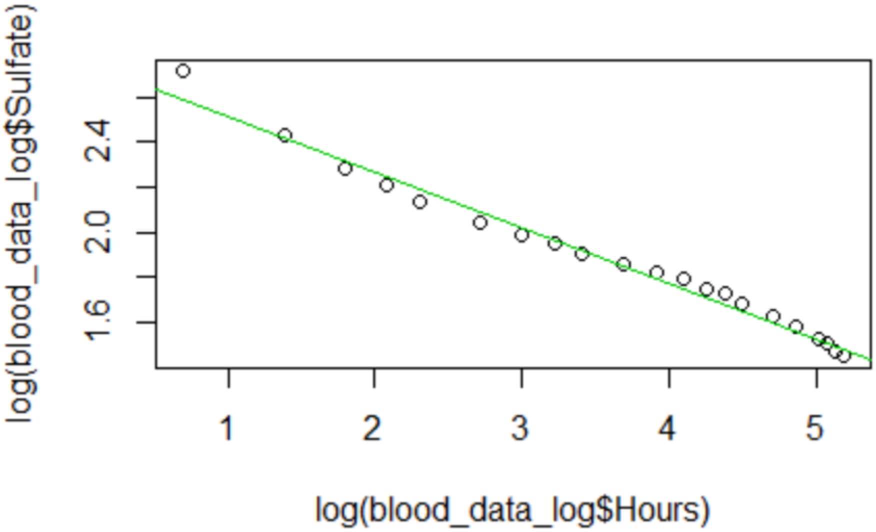
Report for Project 6- CS498 Applied Machine Learning – Spring 2020

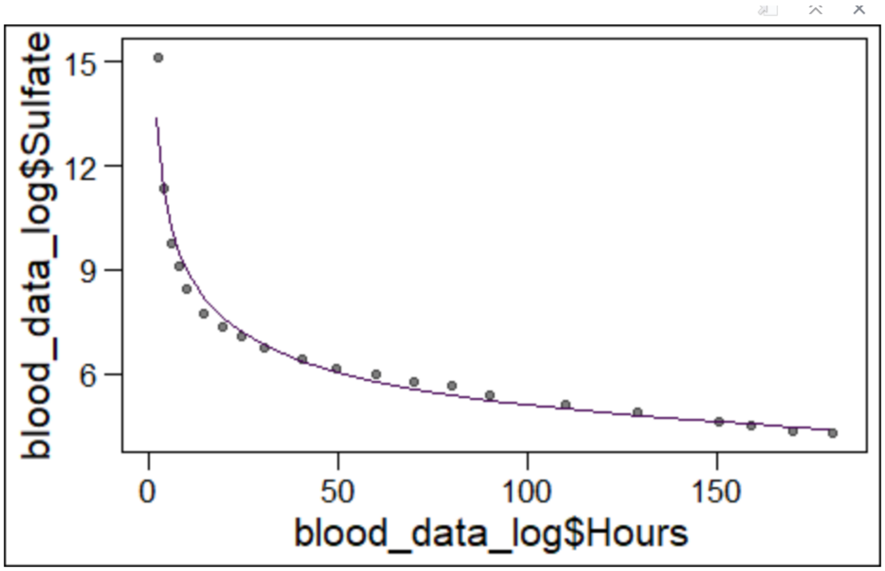
Ishita Ghosh(ighosh4)

Q10.9

a) Below is the plot for Regression line along with Data points in log-log coordinates.



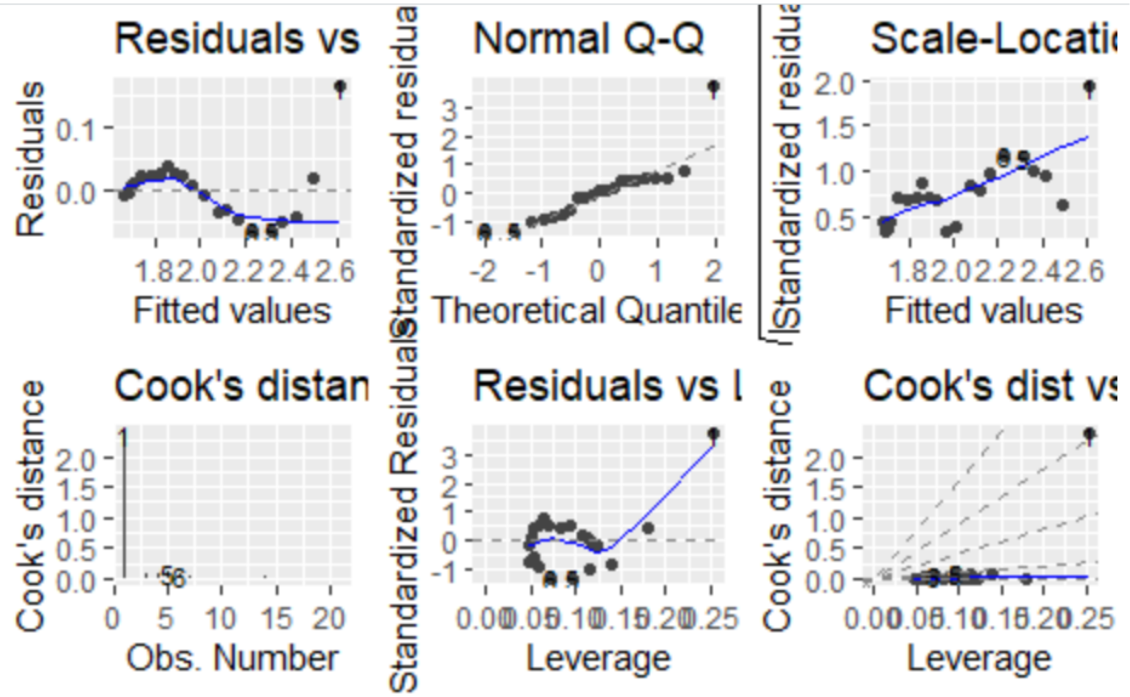
b) Below is the plot for Regression line with the data points in the original coordinates. The data file is named as “blood_data_log” hence the axis labels are in the name of “blood_data_log$Hours” and “blood_data_log$Sulfate”.



