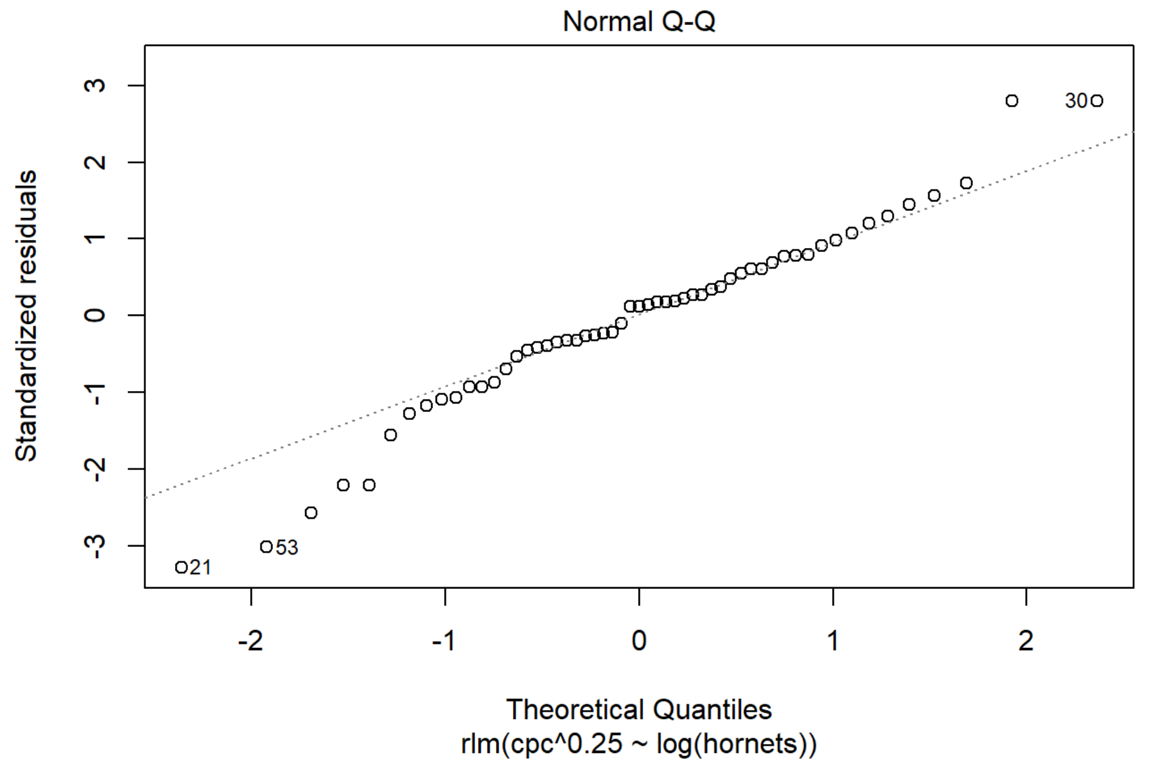

Scale-Location
residuals|
1.5
53
30
21

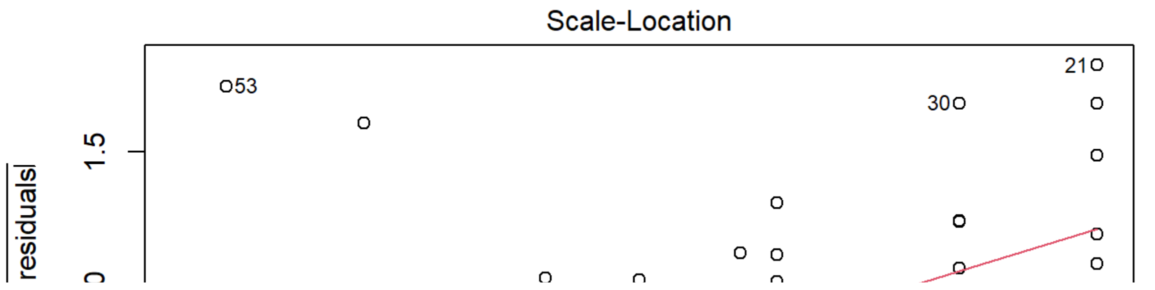

√|Standardized
1.0
0.5
0.0
0.8
1.0
1.2
1.4
1.6
Fitted values
rlm(cpc^0.25 ~ log(hornets))

