Multi level Project0

kallakuri radhakrishna

16/07/2020

#example 1

Data Access and changing the names of columns which we are interested in.

```
data <-  read.table("aimu.dat")
colnames(data) <- c("subject number","Year of Treatment","Age","Height","Weight","VC in ML","FEV1","FEV
#View(data)
colnames(data[,c(3,4,5,7)])
```

```
## [1] "Age"    "Height" "Weight" "FEV1"
```

#example 2 The skewness values are pretty close to zero if skewness is between -0.5 and 0.5, the distribution is approximately symmetric. As the values of kurtosis are also less than -1 we can say that variables can be considered as normal. It is prescribed to go for further testing.

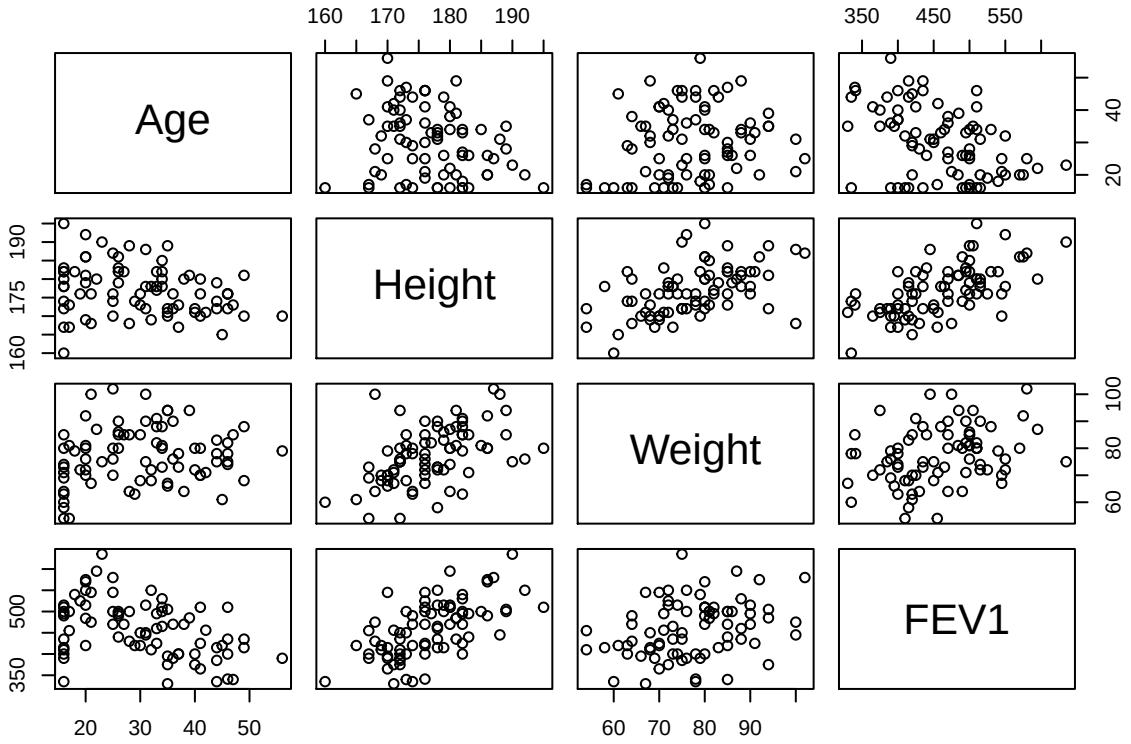
```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.6.2
```

```
describe(data[, c(3, 4, 5, 7)])
```

```
##        vars  n   mean    sd median trimmed   mad min max range skew
## Age       1 79  30.28 10.47     31   29.86 13.34  16  56    40 0.23
## Height    2 79 176.92  6.75    176  176.69  7.41 160 195    35 0.27
## Weight    3 79  77.33 10.47     78   77.23 10.38  54 102    48 0.09
## FEV1      4 79 459.70 65.93    460  459.08 66.72 330 635   305 0.13
##        kurtosis   se
## Age       -1.00 1.18
## Height    -0.18 0.76
## Weight    -0.34 1.18
## FEV1      -0.46 7.42
```

```
plot(data[, c(3, 4, 5, 7)])
```



#example 3

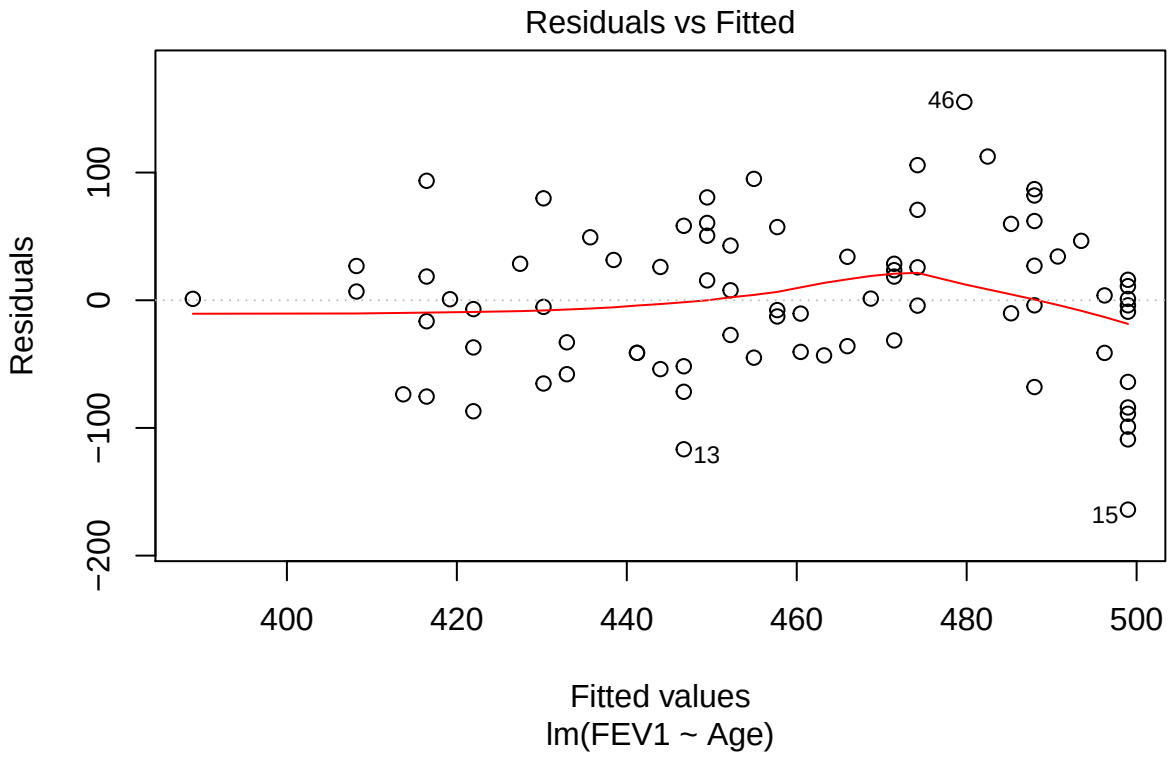
#checking for linearity Ideally, the residual plot will show no fitted pattern. That is, the red line should be approximately horizontal at zero. The presence of a pattern may indicate a problem with some aspect of the linear model.

In our example, there is no pattern in the residual plot. This suggests that we can assume linear relationship between the predictors and the outcome variables individualy, but taken together as in model4 accepting a perfectly linear relationship may not be correct.

```
model1 <- lm(FEV1 ~ Age, data = data)
model1
```

```
##
## Call:
## lm(formula = FEV1 ~ Age, data = data)
##
## Coefficients:
## (Intercept)          Age
##     543.002       -2.751
```

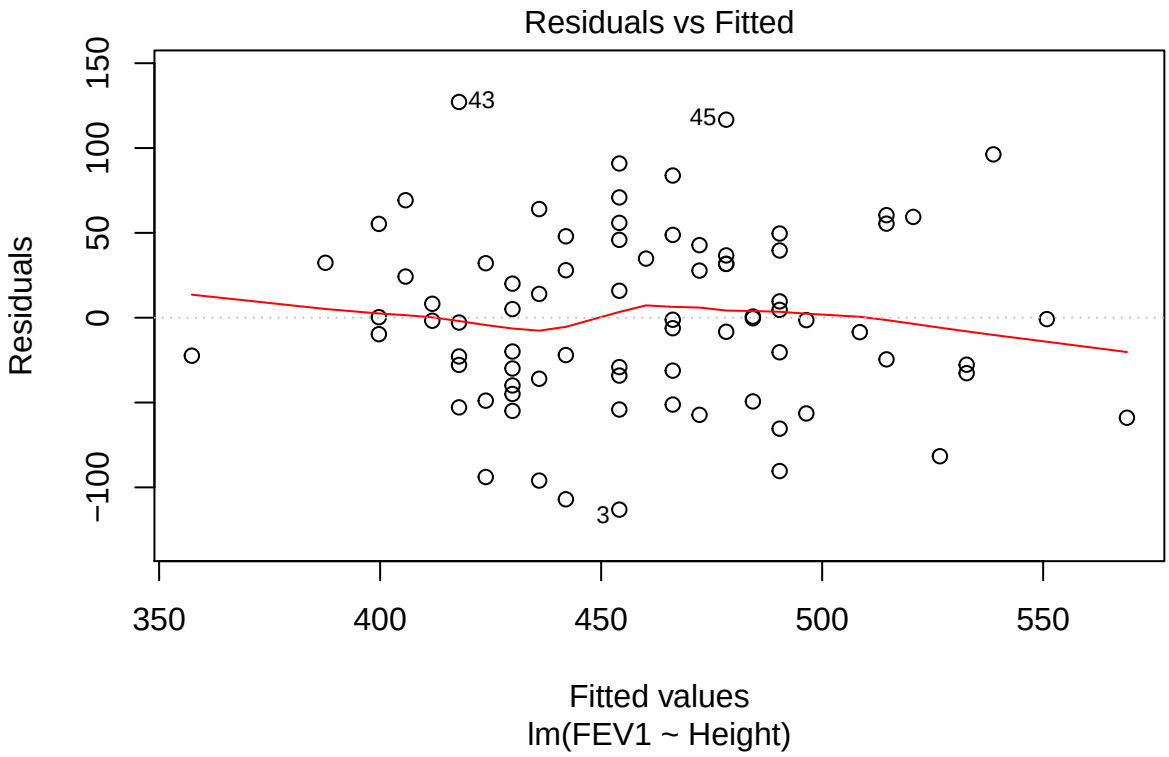
```
plot(model1, 1)
```




```
model2 <- lm(FEV1 ~ Height, data = data)
model2
```

```
##
## Call:
## lm(formula = FEV1 ~ Height, data = data)
##
## Coefficients:
## (Intercept)       Height
##    -609.715        6.044
```

