Practice Final Exam

Instructions

Open book and open notes exam (textbooks (print or pdf), lecture slides, notes, practice exam, homework solutions, and TA slides, including all Rmd's*).

You are allowed to use RStudio Cloud (https://rstudio.cloud.) , Microsoft Word, Power Point and PDF reader, and canvas on your laptop.

Proctorio is required to start this exam. If you are prompted for an access code, you must Configure Proctorio on your machine.

Please read the list of recording and restrictions provided by Proctorio carefully before taking the exam

Please pay attention any timing and technical warnings that popped up your screen

The exam will be available from Monday December 14th at 8 am EST through Tuesday December 15th at 8:00pm EST. Once you start the exam, you have to complete the exam in 3 hours or by Tuesday December 15th at 8:00pm EST, whichever comes first.

In order to receive full credit, please provide full explanations and calculations for each questions

Make sure that you are familiar with the procedures for troubleshooting exam issues Preview the document Make sure you submit both .Rmd and (knitted) pdf or html files.

You need to have a camera on your laptop.

Problem 1

Use the question1 data, fit the regression model on Y by using all the variables (X6 and X7 are categorical variables). Create development sample (70% of the data) and hold-out sample (30% of the data). Perform statistical tests, use graphs or calculate the measures (e.g. VIF, Leverage Points, Cook's Distance) for questions below. Use the development sample for part a to d. Use the hold-out sample for part e. Use seed 1234.

a-) Is the model significant? Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance?

X_8 is significantly correlated with X_9 and X_{10}

X_1, X_2, X_7, X_8, and X_9 are significant variables. R^2 is 70%. QQ plot indicates S function and suggests that Y should be transformed. Errors have constant variance. There are outliers in the data set. There is a multicollinearity issue in the data set. X_8 should be dropped from the model.

```
library(datasets)


PF.Q1.Dat <- read.csv("/cloud/project/Practice Final Question 1.csv")
round(cor(PF.Q1.Dat),2)
```

```
##         Y    X1    X2    X3    X4    X5    X6    X7    X8    X9   X10
## Y    1.00  0.19  0.53  0.33  0.38  0.41 -0.30 -0.49  0.47  0.34  0.36
## X1   0.19  1.00  0.00 -0.23 -0.02 -0.06  0.15 -0.02 -0.05 -0.08 -0.04
## X2   0.53  0.00  1.00  0.56  0.45  0.36 -0.23 -0.19  0.38  0.39  0.41
## X3   0.33 -0.23  0.56  1.00  0.42  0.14 -0.24 -0.31  0.14  0.20  0.19
## X4   0.38 -0.02  0.45  0.42  1.00  0.05 -0.09 -0.30  0.06  0.08  0.11
## X5   0.41 -0.06  0.36  0.14  0.05  1.00 -0.59 -0.11  0.98  0.92  0.79
## X6  -0.30  0.15 -0.23 -0.24 -0.09 -0.59  1.00  0.10 -0.61 -0.59 -0.52
## X7  -0.49 -0.02 -0.19 -0.31 -0.30 -0.11  0.10  1.00 -0.15 -0.11 -0.21
## X8   0.47 -0.05  0.38  0.14  0.06  0.98 -0.61 -0.15  1.00  0.91  0.78
## X9   0.34 -0.08  0.39  0.20  0.08  0.92 -0.59 -0.11  0.91  1.00  0.78
## X10  0.36 -0.04  0.41  0.19  0.11  0.79 -0.52 -0.21  0.78  0.78  1.00
```

```
RNGversion("3.5.2")
```

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```
set.seed(994)

n<-dim(PF.Q1.Dat)[1]
#dummy variables
table(PF.Q1.Dat$X6)
```

```
##
##  1  2
## 17 96
```

```
table(PF.Q1.Dat$X7)
```

```
##
##  1  2  3  4
## 28 32 37 16
```

```
#X7 gas 4 levels and need to create dummy variables or use factor command
#creating the dummy variables
library('fastDummies')
Q1.Dat<-dummy_cols(PF.Q1.Dat, select_columns = 'X6')
Q1.Dat<-dummy_cols(Q1.Dat, select_columns = 'X7')
#this will create dummy variables for all levels, but we need to drop one level and drop X6 and X7
Q1.Dat<-Q1.Dat[,-c(7,8,12,14)]
head(Q1.Dat)
```

```
##       Y   X1  X2   X3    X4  X5  X8  X9 X10 X6_2 X7_2 X7_3 X7_4
## 1  7.13 55.7 4.1  9.0  39.6 279 207 241  60    1    0    0    1
## 2  8.82 58.2 1.6  3.8  51.7  80  51  52  40    1    1    0    0
## 3  8.34 56.9 2.7  8.1  74.0 107  82  54  20    1    0    1    0
## 4  8.95 53.7 5.6 18.9 122.8 147  53 148  40    1    0    0    1
## 5 11.20 56.5 5.7 34.5  88.9 180 134 151  40    1    0    0    0
## 6  9.76 50.9 5.1 21.9  97.0 150 147 106  40    1    1    0    0
```

```
IND=sample(c(1:n),n*0.7)
Q1.Dev<-Q1.Dat[IND,]
Q1.Hold<-Q1.Dat[-IND,]

