Asgm3, Choi

Yeonsil Choi

07/11/2021

1

(a) Multicollinearity affects the interpretation of the regression coefficients.

TRUE because multicollinearity reduces the precision of the estimated coefficients, which weakens the statistical power of regression model. Also, it can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model. Therefore, the coefficient estimates are unstable and difficult to interpret.

(b)

False. The variance inflation factor of $\hat{\beta}_j$ depends on the R^2 that is from the regression of x_j on all other predictors.

(c) A high leverage point is always highly influential.

FALSE since an observation with a high leverage is a point that "could" have a large influence. That is, leverage measures "potential" influence.

2

```
# import the data
dataset = read.csv("https://raw.githubusercontent.com/hgweon2/data/main/hw3-data.txt")
head(dataset)
```

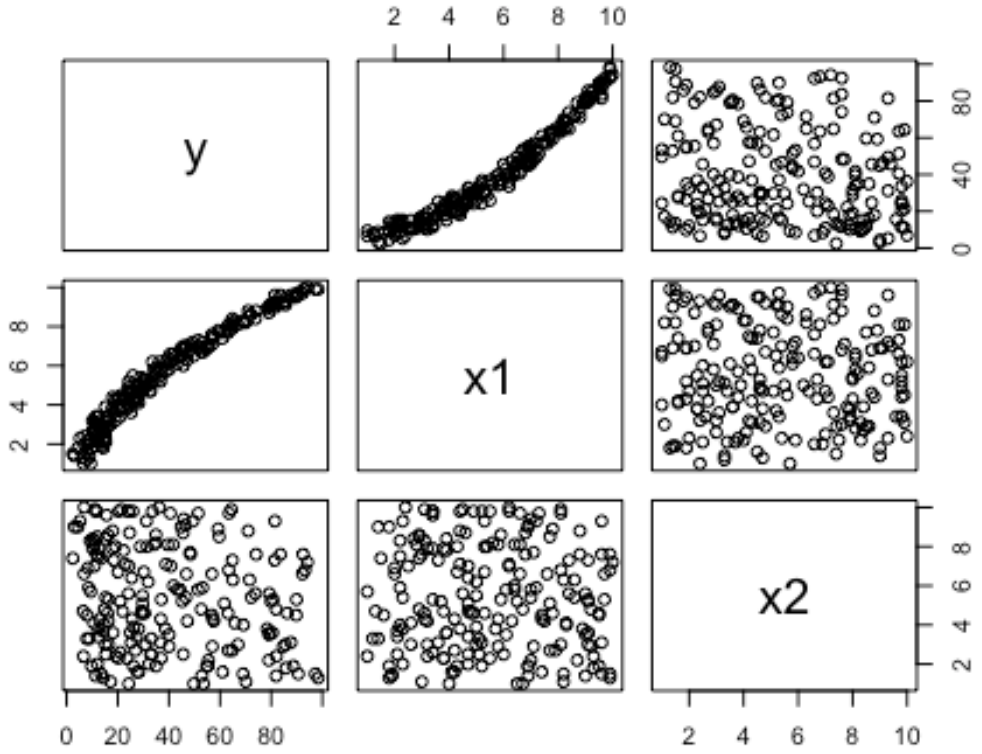
```
##           y  x1  x2
## 1 25.265621 3.6 3.1
## 2 63.512826 8.1 9.7
## 3 25.441710 4.7 6.4
## 4 79.286388 9.0 5.6
## 5 86.271320 9.5 4.6
## 6  3.015506 1.4 9.0
```

```
str(dataset)
```

```
## 'data.frame':    200 obs. of  3 variables:
##  $ y : num  25.3 63.5 25.4 79.3 86.3 ...
##  $ x1: num  3.6 8.1 4.7 9 9.5 1.4 5.8 9.1 6 5.1 ...
##  $ x2: num  3.1 9.7 6.4 5.6 4.6 9 4.3 3.6 2.5 2.5 ...
```

(a) Plot a scatterplot matrix and briefly discuss the relationships between the variables.

```
pairs(~.,data = dataset)
```



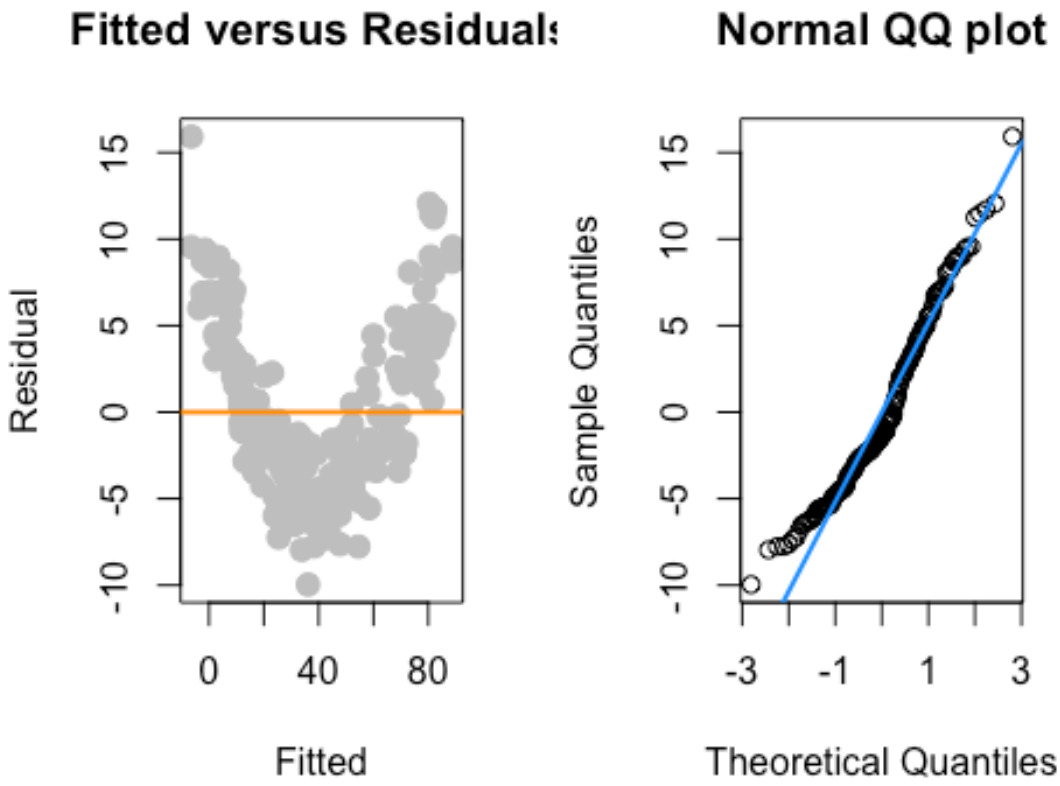
From the scatter plot, we observe a increasing pattern between y and x1. There is no pattern between y and x2 since the points on the scatter plot seem to be scattered randomly. That is, there is no relationship between y and x2. We also see no relationship between x1 and x2.