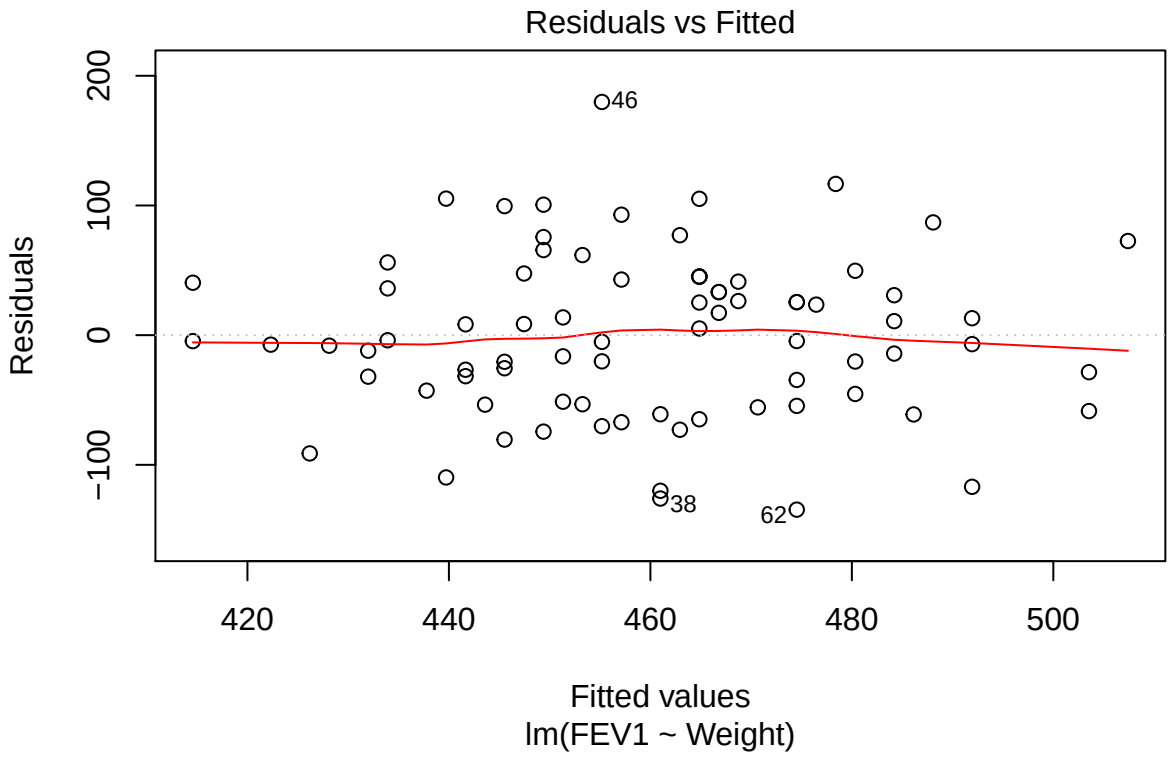
```
plot(model2, 1)
```




```
model3 <- lm(FEV1 ~ Weight, data = data)
model3
```

```
##
## Call:
## lm(formula = FEV1 ~ Weight, data = data)
##
## Coefficients:
## (Intercept)       Weight
##     310.148        1.934
```

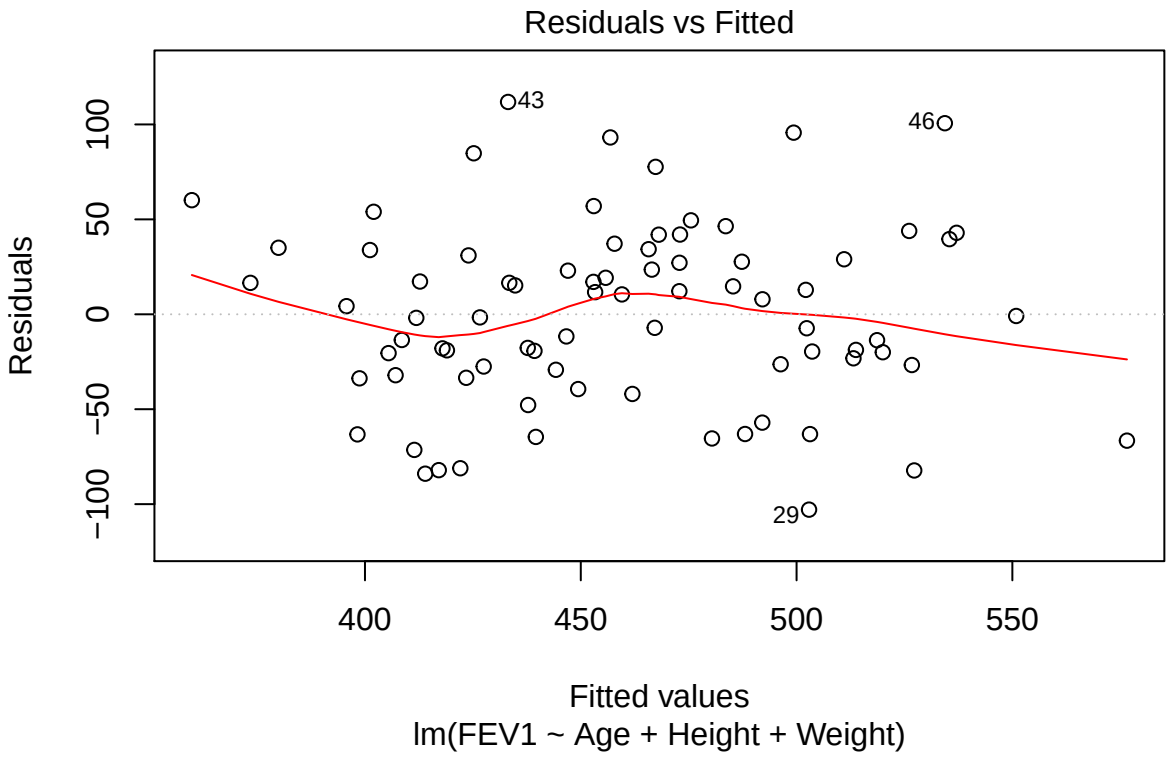
```
plot(model3, 1)
```




```
model4 <- lm(FEV1 ~ Age+Height+Weight, data = data)
model4
```

```
##
## Call:
## lm(formula = FEV1 ~ Age + Height + Weight, data = data)
##
## Coefficients:
## (Intercept)          Age       Height       Weight
##   -357.9860      -2.1517       4.6503       0.7769
```

```
plot(model4,1)
```

