SS3859A Assignment 3

Bradley Assaly-Nesrallah

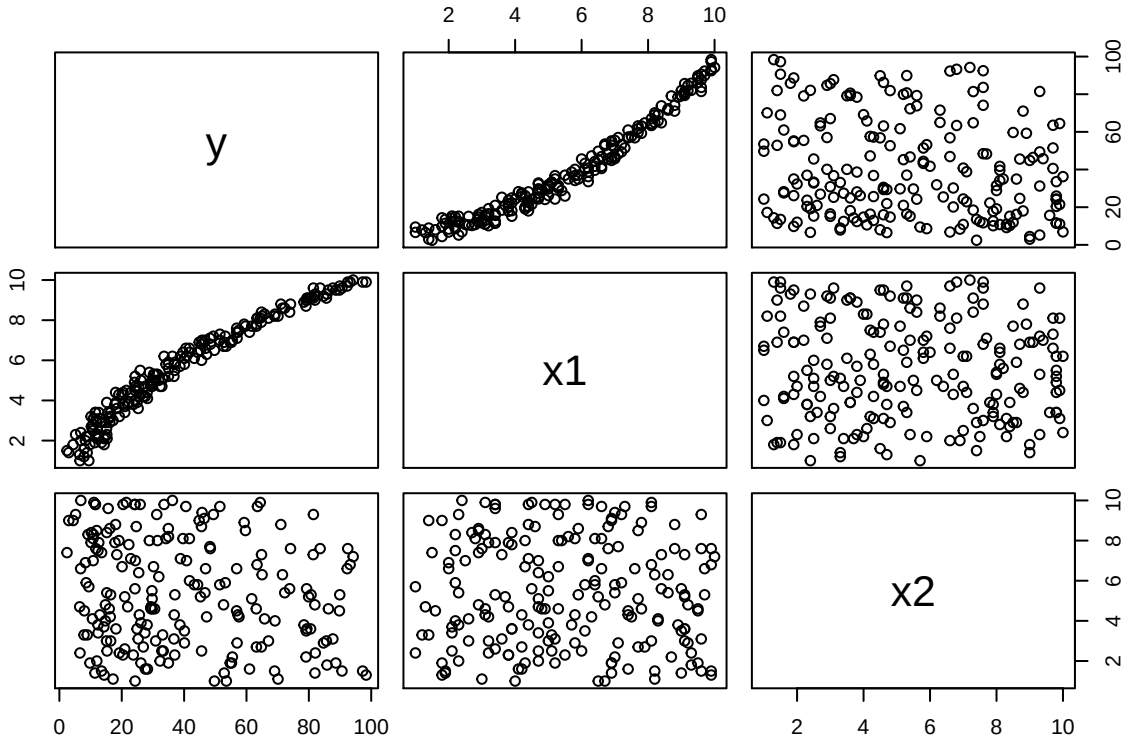
11/11/2020

Question 1

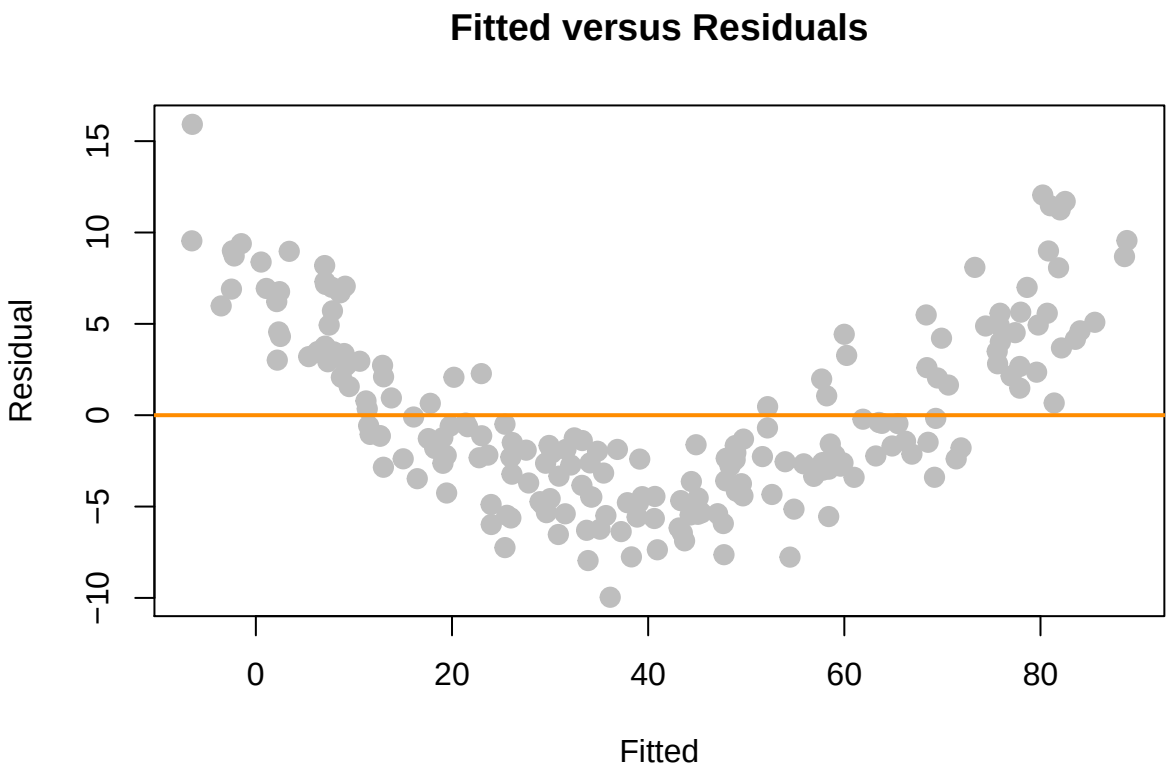
```
##a. Increasing the number of predictors variables will never decreasee the R^2 value :
## True, as R^2 is nondecreasing for MLR first consider the formula for R^2
## R^2=1-SSE/SST, and the objective is to minimize SSE
## However as you add predictors SSE can only decrease
## as the LS of SSE is sum(yi-b0-b1x1-...bpxp-bp+1xp+1)^2
## where there are p predictors and we add a p+1th predictor
##if beta is nonzero SSE decreases so R^2 inreases
##if beta is zero SSE stays the same so R^2 stays the same
## Thus increasing predictors never decreases R^2

##b. Multicolinearity affects the interpretation of regression coefficients :
## True, multicolinearity means that each independent variable is dependent on each other
## making it difficult to interpret which independent variables effect the dependent
## With multicoliearity we are unable to determine if the effect of a variable is based
##on itself or
## the variables that are also effecting each other thus we cant trust the pvalues
## thus as the p-values dont hold as much significance multicollinearity
## affect the interpretation coefficients

##c. The variance inflation factor beta_hatj depends on the R^2 of the regression