(b) Obtain the fitted model. Check the model assumptions using appropriate graphical and testings approaches.

```
fitted_model = lm(y ~ x1 + x2, data = dataset)
summary(fitted_model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = dataset)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.963 -3.503 -1.347  3.473 15.919
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9.5112     1.1359  -8.374 1.03e-14 ***
## x1           10.0947     0.1402  71.983  < 2e-16 ***
## x2           -1.2387     0.1309  -9.461  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.927 on 197 degrees of freedom
## Multiple R-squared:  0.9646, Adjusted R-squared:  0.9642
## F-statistic:  2681 on 2 and 197 DF,  p-value: < 2.2e-16
```

```r
# residual plot and normal QQ plot
par(mfrow=c(1,2))
plot(fitted(fitted_model), resid(fitted_model), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(fitted_model), main = "Normal QQ plot")
qqline(resid(fitted_model), col = "dodgerblue", lwd = 2)
```



From the residual plot: the (vertical) spread of the residuals is roughly the same at any fitted value. Hence the constant variance assumption holds. On the other hand, the mean of the residuals varies with the fitted value. At left and right sides of fitted values the residuals were mostly positive, whereas most of the residuals at medium fitted values were negative. This is evidence that the relationship between y and x1 and x2 is not linear. Hence, the linearity assumption is violated. From the normal QQ plot, the points around the left edge is distant from the linear line. Hence, this is evidence against the normal assumption. There are some points that might be outliers on both side of edges.

```
# bptest
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

bptest(fitted_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fitted_model
## BP = 0.094601, df = 2, p-value = 0.9538
```

The BP test is used for the null hypothesis: true errors have the same (constant) variance. The large p-value (0.9538) of the test confirms that there was no evidence against the null hypothesis.

```
# Shapiro test
shapiro.test(resid(fitted_model))

##
##  Shapiro-Wilk normality test
##
## data:  resid(fitted_model)
## W = 0.95915, p-value = 1.603e-05

1.603e-05 < 0.05

## [1] TRUE
```

Since the p-value (1.603e-05) of the Shapiro test was lower than 0.05, we reject the null hypothesis and conclude that the normal assumption is violated.

(c) Was there any influential point? Use Cook's distance with threshold = 4/n. Report the indices of the influential points.

```
# Obtains Cook's distances
cd_fit = cooks.distance(fitted_model)

# We have 14 influential observations
sum(cd_fit > 4/length(cd_fit))

## [1] 14

influ_inx = which(cd_fit > 4/length(cd_fit))
influ_inx

##   6  18  24  31  35  51  74  87 111 126 128 139 143 193
##   6  18  24  31  35  51  74  87 111 126 128 139 143 193
```

6, 18, 24, 31, 35, 51, 74, 87, 111, 126, 128, 139, 143, and 193 are influential points.

(d) Among the influential points, how many of them are also considered outliers (whose absolute standardized residuals are greater than 2)?

```
# checking outliers
rstandard(fitted_model) #standardized residuals
```

```
##           1           2           3           4           5           6
##  0.46482585  0.67241728 -0.92939568  0.99832757  1.14303960  1.96356283
##           7           8           9          10          11          12
## -1.39997027  0.54612431 -0.48134436 -1.13801754  1.65337607 -0.97608418
##          13          14          15          16          17          18
## -0.32164277 -1.11267867  0.71424992  1.65759977 -0.26113826  1.78740283
##          19          20          21          22          23          24
## -0.86995687  1.15029980  0.43937199 -0.14292923 -0.26658742  2.40339748
##          25          26          27          28          29          30
## -1.13585250 -0.56819619 -1.31125583 -1.20702062  0.13265141  0.61906256
##          31          32          33          34          35          36
##  2.47128093  0.92424105 -0.54217045 -0.49062760  1.92927098 -1.57840805
##          37          38          39          40          41          42
## -0.43544739  0.55916984 -0.47012528 -0.11786438  0.65541957 -0.99730233
##          43          44          45          46          47          48
## -0.40283799 -1.11629618  0.69043647  1.83609094 -0.02004799 -1.33791865
##          49          50          51          52          53          54
## -0.23662647  0.86314841  1.72142725 -1.29910267 -0.09334465  1.01048860
##          55          56          57          58          59          60
## -0.32908329  0.42440848  1.37456374 -0.08071760  0.82048804 -1.22291827
##          61          62          63          64          65          66
## -0.76831382  1.41830909 -0.65792062  0.55972768  1.12021659 -0.28236207
##          67          68          69          70          71          72
## -0.03556778 -0.69369997 -0.36716283 -0.90991413  0.21697389 -0.68611164
##          73          74          75          76          77          78
## -0.45680862  1.85401713 -0.53233988  0.07113604 -0.47729815 -0.89851262
##          79          80          81          82          83          84
## -0.68305846  1.27267021 -0.58222793 -0.60358480 -0.78203007 -0.30309944
##          85          86          87          88          89          90
##  1.17622990 -0.91175122  1.78894192  1.43325754  0.48408115  0.77300135
##          91          92          93          94          95          96
##  1.46610601 -0.53008000 -0.33920518 -0.85139806  0.42386805  0.59912048
##          97          98          99         100         101         102
##  0.91067766  0.71674915 -1.15561795 -0.38238537 -1.10698107 -1.09441148
##         103         104         105         106         107         108
## -1.27274343  1.15768584 -1.11859317  1.00733332  0.13691302 -1.05371858
##         109         110         111         112         113         114
## -0.38088084  1.67376193  1.84093271 -0.08810475  1.22689922  1.04679790
##         115         116         117         118         119         120
## -0.69331046  1.38621818 -0.91055440  0.85203774 -0.33841287 -0.30586609
##         121         122         123         124         125         126
## -0.57022274 -0.10087128 -1.15030985  0.60329548 -1.10380934  1.97087263
```

```
##         127         128         129         130         131         132
##  0.93770543  1.50607022  1.43247573 -0.46386002 -0.41862969  0.30299388
##         133         134         135         136         137         138
## -0.88544822  0.40546721 -0.95210942 -1.58184967 -0.29070104 -0.09110420
##         139         140         141         142         143         144
##  2.30646260 -0.76009036 -0.22953953 -0.64726366  3.26712823  0.43012016
##         145         146         147         148         149         150
##  0.33701440 -0.70914067  0.32192563  1.42158867 -0.53797789 -0.44561256
##         151         152         153         154         155         156
##  0.71629522 -1.63015550 -0.96351007 -0.12149528  0.88207369 -0.44776752
##         157         158         159         160         161         162
## -0.73951438  0.19275155 -2.02934285  1.44564338 -1.26008167 -0.53656472
##         163         164         165         166         167         168
## -0.49828134 -0.42003700 -0.38970951 -0.98180817 -0.34433536 -0.45230097
##         169         170         171         172         173         174
## -0.55500792 -0.37100191  0.09531982 -0.21467809  0.53432637 -0.04438908
##         175         176         177         178         179         180
## -0.51874038 -0.92836722 -0.99755359 -1.09002634  0.57583611 -1.50957643
##         181         182         183         184         185         186
##  0.41641159 -0.25121236 -0.53112556 -0.48764714 -0.73080089 -0.25402795
##         187         188         189         190         191         192
## -0.12875641 -1.55557309  0.75645510  1.01188675 -0.22935204 -1.47804729
##         193         194         195         196         197         198
##  2.35603753 -1.10052291  0.94937745 -0.48961760 -1.28399123 -0.61533223
##         199         200
##  0.16046042 -0.91184733
```

```
#abs(rstandard(fitted_model)) > 2
#out_i = which(abs(rstandard(fitted_model)) > 2)
#rstandard(fitted_model)[out_i]

sum(abs(rstandard(fitted_model)[influ_inx]) > 2)
```

```
## [1] 5
```

Among the influential points, 5 of them are also considered outliers.