c)

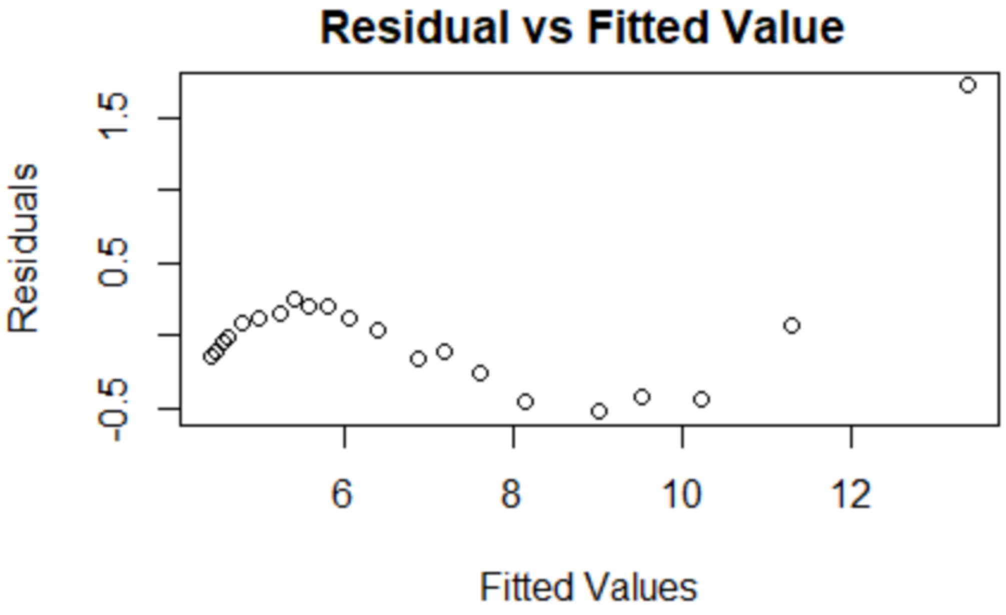
Residuals vs Fitted values in log-log coordinates.

The below plot shows the Residual vs Fitted values in log -log coordinates.



Residuals vs Fitted in the original coordinate:

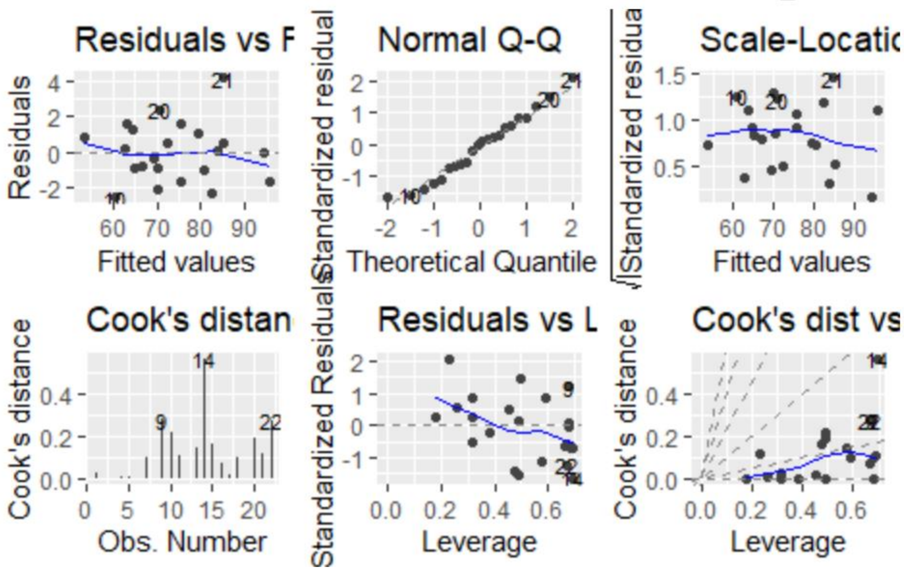
Below is the plot showing Residual vs Fitted in the original coordinate.



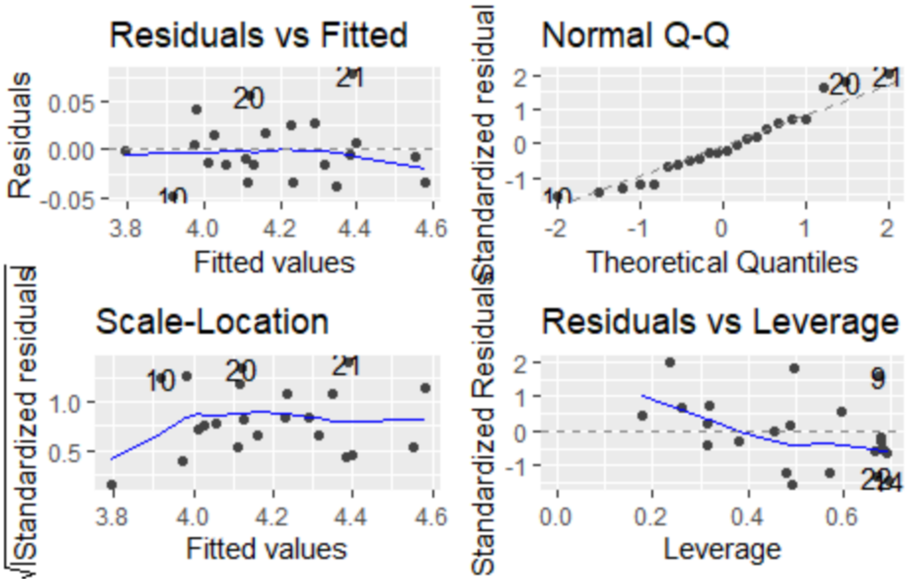
d) The linear regression with log of concentration and log of time is good. Looking at the plots from a) and b) the regression line fits the data point quite accurately, leaving the errors (distance of some of the data points from the regression line) at minimum. The log-log transformed regression model fit better as the log transformed data values are more symmetrical. Also looking at the residual vs fitted plots, the line is aligned with the points, hence this regression is good.