f1<-lm(Y~ X1+X2+X3+X4+X5+X6_2+X7_2+X7_3+X7_4+X8+X9+X10 ,data=Q1.Dev)
```

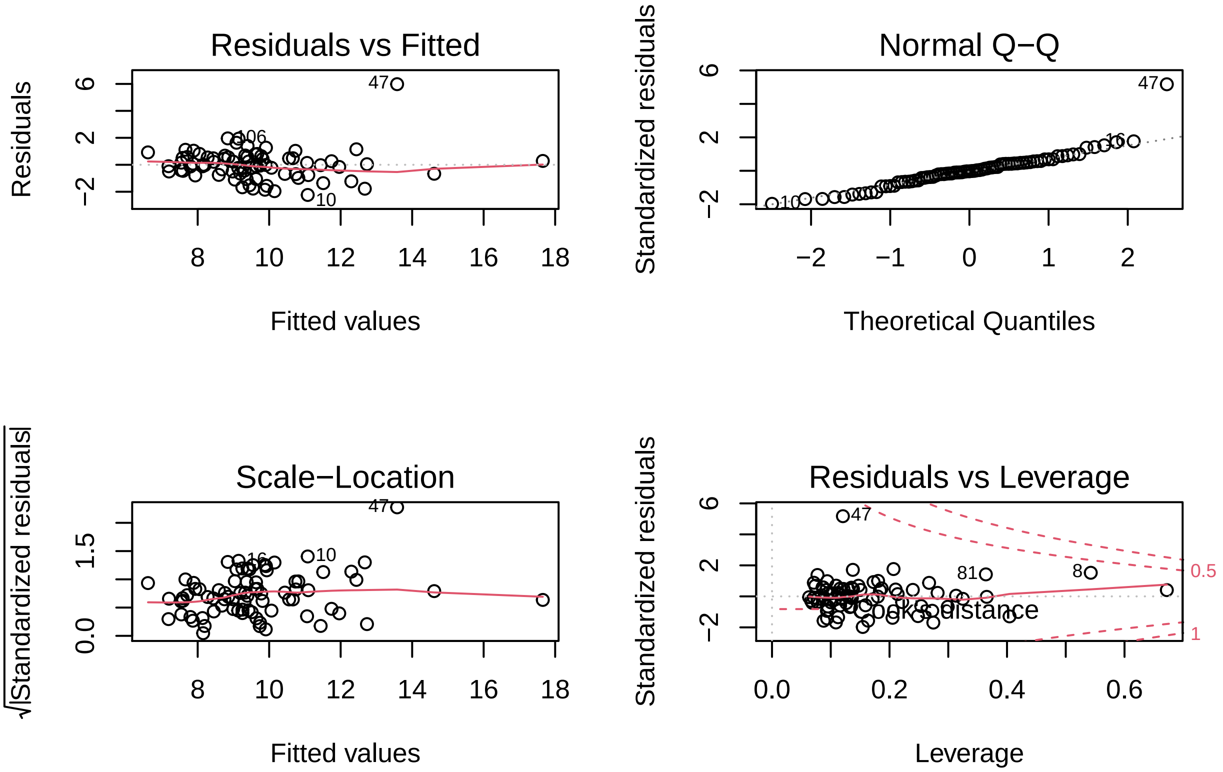
```
summary(f1)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6_2 + X7_2 + X7_3 +
##     X7_4 + X8 + X9 + X10, data = Q1.Dev)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2383 -0.6587 -0.0301  0.5473  5.9806
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.702989   2.097476   1.289 0.202006
## X1           0.101113   0.033131   3.052 0.003274 **
## X2           0.435583   0.158670   2.745 0.007783 **
## X3          -0.003797   0.019312  -0.197 0.844726
## X4           0.011028   0.008873   1.243 0.218340
## X5          -0.007166   0.004469  -1.604 0.113590
## X6_2        -0.694743   0.602269  -1.154 0.252848
## X7_2        -0.677516   0.430622  -1.573 0.120422
## X7_3        -1.342212   0.427386  -3.141 0.002524 **
## X7_4        -2.259871   0.548626  -4.119 0.000108 ***
## X8           0.018641   0.005418   3.441 0.001011 **
## X9          -0.006406   0.002815  -2.275 0.026133 *
## X10         -0.005829   0.015827  -0.368 0.713843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.233 on 66 degrees of freedom
## Multiple R-squared:  0.7125, Adjusted R-squared:  0.6602
## F-statistic: 13.63 on 12 and 66 DF,  p-value: 1.341e-13
```

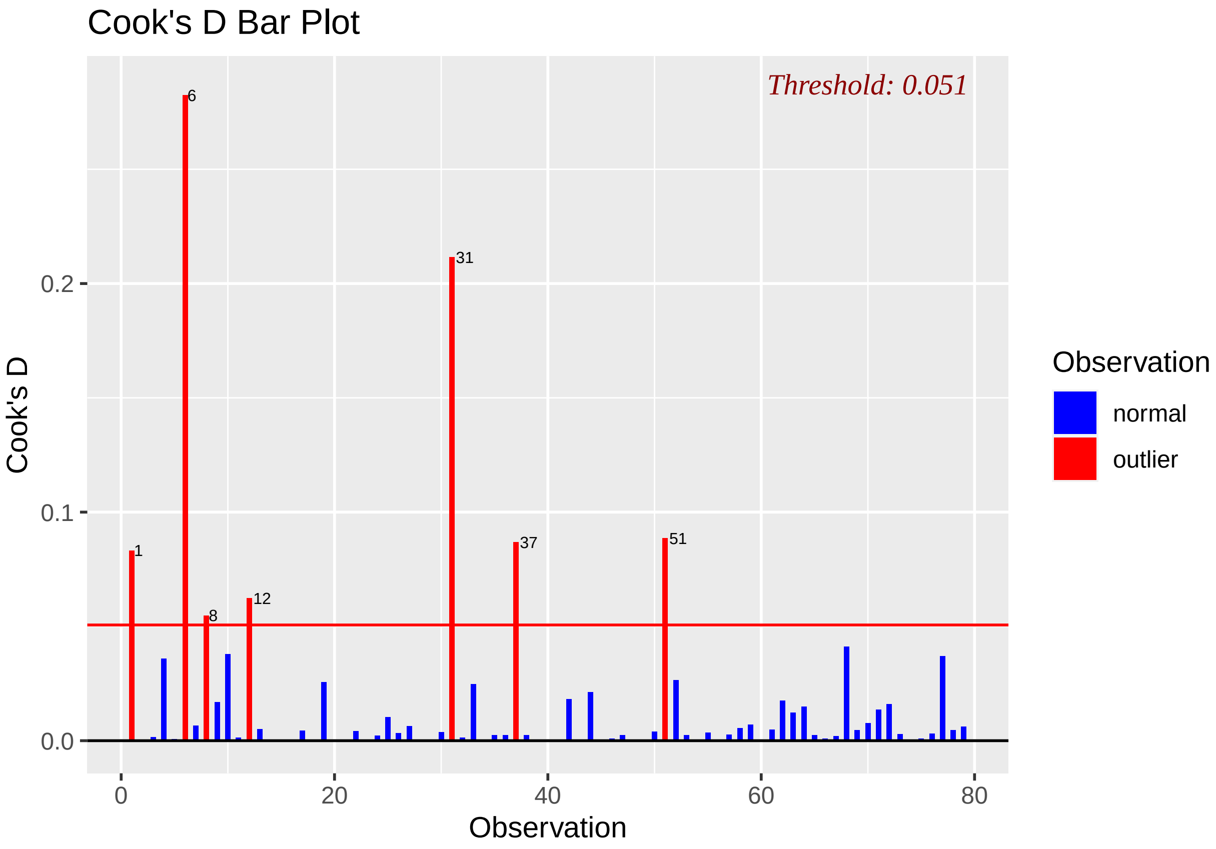
```
par(mfrow=c(2,2))
plot(f1)

