Final Exam

Neelam Purswani

December 1, 2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#Importing required libraries
library(tidyverse) #for data cleaning

## -- Attaching packages ---------------------------------------------------
------------------ tidyverse 1.2.1 --

## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------
------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(dplyr) #for data wrangling
library(caTools)
#library(caTools)
library(pROC) #For ROC calculation: Specificity versus sensitivity

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

library(randomForest) # random forest

## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

library(bestglm) # for finding out best subsets of variables

## Loading required package: leaps

library(MASS) # for statistical functions

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

library(car) #for Scatterplot matrix

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some

library(caret) #for modeling and cross validation

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

library(tree) # for creating decision trees
library(RCurl) #for fetching the data using URL
```

```
## Loading required package: bitops

##
## Attaching package: 'RCurl'

## The following object is masked from 'package:tidyr':
##
##     complete

library(rpart.plot)

## Loading required package: rpart

#read the csv file containing heart Data
heartData <- read.csv("heartData.csv")

#previewing the data
head(heartData)

##   X patient_id slope_of_peak_exercise_st_segment             thal
## 1 1     02cipp                                1           normal
## 2 2     08usun                                1 reversible_defect
## 3 3     0g192k                                2 reversible_defect
## 4 4     0n5fu0                                1           normal
## 5 5     0ryxtv                                2           normal
## 6 6     0xw93k                                1           normal
##   resting_blood_pressure chest_pain_type num_major_vessels
## 1                    140               1                 2
## 2                    120               4                 0
## 3                    128               4                 1
## 4                    180               4                 0
## 5                    102               4                 0
## 6                    124               3                 2
##   fasting_blood_sugar_gt_120_mg_per_dl resting_ekg_results
## 1                                    0                   0
## 2                                    0                   0
## 3                                    0                   0
## 4                                    0                   0
## 5                                    0                   2
## 6                                    1                   0
##   serum_cholesterol_mg_per_dl oldpeak_eq_st_depression sex age
## 1                         239                      1.8   0  69
## 2                         177                      0.4   1  65
## 3                         263                      0.2   1  64
## 4                         325                      0.0   0  64
## 5                         265                      0.6   0  42
## 6                         255                      0.0   1  48
##   max_heart_rate_achieved exercise_induced_angina heart_disease_present
## 1                     151                       0                      0
## 2                     140                       0                      0
## 3                     105                       1                      0
```

```
## 4                       154                       1                       0
## 5                       122                       0                       0
## 6                       175                       0                       0
```

(a) Describe the participants (you must include a written response with your code output). Use descriptive, summarization, and exploratory techniques to describe the participants in the study. For example, what proportion of participants are female? What is the average age of participants?

```
#looking at the variable names and data types
str(heartData)
```

```
## 'data.frame':    180 obs. of  16 variables:
##  $ X                                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ patient_id                         : Factor w/ 180 levels
"02cipp","08usun",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ slope_of_peak_exercise_st_segment  : int  1 1 2 1 2 1 1 1 2 1 ...
##  $ thal                               : Factor w/ 3 levels
"fixed_defect",..: 2 3 3 2 2 2 2 3 3 2 ...
##  $ resting_blood_pressure             : int  140 120 128 180 102 124 128
94 120 130 ...
##  $ chest_pain_type                    : int  1 4 4 4 4 3 2 3 2 3 ...
##  $ num_major_vessels                  : int  2 0 1 0 0 2 0 1 1 0 ...
##  $ fasting_blood_sugar_gt_120_mg_per_dl: int  0 0 0 0 0 1 0 0 0 0 ...
##  $ resting_ekg_results                : int  0 0 0 0 2 0 2 0 2 0 ...
##  $ serum_cholesterol_mg_per_dl        : int  239 177 263 325 265 255 308
227 281 275 ...
##  $ oldpeak_eq_st_depression           : num  1.8 0.4 0.2 0 0.6 0 0 0 1.4
0.2 ...
##  $ sex                                : int  0 1 1 0 0 1 1 1 1 0 ...
##  $ age                                : int  69 65 64 64 42 48 45 51 62
48 ...
##  $ max_heart_rate_achieved            : int  151 140 105 154 122 175 170
154 103 139 ...
##  $ exercise_induced_angina            : int  0 0 1 1 0 0 0 1 0 0 ...
##  $ heart_disease_present              : int  0 0 0 0 0 0 0 0 1 0 ...
```

```
#counting the number of observations
nrow(heartData)
```

```
## [1] 180
```

```
#counting the number of variables
ncol(heartData)
```

```
## [1] 16
```

```
#dropping column 1 and 2
heartData <- dplyr::select(heartData, -c(1,2))