Q 10.10

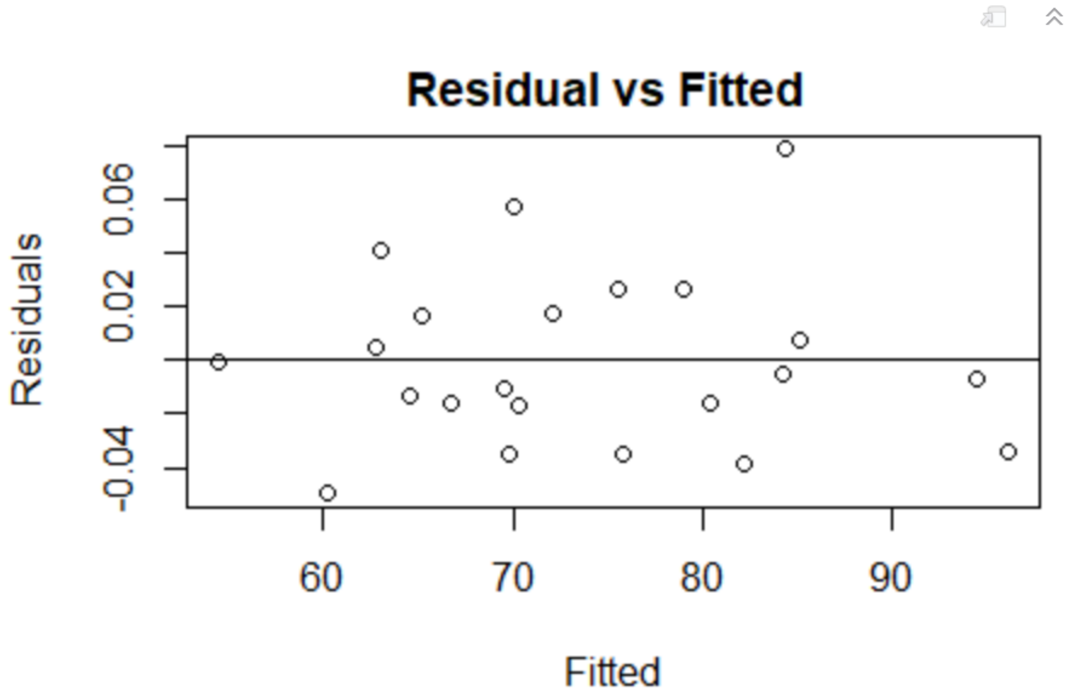
a) Below is the plot of residual vs fitted for the linear regression model, predicting the body mass from the diameters of body_data_orig$Fore,body_data_orig$Bicep,body_data_orig$Chest,body_data_orig$Neck,body_data_orig$Shoulder,body_data_orig$Waist,body_data_orig$Height,body_data_orig$Calf,body_data_orig$Thigh,body_data_orig$Head.



b) Below is the Residual vs Fitted plot for the cube root of mass model against the other diameters, and the plot is in cube root coordinates.



Below is the Residual vs Fitted plot for the cube root of mass model against the other diameters, and the plot is in original coordinates.



c) From the results and plots, the regression model with cube root of mass against the other diameters is better. Looking at the residual vs fitted plot for Linear regression model, and looking at the residual vs fitted plot for the regression model with cube root of mass against other diameters model, the residual plot for the cube root of mass against other diameters regression model is more ideal both in cube root coordinates and in original coordinates. Though these changes are minor from the linear regression model, but considering this minor changes the cube root of mass regression model is better in terms of minimizing the error.

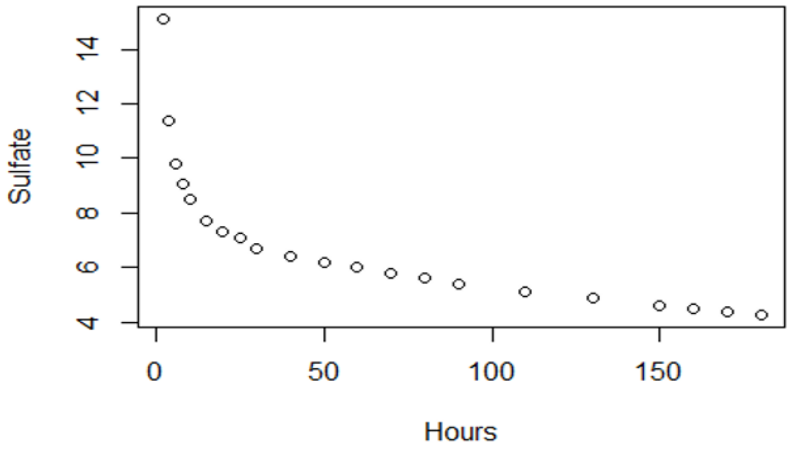
Appendix:

Q10.9 a)

Source Code:

```
require(ISLR)

## Loading required package: ISLR

require(CRAN)

## Loading required package: CRAN

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'CRAN'

library(ggfortify)

## Warning: package 'ggfortify' was built under R version 3.6.3

## Loading required package: ggplot2

  library(ggplot2)
  library(readr)
  library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
blood_data_orig= read.table("brunhild.txt",
  sep="\t", header=TRUE)
blood_data_log=data.frame(blood_data_orig)

plot(blood_data_orig)
```



```
log_y=log(blood_data_log$Sulfate)
log_x=log(blood_data_log$Hours)
log_linear_regression_model= lm(log_y~log_x, data=blood_data_log)
summary(log_linear_regression_model)

##
## Call:
## lm(formula = log_y ~ log_x, data = blood_data_log)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.059282 -0.031323 -0.001922  0.022676  0.120759
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.765837   0.027682   99.91   <2e-16 ***
## log_x       -0.247046   0.007244  -34.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04348 on 19 degrees of freedom
## Multiple R-squared:  0.9839, Adjusted R-squared:  0.9831
## F-statistic:  1163 on 1 and 19 DF,  p-value: < 2.2e-16
```