## of the response variable y on the predictor variable x_j:
## False the VIF seeks to determine the how much beta_j is inflated
##by correlation among the predictor variables
## Thus VIF bhat_j depends on R^2_j which is the r^2 obtained by
## regressing the j_th predictor on the remaining predictor variables
## thus false so we are done

##d. A high leverage point is always highly influential:
## False we consider the case where a point is very far
## from the observations but farther along the fitted line
## in this case it is high leverage but will not
## impact the fitted model, thus we have a high leverage point not highly influential
```

Question 2

```r
data_q2=read.csv("https://raw.githubusercontent.com/hgweon2/data/main/hw3-data.txt")

##a. We plot the scatterplot matrix and briefly discuss the relationship between variables
pairs(data_q2)
```




```r
## y and x1 appear to have an approximately linear relationship
## x1 and x2 appear to have a nonlinear relationship
## y and x2 appear to have a nonlinear relationship

##b. We obtain the fitted model, and check the linearity, equal variance,
## and normality assumptions:

lm_q2=lm(y~x1+x2,data=data_q2)
plot(fitted(lm_q2), resid(lm_q2), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```

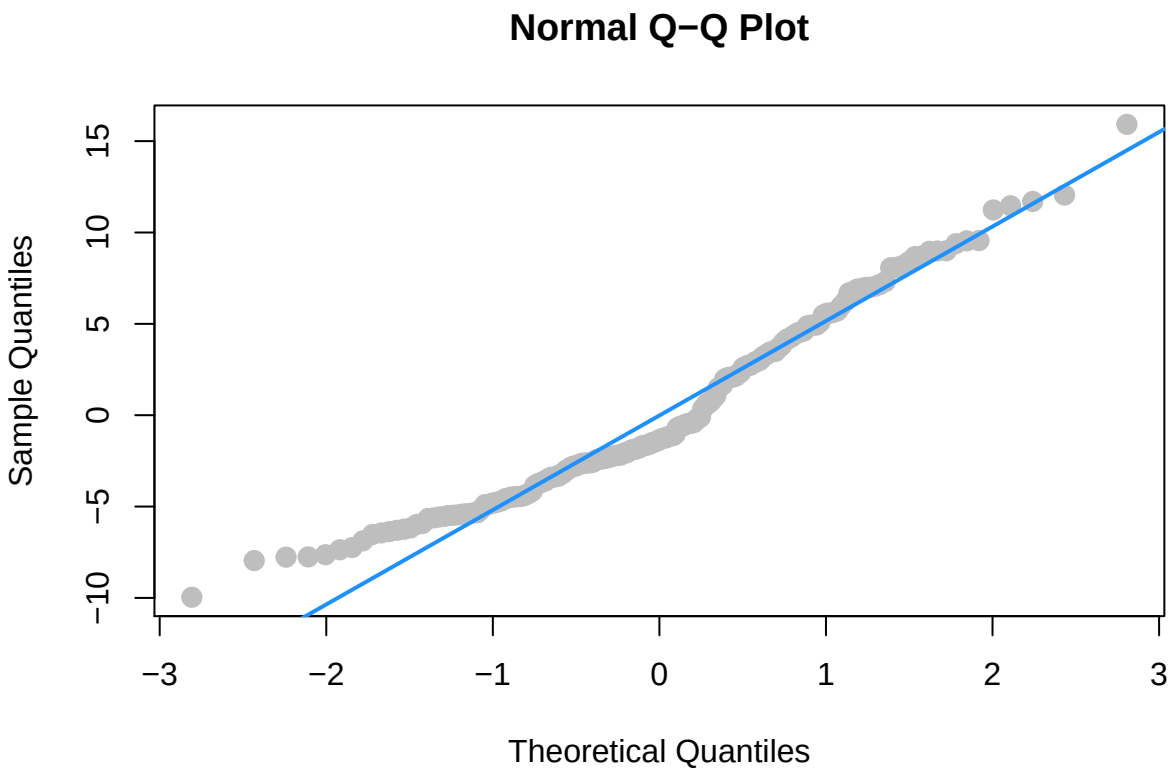