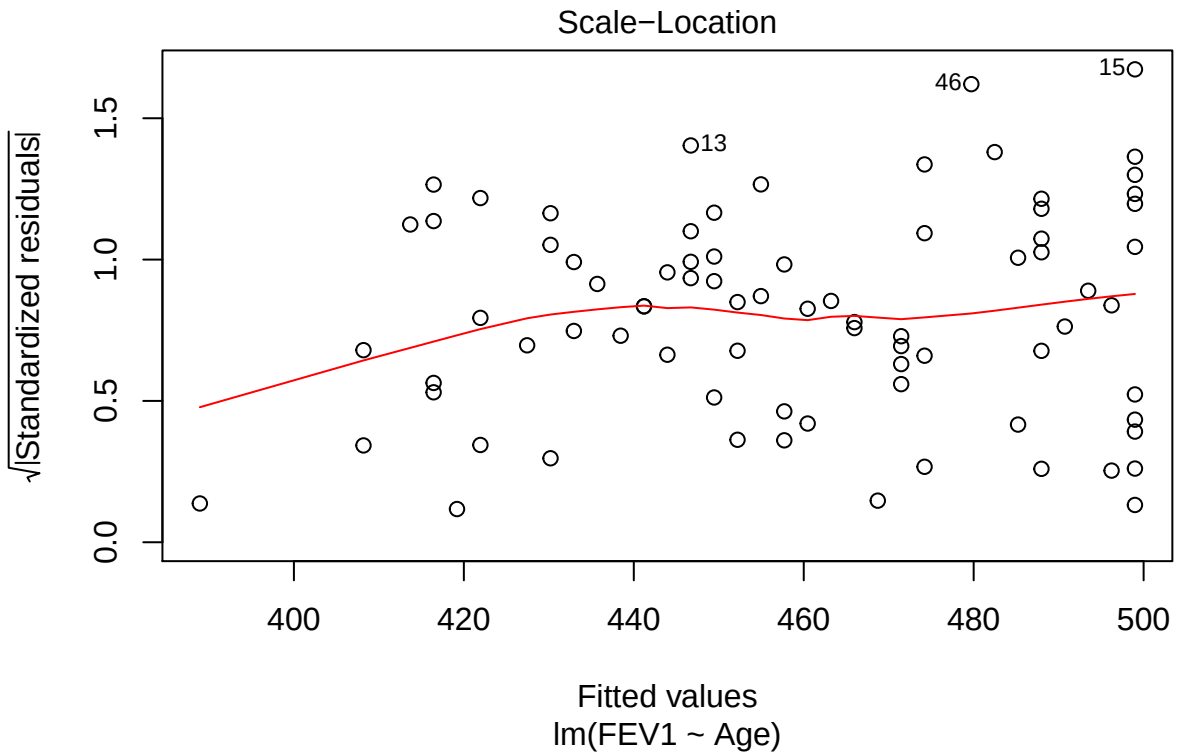


#checking for constant variance

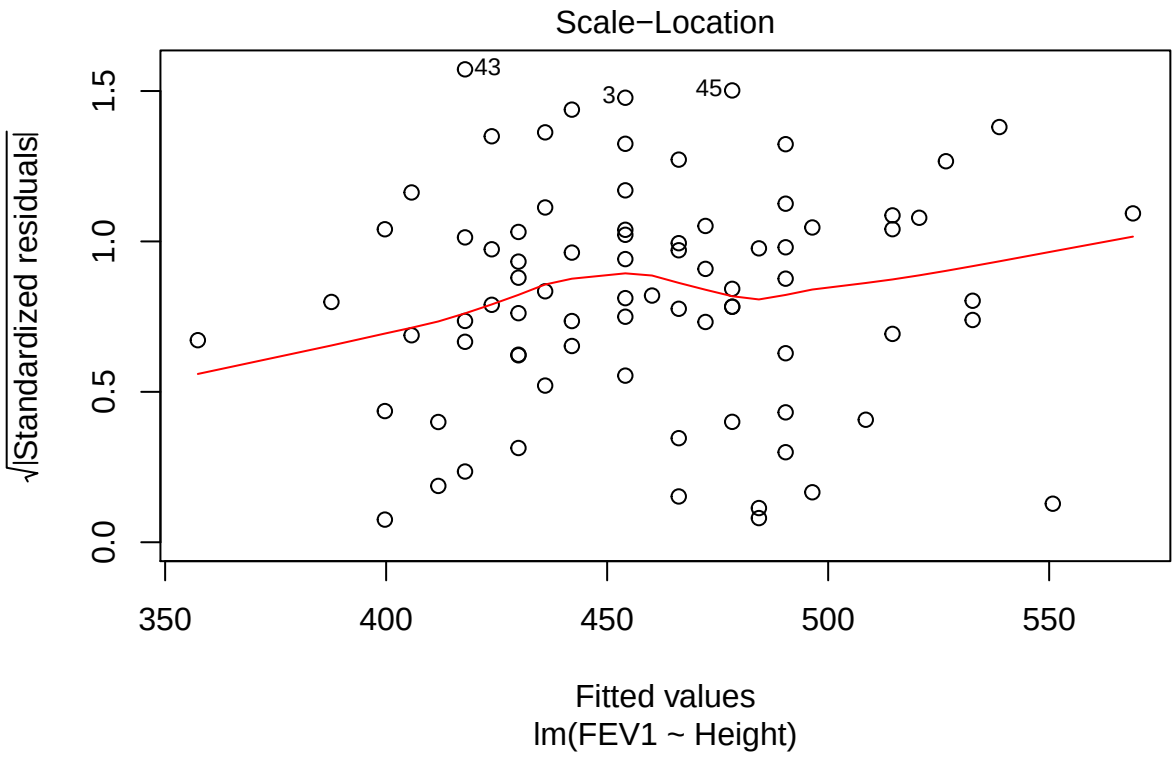
checking Homogeneity of residuals variance. The residuals are assumed to have a constant variance (homoscedasticity).This plot shows if residuals are spread equally along the ranges of predictors. It's good if you see a horizontal line with equally spread points. In our example, this is not the case. It can be seen that the variability (variances) of the residual points increases with the value of the fitted outcome variable, suggesting non-constant variances in the residuals errors (or heteroscedasticity).

A possible solution to reduce the heteroscedasticity problem is to use a log or square root transformation of the outcome variable (y).

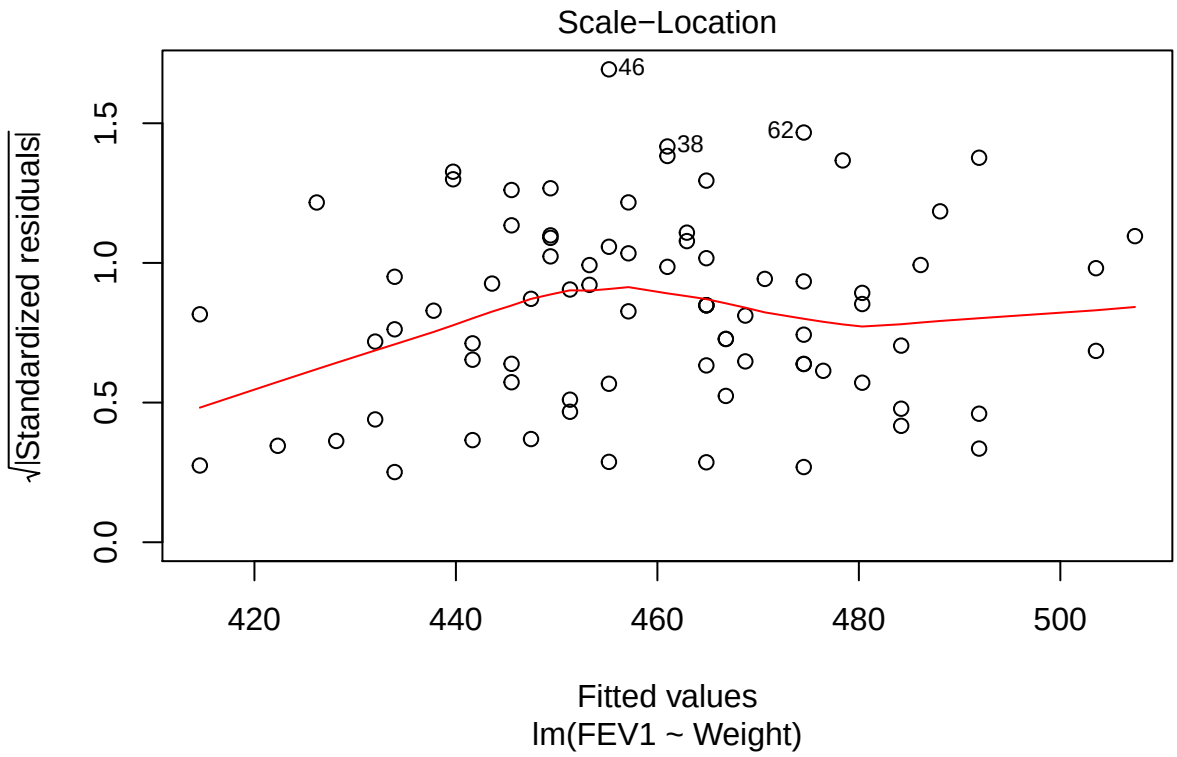
```
plot(model1, 3)
```



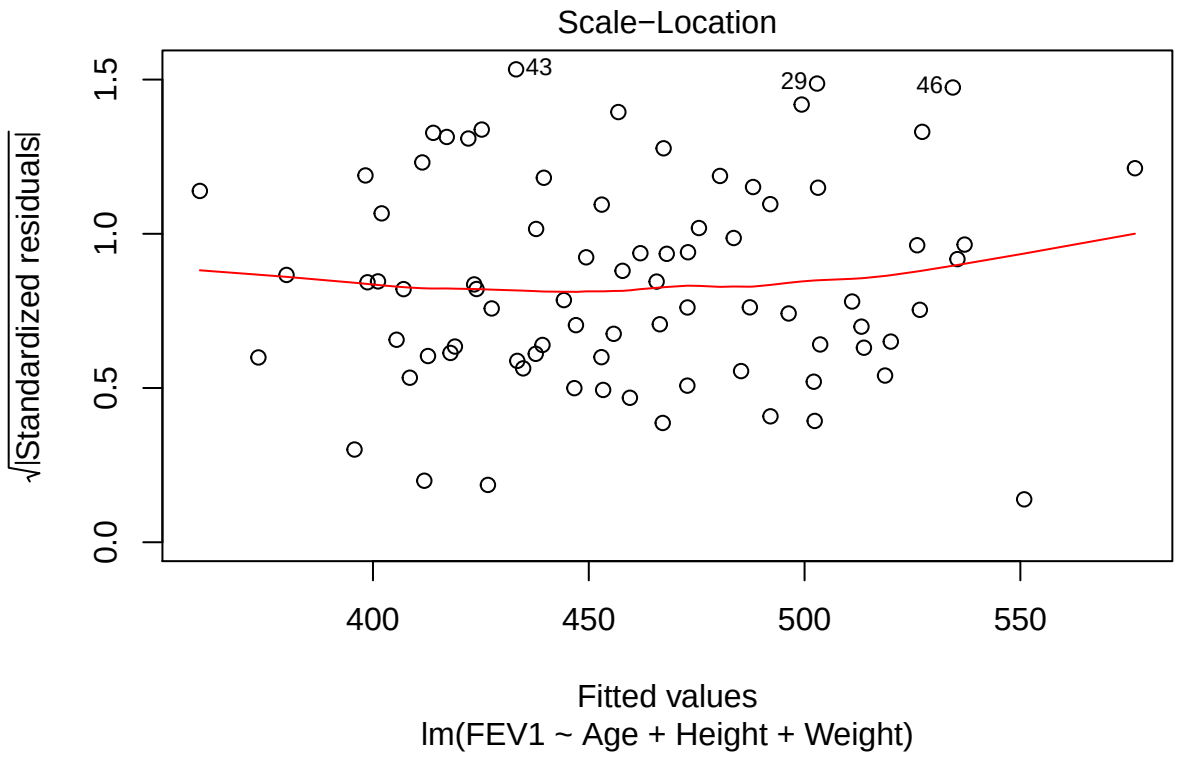

```
plot(model2, 3)
```




```
plot(model3, 3)
```