10.9 a)

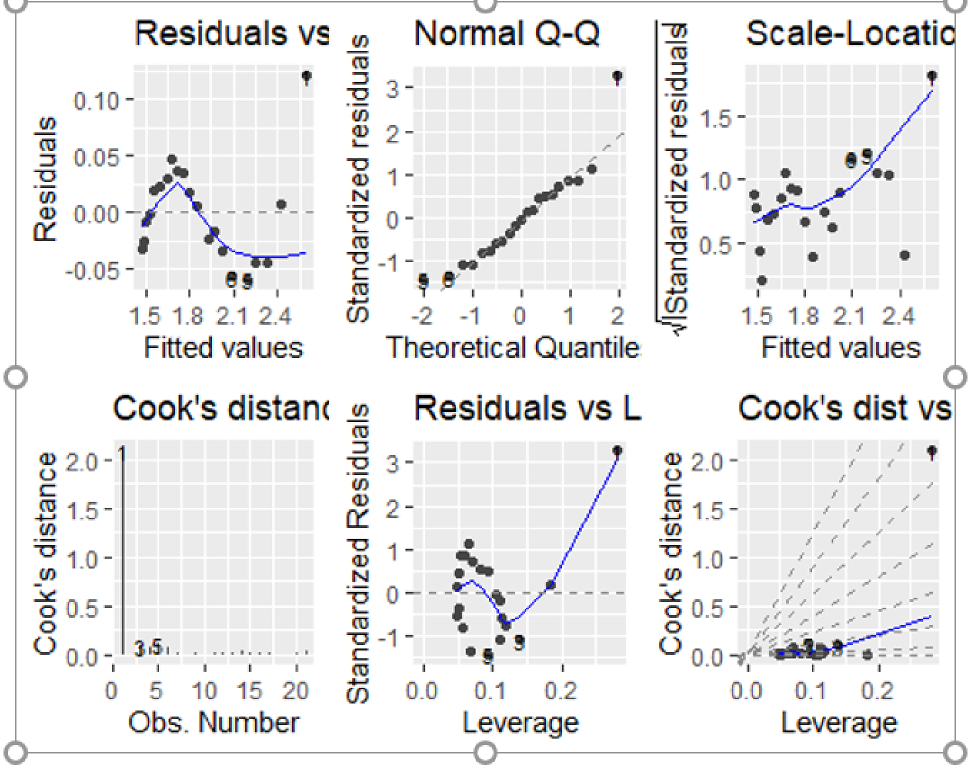
```
plot(log(blood_data_log$Sulfate)~log(blood_data_log$Hours),data=blood_data_lo
g)
abline(log_linear_regression_model, col=3)
```

10.9 b)

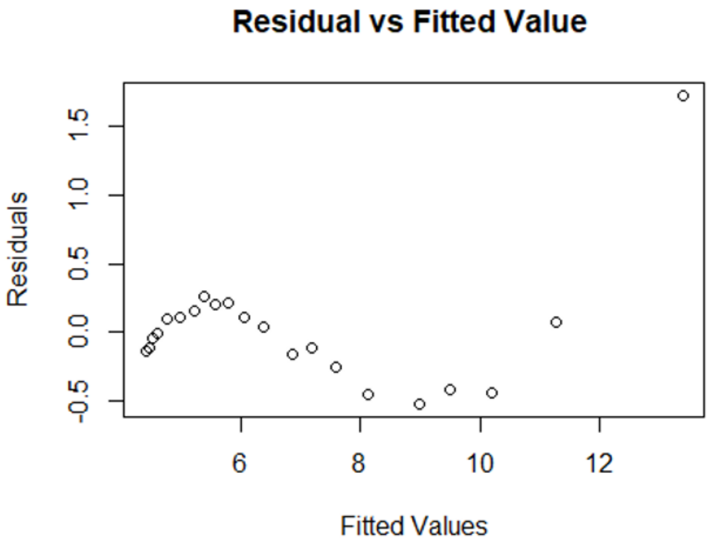
```
ggplot(blood_data_log, aes(x = blood_data_log$Hours, y =
blood_data_log$Sulfate)) +
    geom_jitter(alpha = 0.5) +
    geom_line(data = data.frame(x =
exp(log_linear_regression_model$model$log_x),
                                y =
exp(predict(log_linear_regression_model))),
              aes(x = x, y = y),
              color = viridis(1, end = 0), size = 0.7, linetype="solid")  +
    theme_base()
```

10.9 c)

```
autoplot(log_linear_regression_model, which = 1:6, ncol = 3, label.size = 3)
```