library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

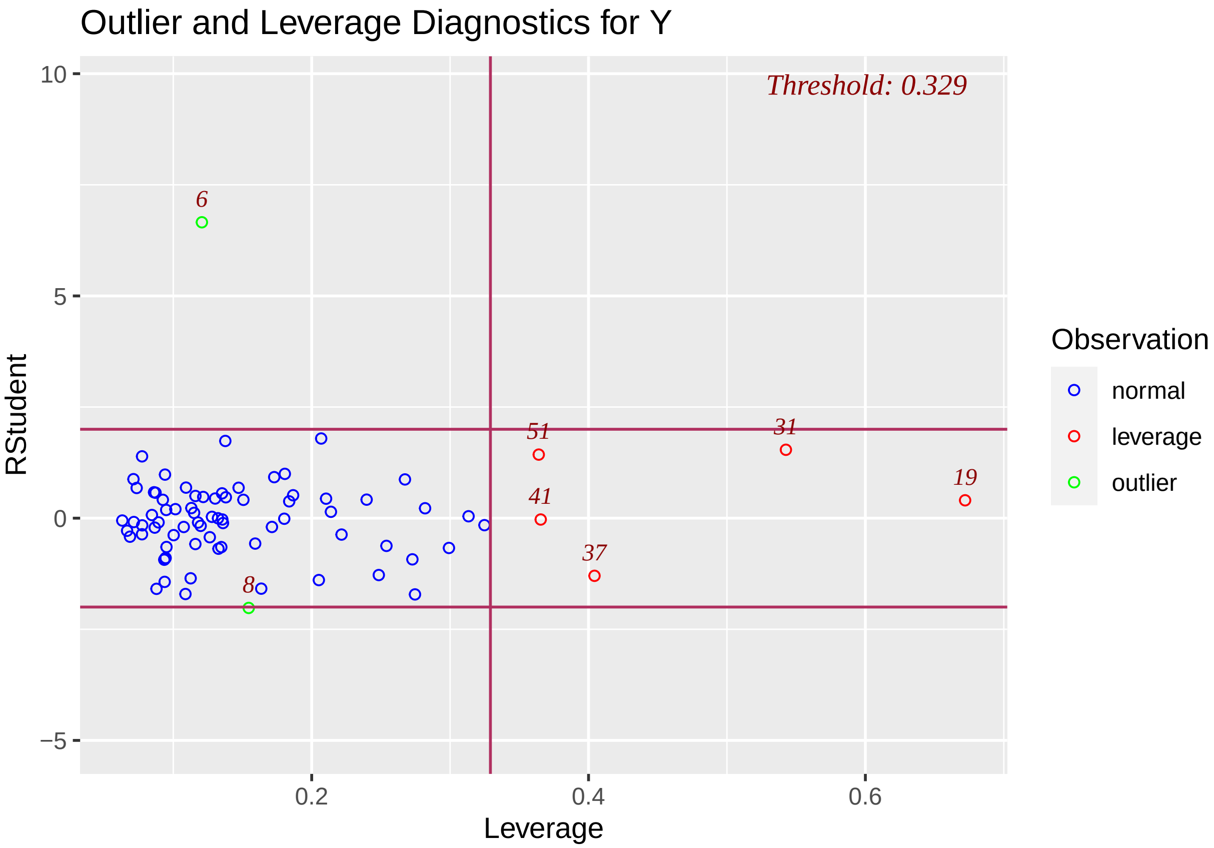
```
## The following object is masked from 'package:datasets':
##
##     rivers
```



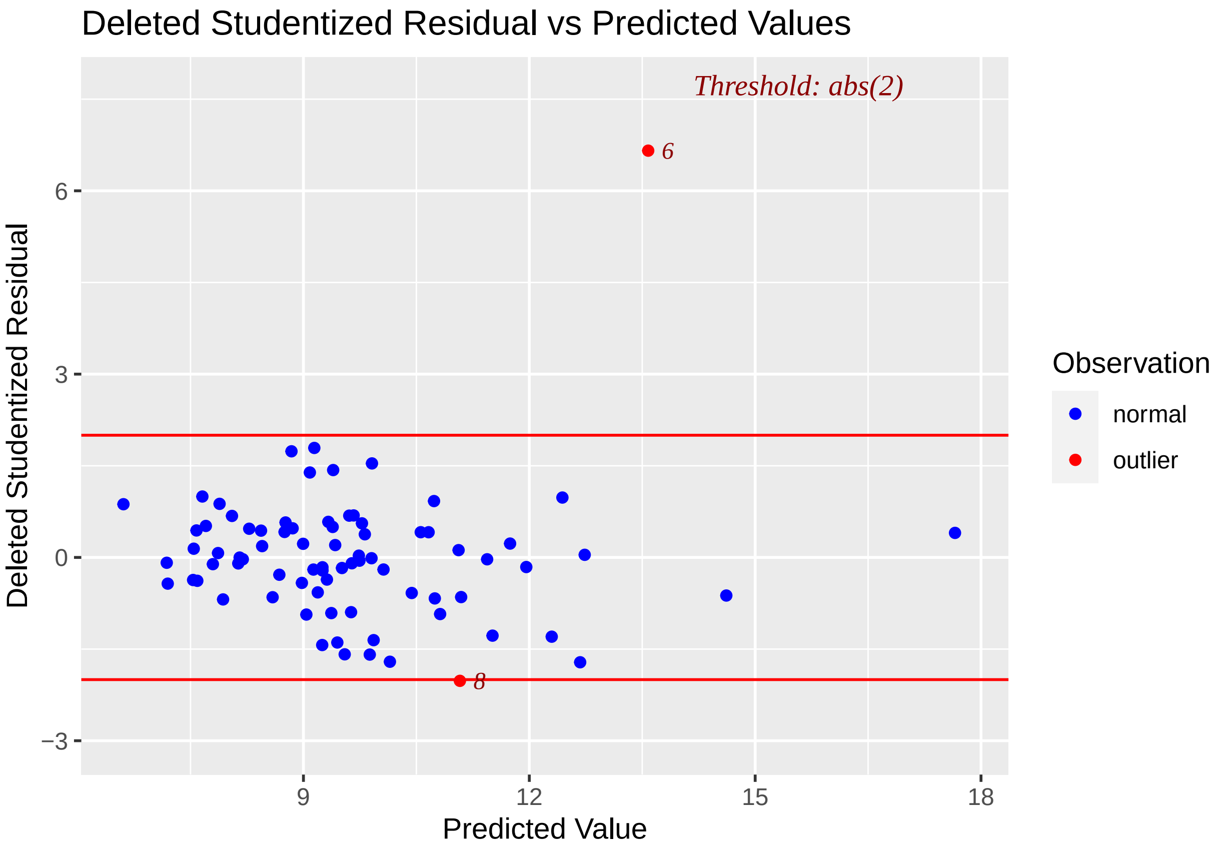
```r
ols_plot_cooksd_bar(f1)
```



```r
ols_plot_resid_lev(f1)
```



```r
ols_plot_resid_stud_fit(f1)
```



```
library(faraway)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

##
## Attaching package: 'faraway'

## The following object is masked from 'package:olsrr':
##
##     hsb
```

```
vif(f1)
```

```
##        X1        X2        X3        X4        X5      X6_2      X7_2      X7_3
##  1.170210  2.541432  2.121925  1.702076 33.035100  1.902758  1.880628  2.135358
##      X7_4        X8        X9       X10
##  1.729277 31.335687  6.687057  3.089231
```

b-) Are there any influential or outlier observations?