```
qqnorm(resid(lm_q2), col = "grey",pch=20,cex=2)
qqline(resid(lm_q2), col = "dodgerblue", lwd = 2)
##based on the qqnorm it appears to be somewhat normal but needs more info
##based on the residual plot linearity fails but equal variance appears to hold
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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




```
bptest(lm_q2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_q2
## BP = 0.094601, df = 2, p-value = 0.9538
```

```
##accept null hypothesis equal variance
shapiro.test(resid(lm_q2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_q2)
## W = 0.95915, p-value = 1.603e-05
```

```
##reject normality

##c. Yes there were influential points, we check with cooks distance with threshold=4/n
sum(cooks.distance(lm_q2) > 4 / length(cooks.distance(lm_q2)))
```

```
## [1] 14
```

```r
##there were in fact 14 influential points with indices
out_i= which(cooks.distance(lm_q2) > 4 / length(cooks.distance(lm_q2)))==TRUE
out_i
```

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

```r
##the indices of the influential points are the numerical val above

##d.We check how many influence points are outliers

out_j = which(abs(rstandard(lm_q2)) > 2)
rstandard(lm_q2)[out_j]
```

```
##        24        31       139       143       159       193
##  2.403397  2.471281  2.306463  3.267128 -2.029343  2.356038
```