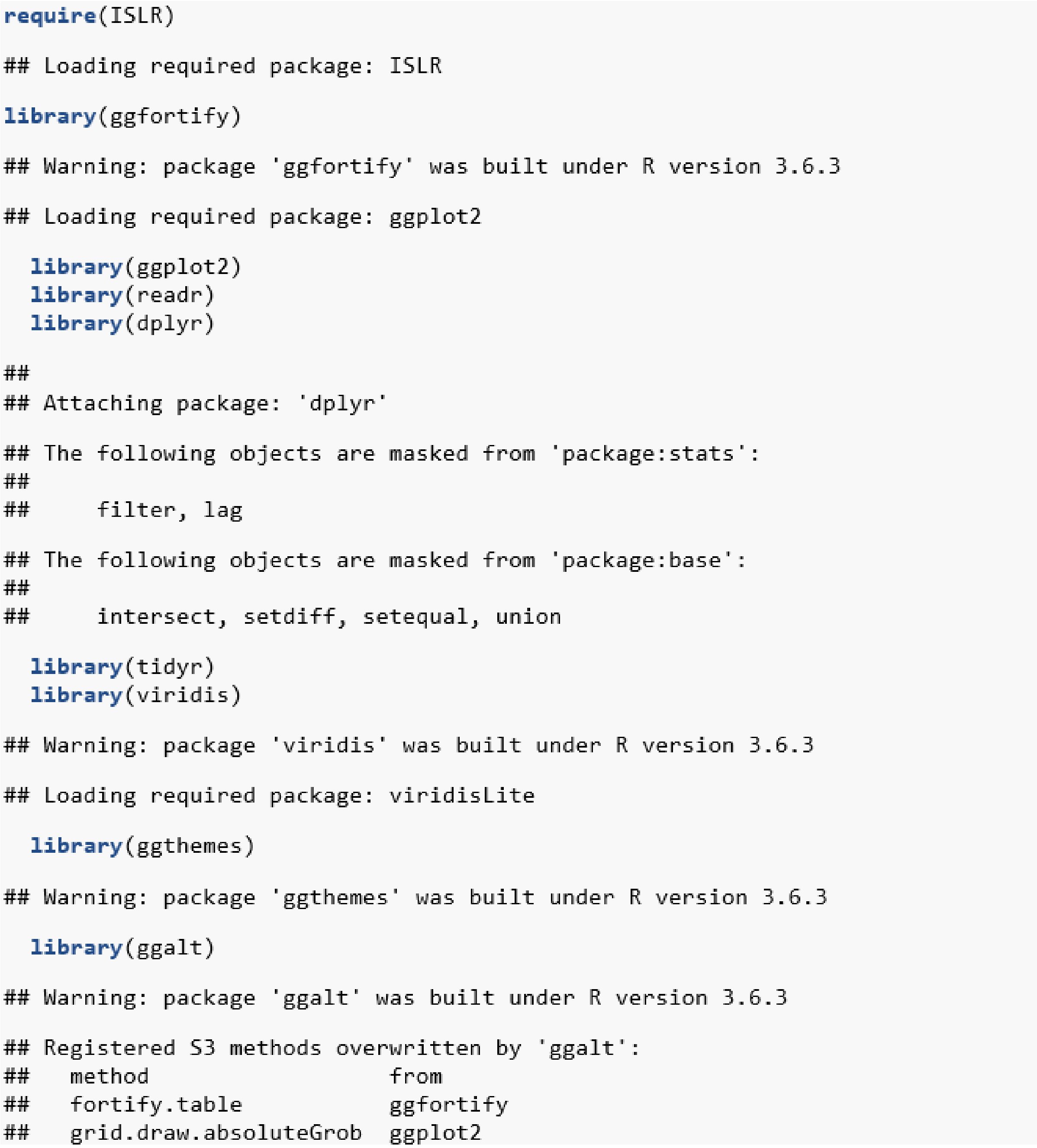


```
y_hat= exp(predict(log_linear_regression_model))
residuals_calculated= blood_data_orig$Sulfate-y_hat
plot(y_hat,residuals_calculated ,
    ylab="Residuals", xlab="Fitted Values",
    main="Residual vs Fitted Value ")
```