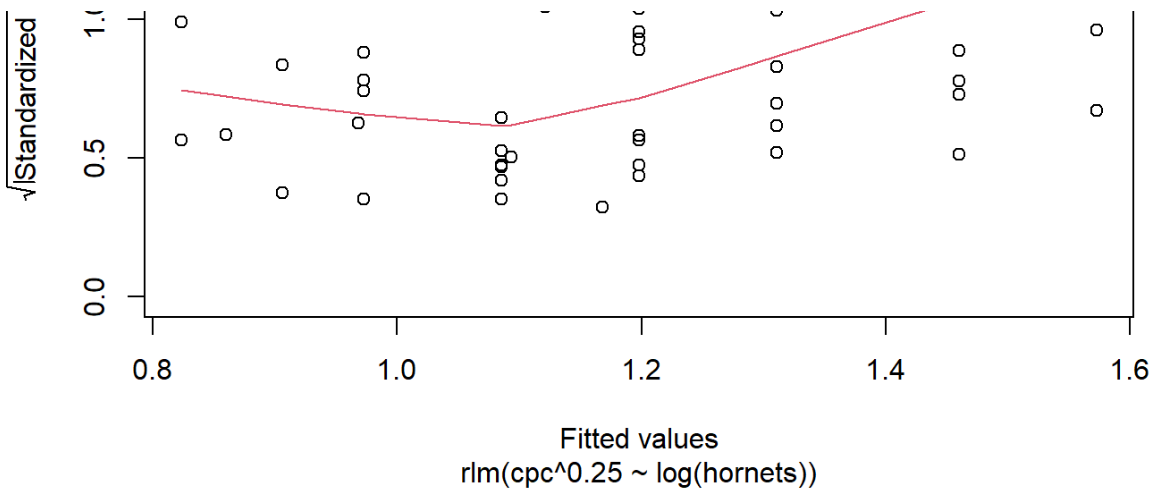

Residuals vs Leverage
Standardized residuals
2
0
-2
-4
0.5
0.5
3
2
1
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
rlm(cpc^0.25 ~ log(hornets))

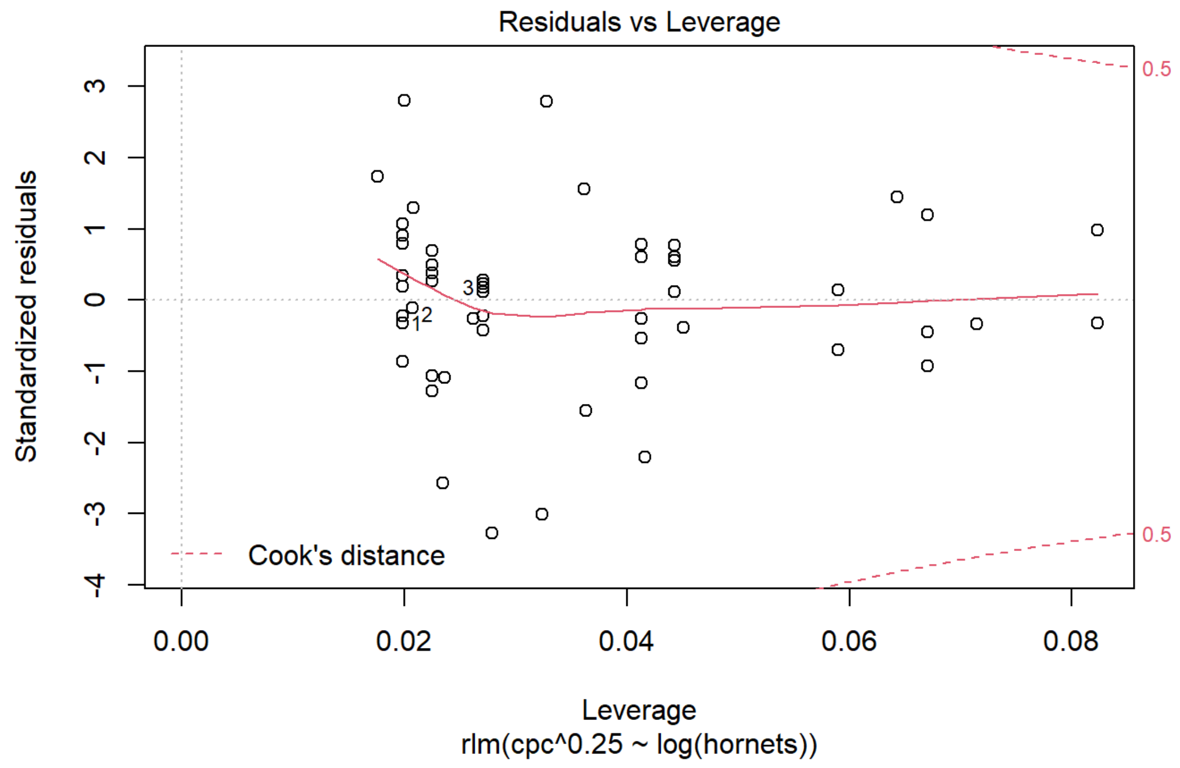
```
plot(huber)
```


Residuals vs Fitted
Residuals
Fitted values
rlm(cpc^0.25 ~ log(hornets))
30
53
21
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0.8
1.0
1.2
1.4
1.6