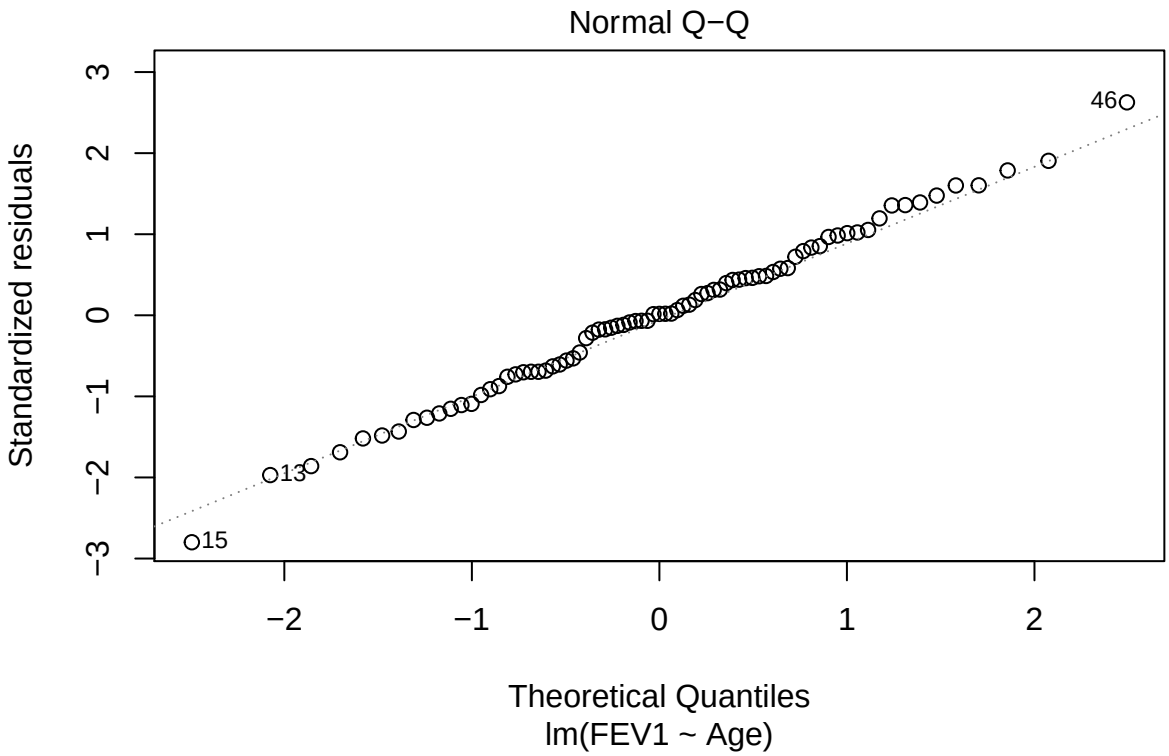



```
plot(model4,3)
```

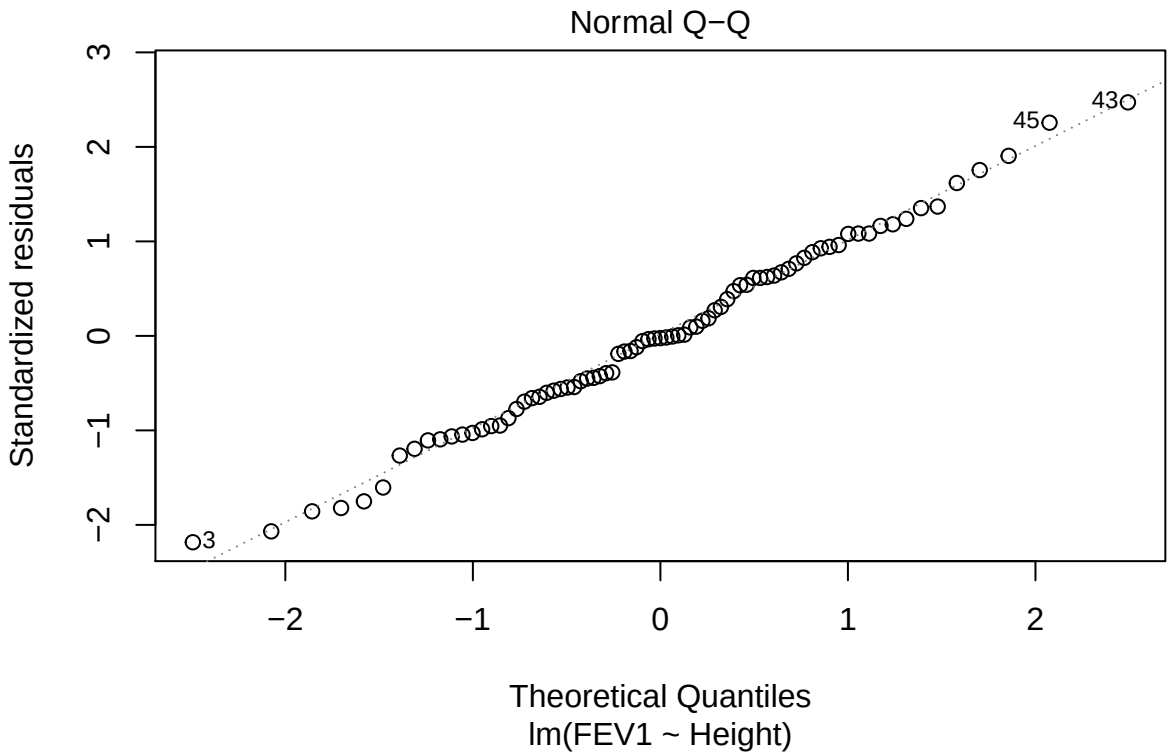


#Normality of residuals The QQ plot of residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow a straight line. In our example, all the points fall approximately along this reference line, so we can assume normality.

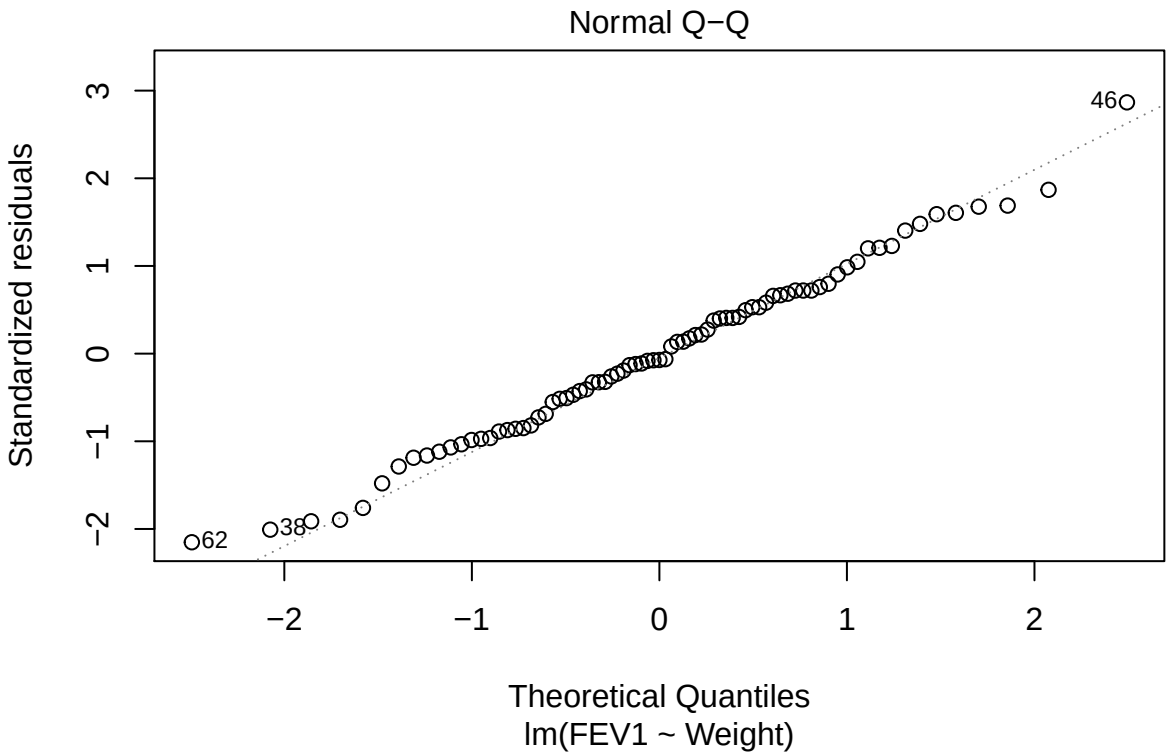
```
plot(model1, 2)
```



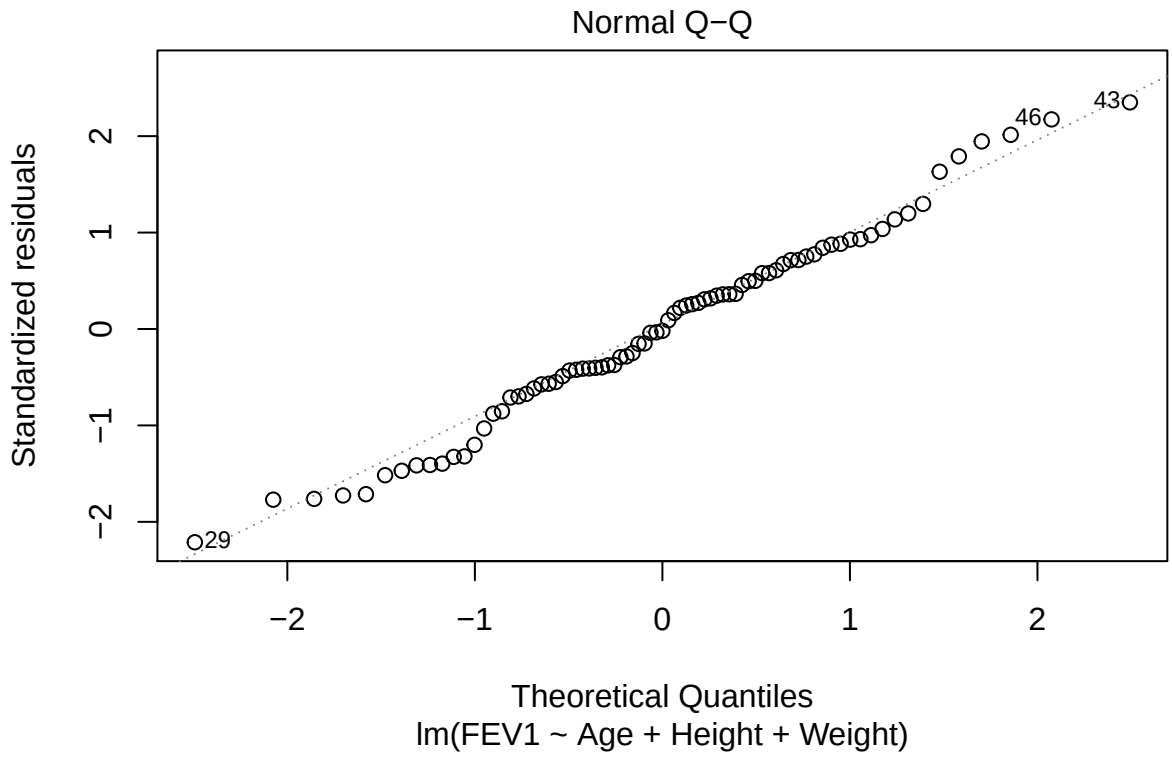

```
plot(model2, 2)
```




```
plot(model3, 2)
```




```
plot(model4,2)
```