Yes, based on the Cook's distance observations 6 and 31 are influential points.

c-) Can X5, X6, and X7 be dropped from the model? Perform the statistical test and state your final model.

Ho: Variables can be dropped Ha: Variables cannot be dropped

Reject Ho. Variables cannot be dropped from the model.

```
f1.r<-lm(Y~ X1+X2+X3+X4+X8+X9+X10,data=Q1.Dev)
anova(f1.r,f1)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X1 + X2 + X3 + X4 + X8 + X9 + X10
## Model 2: Y ~ X1 + X2 + X3 + X4 + X5 + X6_2 + X7_2 + X7_3 + X7_4 + X8 +
##     X9 + X10
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     71 138.75
## 2     66 100.33  5    38.418 5.0542 0.0005555 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

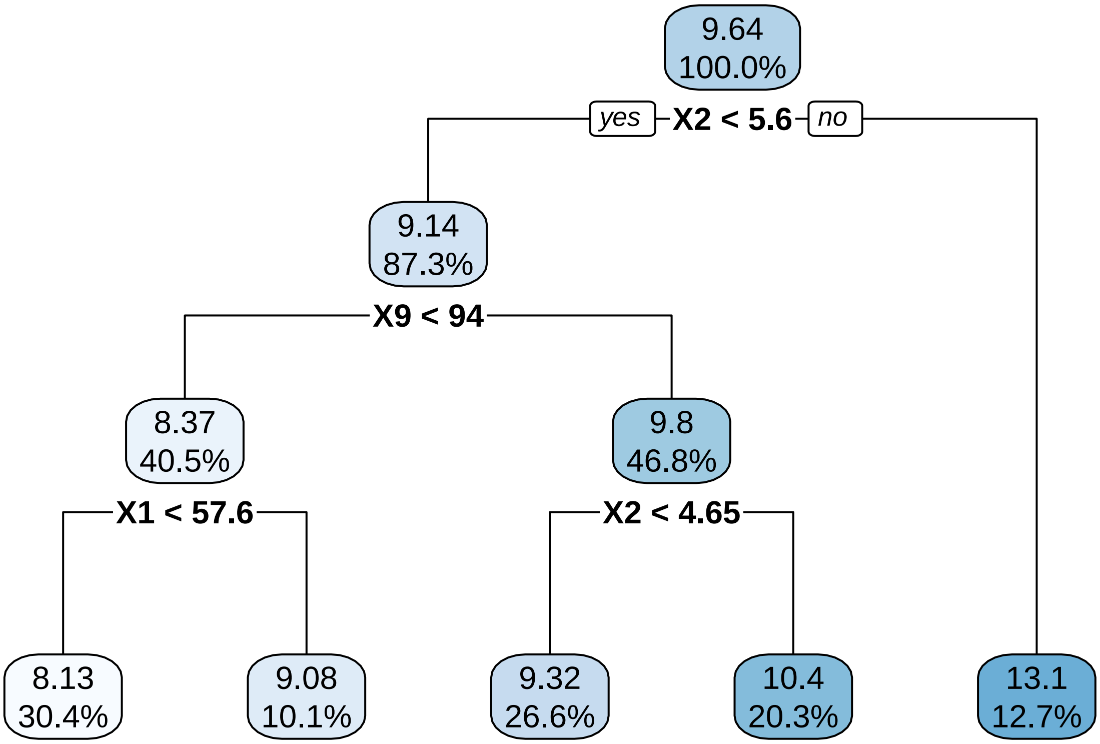
d-) Develop an alternative model by using the Regression Tree and compare the performance against the regression model built in part a.

In the hold out sample, regression model performs better than the tree method using R^2, it has also smallest SSE.

```
library(rpart)
```

```
##
## Attaching package: 'rpart'

## The following object is masked from 'package:faraway':
##
##     solder
```

```
q1.tr<-rpart(Y~.,data=Q1.Dev)
library(rpart.plot)
par(mfrow=c(1,1))
rpart.plot(q1.tr,digits = 3)
```



```
p.tree.dev<-predict(q1.tr,Q1.Dev)
p.reg.dev<-predict(f1,Q1.Dev)

#Measuring performance with the RSquare

R2 <- function(actual, predicted) {sum((actual - predicted)^2)/((length(actual)-1)*var(actual))}
R2(Q1.Dev$Y,p.tree.dev)
```

```
## [1] 0.4544507
```

```
cbind((1-R2(Q1.Dev$Y,p.tree.dev)),summary(f1)$adj.r.squared)
```

```
##           [,1]      [,2]
## [1,] 0.5455493 0.6601881
```

```
#Measuring performance with the SSE
SSE.Tree.Dev<-sum((predict(q1.tr)-Q1.Dev$Y)^2)
SSE.Tree.Dev
```

```
## [1] 158.5817
```

```
SSE.Reg.Dev<-anova(f1)$`Sum Sq`[length(anova(f1)$`Sum Sq`)]
cbind(SSE.Tree.Dev,SSE.Reg.Dev)
```

```
##      SSE.Tree.Dev SSE.Reg.Dev
## [1,]     158.5817    100.3354
```

f-) Score the model on hold-out sample and compare the results against the final model derived in part c.

Since we are comparing two models on the hold out sample. We can use SSE as a measure to compare both models. The regresssion has the smallest SSE and has to be chosen.

```
p.tree.hold<-predict(q1.tr,Q1.Hold)
p.reg.hold<-predict(f1,Q1.Hold)
```

```r
#Measuring performance with the SSE
SSE.Tree.Hold<-sum((p.tree.hold-Q1.Hold$Y)^2)
SSE.Reg.Hold<-sum((p.reg.hold-Q1.Hold$Y)^2)
cbind(SSE.Tree.Hold,SSE.Reg.Hold)
```

```
##      SSE.Tree.Hold SSE.Reg.Hold
## [1,]      111.6806      63.8573
```

Problem 2

Use the question2 data set to answer this question. We are interested in predicting (Y) the number of customers who complained about the service.

a-) Build a model to predict the number of complaints, perform the statistical tests that shows that model is significant

It is a poisson regression model since the dependent variable is a count data. All variables and model are significant.