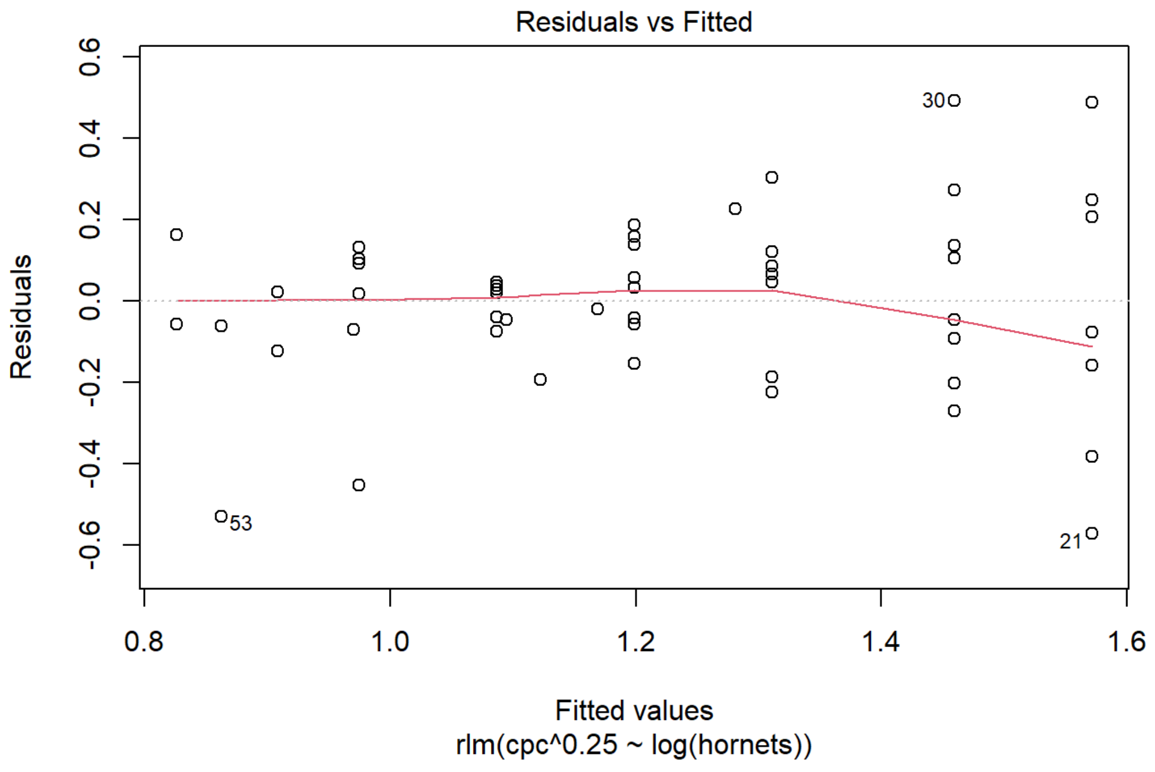

Normal Q-Q
Standardized residuals
Theoretical Quantiles
rlm(cpc^0.25 ~ log(hornets))
30
53
21
3
2
1
0
-1
-2
-3