(e) Suppose that the influential points identified in (c) were simple measurement errors. Remove the influential points from the data and repeat (b) using the updated data set. Was the removal of the influential points helpful for correcting the model assumptions?

```
# ASSUME that those points are simple measurement errors
# Eliminate the points from dataset and store the rest into dataset2
inf_i = which(cd_fit > 4/length(cd_fit))
dataset2 = dataset[-inf_i,]
```

```r
# Check if the influential points are removed
head(dataset)
```

```
##           y  x1  x2
## 1 25.265621 3.6 3.1
## 2 63.512826 8.1 9.7
## 3 25.441710 4.7 6.4
## 4 79.286388 9.0 5.6
## 5 86.271320 9.5 4.6
## 6  3.015506 1.4 9.0
```

```r
head(dataset2)
```

```
##          y  x1  x2
## 1 25.26562 3.6 3.1
## 2 63.51283 8.1 9.7
## 3 25.44171 4.7 6.4
## 4 79.28639 9.0 5.6
## 5 86.27132 9.5 4.6
## 7 36.83479 5.8 4.3
```

```r
nrow(dataset)
```

```
## [1] 200
```

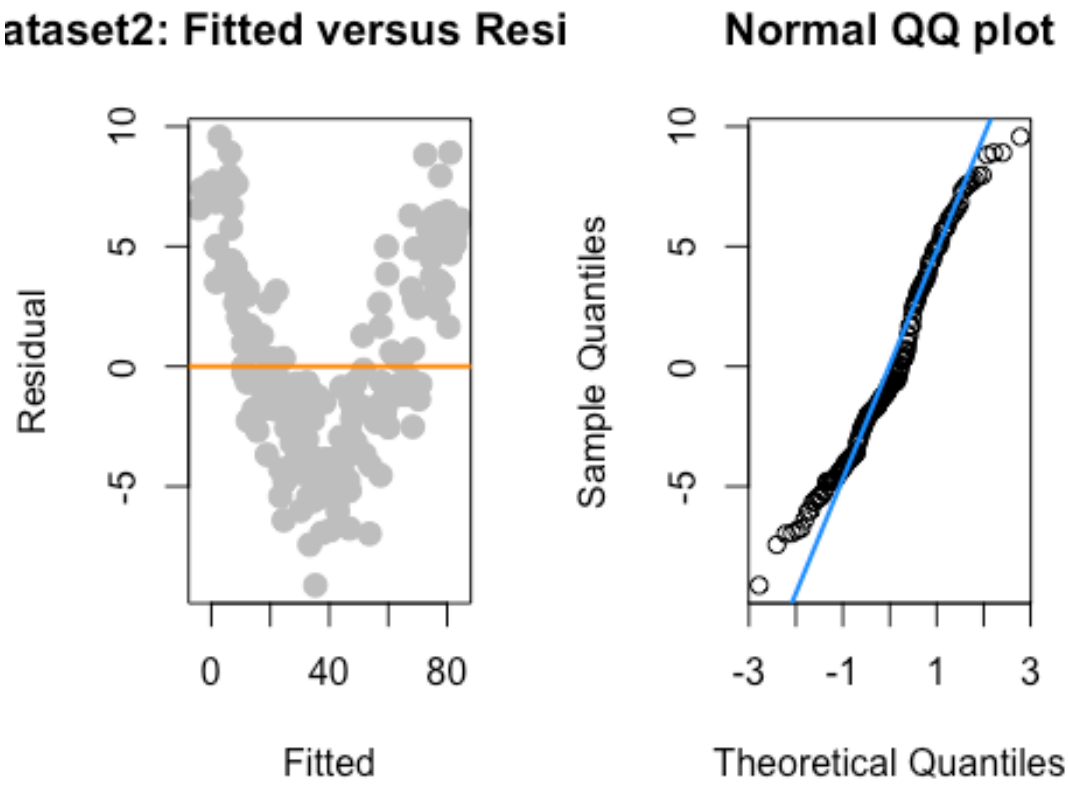
```r
nrow(dataset2)
```

```
## [1] 186
```

```r
# Fit dataset2
fitted_model2 = lm(y ~ x1 + x2, data = dataset2)
summary(fitted_model2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = dataset2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1238 -3.1455 -0.7818  3.2732  9.5680
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.4842     1.0544  -9.944   <2e-16 ***
## x1           10.0788     0.1332  75.662   <2e-16 ***
## x2           -1.1841     0.1146 -10.329   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.161 on 183 degrees of freedom
## Multiple R-squared:  0.9704, Adjusted R-squared:  0.9701
## F-statistic:  3003 on 2 and 183 DF,  p-value: < 2.2e-16
```

```r
# Residual plot and normal qq plot
par(mfrow=c(1,2))
plot(fitted(fitted_model2), resid(fitted_model2), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "dataset2: Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(fitted_model2), main = "Normal QQ plot")
qqline(resid(fitted_model2), col = "dodgerblue", lwd = 2)
```



From the residual plot: the (vertical) spread of the residuals is roughly the same at any fitted value. Hence the constant variance assumption holds. On the other hand, the mean of the residuals varies with the fitted value. At left and right sides of fitted values the residuals were mostly positive, whereas most of the residuals at medium fitted values were negative. This is evidence that the relationship between y and x1 and x2 is not linear. Hence, the linearity assumption is violated. From the normal QQ plot, the points around the both edges are a little bit distant from the linear line. Hence, this is evidence against the normal assumption.

```r
# bptest
bptest(fitted_model2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fitted_model2
## BP = 0.78179, df = 2, p-value = 0.6764
```

The BP test is used for the null hypothesis: true errors have the same (constant) variance.nThe large p-value (0.6764) of the test confirms that there was no evidence against the null hypothesis. Even though p-value is still large, we can find p-value is decreased after removing the influential points.

```
# Shapiro test
shapiro.test(resid(fitted_model2))
```