#regression model

Weight is insignificant

```
summary(model4)
```

```
##
## Call:
## lm(formula = FEV1 ~ Age + Height + Weight, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.891  -28.369   -0.887   32.424  111.840
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -357.9860   162.1985  -2.207 0.030367 *
## Age           -2.1517     0.5684  -3.786 0.000307 ***
## Height         4.6503     1.0032   4.635 1.47e-05 ***
## Weight         0.7769     0.6351   1.223 0.225006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.41 on 75 degrees of freedom
## Multiple R-squared:  0.4817, Adjusted R-squared:  0.461
## F-statistic: 23.24 on 3 and 75 DF,  p-value: 9.713e-11
```

```
mod.log <- update(model4, .~. -Weight+log(Weight))
summary(mod.log)
```

```
##
## Call:
## lm(formula = FEV1 ~ Age + Height + log(Weight), data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -101.453  -28.732   -0.567   33.578  111.228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -564.6019   185.3609  -3.046 0.003200 **
## Age           -2.1947     0.5734  -3.828 0.000266 ***
## Height         4.5664     1.0122   4.511 2.34e-05 ***
## log(Weight)   65.1884    48.9839   1.331 0.187283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.32 on 75 degrees of freedom
## Multiple R-squared:  0.4836, Adjusted R-squared:  0.4629
## F-statistic: 23.41 on 3 and 75 DF,  p-value: 8.508e-11
```

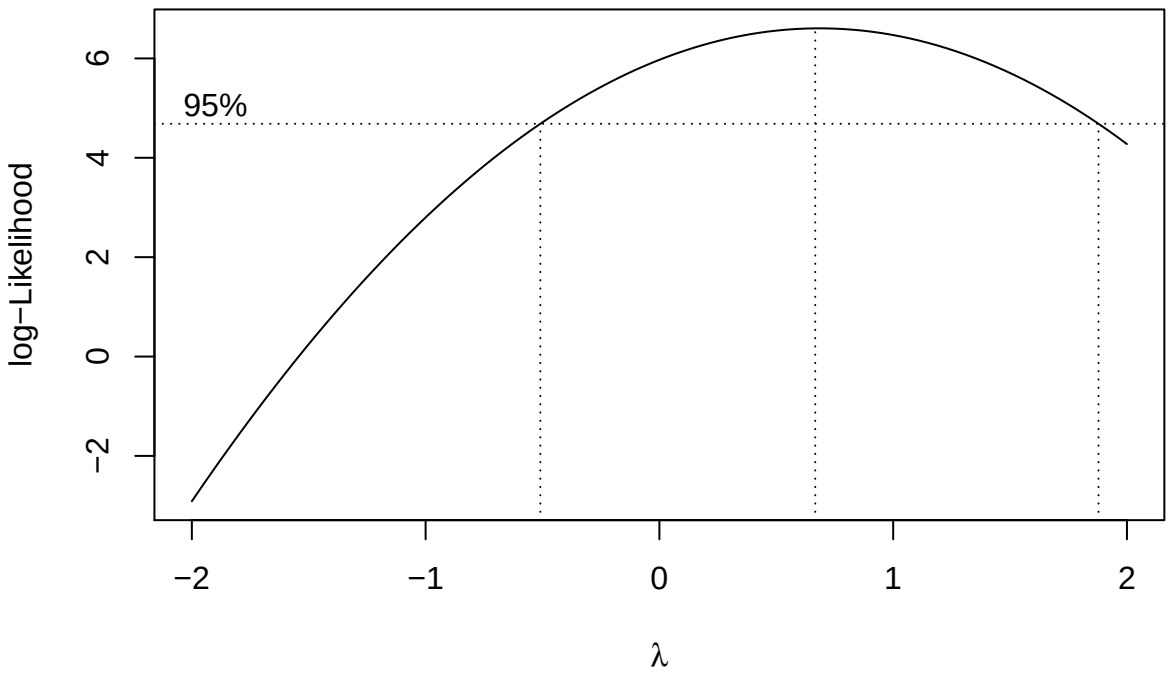
#example 4

#optimal boxcox transformation There is not much difference as compared to the earlier model. There is slight improvement in R-square. Weight is still insignificant. They are normal with homoscedacity being satisfied but the condition of linearity is not satisfied.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.2
```

```
 bc.null <- boxcox(model4)
```



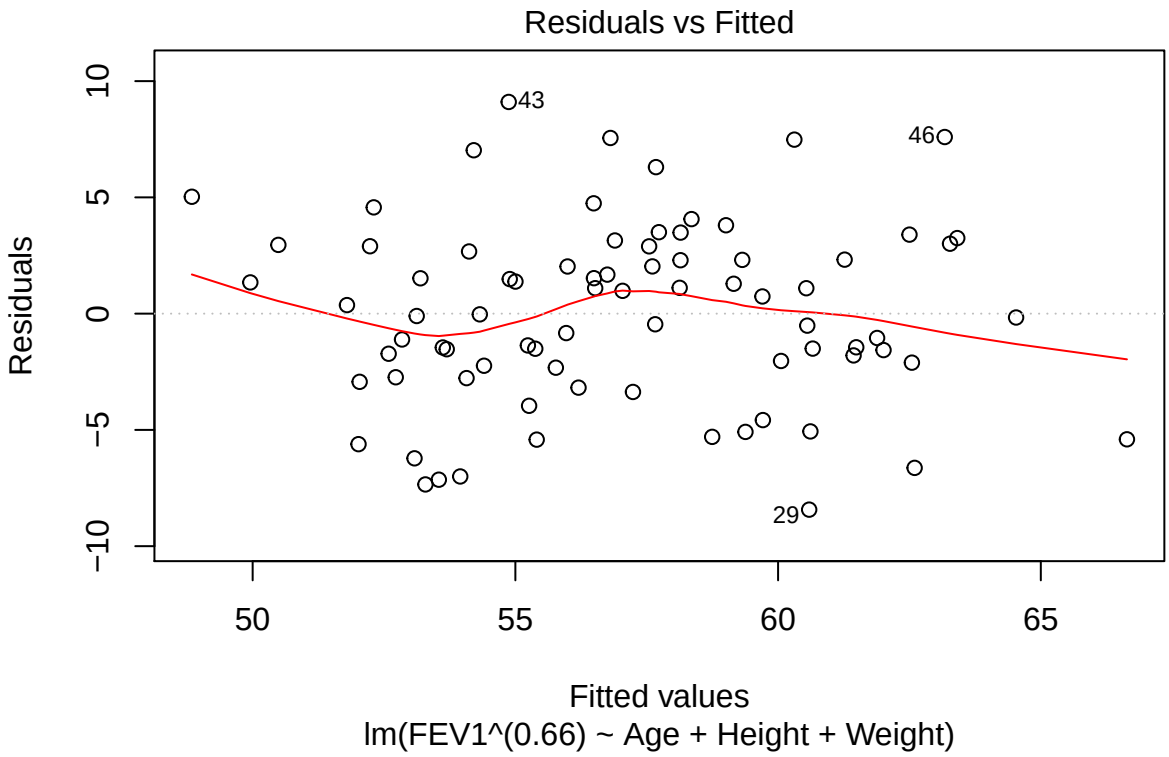

```
bc.null.opt <- bc.null$x[which.max(bc.null$y)]

paste("ML Box-Cox estimate for null model:",bc.null.opt)
```

```
## [1] "ML Box-Cox estimate for null model: 0.666666666666667"
```

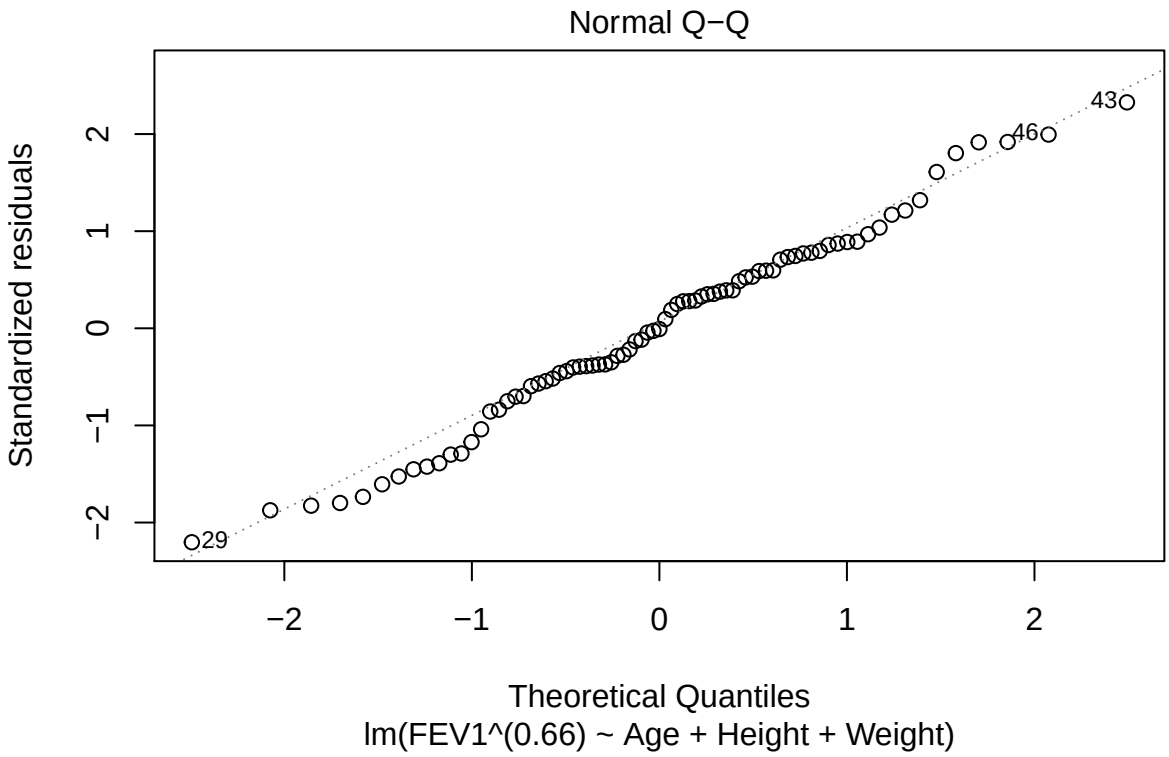
```
# lambda=0.66
bc1.mod <- lm(FEV1^(0.66) ~ Age+Height+Weight, data = data)
bc1.mod
```