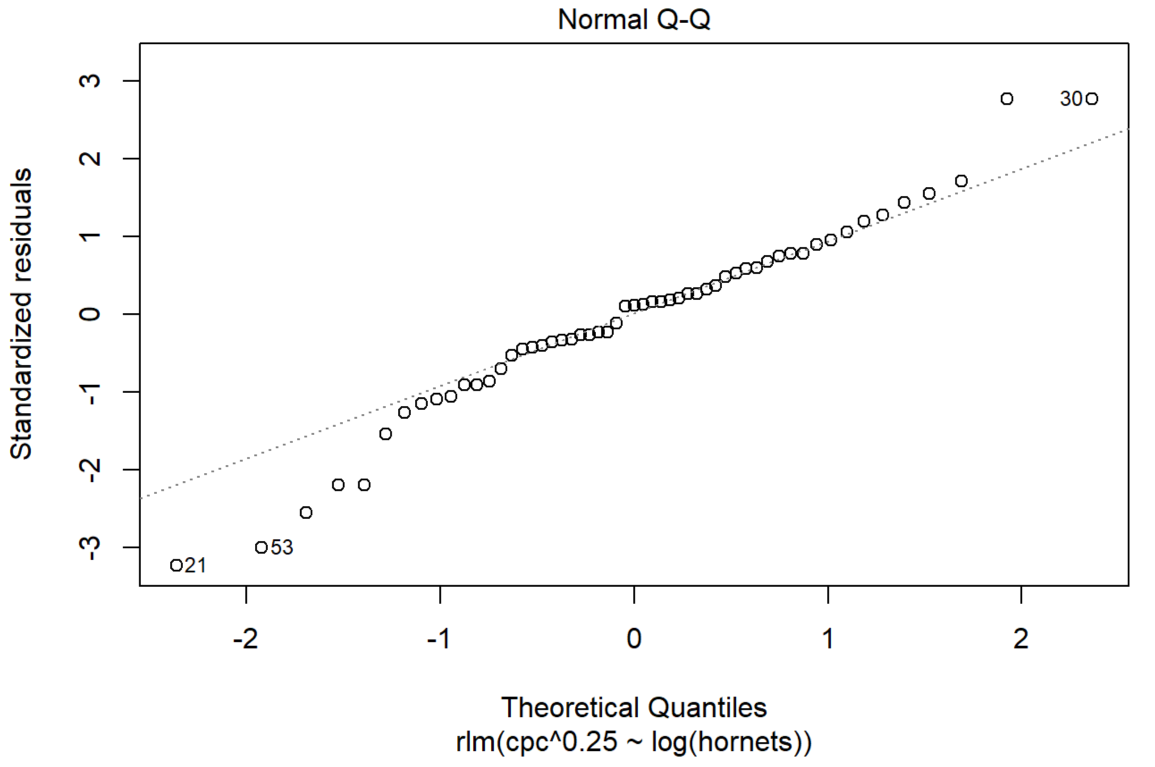

Scale-Location
residuals
1.5
53
21
30

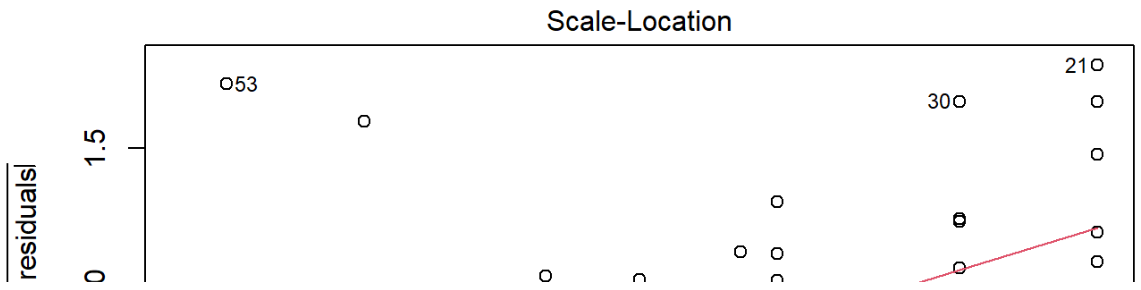

√|Standardized
1.0
0.5
0.0
0.8
1.0
1.2
1.4
1.6
Fitted values
rlm(cpc^0.25 ~ log(hornets))

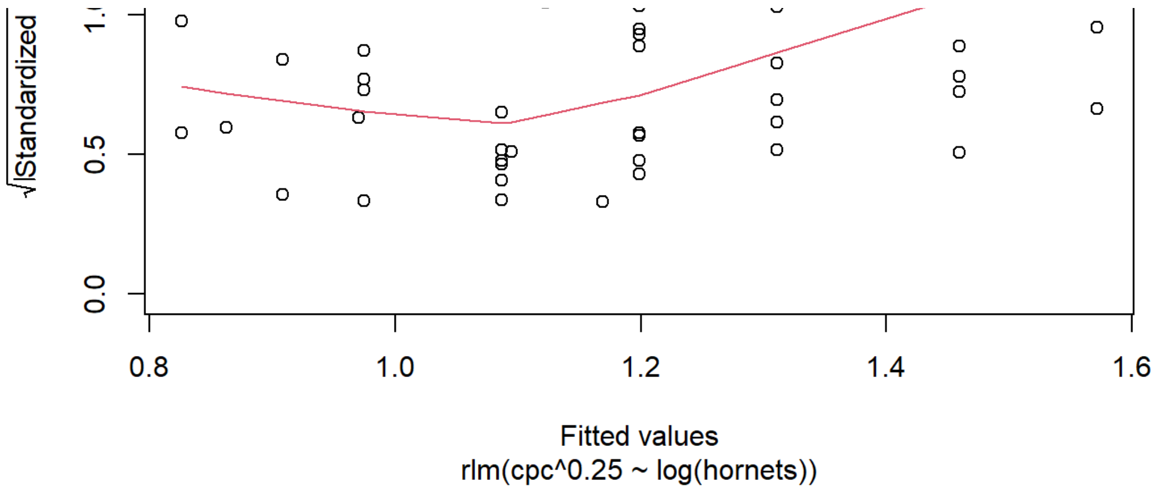

Residuals vs Leverage
Standardized residuals
3
2
1
0
-1
-2
-3
-4
0.5
0.5
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
rlm(cpc^0.25 ~ log(hornets))

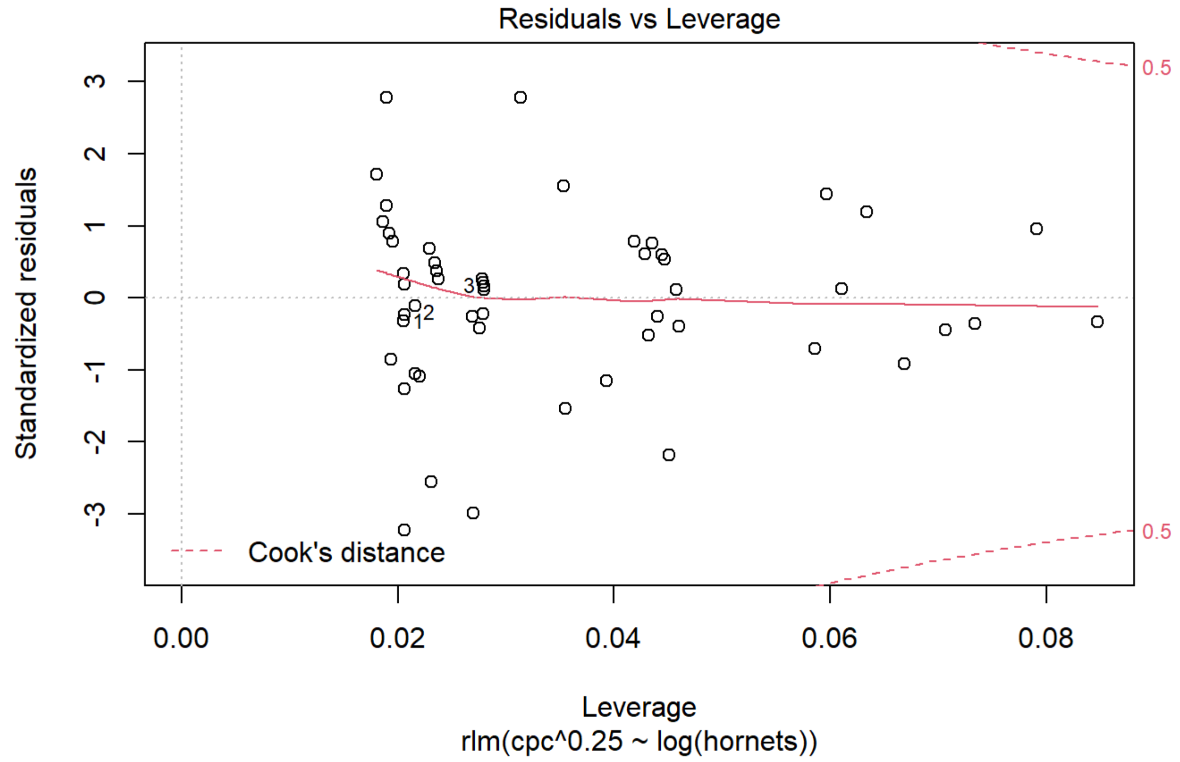
```
plot(tukey)
```


Residuals vs Fitted
Residuals
Fitted values
rlm(cpc^0.25 ~ log(hornets))
30
53
21
0.8
1.0
1.2
1.4
1.6
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6