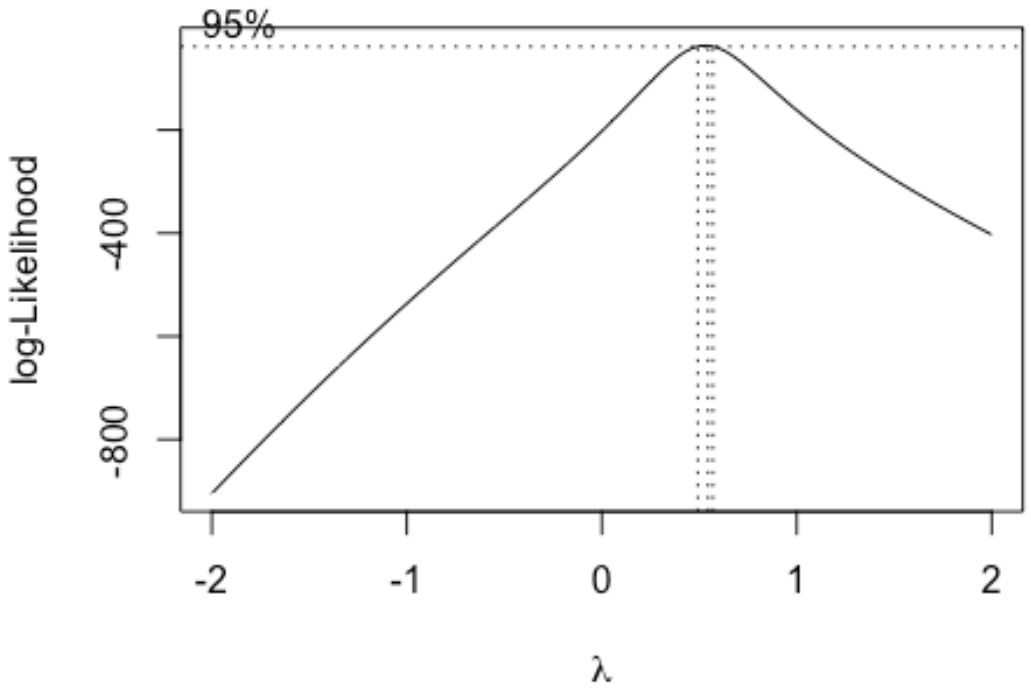
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fitted_model2)
## W = 0.96638, p-value = 0.0001911
```

Since the p-value (0.0001911) of the Shapiro test was lower than 0.05, we reject the null hypothesis and conclude that the normal assumption is violated. Even though p-value is still small, we can find p-value is increased after removing the influential points.

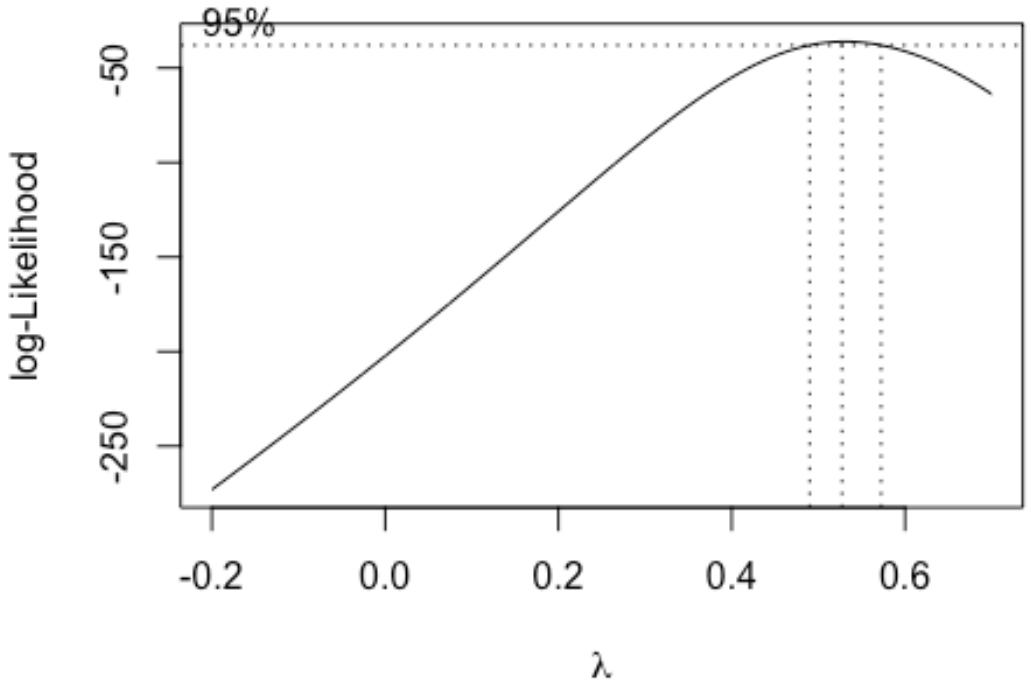
The removal of the influential points was not that much helpful for correcting the model assumptions.

(f) Use the Box-Cox method to determine the best transformation on the response variable y

```
# We need the MASS package for the boxcox function
# No need to install the package this time,
# as it is installed as default.
library(MASS)

# Run the boxcox
# Input: lm object where Y is used as response
# The optimal lambda is around 0 -> log transformation was a reasonable
choice.
par(mfrow=c(1,1))
boxcox(fitted_model)
```

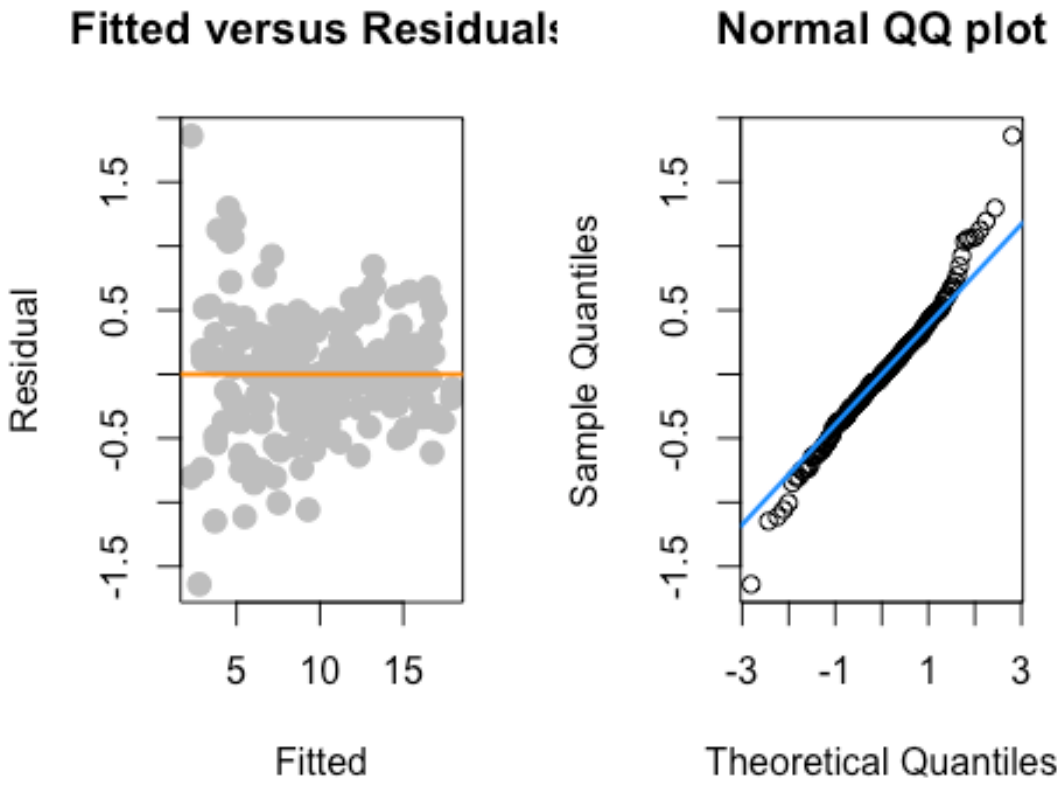



```r
# Specify the range of lambda
boxcox(fitted_model, lambda = seq(-0.2, 0.7, by = 0.1))
```




```
#bc <- boxcox(fitted_model)
#lambda <- bc$x[which.max(bc$y)]

# Lets transform Y using lambda = 0.5
lambda = 0.5
transfer_fit <- lm(((y^(lambda)-1)/(lambda))~ x1 + x2, data = dataset)
summary(transfer_fit)

##
## Call:
## lm(formula = ((y^(lambda) - 1)/(lambda)) ~ x1 + x2, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63948 -0.26509 -0.00651  0.26212  1.86093
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.78737    0.10981   16.28   <2e-16 ***
## x1           1.65931    0.01356  122.40   <2e-16 ***
## x2          -0.20386    0.01266  -16.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4763 on 197 degrees of freedom
## Multiple R-squared:  0.9875, Adjusted R-squared:  0.9873
## F-statistic:  7751 on 2 and 197 DF,  p-value: < 2.2e-16

#lm(formula = y ~ ., data = dataset)

# residual plot and normal QQ plot
par(mfrow=c(1,2))
plot(fitted(transfer_fit), resid(transfer_fit), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(transfer_fit), main = "Normal QQ plot")
qqline(resid(transfer_fit), col = "dodgerblue", lwd = 2)
```



From the residual plot: the mean of the residuals is (roughly) close to zero at any fitted value. Hence the linearity assumption holds. On the other hand, the (vertical) spread of the residuals decreases as the fitted value gets large. This is evidence that the variance of true errors is dependent on the mean response value. Hence, the equal variance assumption is violated.

From the normal QQ plot: the points around the two edges are distant from the linear line. Hence, this is evidence against the normal assumption. There are some points that might be outliers on both side of edges.