```
##
## Call:
## lm(formula = FEV1^(0.66) ~ Age + Height + Weight, data = data)
##
## Coefficients:
## (Intercept)          Age       Height       Weight
##    -9.98203     -0.17730      0.38109      0.06432
```

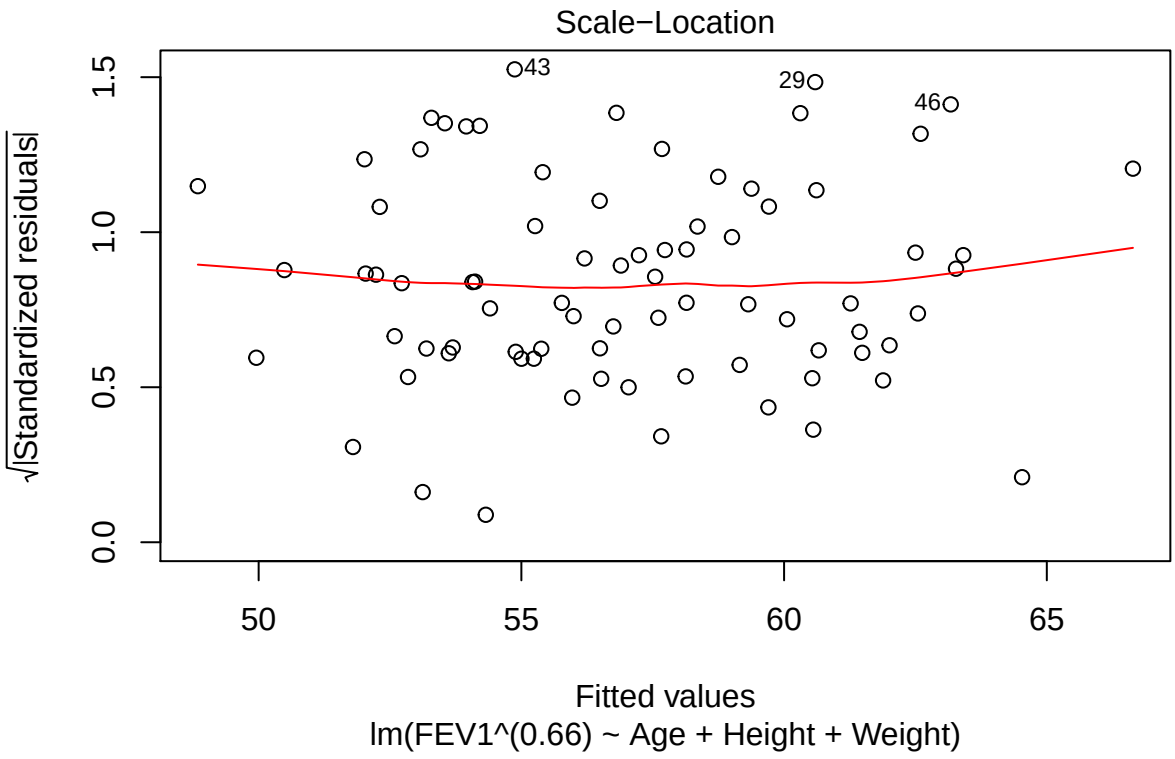
```
plot(bc1.mod,1)
```




```
plot(bc1.mod,2)
```




```
plot(bc1.mod,3)
```




```
summary(bc1.mod)
```

```
##
## Call:
## lm(formula = FEV1^(0.66) ~ Age + Height + Weight, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4289 -2.2826 -0.0306  2.7835  9.1037
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.98203   13.33931  -0.748 0.456609
## Age         -0.17730    0.04674  -3.793 0.000299 ***
## Height       0.38109    0.08251   4.619 1.57e-05 ***
## Weight       0.06432    0.05223   1.232 0.221961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.981 on 75 degrees of freedom
## Multiple R-squared:  0.4812, Adjusted R-squared:  0.4605
## F-statistic: 23.19 on 3 and 75 DF,  p-value: 1.008e-10
```

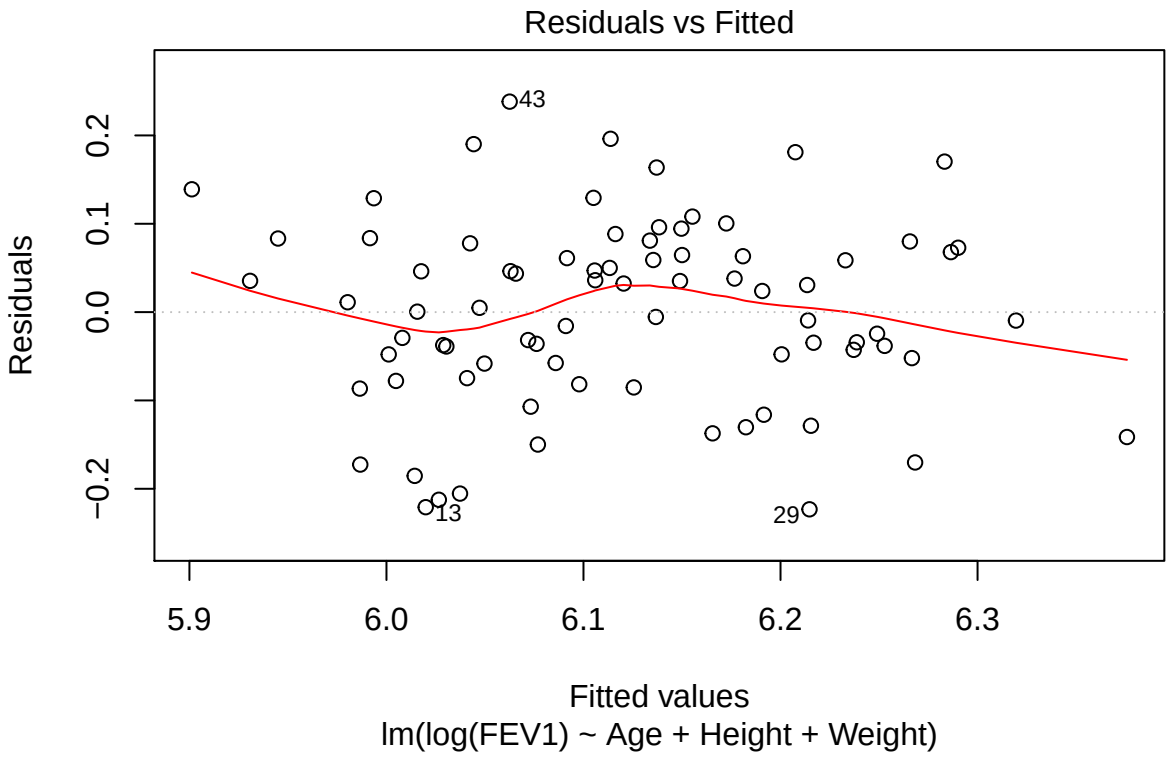
#lambda=0

Significance values are same R-square values have fallen down. the optimal box cox model is better, but intercept is significant in this model.

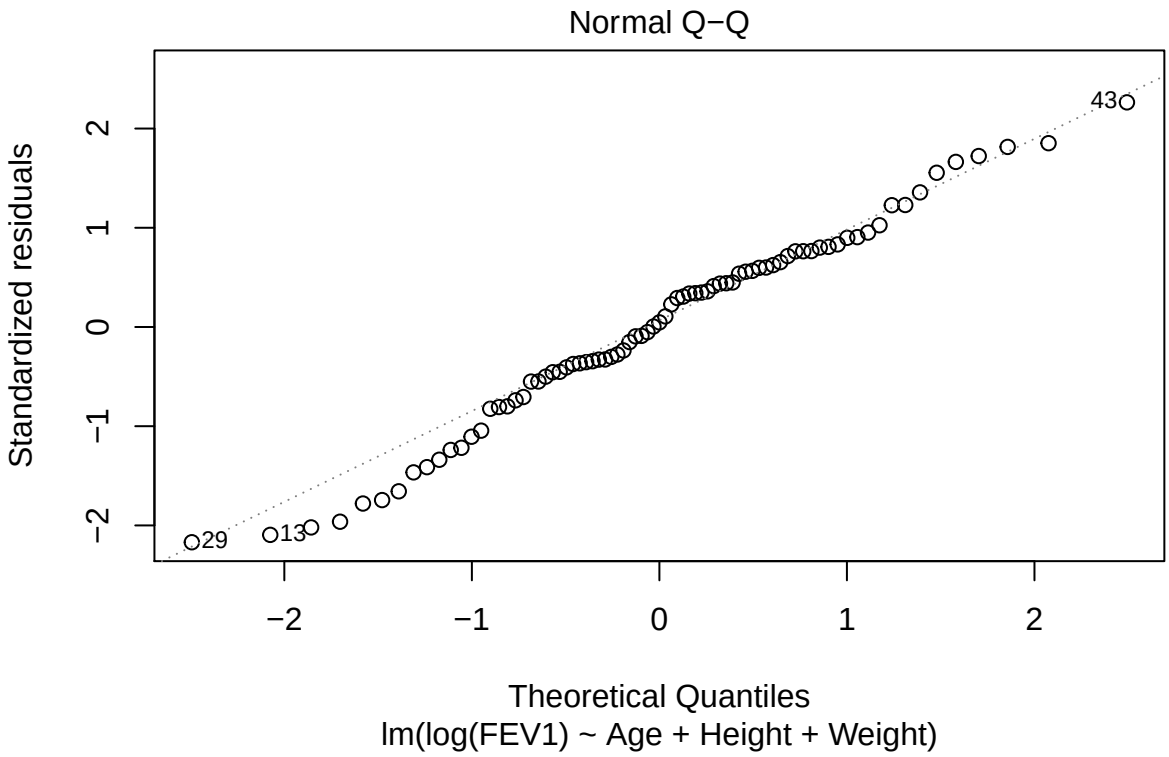
```
bc2.mod <- lm(log(FEV1) ~ Age+Height+Weight, data = data)
 bc2.mod
```