#converting thal to numeric for analysis
heartData$thal <- as.numeric(heartData$thal)
```

The heart dataset consists of 180 observations and 14 variables, the 13 predictor variables help us to determine the correlation between their values and presence of heart disease in a participant. Here are more details about each field: 1. X: record identification number 2. id: patient identification number 3. slope_of_peak_exercise_st_segment:the slope of the peak exercise ST segment – Value 1: upsloping – Value 2: flat – Value 3: downsloping 4. thal:thalassemia? 3 = normal; 6 = fixed defect; 7 = reversable defect 5. resting_blood_pressure: blood pressure while resting 6. chest_pain_type:type of chest pain 7. num_major_vessels: num: diagnosis of heart disease (angiographic disease status) – Value 0: < 50% diameter narrowing – Value 1: > 50% diameter narrowing (in any major vessel: attributes 59 through 68 are vessels) 8. fasting_blood_sugar_gt_120_mg_per_dl: blood sugar level 9. resting_ekg_results:resting ecg results 10. serum_cholesterol_mg_per_dl: serum cholesterol level 11. oldpeak_eq_st_depression: old peak standard depression 12. sex: sex (1 = male; 0 = female) 13. age: age in years 14. max_heart_rate_achieved: maximum heart rate acheived by participants 15. exercise_induced_angina: chest pain induced from exercise 16. heart_disease_present: whether a participant was diagnosed with heart disease or not

Using http://archive.ics.uci.edu/ml/datasets/Heart+Disease, lets understand the data in further detail and try to see things in context so that we can make a decision about what data types need to be changed. Conducting Descriptive data analysis: Lets look at the number of male and female population involved in the study:

```
#Since Sex 1 indicates male, we can filter those records and take a count,
and store the results in variable named number_of_males
number_of_males <- heartData %>%
  filter(sex==1) %>%
  summarise(count=n())
number_of_males

##   count
## 1   124

# For getting the count of number of females, we can filter on sex 0
number_of_females <- heartData %>%
  filter(sex==0) %>%
  summarise(count=n())
number_of_females

##   count
## 1    56

#Percentage of males and females
male_prop <- (number_of_males/(number_of_males+number_of_females))*100
male_prop

##      count
## 1 68.88889

female_prop  <- (number_of_females/(number_of_males+number_of_females))*100
female_prop
```

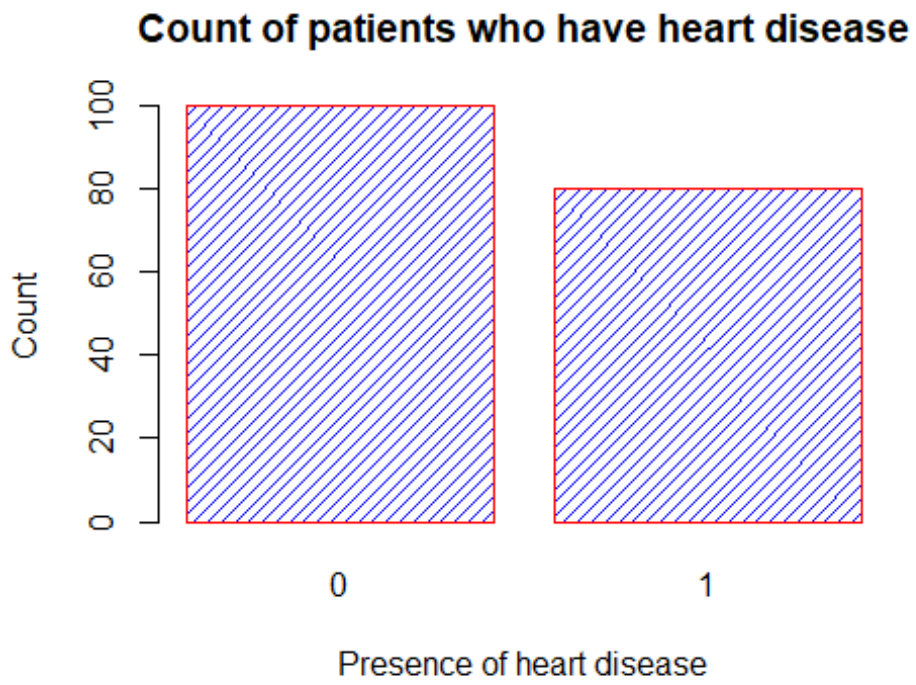
```
##       count
## 1 31.11111
```

Close to 69% of the population is male And 31% is female

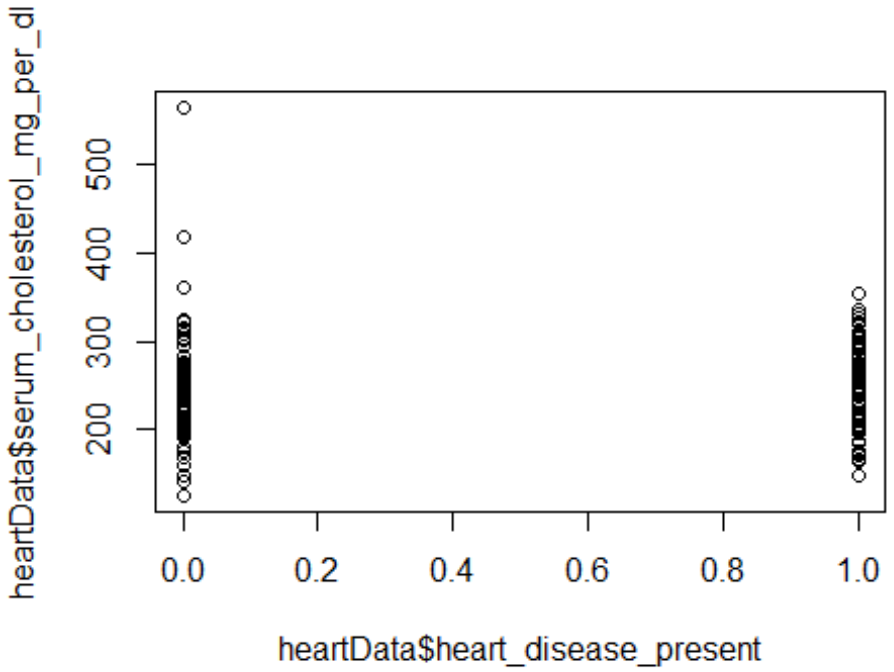
```
#What is the average age of participants?
average_age_of_participants <- heartData %>%
  summarise(mean(age))
average_age_of_participants