```r
PF.Q2.Dat <- read.csv("/cloud/project/Practice Final Question 2.csv")
f2<-glm(Y~.,data=PF.Q2.Dat,family=poisson)
summary(f2)
```

```
## 
## Call:
## glm(formula = Y ~ ., family = poisson, data = PF.Q2.Dat)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.93195  -0.58868  -0.00009   0.59269   2.23441  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  2.942e+00  2.072e-01  14.198  < 2e-16 ***
## X1           6.058e-04  1.421e-04   4.262 2.02e-05 ***
## X2          -1.169e-05  2.112e-06  -5.534 3.13e-08 ***
## X3          -3.726e-03  1.782e-03  -2.091   0.0365 *  
## X4           1.684e-01  2.577e-02   6.534 6.39e-11 ***
## X5          -1.288e-01  1.620e-02  -7.948 1.89e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 422.22  on 109  degrees of freedom
## Residual deviance: 114.99  on 104  degrees of freedom
## AIC: 571.02
## 
## Number of Fisher Scoring iterations: 4
```

b-) Find the predicted number complaints given the independent variables below and predict 95% confidence interval

X1=606 X2=41393 X3=3 X4=3.04 X5=6.32

Please see below

```
test.dat<-data.frame(X1=606,X2=41393,X3=3,X4=3.04,X5=6.32)
pred<-predict(f2,test.dat,type="link",se.fit = TRUE)

exp(pred$fit)
```

```
##        1
## 12.33778
```

```
critval <- round(qnorm(1-.05/2),2)#1.96 approx 95% CI
critval
```

```
## [1] 1.96
```

```
upr <- exp(pred$fit + (critval * pred$se.fit))
lwr <- exp(pred$fit - (critval * pred$se.fit))
cbind(lwr,upr)
```

```
##        lwr      upr
## 1 11.08404 13.73332
```

Problem 3

Use question 3 data sets. Monthly data on amount of billings (Y) and on number of hours of staff time (X) for the 20 most recent months are recorded.

a-) Build a model to predict Y based on the independent variables and test if there is an autocorrelation persists in the data. If autocorrelation persists, remediate the autocorrelation.

There is autocorrelation in the data set. I will use Cochrane-Orcutt procedure to eliminate it. The suggested ρ is 0.33 and autocorrelation is remediated.

```
PF.Q3.Dat <- read.csv("/cloud/project/Practice Final Question 3.csv")
f3<-lm(Y~X,data=PF.Q3.Dat)
summary(f3)
```

```
## 
## Call:
## lm(formula = Y ~ X, data = PF.Q3.Dat)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.55515 -0.23700  0.05229  0.56250  0.80657 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  93.6865     0.8229   113.8   <2e-16 ***
## X            50.8801     0.2634   193.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.631 on 18 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995 
## F-statistic: 3.73e+04 on 1 and 18 DF,  p-value: < 2.2e-16
```

```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
dwtest(f3)
```

```
##
##  Durbin-Watson test
##
## data:  f3
## DW = 0.97374, p-value = 0.002891
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
#manual solution
library(Hmisc)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:faraway':
##
##     melanoma

## Loading required package: survival

##
## Attaching package: 'survival'

## The following objects are masked from 'package:faraway':
##
##     rats, solder

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
et<-f3$residuals
et1<-Lag(et, shift = 1)

d1<-sum(na.omit(et1*et))
d2<-sum(na.omit(et1)^2)
rho<-d1/d2

Ytnew=PF.Q3.Dat$Y - rho*Lag(PF.Q3.Dat$Y , shift = 1)
Xtnew=PF.Q3.Dat$X - rho*Lag(PF.Q3.Dat$X , shift = 1)
```

```r
f3.1<-lm(Ytnew~Xtnew)
summary(f3.1)
```

```
##
## Call:
## lm(formula = Ytnew ~ Xtnew)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95813 -0.29553 -0.02312  0.34451  0.60490
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  63.3840     0.5592   113.4   <2e-16 ***
## Xtnew        50.5470     0.2622   192.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4546 on 17 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 3.715e+04 on 1 and 17 DF,  p-value: < 2.2e-16
```

```r
dwtest(f3.1)
```

```
##
##  Durbin-Watson test
##
## data:  f3.1
## DW = 1.7612, p-value = 0.2337
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
#Aletnatively
#use the function
library(orcutt)
coch<- cochrane.orcutt(f3)
summary(coch)
```

```
## Call:
## lm(formula = Y ~ X, data = PF.Q3.Dat)
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 95.16377    0.91343  104.18 < 2.2e-16 ***
## X           50.46593    0.28415  177.61 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4514 on 17 degrees of freedom
## Multiple R-squared:  0.9995 ,  Adjusted R-squared:  0.9994
## F-statistic: 31543.9 on 1 and 17 DF,  p-value: < 3.137e-29
##
## Durbin-Watson statistic
## (original):    0.97374 , p-value: 2.891e-03
## (transformed): 1.96762 , p-value: 4.079e-01
```

b-) X (Staff time) in month 21 is expected to be 3.625 thousand hours. Predict the amount of billings in constant dollars for month 21, using a 99 percent prediction intervaL Interpret your interval.

```
b0 <- summary(f3.1)[[4]][1,1]/(1-rho)
b1 <- summary(f3.1)[[4]][2,1]
MSE<-summary(f3.1)$sigma^2

X.prime<-Xtnew
X.bar.prime <- mean(X.prime[-1])

X.n.plus.1 <- 3.625
X.n <- rev(PF.Q3.Dat$X)[1]
X.n.plus.1.prime <- X.n.plus.1 - rho*X.n

# Point forecast:

Y.hat.n.plus.1 <- b0 + b1*X.n.plus.1
Y.n <- rev(PF.Q3.Dat$Y)[1]
e.n <- Y.n - (b0 + b1*X.n)
Y.hat.FORECAST.n.plus.1 <- Y.hat.n.plus.1 + rho*e.n

print(paste("forecasted response at time n+1 is:", round(Y.hat.FORECAST.n.plus.1,4) ))
```

```
## [1] "forecasted response at time n+1 is: 278.3537"
```

```
# Prediction interval:

alpha <- 0.01
n<-length(PF.Q3.Dat$X)
s.pred <- sqrt(MSE*(1 + (1/n) + (X.n.plus.1.prime -X.bar.prime)^2/(sum((X.prime[-1]-X.bar.prime)^2))))
s.pred
```

```
## [1] 0.4737689
```

```
pred.L <- Y.hat.FORECAST.n.plus.1 - qt(1-alpha/2,df=n-3)*s.pred
pred.U <- Y.hat.FORECAST.n.plus.1 + qt(1-alpha/2,df=n-3)*s.pred

print(paste(100*(1-alpha) ,"percent PI for response at time n+1 is:", round(pred.L,4), ",", round(pred.U
```