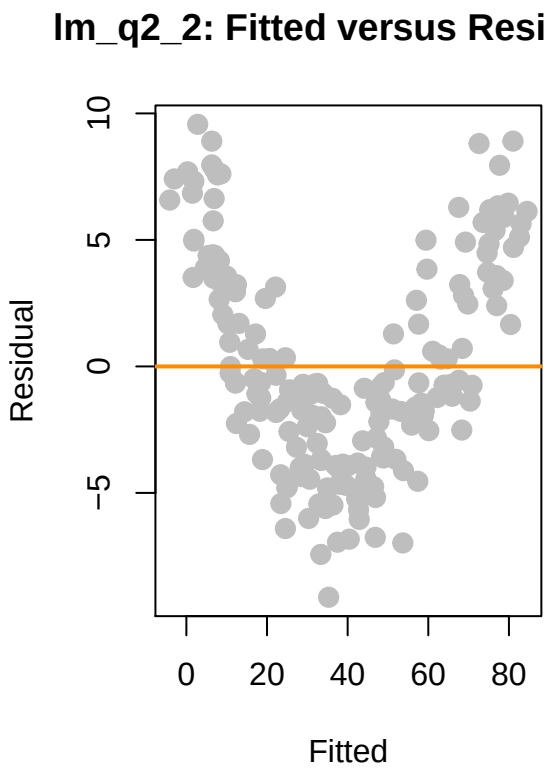
```r
##these are the outliers
out_i
```

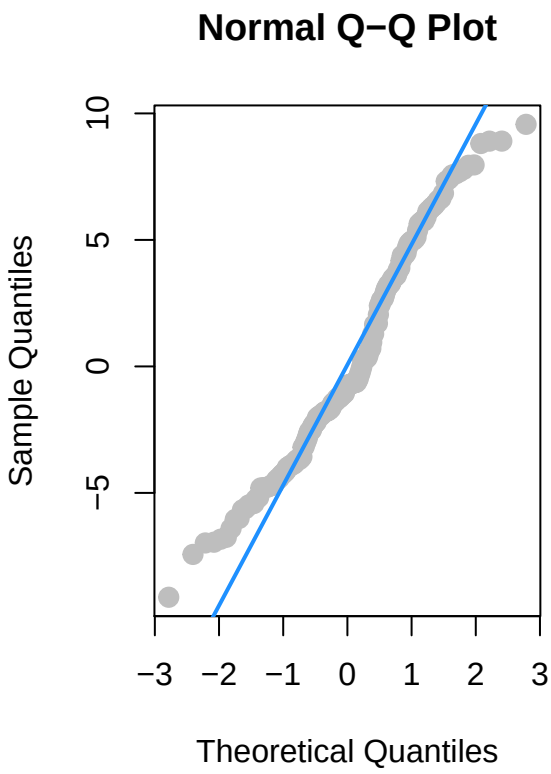
```
##     6    18    24    31    35    51    74    87   111   126   128   139   143
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   193
## FALSE
```

```r
out_j
```

```
##  24  31 139 143 159 193
##  24  31 139 143 159 193
```

```r
## By inpecting the indices of outliers and influence points 24 31 139 143 193 are both
## so there are 5 that are outliers and influential

##e.We remove influential points from the data and check if it corrects model assumptions
lm_q2_cd=cooks.distance(lm_q2)
inf_i = which(lm_q2_cd > 4/length(lm_q2_cd))
data_q2_2=data_q2[-inf_i,]
lm_q2_2=lm(y~x1+x2,data=data_q2_2)
# Residual plot and noremal qq plot
par(mfrow=c(1,2))
plot(fitted(lm_q2_2), resid(lm_q2_2), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residual",cex=2,
     main = "lm_q2_2: Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
qqnorm(resid(lm_q2_2), col = "grey",pch=20,cex=2)
qqline(resid(lm_q2_2), col = "dodgerblue", lwd = 2)
```





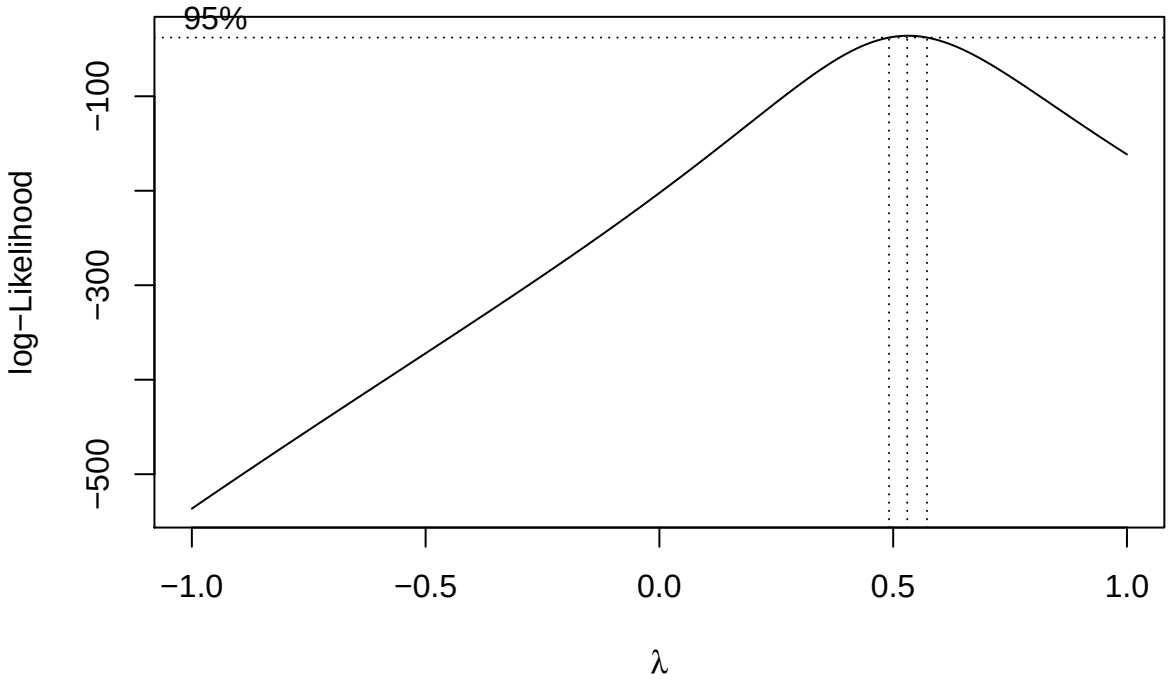

```
##appears slightly more normal on qqplot
##visually the linearity assumption does not appear to hold,
## but equal variance does apper to hold
# bptest
bptest(lm_q2_2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_q2_2
## BP = 0.78179, df = 2, p-value = 0.6764
```