Normal Q-Q
Standardized residuals
Theoretical Quantiles
rlm(cpc^0.25 ~ log(hornets))
30
53
21
-2
-1
0
1
2
-3
-2
-1
0
1
2
3


Scale-Location
|residuals|
53
21
30
1.5
0





```
tibble("humple" = tidy(hampel)$estimate,
       "huber" = tidy(huber)$estimate,
       "tukey" = tidy(tukey)$estimate,
       "mod_2" = tidy(mod_2)$estimate)
```

```
## # A tibble: 2 x 4
##   humple  huber  tukey  mod_2
##    <dbl>  <dbl>  <dbl>  <dbl>
## 1   1.57   1.57   1.57   1.57
## 2 -0.164 -0.163 -0.162 -0.168
```

all robust models look rather the same, the estimates are the same, and the plots are similar

Q3

Question 3.

The file Puromycin.dat (Puromycin.dat) contains the data on the substrate concentration of Puromycin, x (parts per million, ppm) and the initial rate, or "velocity", y, of the enzymatic reaction (counts/min^2) in the presence of Puromycin. The velocity is assumed to depend on the substrate concentration according to the Michaelis-Menten equation: $y=\theta_1 x/(\theta_2+x)$.

1. What is the physical/mathematical meaning of the parameters θ_1 and θ_2?
2. Using transformations of x and y transform the original nonlinear model to a linear one. Fit the corresponding linear model. Does it seem to be adequate?
3. Find the gradient matrix D for the Michaelis-Menten original nonlinear model. Fit the nonlinear model using the results of the previous paragraph for obtaining initial values for the parameters, and check the adequacy of the nonlinear model.
4. Test the hypothesis θ_1=200 applying F- and t-tests, comment the results.
5. Predict velocity of the enzyme reaction when the substrate concentration of Puromycin is at level 0.5. Give the corresponding confidence and prediction intervals.

1

$$\theta_1 \text{ is the maxsimum achivable velocity}$$
$$\theta_2 \text{ is a decaying paramater probabbly connected to the substrate concentration}$$

2

$$\frac{1}{y} = \frac{\theta_2}{\theta_1}\frac{1}{x} + \frac{1}{\theta_1}$$

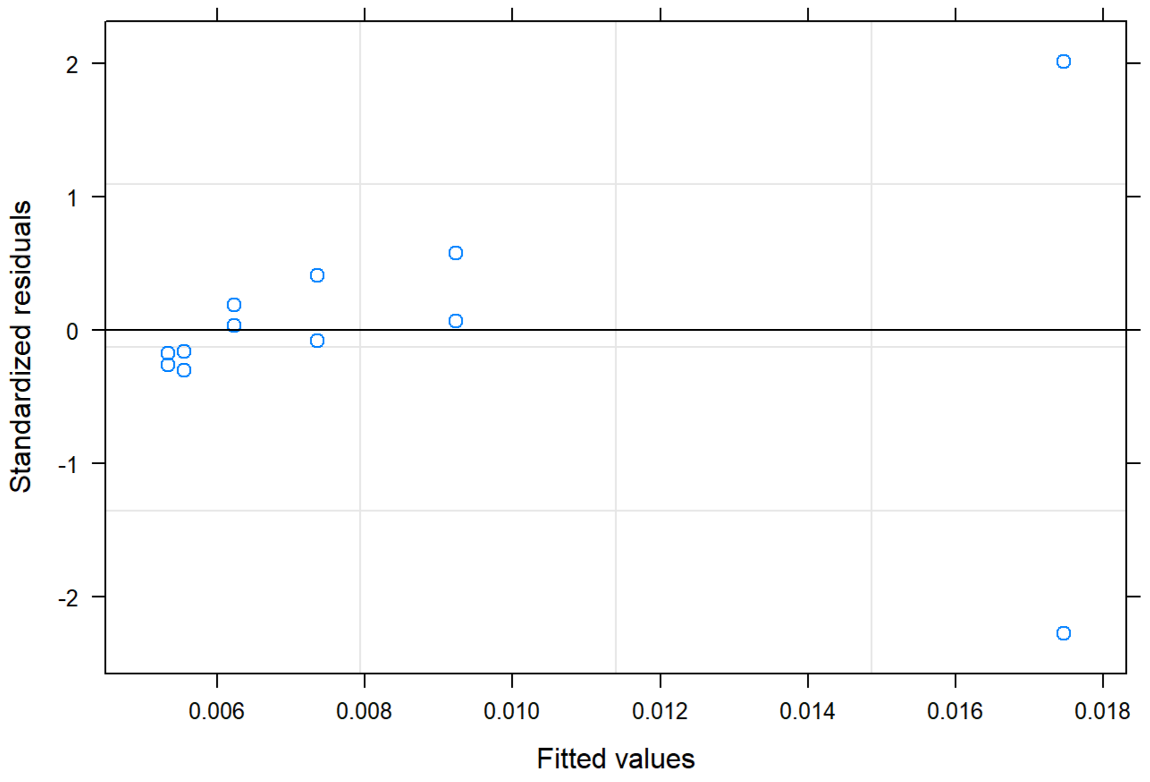
```
puro <- read.table("Puromycin.dat")
#calc prediction
y_hat <- function(x,theta1,theta2){
  return(theta1*x/(x+theta2))
}

nls_mod <- nls(1/V2 ~ t2/t1 * 1/V1 + 1/t1,data = puro)
```

```
## Warning in nls(1/V2 ~ t2/t1 * 1/V1 + 1/t1, data = puro): No starting values specified for some paramet
ers.
## Initializing 't2', 't1' to '1.'.
## Consider specifying 'start' or using a selfStart model
```