```
##
## Call:
## lm(formula = log(FEV1) ~ Age + Height + Weight, data = data)
##
## Coefficients:
## (Intercept)          Age       Height       Weight
##    4.338455    -0.004754     0.010127     0.001733
```

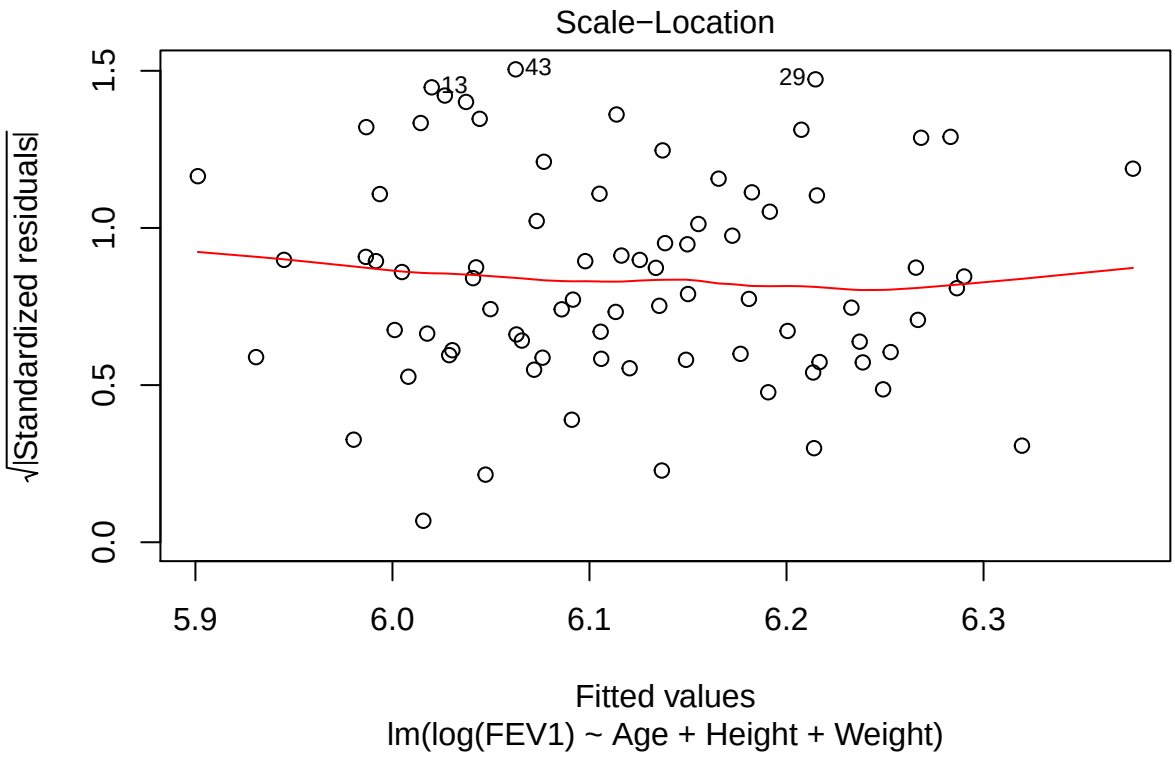
```
 plot(bc2.mod,1)
```




```
plot(bc2.mod,2)
```




```
plot(bc2.mod,3)
```




```
summary(bc2.mod)
```

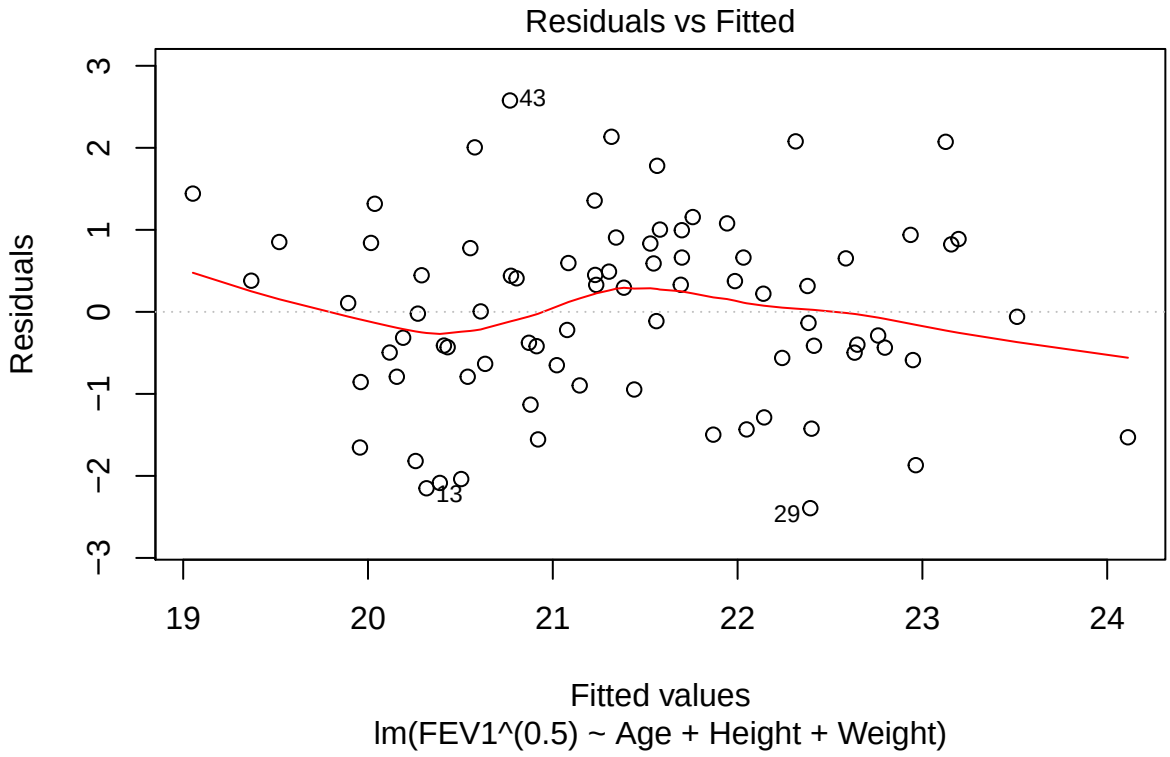
```
##
## Call:
## lm(formula = log(FEV1) ~ Age + Height + Weight, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.223245 -0.057945  0.004851  0.070380  0.238249
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.338455   0.358747  12.093  < 2e-16 ***
## Age         -0.004754   0.001257  -3.781 0.000311 ***
## Height       0.010127   0.002219   4.564 1.93e-05 ***
## Weight       0.001733   0.001405   1.234 0.221012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1071 on 75 degrees of freedom
## Multiple R-squared:  0.4772, Adjusted R-squared:  0.4563
## F-statistic: 22.82 on 3 and 75 DF,  p-value: 1.341e-10
```

#lambda=0.5

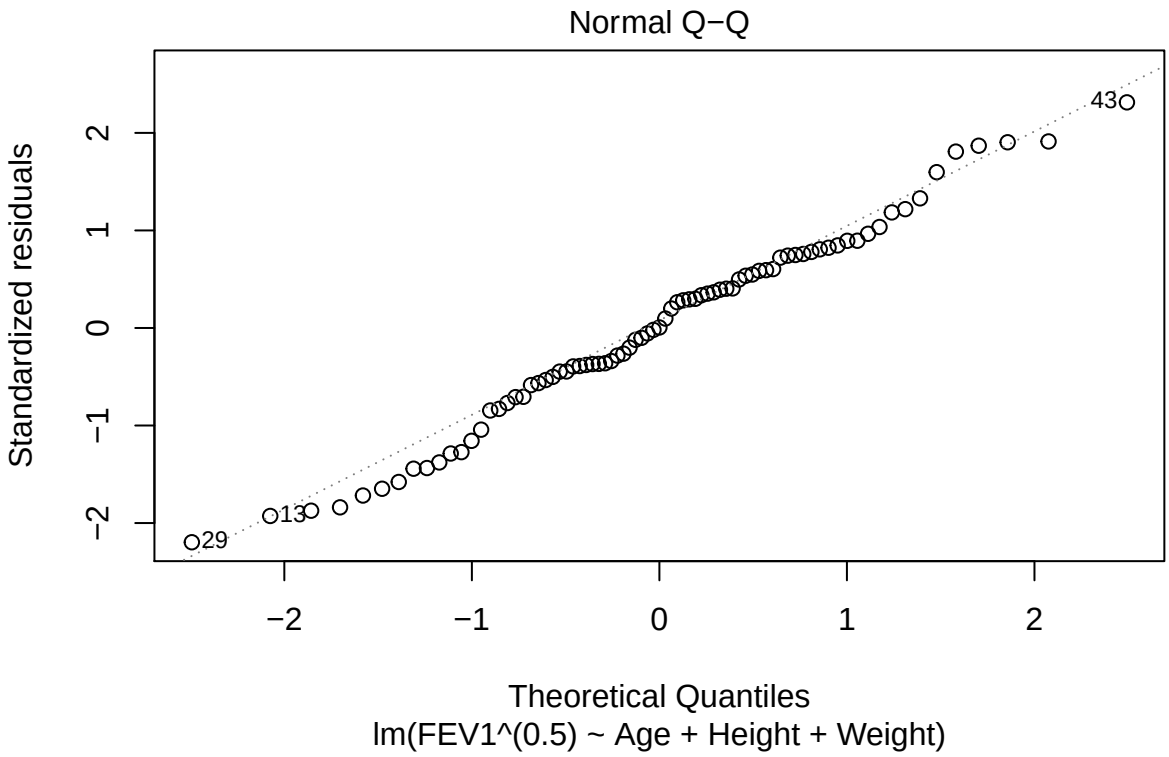
```
bc3.mod <- lm(FEV1^(0.5) ~ Age+Height+Weight, data = data)
bc3.mod
```