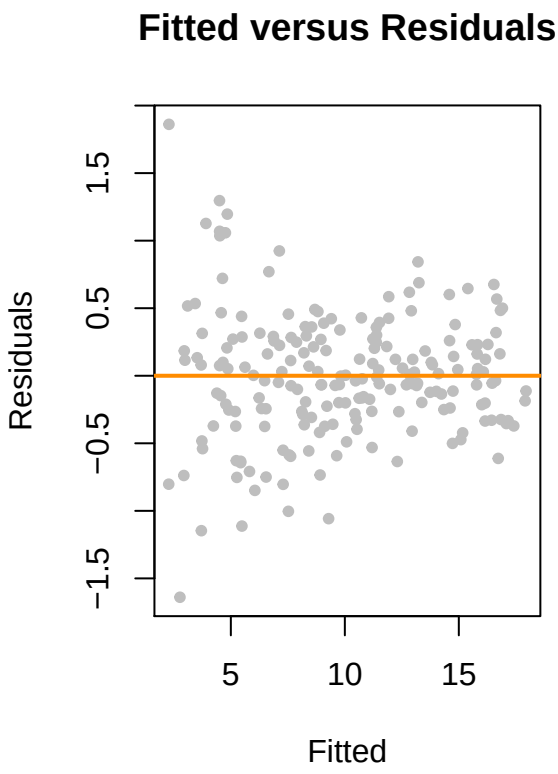
```
##thus equal variance still holds

# Shapiro test
shapiro.test(resid(lm_q2_2))
```

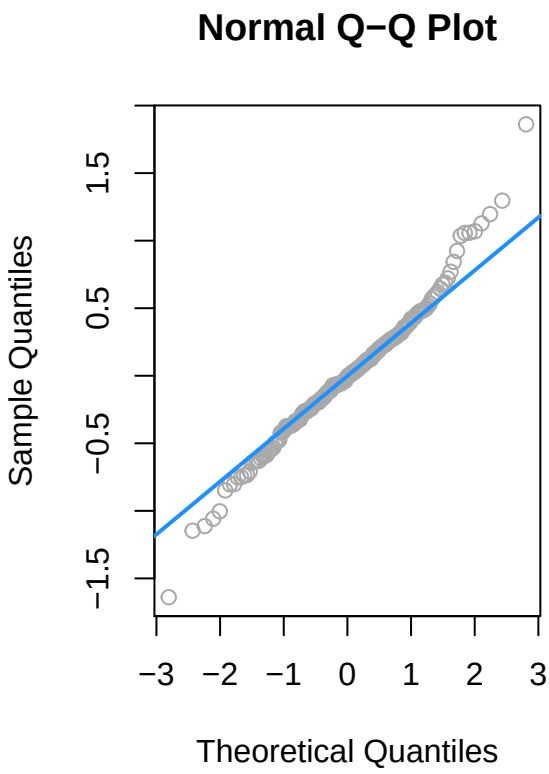
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_q2_2)
## W = 0.96638, p-value = 0.0001911
```

```
##thus removing the influential points did not correct the normality assumption or linearity,
##however the p-values in the test scores were slightly better for shapiro
##but still reject assumption as p val far below alpha=0.01

##f. we use Box-Cot to obtain the desired transmorm of variable and test its assumptions
library(MASS)
par(mfrow=c(1,1))
boxcox(lm_q2, lambda = seq(-1, 1, by = 0.01))
```




```
##by inspection lambda is approximately 0.5 by boxcox
lambda = 0.5
lm_q2_tf <- lm(((y^(lambda)-1)/(lambda))~x1+x2,data=data_q2)

par(mfrow=c(1,2))
plot(resid(lm_q2_tf)~fitted(lm_q2_tf), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

# Normal qq plot - Looks much better
qqnorm(resid(lm_q2_tf), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(lm_q2_tf), col = "dodgerblue", lwd = 2)
```