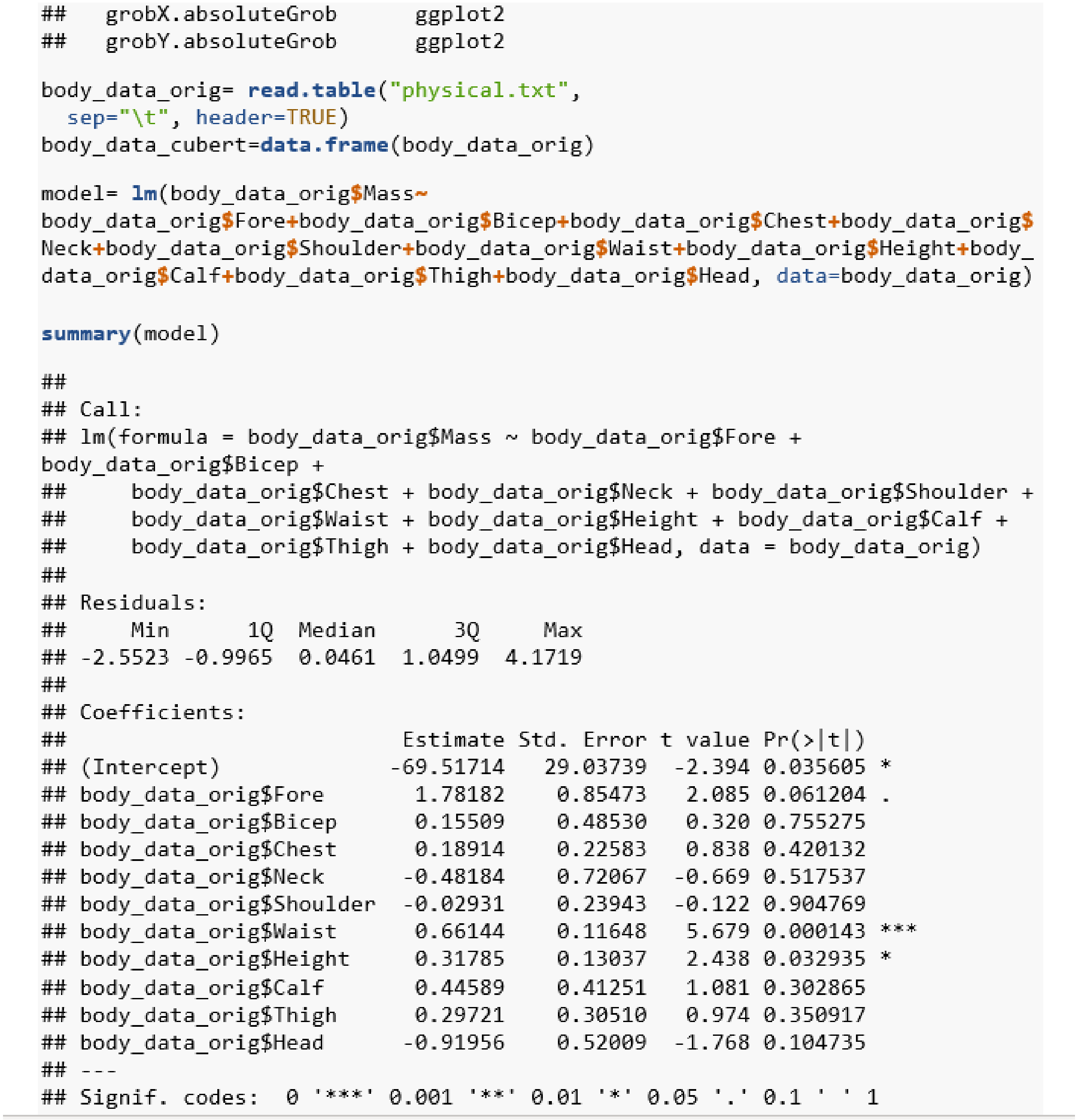


Q10.10)

a)

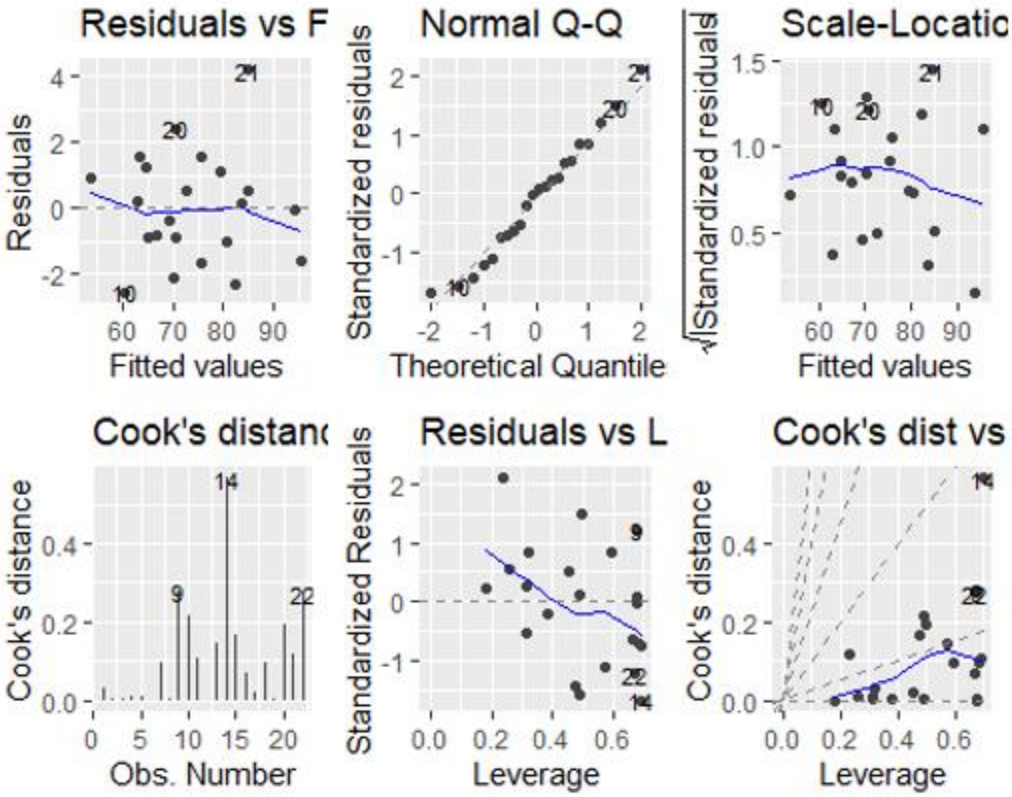


```
require(ISLR)

## Loading required package: ISLR

library(ggfortify)

## Warning: package 'ggfortify' was built under R version 3.6.3

## Loading required package: ggplot2

  library(ggplot2)
  library(readr)
  library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

  library(tidyr)
  library(viridis)

## Warning: package 'viridis' was built under R version 3.6.3

## Loading required package: viridisLite

  library(ggthemes)

## Warning: package 'ggthemes' was built under R version 3.6.3

  library(ggalt)

## Warning: package 'ggalt' was built under R version 3.6.3

## Registered S3 methods overwritten by 'ggalt':
##   method                  from
##   fortify.table           ggfortify
##   grid.draw.absoluteGrob  ggplot2
```