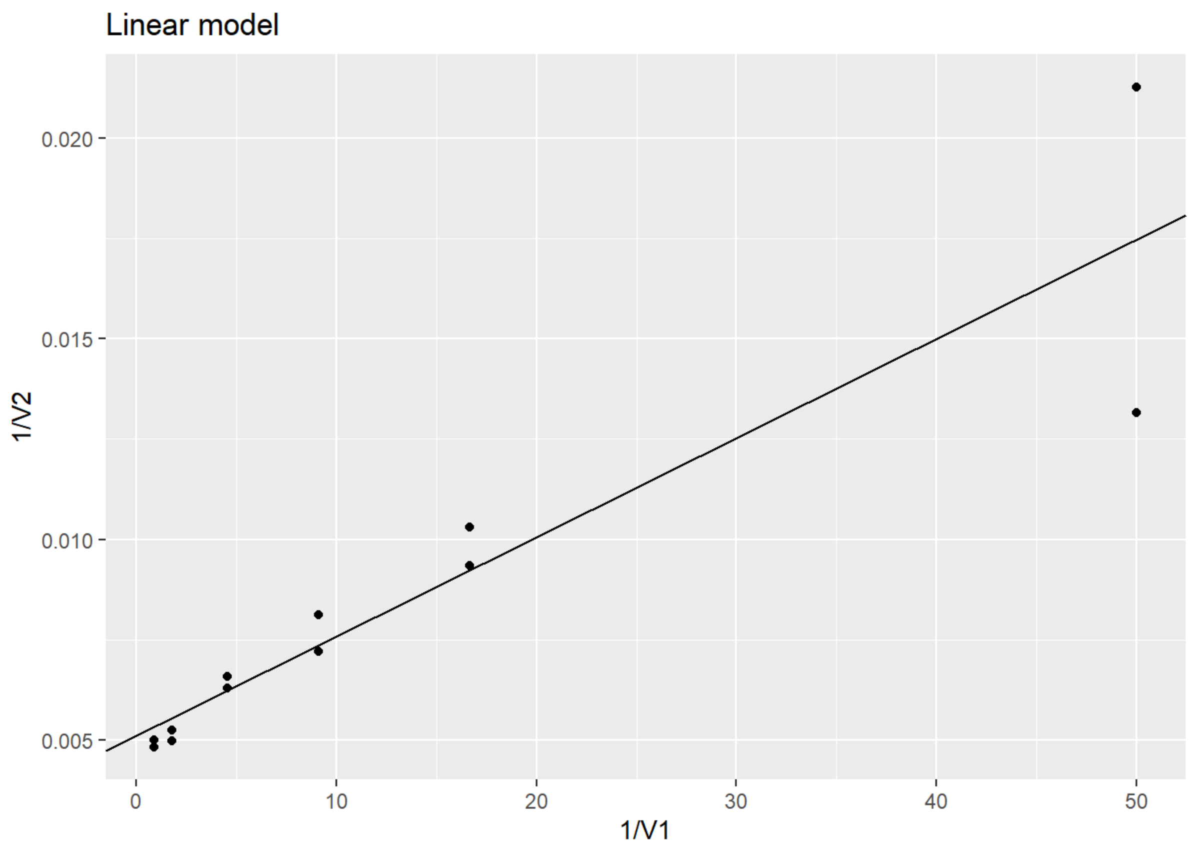
```
summary(nls_mod)
```

```
##
## Formula: 1/V2 ~ t2/t1 * 1/V1 + 1/t1
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## t2   0.04841    0.01170   4.136  0.00202 **
## t1 195.80271   26.99038   7.255 2.74e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001892 on 10 degrees of freedom
##
## Number of iterations to convergence: 12
## Achieved convergence tolerance: 5.272e-08
```

```
theta <- tidy(nls_mod) %>% pluck("estimate")
theta2 = theta[1]
theta1 = theta[2]
plot(nls_mod)
```