```
# Both normality appears to hold but equal variance appears to fail
# Equal variance needs more information
bptest(lm_q2_tf)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_q2_tf
## BP = 26.212, df = 2, p-value = 2.033e-06
```

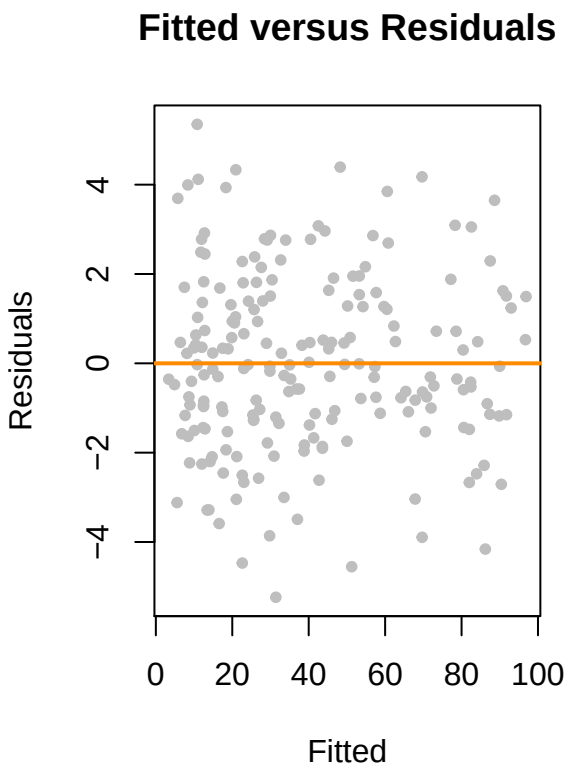
```
#equal variance now fails as p very low by bp test far below alpha=0.01
shapiro.test(resid(lm_q2_tf))
```

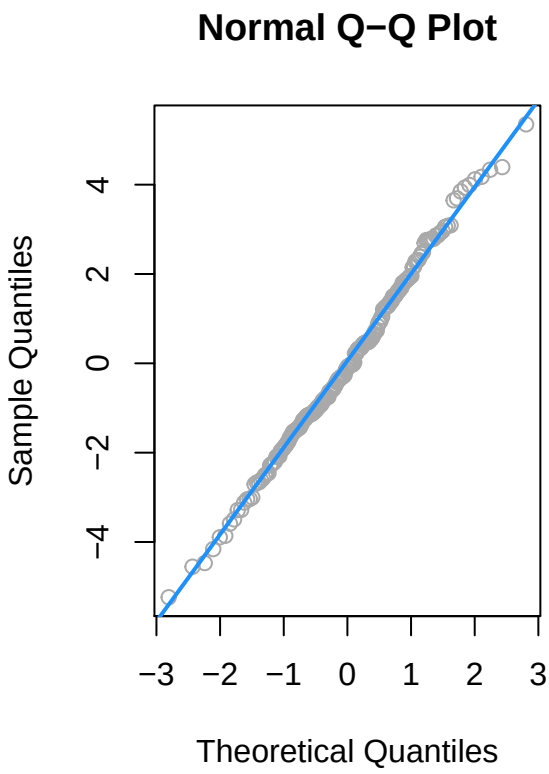
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_q2_tf)
## W = 0.9816, p-value = 0.01006
```

```
##normality holds now for alpha=0.01

#g. This time we obtain a polynomial model and check if the assumptions hold
lm_q2_poly=lm(y~x1+x2+I(x1^2)+I(x2^2),data=data_q2)
summary(lm_q2_poly)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + I(x1^2) + I(x2^2), data = data_q2)
##
## 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
#poly terms are statistically significant
par(mfrow=c(1,2))
plot(resid(lm_q2_poly)~fitted(lm_q2_poly), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
##normality and equal variance appears to hold, and linearity
# Normal qq plot - Looks much better
qqnorm(resid(lm_q2_poly), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(lm_q2_poly), col = "dodgerblue", lwd = 2)
```






```
bptest(lm_q2_poly)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_q2_poly
## BP = 2.6009, df = 4, p-value = 0.6267
```

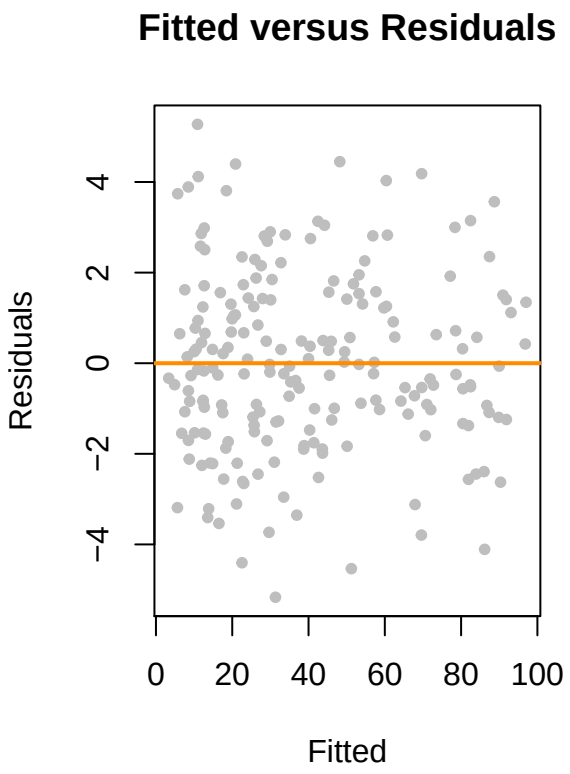
```
#equal variance assumption now holds as p value high
shapiro.test(resid(lm_q2_poly))
```