```
# bptest
library(lmtest)
bptest(transfer_fit)

##
##  studentized Breusch-Pagan test
##
## data:  transfer_fit
## BP = 26.212, df = 2, p-value = 2.033e-06
```

The BP test is used for the null hypothesis: true errors have the same (constant) variance. The small p-value (2.033e-06) of the test confirms that equal variance assumption is violated.

```
# Shapiro test
shapiro.test(resid(transfer_fit))

##
##  Shapiro-Wilk normality test
##
## data:  resid(transfer_fit)
## W = 0.9816, p-value = 0.01006
```

Since the p-value (0.01006) of the Shapiro test was lower than 0.05, we reject the null hypothesis and conclude that the normal assumption is violated.

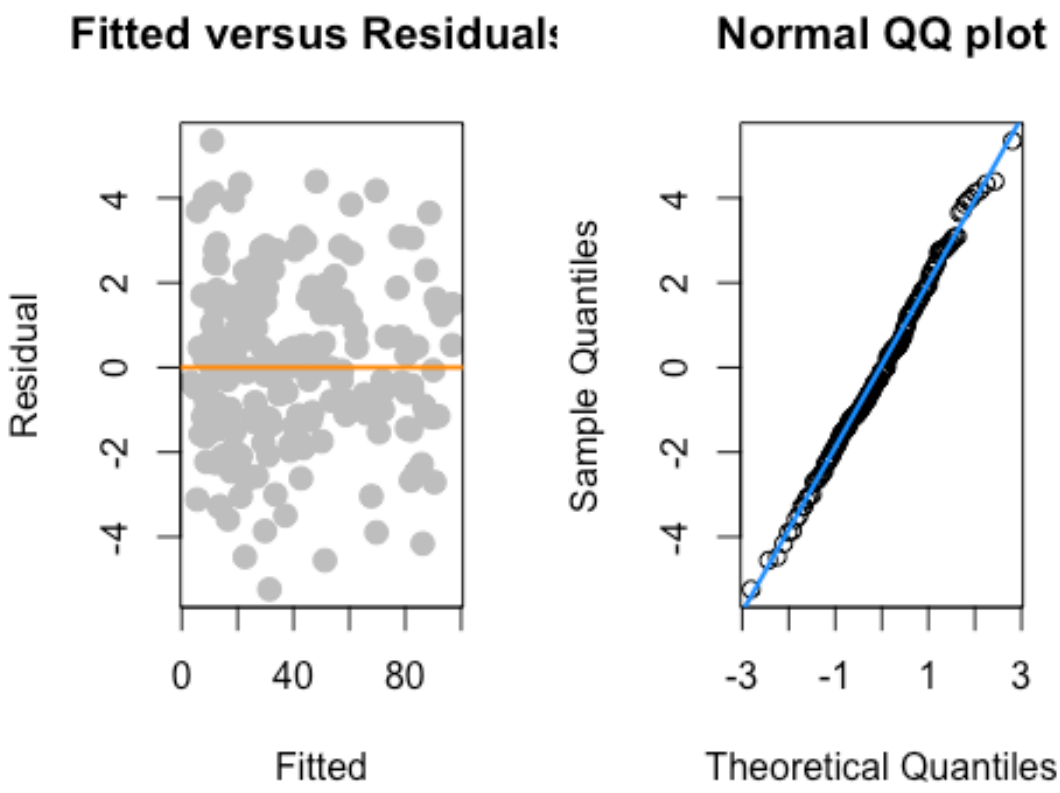
This transformation was not that helpful for correcting the model assumptions.

(g) This time, obtain the polynomial model. Is this polynomial model preferable to the resulting models in (b) and (f)? Justify your answer.

```
poly_fit = lm(y ~ x1 + x2 + I(x1^2) + I(x2^2), data = dataset)

#The polynomial model is significant.
summary(poly_fit)

##
## Call:
## lm(formula = y ~ x1 + x2 + I(x1^2) + I(x2^2), data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2370 -1.2533 -0.0942  1.3701  5.3505
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  8.65216    0.93122   9.291  < 2e-16 ***
## x1           1.30413    0.28367   4.597 7.68e-06 ***
## x2          -0.72887    0.25617  -2.845  0.00491 **
## I(x1^2)      0.77857    0.02463  31.614  < 2e-16 ***
## I(x2^2)     -0.02560    0.02259  -1.133  0.25854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.995 on 195 degrees of freedom
## Multiple R-squared:  0.9942, Adjusted R-squared:  0.9941
## F-statistic:  8422 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
# residual plot and normal QQ plot
par(mfrow=c(1,2))
plot(fitted(poly_fit), resid(poly_fit), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(poly_fit), main = "Normal QQ plot")
qqline(resid(poly_fit), col = "dodgerblue", lwd = 2)
```



The (vertical) spread of the residuals is roughly constant at any fitted value. Hence, the equal variance holds. Also the mean of the residuals is around zero at any fitted value (i.e. the

linearity holds). Most of the points in the normal qq plot are close to the linear line, which suggests that the residuals follow a normal distribution (i.e. the normal assumption holds).

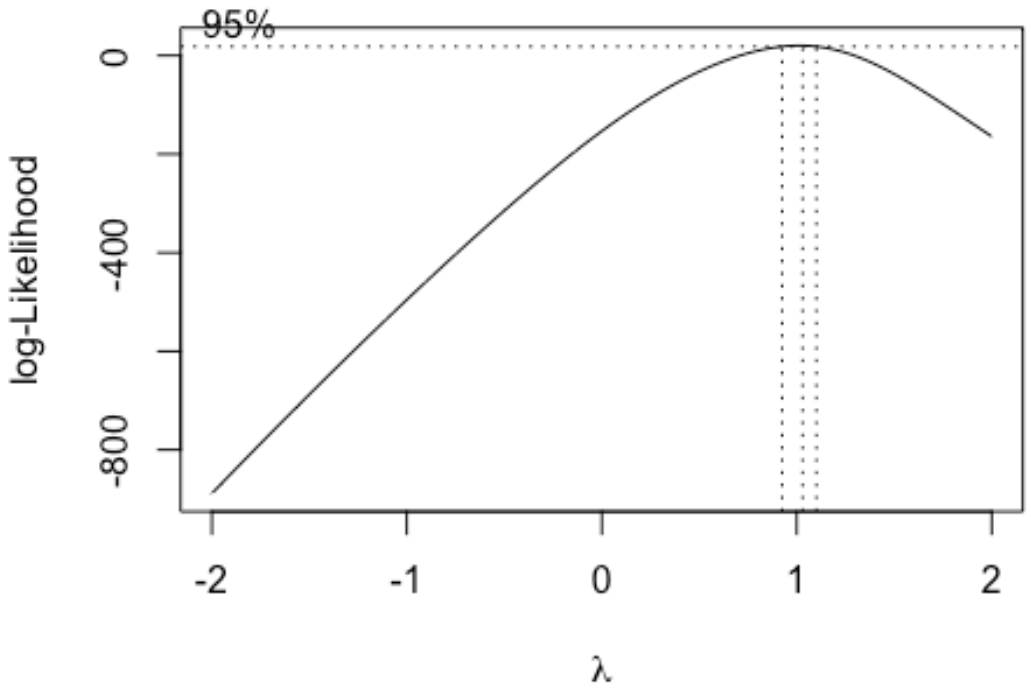
```
# bptest
library(lmtest)
bptest(poly_fit)