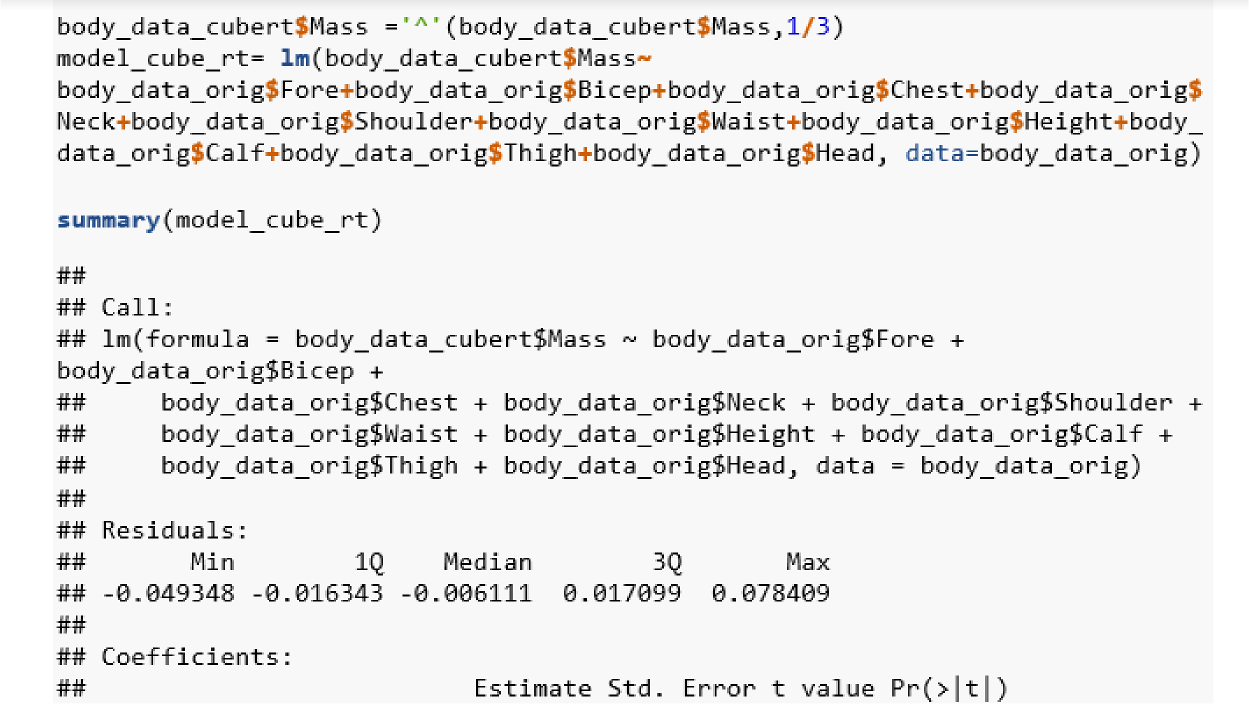


```
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2

body_data_orig= read.table("physical.txt",
  sep="\t", header=TRUE)
body_data_cubert=data.frame(body_data_orig)

model= lm(body_data_orig$Mass~
body_data_orig$Fore+body_data_orig$Bicep+body_data_orig$Chest+body_data_orig$
Neck+body_data_orig$Shoulder+body_data_orig$Waist+body_data_orig$Height+body_
data_orig$Calf+body_data_orig$Thigh+body_data_orig$Head, data=body_data_orig)

summary(model)

##
## Call:
## lm(formula = body_data_orig$Mass ~ body_data_orig$Fore +
body_data_orig$Bicep +
##     body_data_orig$Chest + body_data_orig$Neck + body_data_orig$Shoulder +
##     body_data_orig$Waist + body_data_orig$Height + body_data_orig$Calf +
##     body_data_orig$Thigh + body_data_orig$Head, data = body_data_orig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5523 -0.9965  0.0461  1.0499  4.1719
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -69.51714   29.03739  -2.394 0.035605 *
## body_data_orig$Fore       1.78182    0.85473   2.085 0.061204 .
## body_data_orig$Bicep      0.15509    0.48530   0.320 0.755275
## body_data_orig$Chest      0.18914    0.22583   0.838 0.420132
## body_data_orig$Neck      -0.48184    0.72067  -0.669 0.517537
## body_data_orig$Shoulder  -0.02931    0.23943  -0.122 0.904769
## body_data_orig$Waist      0.66144    0.11648   5.679 0.000143 ***
## body_data_orig$Height     0.31785    0.13037   2.438 0.032935 *
## body_data_orig$Calf       0.44589    0.41251   1.081 0.302865
## body_data_orig$Thigh      0.29721    0.30510   0.974 0.350917
## body_data_orig$Head      -0.91956    0.52009  -1.768 0.104735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

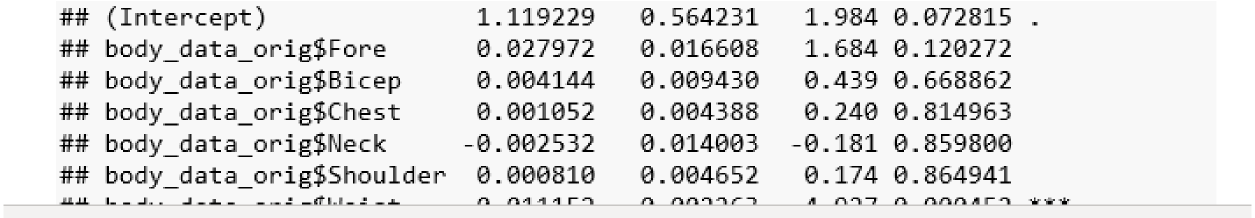
```
autoplot(model,which = 1:6, ncol = 3, label.size = 3)
```



10.10 b)

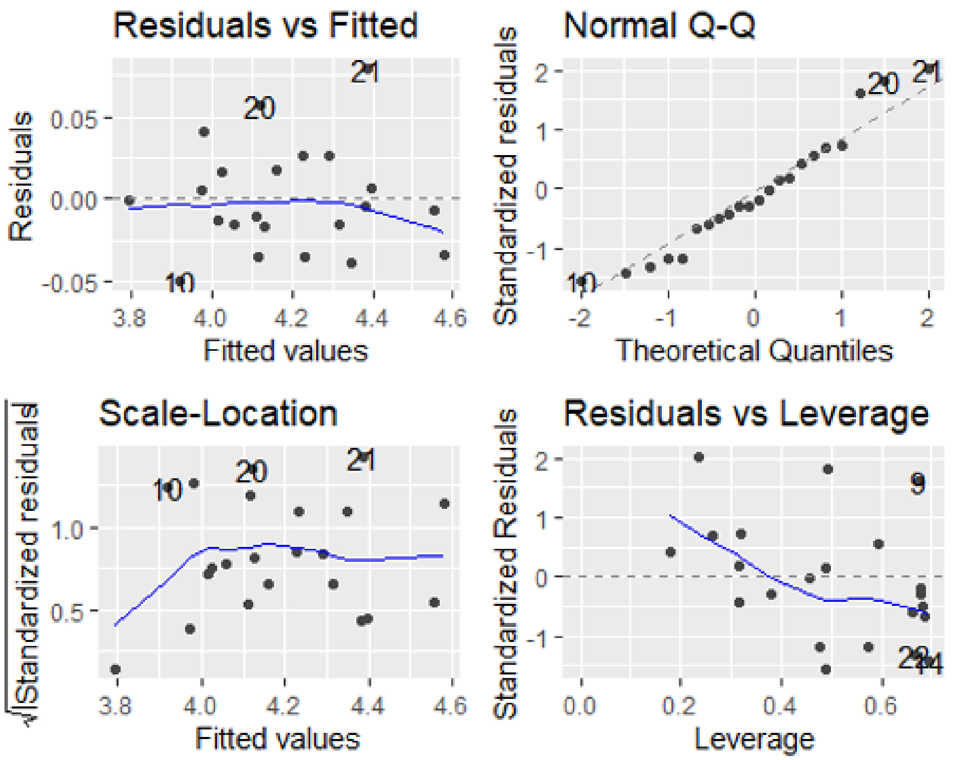