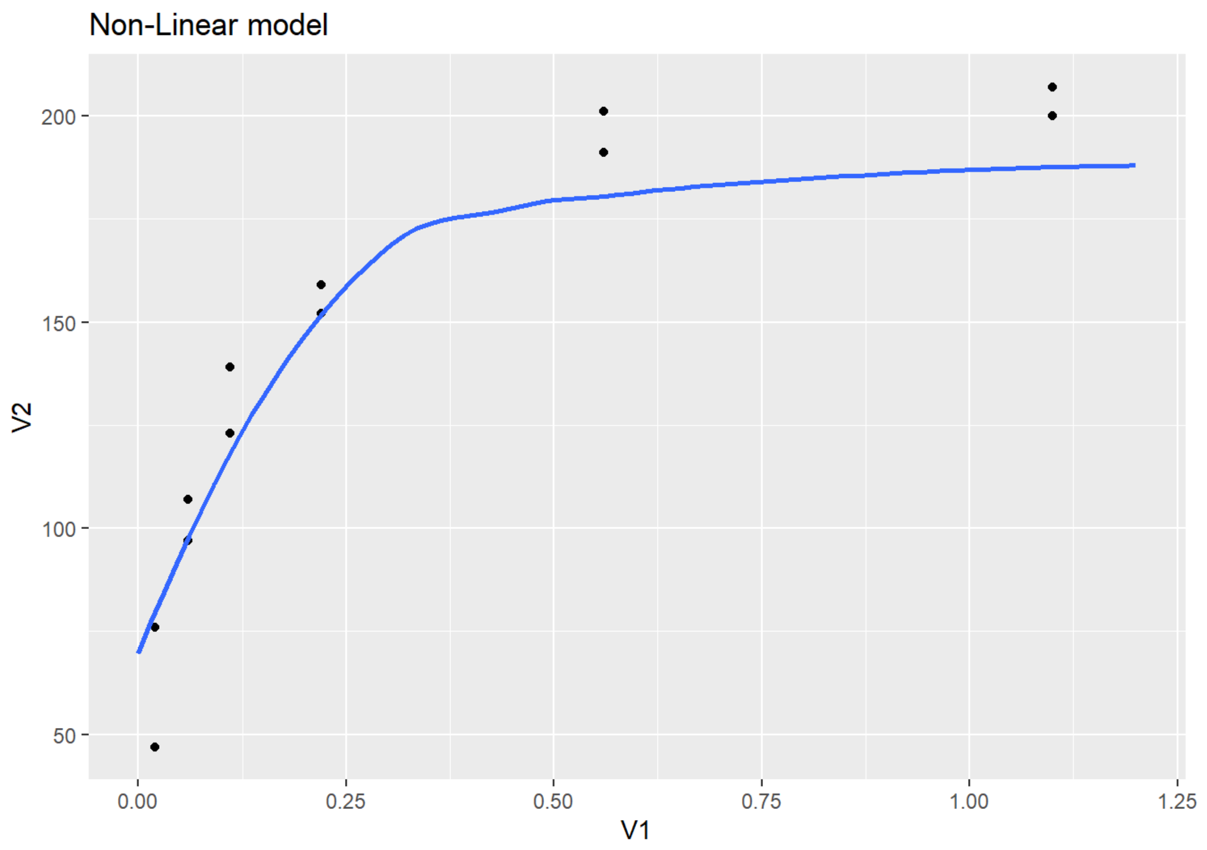


```
puro %>%
  ggplot(aes(x = 1/V1,y = 1/V2)) +
  geom_point()+
  geom_abline(slope = theta2/theta1,intercept = 1/theta1)+
  ggtitle("Linear model")
```



```
x_range <- seq(0,1.2,0.01)
y_pred <- y_hat(x_range,theta1,theta2)

puro %>%
  ggplot(aes(x = V1,y = V2)) +
  geom_point()+
  geom_smooth(aes(x = x_range, y = y_pred),data = tibble("x_range" = x_range,"y_pred" = y_pred),se = F)+
  ggtitle("Non-Linear model")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



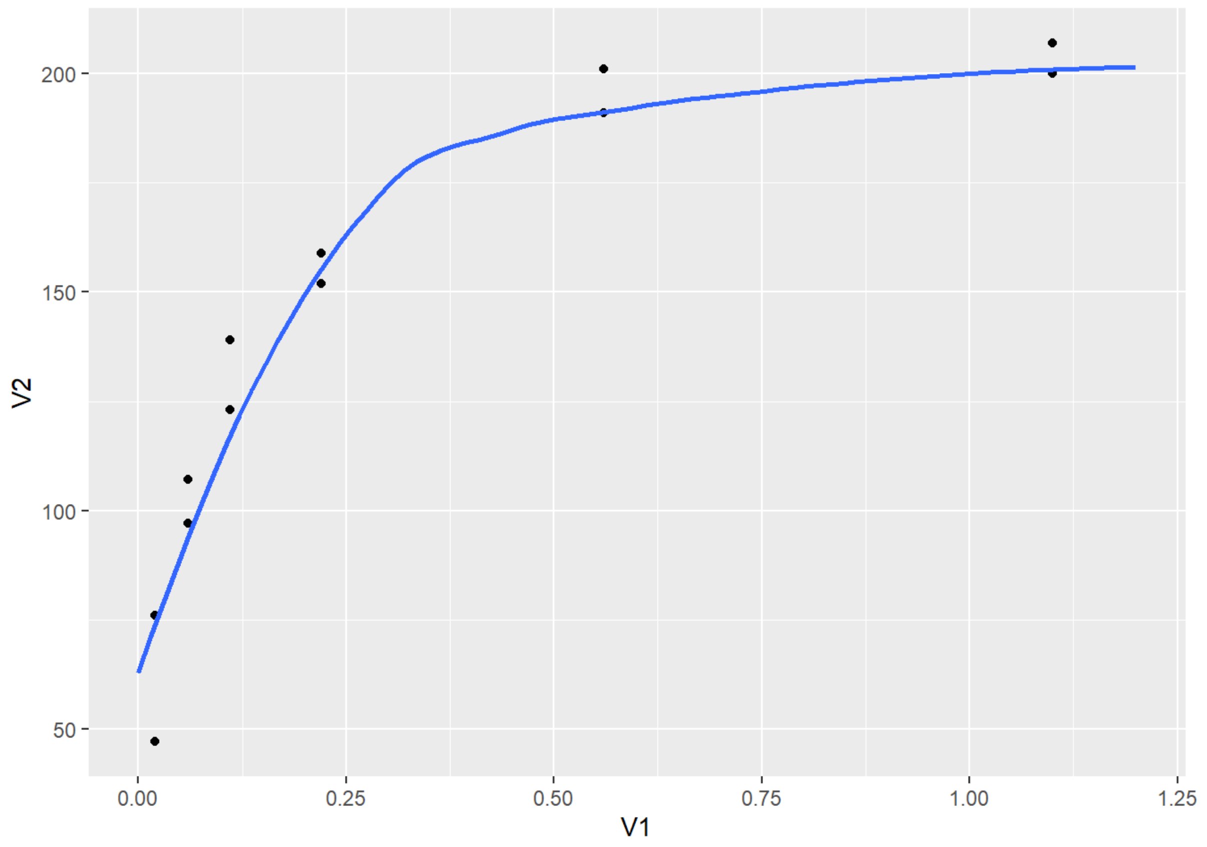
looks like the model is a very good fit to the linear model, but when we convert back to the nonlinear model we can see that the fit is not as good

3

$$\frac{\partial y}{\partial \theta_1} = \frac{x}{\theta_2 + x}$$
$$\frac{\partial y}{\partial \theta_2} = -\frac{\theta_1 x}{(\theta_2 + x)^2}$$
$$D = \begin{bmatrix} \frac{x_1}{\theta_2+x_1} & -\frac{\theta_1 x_1}{(\theta_2+x_1)^2} \\ \vdots & \vdots \\ \frac{x_n}{\theta_2+x_n} & -\frac{\theta_1 x_n}{(\theta_2+x_n)^2} \end{bmatrix}$$