##
##  studentized Breusch-Pagan test
##
## data:  poly_fit
## BP = 2.6009, df = 4, p-value = 0.6267
```

The BP test is used for the null hypothesis: true errors have the same (constant) variance. The large p-value (0.6267) of the test confirms that there was no evidence against the null hypothesis.

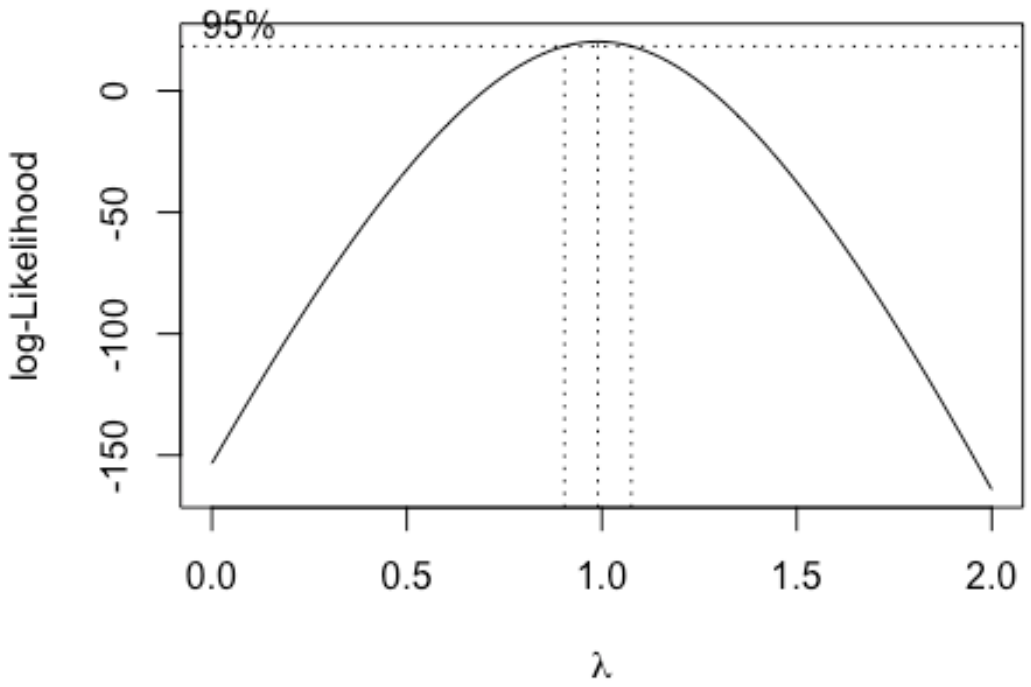
```
# Shapiro test
shapiro.test(resid(poly_fit))

##
##  Shapiro-Wilk normality test
##
## data:  resid(poly_fit)
## W = 0.9956, p-value = 0.8331

0.8331 > 0.05

## [1] TRUE
```

The Shapiro test is used for the null hypothesis: true errors follow a normal distribution. Since the p-value (0.8331) of the test was greater than 0.05, we fail to reject the null hypothesis. There was no evidence to say that the normal assumption is violated.

```
# Run the boxcox
# Input: lm object where Y is used as response
par(mfrow=c(1,1))
boxcox(poly_fit)
```




```r
# Specify the range of lambda
boxcox(poly_fit, lambda = seq(0, 2, by = 0.5))
```




```
# optimal lambda is 1
# If the optimal value for lambda is 1, then the data is already normally
distributed,
# and the Box-Cox transformation is unnecessary.
```

This polynomial model is preferable to the resulting models in (b) and (f) since the equal variance assumption holds from bp test and a residual plot, the normal assumption holds from Shapiro test and the normal qq plot, the linearity holds, boxcox shows the data is already normally distributed.

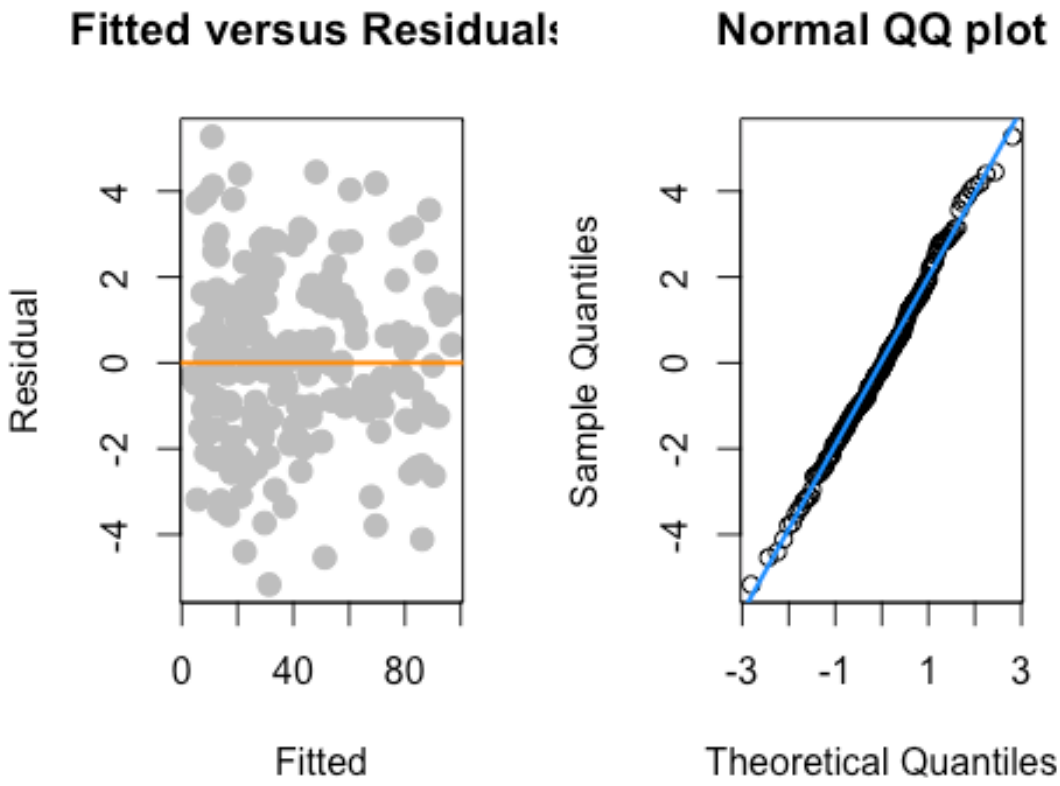
(h) Add the cubic terms to the model. Would this cubic model be preferred to the quadratic one in (g)?

```
cubic_fit = lm(y ~ x1 + x2 + I(x1^2) + I(x2^2) + I(x1^3) + I(x2^3), data =
dataset)
summary(cubic_fit)

##
## Call:
## lm(formula = y ~ x1 + x2 + I(x1^2) + I(x2^2) + I(x1^3) + I(x2^3),
##     data = dataset)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.166 -1.281 -0.122  1.359  5.273
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.065491   1.810742   5.007 1.25e-06 ***
## x1           1.420580   0.929225   1.529    0.128
## x2          -1.182477   0.801651  -1.475    0.142
## I(x1^2)      0.755965   0.182125   4.151 4.97e-05 ***
## I(x2^2)      0.069683   0.161015   0.433    0.666
## I(x1^3)      0.001279   0.010753   0.119    0.905
## I(x2^3)     -0.005755   0.009623  -0.598    0.551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.004 on 193 degrees of freedom
## Multiple R-squared:  0.9943, Adjusted R-squared:  0.9941
## F-statistic:  5568 on 6 and 193 DF,  p-value: < 2.2e-16
```

The cubic model is also significant, but not that different with the polynomial model. polynomial model's r-squared: 0.9942 cubic model's r-squared: 0.9943 There is only 0.0001 difference.