```
## [1] "99 percent PI for response at time n+1 is: 276.9807 , 279.7268"
```

Problem 4

Use question 4 data set, Create development sample (70% of the data) and hold-out sample (30% of the data) use set.seed(1023) before creating the samples.

a-) Use the development sample , fit a linear regression model, regression tree and Neural Network Model, and calculate the SSE for each model, which method has the lowest SSE?

Reg Model: X_1,X_2,X_3 and X_4 are significant and R^2 square is 56%. QQ plot indicates S shape suggesting that transformation is needed. Residual vs Fitted graph suggest that furhter testing for unequal variances are needed. However, there is no multicolinearity in the data. SSE is 10128646733.

Tree: It is based only 2 variables (X_1 and X_2).

Neural Network:

I used 2 hidden layers with 5 nodes each, you can also try single layer or multiple layer. SSE is 22389967988. It has the lowest SSE. Highest R^2

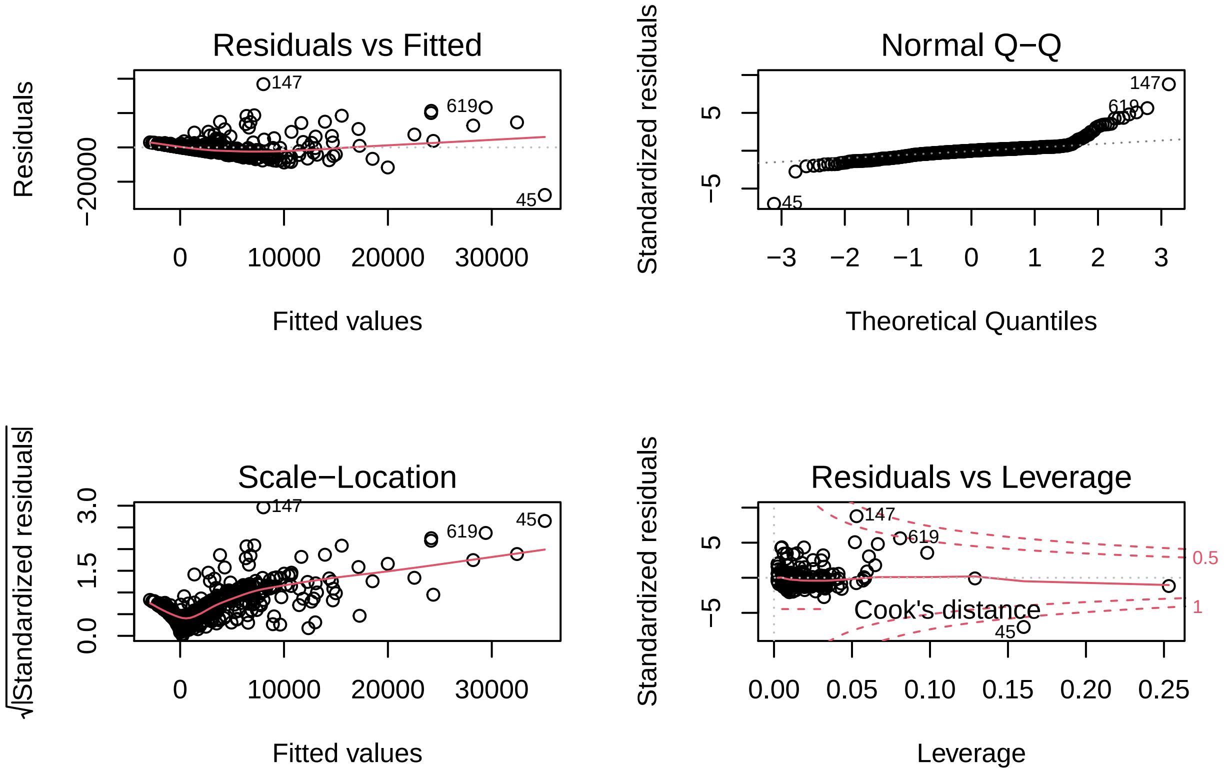
```r
PF.Q4.Dat<- read.csv("/cloud/project/Practice Final Question 4.csv")
n<-dim(PF.Q4.Dat)[1]

RNGversion("3.5.2")
```

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```r
set.seed(1023)
IND=sample(c(1:n),n*0.7)

Q4.Dev<-PF.Q4.Dat[IND,]
Q4.Hold<-PF.Q4.Dat[-IND,]

#regression model
f4<-lm(Y~X1+X2+X3+X4+X5,data=Q4.Dev)
summary(f4)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = Q4.Dev)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27714  -1610      0   1116  36826
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2689.67     335.28  -8.022 6.40e-15 ***
## X1            749.07      35.52  21.089  < 2e-16 ***
## X2           -481.61     206.68  -2.330   0.0202 *
## X3            535.33      91.07   5.878 7.23e-09 ***
## X4            557.50     727.24   0.767   0.4437
## X5             68.22      33.62   2.029   0.0430 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4311 on 545 degrees of freedom
## Multiple R-squared:  0.5647, Adjusted R-squared:  0.5607
## F-statistic: 141.4 on 5 and 545 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(f4)
```



```r
vif(f4)
```

```
##       X1       X2       X3       X4       X5 
## 1.210194 1.489820 1.607477 1.078608 1.038691
```

```r
a<-summary(f4)$adj.r.squared
SSE.Reg.Dev<-anova(f4)$`Sum Sq`[length(anova(f4)$`Sum Sq`)]
SSE.Reg.Dev
```