```
body_data_cubert$Mass ='^'(body_data_cubert$Mass,1/3)
model_cube_rt= lm(body_data_cubert$Mass~
body_data_orig$Fore+body_data_orig$Bicep+body_data_orig$Chest+body_data_orig$
Neck+body_data_orig$Shoulder+body_data_orig$Waist+body_data_orig$Height+body_
data_orig$Calf+body_data_orig$Thigh+body_data_orig$Head, data=body_data_orig)

summary(model_cube_rt)

##
## Call:
## lm(formula = body_data_cubert$Mass ~ body_data_orig$Fore +
body_data_orig$Bicep +
##     body_data_orig$Chest + body_data_orig$Neck + body_data_orig$Shoulder +
##     body_data_orig$Waist + body_data_orig$Height + body_data_orig$Calf +
##     body_data_orig$Thigh + body_data_orig$Head, data = body_data_orig)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.049348 -0.016343 -0.006111  0.017099  0.078409
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
```

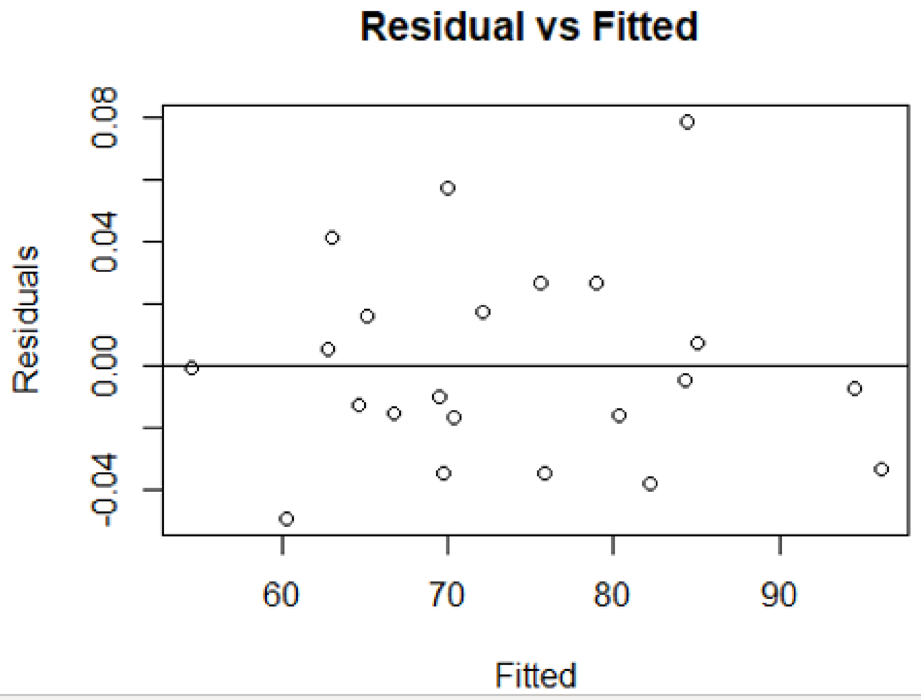


```
## (Intercept)            1.119229   0.564231   1.984 0.072815 .
## body_data_orig$Fore      0.027972   0.016608   1.684 0.120272
## body_data_orig$Bicep     0.004144   0.009430   0.439 0.668862
## body_data_orig$Chest     0.001052   0.004388   0.240 0.814963
## body_data_orig$Neck     -0.002532   0.014003  -0.181 0.859800
## body_data_orig$Shoulder  0.000810   0.004652   0.174 0.864941
## body_data_orig$Waist     0.011152   0.002263   4.927 0.000453 ***
## body_data_orig$Bicep     0.004144   0.009430   0.439 0.668862
## body_data_orig$Chest     0.001052   0.004388   0.240 0.814963
## body_data_orig$Neck     -0.002532   0.014003  -0.181 0.859800
## body_data_orig$Shoulder  0.000810   0.004652   0.174 0.864941
## body_data_orig$Waist     0.011152   0.002263   4.927 0.000452 ***
## body_data_orig$Height    0.005774   0.002533   2.279 0.043593 *
## body_data_orig$Calf      0.010656   0.008016   1.329 0.210608
## body_data_orig$Thigh     0.007919   0.005928   1.336 0.208613
## body_data_orig$Head     -0.012452   0.010106  -1.232 0.243584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04444 on 11 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9539
## F-statistic: 44.44 on 10 and 11 DF,  p-value: 1.929e-07
```

```
autoplot(model_cube_rt)
```



```r
model_cube_rt_res = resid(model_cube_rt)
fitted_cube_model='^'(fitted(model_cube_rt),3)

 plot(fitted_cube_model, residuals(model_cube_rt),
    ylab="Residuals", xlab="Fitted",
    main="Residual vs Fitted ")
abline(0, 0)
```


Residual vs Fitted
Residuals
Fitted