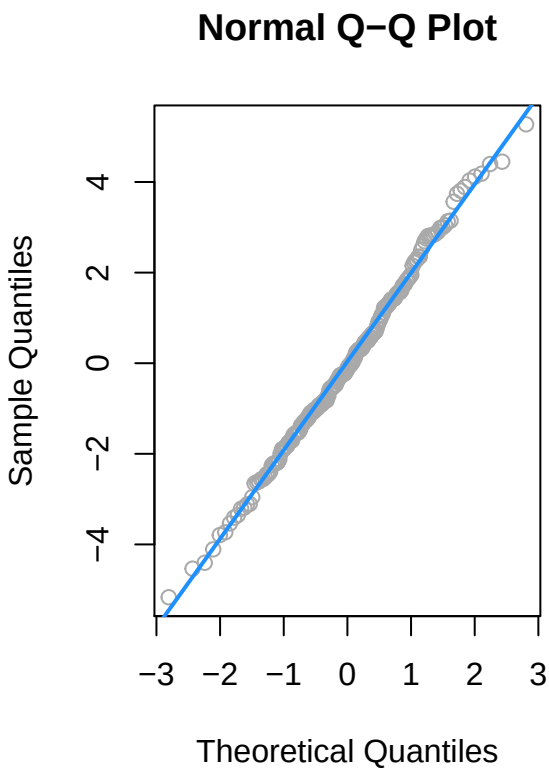
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_q2_poly)
## W = 0.9956, p-value = 0.8331
```

```
#normality assumption now holds as p value high
##this model appears much better than the previous models as
##linearity, normality and equal variance holds

#h.We add cubic terms to the model and compare to the previous model
lm_q2_cubic=lm(y~x1+x2+I(x1^2)+I(x2^2)+I(x1^3)+I(x2^3),data=data_q2)
summary(lm_q2_cubic)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + I(x1^2) + I(x2^2) + I(x1^3) + I(x2^3),
##     data = data_q2)
##
## 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
```

```r
#cubic terms do not appear to be statistically significant
par(mfrow=c(1,2))
plot(resid(lm_q2_cubic)~fitted(lm_q2_cubic), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
##lienarity and equal variance appears to hold, vut need more info
# Normal qq plot - Looks much better
qqnorm(resid(lm_q2_cubic), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(lm_q2_cubic), col = "dodgerblue", lwd = 2)
```






```
##bp test and shapriro for ev assumption and normality
bptest(lm_q2_cubic)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_q2_cubic
## BP = 4.2839, df = 6, p-value = 0.6383
```

```
#equal variance assumption now holds as p value high
shapiro.test(resid(lm_q2_cubic))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_q2_cubic)
## W = 0.99579, p-value = 0.8581
```

```
#normality assumption now holds as p value high
##this model appears much better than the previous models as
##linearity, normality and equal variance holds
```

Question 3

```
#a.We fitter a regression model using cyl,disp,hp,wt,drat as predictors
lm_q3=lm(mpg~cyl+disp+hp+wt+drat,data=mtcars)
lm_disp=lm(disp~cyl+hp+wt+drat,data=mtcars)
##we obtain the VIF for each predictor using vif
library(faraway)
vif(lm_q3)
```

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

```
##colinearity exists, disp has VIF higher than 10
##collinearity affects regression analysis by having each indepedent
##variable effected by another
##thus we cannot determine the relationship between each ind variable
##and the dependent effectively
##so we cannot trust the p values as much to determine which independent
##variables are valuable to the model

#b. We remove the disp var from the model as it had the highest vif
##and compute the VIF again
lm_q3_2=lm(mpg~cyl+hp+wt+drat,data=mtcars)
lm_cyl=lm(cyl~hp+wt+drat,data=mtcars)
lm_hp=lm(hp~cyl+wt+drat,data=mtcars)
lm_wt=lm(wt~hp+cyl+drat,data=mtcars)
lm_drat=lm(drat~wt+hp+cyl,data=mtcars)
VIF_cyl=1/(1-summary(lm_cyl)$r.squared)
VIF_hp=1/(1-summary(lm_hp)$r.squared)
VIF_wt=1/(1-summary(lm_wt)$r.squared)
VIF_drat=1/(1-summary(lm_drat)$r.squared)
VIF_cyl
```

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

```
VIF_drat
```

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

```
VIF_hp
```

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

```
VIF_wt
```

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

```
##There exists mild colinearity as there is a VIF above 5 for cyl
##but there are none above 10
##Thus there is mild collinearity

##c.We find the best subset of predictors to mpg using AIC
m<-lm(mpg~1,mtcars)
stepAIC(m,direction="forward",scope =list(lower=m,upper=~cyl+drat+wt+cyl+hp),trace=0)
```

```
##
## Call:
## lm(formula = mpg ~ wt + cyl + hp, data = mtcars)
##
## Coefficients:
## (Intercept)           wt          cyl           hp
##    38.75179     -3.16697     -0.94162     -0.01804
```

```
##thus the best subset of predictors by forward AIC is wt,cyl,hp as predictors

##d. We find the best subset of predictors using backward BIC
step(lm_q3,criterion="BIC", direction = "backward", k=log(nrow(mtcars)),trace=0)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + wt, data = mtcars)
##
## Coefficients:
## (Intercept)          cyl           wt
##      39.686       -1.508       -3.191
```