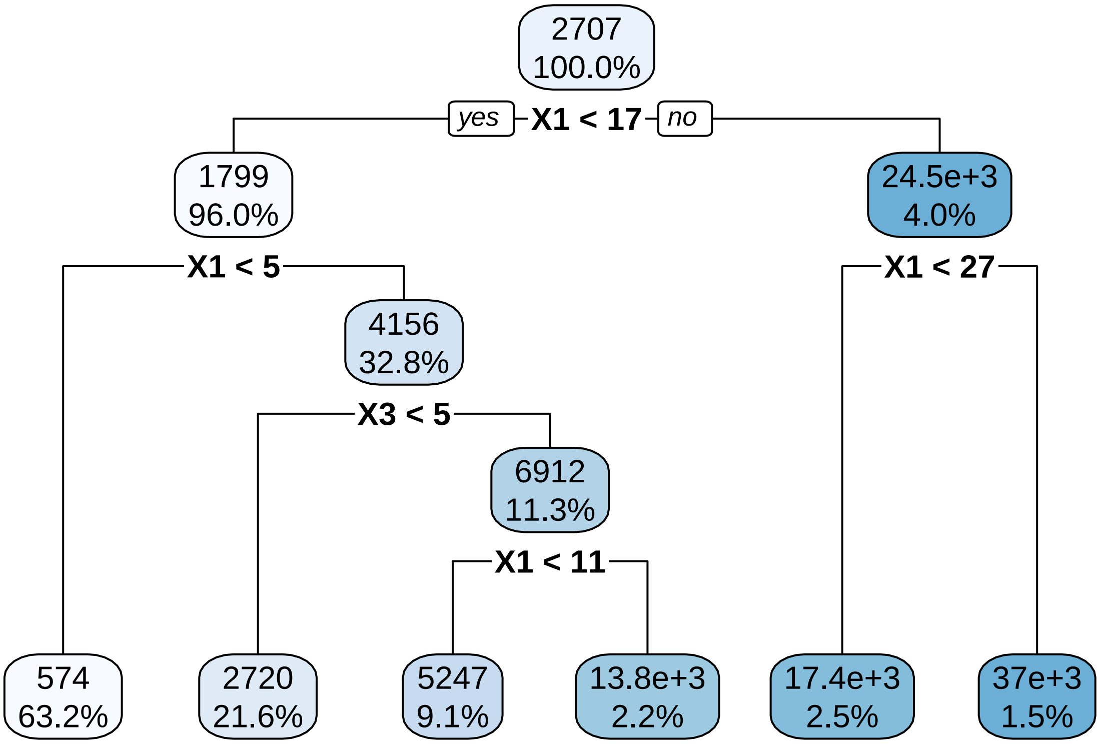
```
## [1] 10128646733
```

```r
summary(f4)
```

```
## 
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = Q4.Dev)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -27714  -1610      0   1116  36826 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2689.67     335.28  -8.022 6.40e-15 ***
## X1            749.07      35.52  21.089  < 2e-16 ***
## X2           -481.61     206.68  -2.330   0.0202 *  
## X3            535.33      91.07   5.878 7.23e-09 ***
## X4            557.50     727.24   0.767   0.4437    
## X5             68.22      33.62   2.029   0.0430 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4311 on 545 degrees of freedom
## Multiple R-squared:  0.5647, Adjusted R-squared:  0.5607
## F-statistic: 141.4 on 5 and 545 DF,  p-value: < 2.2e-16
```

```
#Tree
library(rpart)
q4.tr<-rpart(Y~X1+X2+X3+X4+X5,data=Q4.Dev)
library(rpart.plot)
par(mfrow=c(1,1))
rpart.plot(q4.tr,digits = 3)
```



```
#Measuring performance with the RSquare
b<-R2(Q4.Dev$Y,predict(q4.tr))
SSE.Tree.Dev<-sum((predict(q4.tr)-Q4.Dev$Y)^2)
SSE.Tree.Dev
```

```
## [1] 7450047035
```

```
#Neural Network
#install.packages("neuralnet")
library(neuralnet)
normalize <- function(x) {return((x - min(x)) / (max(x) - min(x)))}
scaled.Q4.Dat <- as.data.frame(lapply(PF.Q4.Dat, normalize))
scaled.Q4.Dev<- scaled.Q4.Dat[IND,]
scaled.Q4.Hold<- scaled.Q4.Dat[-IND,]

NN = neuralnet(Y~X1+X2+X3+X4+X5,hidden=c(5,5),scaled.Q4.Dev,linear.output= T )
plot(NN)
predict_testNN= compute(NN, scaled.Q4.Dev[,-c(1)])
#we need to transform it back to orginal scale
predict_testNN1 = (predict_testNN$net.result* (max(PF.Q4.Dat$Y) -min(PF.Q4.Dat$Y))) + min(PF.Q4.Dat$Y)
```

```r
plot(scaled.Q4.Dev$Y, predict_testNN1, col='blue', pch=16, ylab= "Predicted Y", xlab= "Actual Y")
#Measuring performance with the RSquare
c<-R2(Q4.Dev$Y,predict_testNN1)
SSE.NN.Dev<-sum((predict_testNN1-Q4.Dev$Y)^2)
SSE.NN.Dev
```

```
## [1] 4663684919
```

```r
#Measuring performance in the Hold out sample with the RSquare criteria: a=regression, b=reg.tree and c

cbind(a,1-b,1-c)
```

```
##              a
## [1,] 0.5606646 0.679788 0.7995492
```

```r
cbind(SSE.Reg.Dev,SSE.Tree.Dev,SSE.NN.Dev)
```

```
##      SSE.Reg.Dev SSE.Tree.Dev SSE.NN.Dev
## [1,] 10128646733   7450047035 4663684919
```

```r
#10,128,646,733   7,450,047,035   4,663,684,919
```

b-) test the models performances on the hold out sample, which model would you choose?

NN has the lower SSE, I would choose the NN approach. It outperforms other models.

```r
SSE <- function(actual, predicted) {sum((actual - predicted)^2)}
#Regression
f4<-lm(Y~X1+X2+X3+X4+X5,data=Q4.Dev)
reg.predict<-predict(f4,Q4.Hold)
REG=SSE(Q4.Hold$Y,reg.predict)

#Tree
tree.predict<-predict(q4.tr,Q4.Hold)
Tree=SSE(Q4.Hold$Y,tree.predict)

#NN
nn = neuralnet(Y~X1+X2+X3+X4+X5,hidden=c(5,5),scaled.Q4.Hold,linear.output= T )

nn.predict<-compute(nn, scaled.Q4.Hold)
nn.predict1 = (nn.predict$net.result*(max(PF.Q4.Dat$Y) -min(PF.Q4.Dat$Y))) + min(PF.Q4.Dat$Y)
NN=SSE(Q4.Hold$Y,nn.predict1)

cbind(REG,Tree,NN)
```

```
##             REG       Tree         NN
## [1,] 5958240574 5497451623 1279306638
```

```r
#5,958,240,574 5,497,451,623 1,279,306,638
```

Problem 5

Use Question 5 dataset, Y is a dichotomous response variable and X2, X3, and X4 are categorical variables.

a-) Fit a regression model containing the predictor variables in first-order terms and interaction terms for all pairs of predictor variables on development sample.

Y is a dichotomous response variable. Therefore, we will use logistic regression model. X_2 has 3 levels and other two categorical variables has two levels. X_3 is coded 1-2. X_4 is coded 0-1. We need to create two dummy variables for X_2 and change the levels of X_3 to 0-1.