```
##
## Call:
## lm(formula = FEV1^(0.5) ~ Age + Height + Weight, data = data)
##
## Coefficients:
## (Intercept)          Age       Height       Weight
##     2.34361     -0.05048      0.10825      0.01835
```

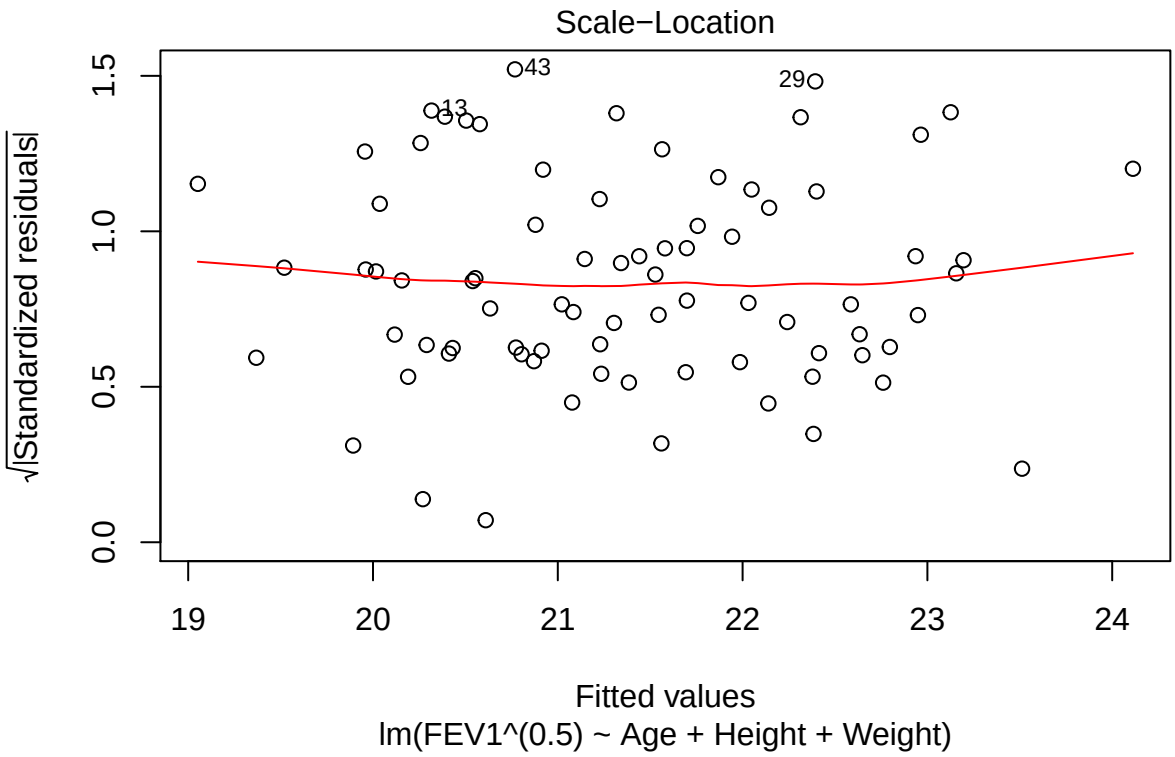
```
plot(bc3.mod,1)
```




```
plot(bc3.mod,2)
```




```
plot(bc3.mod,3)
```




```
summary(bc3.mod)
```

```
##
## Call:
## lm(formula = FEV1^(0.5) ~ Age + Height + Weight, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.39313 -0.64205  0.00558  0.79994  2.57704
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.34361    3.79776   0.617 0.539036
## Age         -0.05048    0.01331  -3.793 0.000299 ***
## Height       0.10825    0.02349   4.608 1.63e-05 ***
## Weight       0.01835    0.01487   1.234 0.221137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.133 on 75 degrees of freedom
## Multiple R-squared:  0.4806, Adjusted R-squared:  0.4598
## F-statistic: 23.13 on 3 and 75 DF,  p-value: 1.052e-10
```

#example 5 We see the word Deviance twice over in the model output. Deviance is a measure of goodness of fit of a generalized linear model. Or rather, it's a measure of badness of fit–higher numbers indicate worse fit.

R reports two forms of deviance – the null deviance and the residual deviance. The null deviance shows how

well the response variable is predicted by a model that includes only the intercept (grand mean).

By looking at deviance measurements Gamma with link log has best fit.

-Fisher Scoring What about the Fisher scoring algorithm? Fisher's scoring algorithm is a derivative of Newton's method for solving maximum likelihood problems numerically. This doesn't really tell you a lot that you need to know, other than the fact that the model did indeed converge, and had no trouble doing it.

- Information Criteria The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models. It's based on the Deviance, but penalizes you for making the model more complicated. Much like adjusted R-squared, it's intent is to prevent you from including irrelevant predictors.

However, unlike adjusted R-squared, the number itself is not meaningful. If you have more than one similar candidate models (where all of the variables of the simpler model occur in the more complex models), then you should select the model that has the smallest AIC.

So it's useful for comparing models, but isn't interpretable on its own. Almost all the three models have sam AIC.

```
m1 <- glm(FEV1 ~ Age+ Height+ Weight,data=data,family=gaussian(link="log") )
summary(m1)
```

```
##
## Call:
## glm(formula = FEV1 ~ Age + Height + Weight, family = gaussian(link = "log"),
##     data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -100.015   -26.933     0.019    30.546   111.505
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.455483   0.344513  12.933  < 2e-16 ***
## Age         -0.004844   0.001292  -3.749 0.000347 ***
## Height       0.009411   0.002102   4.476 2.66e-05 ***
## Weight       0.001967   0.001354   1.453 0.150435
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2362.15)
##
##     Null deviance: 339071  on 78  degrees of freedom
## Residual deviance: 177161  on 75  degrees of freedom
## AIC: 843.71
##
## Number of Fisher Scoring iterations: 4
```

```
m2 <- glm(FEV1 ~ Age+ Height+ Weight,data=data,family=gaussian(link="identity"))
summary(m2)
```

```
##
## Call:
## glm(formula = FEV1 ~ Age + Height + Weight, family = gaussian(link = "identity"),
##     data = data)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -102.891   -28.369    -0.887    32.424   111.840
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -357.9860   162.1985  -2.207 0.030367 *
## Age           -2.1517     0.5684  -3.786 0.000307 ***
## Height         4.6503     1.0032   4.635 1.47e-05 ***
## Weight         0.7769     0.6351   1.223 0.225006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2343.125)
##
##     Null deviance: 339071  on 78  degrees of freedom
## Residual deviance: 175734  on 75  degrees of freedom
## AIC: 843.07
##
## Number of Fisher Scoring iterations: 2
```

```
m3 <- glm(FEV1 ~ Age+ Height+ Weight,data=data,family=Gamma(link="log") )
summary(m3)
```

```
##
## Call:
## glm(formula = FEV1 ~ Age + Height + Weight, family = Gamma(link = "log"),
##     data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.22107  -0.06272  -0.00152   0.06754   0.24186
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.346759   0.355837  12.216  < 2e-16 ***
## Age         -0.004722   0.001247  -3.787 0.000306 ***
## Height       0.010143   0.002201   4.609 1.63e-05 ***
## Weight       0.001646   0.001393   1.182 0.241061
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.01127725)
##
##     Null deviance: 1.62704  on 78  degrees of freedom
## Residual deviance: 0.85393  on 75  degrees of freedom
## AIC: 843.68
##
## Number of Fisher Scoring iterations: 4
```

#prediction regions

```
d <- data$Age

p1<-predict(m1,data.frame(Age=d,Height=180,Weight=80),type="response",data=data)
p1
```

```
##        1        2        3        4        5        6        7        8
## 447.3329 469.5369 438.7482 483.3849 497.6413 432.4179 451.6881 418.0001
##        9       10       11       12       13       14       15       16
## 449.5052 462.7624 438.7482 438.7482 462.7624 507.3784 507.3784 507.3784
##       17       18       19       20       21       22       23       24
## 507.3784 504.9264 507.3784 507.3784 507.3784 507.3784 507.3784 507.3784
##       25       26       27       28       29       30       31       32
## 478.7241 465.0097 474.1083 483.3849 507.3784 453.8816 483.3849 438.7482
##       33       34       35       36       37       38       39       40
## 502.4863 504.9264 440.8788 500.0579 456.0857 443.0198 467.2678 465.0097
##       41       42       43       44       45       46       47       48
## 471.8171 458.3005 485.7323 465.0097 492.8431 490.4613 462.7624 497.6413
##       49       50       51       52       53       54       55       56
## 497.6413 495.2364 460.5261 497.6413 471.8171 474.1083 462.7624 471.8171
##       57       58       59       60       61       62       63       64
## 485.7323 497.6413 497.6413 481.0489 465.0097 436.6278 443.0198 467.2678
##       65       66       67       68       69       70       71       72
## 495.2364 451.6881 443.0198 467.2678 476.4106 469.5369 458.3005 485.7323
##       73       74       75       76       77       78       79
## 485.7323 478.7241 432.4179 460.5261 483.3849 449.5052 449.5052
```

```
p2<- predict(m3,data.frame(Age=d,Height=180,Weight=80),type="response",data=data)
p2
```

```
##        1        2        3        4        5        6        7        8
## 448.5070 470.1921 440.1157 483.7032 497.6025 433.9254 452.7625 419.8177
##        9       10       11       12       13       14       15       16
## 450.6298 463.5787 440.1157 440.1157 463.5787 507.0900 507.0900 507.0900
##       17       18       19       20       21       22       23       24
## 507.0900 504.7013 507.0900 507.0900 507.0900 507.0900 507.0900 507.0900
##       25       26       27       28       29       30       31       32
## 479.1569 465.7728 474.6533 483.7032 507.0900 454.9054 483.7032 440.1157
##       33       34       35       36       37       38       39       40
## 502.3238 504.7013 442.1987 499.9576 457.0584 444.2916 467.9772 465.7728
##       41       42       43       44       45       46       47       48
## 472.4174 459.2216 485.9925 465.7728 492.9256 490.6036 463.5787 497.6025
##       49       50       51       52       53       54       55       56
## 497.6025 495.2586 461.3950 497.6025 472.4174 474.6533 463.5787 472.4174
##       57       58       59       60       61       62       63       64
## 485.9925 497.6025 497.6025 481.4247 465.7728 438.0425 444.2916 467.9772
##       65       66       67       68       69       70       71       72
## 495.2586 452.7625 444.2916 467.9772 476.8998 470.1921 459.2216 485.9925
##       73       74       75       76       77       78       79
## 485.9925 479.1569 433.9254 461.3950 483.7032 450.6298 450.6298
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
df <- data.frame(data$Age,p1,p2)
g <- ggplot(df, aes(data$Age))
g <- g + geom_line(aes(y=p1), colour="red")
g <- g + geom_line(aes(y=p2), colour="green")
g
```