```
#calc gradient matrix
grad <- function(x,theta1,theta2){
  c1 <- x/(x+theta2)
  c2 <- - theta1*x/((x+theta2)^2)
  D <- cbind(c1,c2)
  return(D)
}
#init values
x <- puro$V1
y <- puro$V2

epsilon <- 0.001
cont <- T
while (cont){
  theta1_l <- theta1
  theta2_l <- theta2

  D <- grad(x,theta1_l,theta2_l)
  e <- y - y_hat(x,theta1_l,theta2_l)
  theta <- as.numeric(solve(t(D)%*%D)%*%t(D)%*%e) + c(theta1_l,theta2_l)
  theta1 <- theta[1]
  theta2 <- theta[2]

  cont <- sum((theta - c(theta1_l,theta2_l))^2) > epsilon
}

#does it fit?
x_range <- seq(0,1.2,0.01)
y_pred <- y_hat(x_range,theta1,theta2)
puro %>%
  ggplot(aes(x = V1,y = V2)) +
  geom_point()+
  geom_smooth(aes(x = x_range, y = y_pred),data = tibble("x_range" = x_range,"y_pred" = y_pred),se = F)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



the model fits the data nicly, better than the last model fit the data

4

$$\hat{\theta} \sim N(\theta, \sigma^2 (D^t D)^{-1})$$

```
rss <- mean((y-y_hat(x,theta1,theta2))^2)
sigma_hat <- sqrt(rss)
D <- grad(x,theta1,theta2)
beta_sd_hat <- sigma_hat * sqrt(solve(t(D)%*%D)[1,1])
t_stat <- (theta1-200)/beta_sd_hat
rss_200 <- mean((y-y_hat(x,200,theta2))^2)
delta_rss <- rss_200 - rss
f_stat <- delta_rss/(rss_200/(nrow(puro)-1))
glue("t-statisitc is : {round(t_stat,4)}, pvalue is {round((1-pt(t_stat,nrow(puro)))/2,3)}
     F-statisitc is : {round(f_stat,4)}, pvalue is {round((1-pf(f_stat,1,nrow(puro)-1))/2,3)}")
```

```
## t-statisitc is : 1.9997, pvalue is 0.017
## F-statisitc is : 4.9013, pvalue is 0.024
```

$$\text{in both tests we can reject } H_0 \text{ and conclude at a confidance level of } 5\% \text{ that } \theta_1 \neq 200$$

5

```
v_pred <- y_hat(0.5,theta1,theta2)
ci <- v_pred + c(-1,1)*qnorm(0.975)*beta_sd_hat
pi <- v_pred + c(-1,1)*qnorm(0.975)*sigma_hat*sqrt(as.numeric( grad(0.5,theta1,theta2) %*% solve(t(D)%*%
D) %*% t(grad(0.5,theta1,theta2))))
glue("expected velocity for concentration of 0.5 is: {round(v_pred,3)}
     ci is: ({round(ci[1],4)},{round(ci[2],4)})
     pi is: ({round(pi[1],4)},{round(pi[2],4)})")
```

```
## expected velocity for concentration of 0.5 is: 188.508
## ci is: (176.0786,200.9378)
## pi is: (180.6074,196.409)
```