Only X_2 is significant. Please see below

```
PF.Q5.Dat<- read.csv("/cloud/project/Practice Final Question 5.csv")
table(PF.Q5.Dat$X2)
```

```
##
##  1  2  3
## 77 49 70
```

```
table(PF.Q5.Dat$X3)
```

```
##
##   1   2
## 117  79
```

```
table(PF.Q5.Dat$X4)
```

```
##
##   0   1
## 139  57
```

```
library('fastDummies')
Q5.Dat<-dummy_cols(PF.Q5.Dat, select_columns = 'X2')
Q5.Dat<-dummy_cols(Q5.Dat, select_columns = 'X3')

head(Q5.Dat)
```

```
##   X1 X2 X3 X4 Y X2_1 X2_2 X2_3 X3_1 X3_2
## 1 33  1  1  0 1    1    0    0    1    0
## 2 35  1  1  0 1    1    0    0    1    0
## 3  6  1  1  0 0    1    0    0    1    0
## 4 60  1  1  0 1    1    0    0    1    0
## 5 18  3  1  1 0    0    0    1    1    0
## 6 26  3  1  0 0    0    0    1    1    0
```

```
#drop X2, X3,X2_1, and X3_1
Q5.Dat1<-Q5.Dat[,-c(2,3,6,9)]
head(Q5.Dat1)
```

```
##   X1 X4 Y X2_2 X2_3 X3_2
## 1 33  0 1    0    0    0
## 2 35  0 1    0    0    0
## 3  6  0 0    0    0    0
## 4 60  0 1    0    0    0
## 5 18  1 0    0    1    0
## 6 26  0 0    0    1    0
```

TO GET THE FULL FORMULA USE THE CODE BELOW. RUN THE CODE. CUT & PASTE THE RESULTS AND ADD THE "+" IN BETWEEN EACH TERM. HOWEVER THIS CODE WILL GIVE YOU ALL INTERACTIONS. SO YOU NEED TO MANUALLY DELETE ALL THE INTERACTIONS TERMS FOR THE DUMMY VARIABLES OF THE SAME VARIABLE (I.E. X2_2:X2_3) BEFORE RUNNING THE GLM.

#ff<-lm(Y~.^2,data=Q5.Dat1) #ff

```
f5<-glm(Y~X1+X4+X2_2+X2_3+X3_2+X1:X4+X1:X2_2+X1:X2_3+X1:X3_2+X4:X2_2+X4:X2_3+X4:X3_2+X2_2:X3_2+X2_3:X3_2
f5
```

```
##
## Call:  glm(formula = Y ~ X1 + X4 + X2_2 + X2_3 + X3_2 + X1:X4 + X1:X2_2 +
##     X1:X2_3 + X1:X3_2 + X4:X2_2 + X4:X2_3 + X4:X3_2 + X2_2:X3_2 +
##     X2_3:X3_2, family = binomial, data = Q5.Dat1)
##
## Coefficients:
## (Intercept)           X1           X4         X2_2         X2_3         X3_2
##    0.155908     0.035838    -0.946814    -1.306280    -2.151271     0.916937
##       X1:X4      X1:X2_2      X1:X2_3      X1:X3_2      X4:X2_2      X4:X2_3
##    0.021247     0.008166     0.002890    -0.021077    -0.111640    -0.137603
##     X4:X3_2    X2_2:X3_2    X2_3:X3_2
##    0.930980    -0.131848     0.388653
##
## Degrees of Freedom: 195 Total (i.e. Null);  181 Residual
## Null Deviance:       270.1
## Residual Deviance: 212.8     AIC: 242.8
```

b-)Use the likelihood ratio test to determine whether all interaction terms can be dropped from the regression model; State the alternatives, full and reduced models, decision rule, and conclusion.

Ho: Variables can be dropped Ha: Variables cannot be dropped

Accept Ho, all interaction terms can be dropped.

```
f5r<-glm(Y~X1+X4+X2_2+X2_3+X3_2,data=Q5.Dat1,family=binomial)
anova(f5r,f5, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X4 + X2_2 + X2_3 + X3_2
## Model 2: Y ~ X1 + X4 + X2_2 + X2_3 + X3_2 + X1:X4 + X1:X2_2 + X1:X2_3 +
##     X1:X3_2 + X4:X2_2 + X4:X2_3 + X4:X3_2 + X2_2:X3_2 + X2_3:X3_2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       190     215.36
## 2       181     212.84  9   2.5213   0.9803
```

c-)For logistic regression model in part (a), use backward elimination to decide which predictor variables can be dropped from the regression model. Which variables are retained in the regression model?

Please see below, all interaction terms, X_4 are dropped from the model. All variables are significant.

```
t0<-step(f5,direction="backward",trace=0)
summary(t0)
```

```
##
## Call:
```

```
## glm(formula = Y ~ X1 + X2_2 + X2_3 + X3_2, family = binomial,
##     data = Q5.Dat1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2898  -0.8648   0.3887   0.8149   1.9887
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.054054   0.381415   0.142 0.887302
## X1           0.035471   0.009796   3.621 0.000294 ***
## X2_2        -1.174332   0.417764  -2.811 0.004939 **
## X2_3        -1.953575   0.402550  -4.853 1.22e-06 ***
## X3_2         0.789524   0.348572   2.265 0.023511 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 215.36  on 191  degrees of freedom
## AIC: 225.36
##
## Number of Fisher Scoring iterations: 4
```

d-) Conduct the Hosmer-Lemeshow goodness of fit test for the appropriateness of the logistic regression function by forming five groups. State the alternatives, decision rule, and conclusion.

Using the model in part C, Ho: Fit is good Ha: Fit is not good

Accept null, the fit is good.

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
hoslem.test(t0$y,fitted(t0),g=5)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  t0$y, fitted(t0)
## X-squared = 3.8928, df = 3, p-value = 0.2733
```

e-) Make the prediction for the following two cases and calculate 95% confidence interval

X1=(33,6) X2=(1,1) X3=(1,1) X4=(0,0)

Please see below

```
test.dat<-data.frame(X1=c(33,6),X2=c(1,1),X3=c(1,1),X4=c(0,0))
#however we need to create dummy variables

test.dat<-data.frame(X1=c(33,6),X2_2=c(0,0),X2_3=c(0,0),X3_2=c(0,0),X4=c(0,0))
#to install inv.logit function, we need to boot library
library(boot)
```

```
##
## Attaching package: 'boot'
## The following object is masked from 'package:survival':
##
##     aml

## The following object is masked from 'package:lattice':
##
##     melanoma

## The following objects are masked from 'package:faraway':
##
##     logit, melanoma
```

```
pred<-predict(t0,test.dat,type="link",se.fit = TRUE)

inv.logit(pred$fit)
```

```
##         1         2
## 0.7728740 0.5663273
```

```
critval <- round(qnorm(1-.05/2),2)#1.96 approx 95% CI
critval
```

```
## [1] 1.96
```

```
upr <- inv.logit(pred$fit + (critval * pred$se.fit))
lwr <- inv.logit(pred$fit - (critval * pred$se.fit))
cbind(lwr,upr)
```

```
##         lwr       upr
## 1 0.6383973 0.8677038
## 2 0.3958197 0.7224562
```