```
## using BIC and backward selection the model ysing cyl and wt
lm_c=lm(mpg~wt+cyl+hp,mtcars)
lm_d=lm(mpg~wt+cyl,mtcars)
anova(lm_c,lm_d)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + cyl + hp
## Model 2: mpg ~ wt + cyl
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 176.62
## 2     29 191.17 -1   -14.551 2.3069   0.14
```

```
##thus as the p values is greater than 0.05 the models are
##not statistically different so we are don
```

Question 4

```
library(faraway)
lm_a=lm(lpsa~lcavol+lweight+svi,data=prostate)
```

```r
lm_b=lm(lpsa~lcavol+lweight+svi+lbph,data=prostate)
lm_c=lm(lpsa~lcavol+lweight+svi+lbph+lcp+gleason,data=prostate)

AIC(lm_a,lm_b,lm_c) ##AIC chooses lm_b
```

```
##      df      AIC
## lm_a  5 216.5979
## lm_b  6 215.9223
## lm_c  8 218.9735
```

```r
BIC(lm_a,lm_b,lm_c) # BIC: chooses lm_a
```

```
##      df      BIC
## lm_a  5 229.4714
## lm_b  6 231.3705
## lm_c  8 239.5712
```

```r
# Adjusted_R2: chooses lm_b
summary(lm_a)$adj.r.squared
```

```
## [1] 0.6143899
```

```r
summary(lm_b)$adj.r.squared
```

```
## [1] 0.6208036
```

```r
summary(lm_c)$adj.r.squared
```

```
## [1] 0.6161501
```

```r
## thus best model is b,a,lb by AIC,BIC,R^2 respectively

##b. we find the best model in terms of PRESS
sqrt(sum((resid(lm_a)/(1-hatvalues(lm_a)))^2)/nrow(prostate))
```

```
## [1] 0.7381178
```

```r
sqrt(sum((resid(lm_b)/(1-hatvalues(lm_b)))^2)/nrow(prostate))
```

```
## [1] 0.7355329
```

```r
sqrt(sum((resid(lm_c)/(1-hatvalues(lm_c)))^2)/nrow(prostate))
```

```
## [1] 0.7458586
```

```r
##chooses model b

##c. We find the best model using R^2
summary(lm_a)$r.squared
```

```
## [1] 0.6264403
```

```r
summary(lm_b)$r.squared
```

```
## [1] 0.6366035
```

```r
summary(lm_c)$r.squared
```

```
## [1] 0.6401407
```

```r
##the best model using R^2 is model C
##R^2 is not appropriate for model comparison as it favors models with more predictors
##due to the non decreasing property of R^2
##it is not robust enough for model selection

##d.We find the best model in terms of RMSE using 2-fold

set.seed(10)
rand_index = sample(nrow(prostate))
prostate2 = prostate [rand_index,]

k = 2
RMSE_kcv_a = RMSE_kcv_b = RMSE_kcv_c = numeric(k)

#Create k equally size folds
folds <- cut(1:nrow(prostate),breaks=k,labels=FALSE)

#Perform a k-fold cross validation
for(i in 1:k)
{
  # Find the indices for test data
  smp_size=floor(0.5*nrow(prostate2))
  train_index <- sample(seq_len(nrow(prostate2)),
                        size = smp_size)
  if(i==2){
    test_data = prostate2[train_index, ]
    training_data = prostate2[-train_index, ]
  }else{
    test_data = prostate2[-train_index, ]
    training_data = prostate2[train_index, ]
  }
  # Obtain training/test data

  kcv_a = lm(lpsa~lcavol+lweight+svi, data = training_data)
  kcv_b = lm(lpsa~lcavol+lweight+svi+lbph, data = training_data)
  kcv_c  = lm(lpsa~lcavol+lweight+svi+lbph+lcp+gleason, data = training_data)
```

```
  # Obtain RMSE on the 'test' data
  resid_a = test_data[,2] - predict(kcv_a, newdata=test_data)
  RMSE_kcv_a[i] = sqrt(sum(resid_a^2)/nrow(test_data))

  resid_b = test_data[,2] - predict(kcv_b, newdata=test_data)
  RMSE_kcv_b[i] = sqrt(sum(resid_b^2)/nrow(test_data))

  resid_c = test_data[,2] - predict(kcv_c, newdata=test_data)
  RMSE_kcv_c[i] = sqrt(sum(resid_c^2)/nrow(test_data))
}

# ith value = RMSE_kcv for the ith fold
RMSE_kcv_a
```

```
## [1] 1.474796 1.324250
```

```
# Chooses fit_quad
mean(RMSE_kcv_a)
```

```
## [1] 1.399523
```

```
mean(RMSE_kcv_b)
```

```
## [1] 1.399578
```

```
mean(RMSE_kcv_c)
```

```
## [1] 1.411719
```

```
##the 2-fold CV in terms of RMSE picks model a
```