```
# residual plot and normal QQ plot
par(mfrow=c(1,2))
plot(fitted(cubic_fit), resid(cubic_fit), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(cubic_fit), main = "Normal QQ plot")
qqline(resid(cubic_fit), col = "dodgerblue", lwd = 2)
```



There is no big difference between polynomial and cubic models

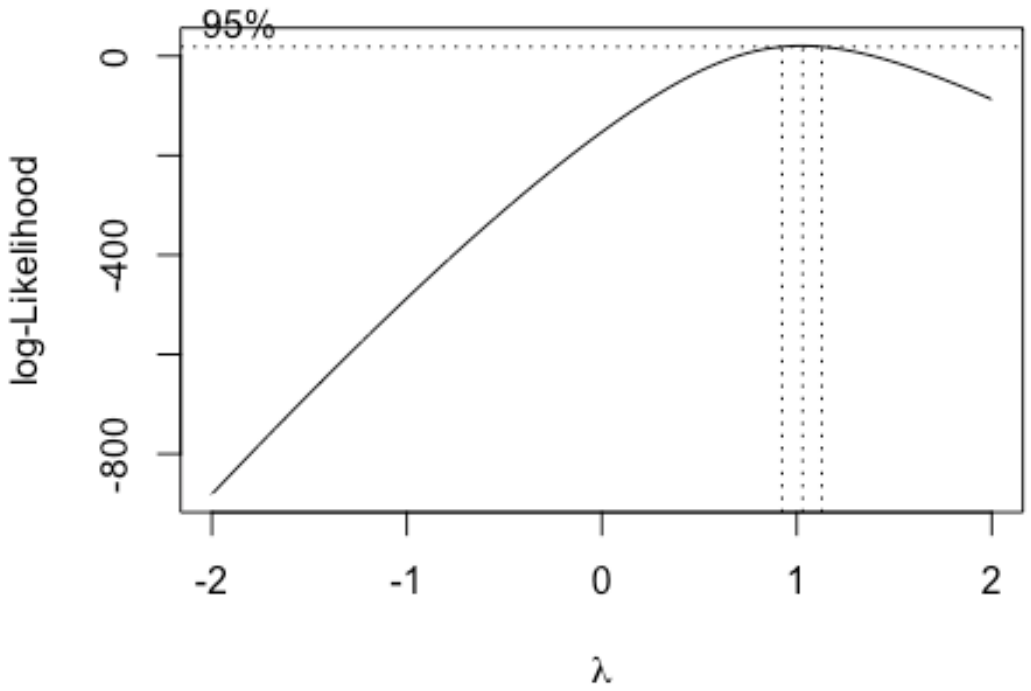
```
# bptest
library(lmtest)
bptest(cubic_fit)

##
##  studentized Breusch-Pagan test
##
## data:  cubic_fit
## BP = 4.2839, df = 6, p-value = 0.6383
```

The BP test is used for the null hypothesis: true errors have the same (constant) variance. The large p-value (0.6383) of the test confirms that there was no evidence against the null hypothesis. There is no big difference between polynomial and cubic models.

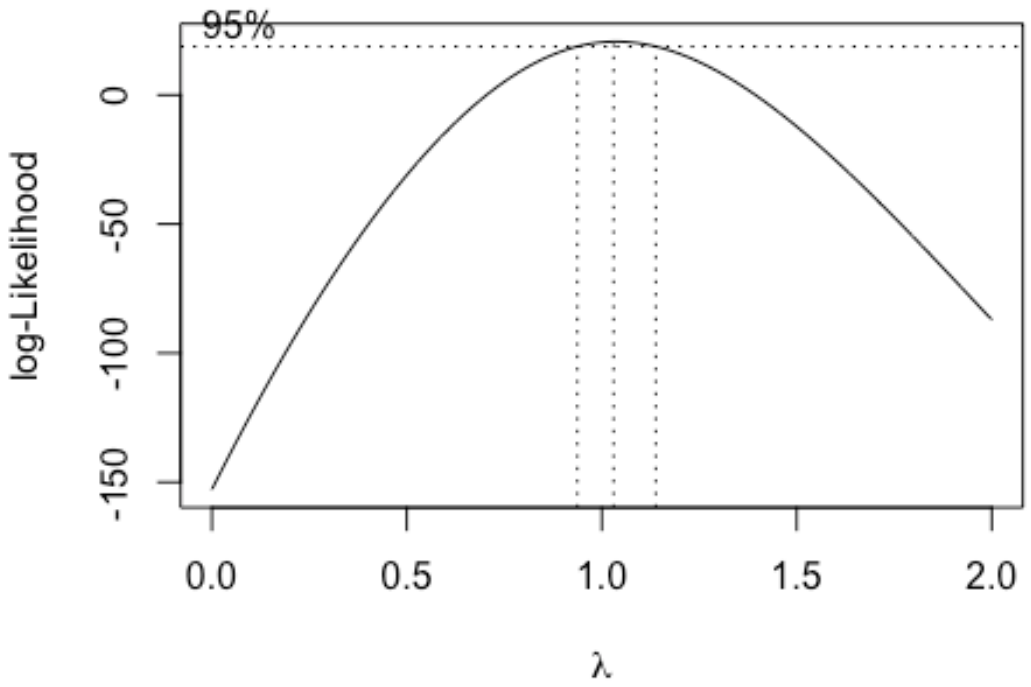
```
# Shapiro test
shapiro.test(resid(cubic_fit))

##
##  Shapiro-Wilk normality test
##
## data:  resid(cubic_fit)
## W = 0.99579, p-value = 0.8581
```

The Shapiro test is used for the null hypothesis: true errors follow a normal distribution. Since the p-value (0.8581) of the test was greater than 0.05, we fail to reject the null hypothesis. There was no evidence to say that the normal assumption is violated. There is no big difference between polynomial and cubic models.

```
# Run the boxcox
# Input: lm object where Y is used as response
par(mfrow=c(1,1))
boxcox(cubic_fit)
```




```
# Specify the range of lambda
boxcox(cubic_fit, lambda = seq(0, 2, by = 0.5))
```




```
# optimal lambda is 1
# If the optimal value for lambda is 1, then the data is already normally
distributed,
# and the Box-Cox transformation is unnecessary.
# There is no big difference between polynomial and cubic models
```

So we may use poly_fit as the final model.

3 use the mtcars data in R.

(a) Fit a regression model (model_a) using mpg as the response and cyl, disp, hp, wt and drat as predictors (Do not include and polynomial or interaction terms). Obtain the Variance Inflation Factor (VIF) for each predictor. (You may use the "vif" function in the faraway package.) Does any collinearity exist? Report all predictors whose VIF are higher than 10. Briefly explain how collinearity affects in the regression analysis.

```
# Consider the following reg model using mtcars
install.packages("car",repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##
/var/folders/7l/hzhbdmks3snf1lxq26d5wcbh0000gn/T//Rtmp4V4WM5/downloaded_packa
ges

library(car)

## Loading required package: carData

model_a = lm(mpg ~ cyl + disp + hp + wt + drat, data = mtcars)
summary(model_a)

##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + wt + drat, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7014 -1.6850 -0.4226  1.1681  5.7263
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.00836    7.57144   4.756  6.4e-05 ***
## cyl         -1.10749    0.71588  -1.547  0.13394
## disp         0.01236    0.01190   1.039  0.30845
## hp          -0.02402    0.01328  -1.809  0.08208 .
## wt          -3.67329    1.05900  -3.469  0.00184 **
## drat         0.95221    1.39085   0.685  0.49964
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.538 on 26 degrees of freedom
## Multiple R-squared:  0.8513, Adjusted R-squared:  0.8227
## F-statistic: 29.77 on 5 and 26 DF,  p-value: 5.618e-10

vif(model_a)

##       cyl      disp        hp        wt      drat
##  7.869010 10.463957  3.990380  5.168795  2.662298

#?vif
```

There exists the collinearity as cyl and disp have pretty high vifs (much bigger than 1) disp's VIF is 10.46 which is higher than 10. Higher VIF results in larger standard error since $VIF_j = (1/(1 - R_j^2))$ where R_j^2 is the R-squared from the regression of x_j on the other predictors. With large standard errors, individual regression coefficients may not be meaningful. Further, because a large standard error means that the corresponding t-ratio is small, it is difficult to detect the importance of a variable.

(b) From the result in (a), remove the predictor with the highest VIF value and fit another regression model using the rest of the predictors. Obtain the Variance Inflation Factor (VIF) for each predictor used for the model. This time, do not use any built-in function in R to compute the VIF values. (You can still use the lm function/object.) Does any collinearity exist? Report all predictors whose VIF are higher than 10.

```
model_a2 = lm(mpg ~ cyl + hp + wt + drat, data = mtcars)
summary(model_a2)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + hp + wt + drat, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6171 -1.5663 -0.6058  1.2612  5.8161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.49588    7.44101   4.636  8.1e-05 ***
## cyl         -0.76229    0.63502  -1.200  0.24040
## hp          -0.02089    0.01295  -1.613  0.11845
## wt          -2.97331    0.81818  -3.634  0.00116 **
## drat         0.81771    1.38684   0.590  0.56034
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.541 on 27 degrees of freedom
## Multiple R-squared:  0.8451, Adjusted R-squared:  0.8222
## F-statistic: 36.84 on 4 and 27 DF,  p-value: 1.438e-10
```

```
# VIF for cyl
cyl_model = lm(cyl ~ hp + wt + drat, data = mtcars)
summary(cyl_model)
```

```
##
## Call:
## lm(formula = cyl ~ hp + wt + drat, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16710 -0.53372 -0.08989  0.60628  1.22337
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.691584   1.817868   3.681 0.000981 ***
```

```
## hp           0.014883   0.002635   5.647 4.74e-06 ***
## wt           0.334046   0.235165   1.420 0.166510
## drat        -1.045972   0.362310  -2.887 0.007415 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7563 on 28 degrees of freedom
## Multiple R-squared:  0.838,  Adjusted R-squared:  0.8207
## F-statistic: 48.29 on 3 and 28 DF,  p-value: 3.418e-11
```

```
1/(1-summary(cyl_model)$r.squared)
```

```
## [1] 6.17356
```

```
# VIF for hp
hp_model = lm(hp ~ cyl + wt + drat, data = mtcars)
summary(hp_model)
```

```
##
## Call:
## lm(formula = hp ~ cyl + wt + drat, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.673 -21.445  -8.728  22.142 121.980
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -252.414     97.540  -2.588   0.0151 *
## cyl           35.780      6.336   5.647 4.74e-06 ***
## wt            10.602     11.770   0.901   0.3754
## drat          39.928     18.777   2.126   0.0424 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.08 on 28 degrees of freedom
## Multiple R-squared:  0.7358, Adjusted R-squared:  0.7075
## F-statistic: 25.99 on 3 and 28 DF,  p-value: 3.041e-08
```

```
1/(1-summary(hp_model)$r.squared)
```

```
## [1] 3.78467
```

```
# VIF for wt
wt_model = lm(wt ~ cyl + hp + drat, data = mtcars)
summary(wt_model)
```

```
##
## Call:
## lm(formula = wt ~ cyl + hp + drat, data = mtcars)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.0566 -0.2998 -0.1528  0.1789  1.2923
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.029787   1.540792   2.615   0.0142 *
## cyl          0.201226   0.141661   1.420   0.1665
## hp           0.002656   0.002949   0.901   0.3754
## drat        -0.680449   0.293387  -2.319   0.0279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.587 on 28 degrees of freedom
## Multiple R-squared:  0.6749, Adjusted R-squared:  0.6401
## F-statistic: 19.38 on 3 and 28 DF,  p-value: 5.322e-07
```

```
1/(1-summary(wt_model)$r.squared)
```

```
## [1] 3.076225
```

```
# VIF for drat
drat_model = lm(drat ~ cyl + hp + wt, data = mtcars)
summary(drat_model)
```

```
##
## Call:
## lm(formula = drat ~ cyl + hp + wt, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67503 -0.22446 -0.01402  0.24571  0.80396
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.204654   0.246380  21.124  < 2e-16 ***
## cyl         -0.219301   0.075963  -2.887  0.00742 **
## hp           0.003482   0.001638   2.126  0.04242 *
## wt          -0.236831   0.102114  -2.319  0.02790 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3463 on 28 degrees of freedom
## Multiple R-squared:  0.6211, Adjusted R-squared:  0.5805
## F-statistic:  15.3 on 3 and 28 DF,  p-value: 4.379e-06
```

```
1/(1-summary(drat_model)$r.squared)
```

```
## [1] 2.639229
```

There does not exist any collinearity since there is no VIF > 10.

4 Use the prostate data in the faraway package. Consider the following three models.

```
library(faraway)

##
## Attaching package: 'faraway'

## The following objects are masked from 'package:car':
##
##     logit, vif

?prostate

model_a = lm(lpsa ~ lcavol + lweight + svi, data = prostate)
model_b = lm(lpsa ~ lcavol + lweight + svi + lbph, data = prostate)
model_c = lm(lpsa ~ lcavol + lweight + svi + lbph + lcp + gleason, data =
prostate)
```

(a) Find the best model in terms of AIC, BIC and adjusted R^2, respectively.

```
AIC(model_a,model_b,model_c) # AIC: chooses model_b

##         df      AIC
## model_a  5 216.5979
## model_b  6 215.9223
## model_c  8 218.9735

BIC(model_a,model_b,model_c) # BIC: chooses model_a

##         df      BIC
## model_a  5 229.4714
## model_b  6 231.3705
## model_c  8 239.5712

# Adjusted_R2: chooses model_b
summary(model_a)$adj.r.squared

## [1] 0.6143899

summary(model_b)$adj.r.squared

## [1] 0.6208036

summary(model_c)$adj.r.squared

## [1] 0.6161501
```

(b) Find the best model using R^2 as the quality criterion. Explain why R^2 is not an appropriate measure for model comparison.

```
# R2: chooses model_c
summary(model_a)$r.squared

## [1] 0.6264403

summary(model_b)$r.squared

## [1] 0.6366035

summary(model_c)$r.squared

## [1] 0.6401407
```

From there it is evident Model C is the best model as it has the highest value of R^2. However R^2 can be misleading sometimes as R^2 always increases as we add more predictors to the model. If we choose the model with the highest R^2, it will always be the model with all possible predictors, even if many of them are useless.