##   mean(age)
## 1  54.81111
```

The average age of participants in the study is 54.81 years.

```
barplot(table(heartData$heart_disease_present),
main="Count of patients who have heart disease",
xlab="Presence of heart disease",
ylab="Count",
border="red",
col="blue",
density=20
)
```




```
#plotting a box plot to see the median cholesterol level of people who were
diagnosed with heart disease
plot(heartData$heart_disease_present,heartData$serum_cholesterol_mg_per_dl)
```



It appears that the median cholesterol level in people with heart disease is higher than in the people without it.

Finding co-relations between variables for further analysis:

```
cor(heartData$age, heartData$heart_disease_present)

## [1] 0.1382547

cor(heartData$sex, heartData$heart_disease_present)

## [1] 0.3354209

cor(heartData$serum_cholesterol_mg_per_dl, heartData$heart_disease_present)

## [1] 0.07977485
```

(b) We want to explore the characteristics of participants who have been diagnosed with heart disease. The data includes a binary outcome variable heart_disease_present. Describe what the values within this variable signify.

```
#heartData$heart_disease_present <- (heartData$heart_disease_present -1)
#heartData$heart_disease_present
```

heart_disease_present = 0 stands for false indicating heart disease is absent 1 is for true indicating heart disease is present

(c) Describe the potential explanatory (independent, predictor) variables in this dataset.

Some of the predictor variables can be: Apart from Patient Id, and X all the other variables like age, cholesterol, chest_pain_type, oldpeak_eq_st_depression, serum_cholesterol_mg_per_dl etc can be predictor variables.

(d) Split your data into a training and test set based on an 70-30 split, in other words, 70% of the observations will be in the training set (you do not need to create a validation set for this exercise).

```
# code adapted from https://rpubs.com/ID_Tech/S1 AND
https://stackoverflow.com/a/31634462

# Set seed for reproducibility
set.seed(112718)
# splits the data in the ratio mentioned in SplitRatio. After splitting marks
these rows as logical
# TRUE and the the remaining are marked as logical FALSE
sample = sample.split(heartData$heart_disease_present, SplitRatio = .7)
# creates a training dataset named train with rows which are marked as TRUE
heartData_train = subset(heartData, sample == TRUE)
# creates a training dataset named test with rows which are marked as FALSE
nrow(heartData_train)

## [1] 126

heartData_test  = subset(heartData, sample == FALSE)
nrow(heartData_test)

## [1] 54
```

(e) Use an appropriate regression model to explore the relationship between having a diagnosis of heart disease (or not) and all other characteristics in your training data. Comment on which covariates seem to be predictive of having heart disease and which do not.

```
#performing logistic regression on the dataset to find out the covariates
#class(heartData_train$thal)
str(heartData_train)

## 'data.frame':    126 obs. of  14 variables:
##  $ slope_of_peak_exercise_st_segment   : int  1 2 1 1 1 2 1 2 2 1 ...
##  $ thal                                : num  3 3 2 2 3 3 2 2 2 3 ...
##  $ resting_blood_pressure              : int  120 128 124 128 94 120 130
138 120 128 ...
##  $ chest_pain_type                     : int  4 4 3 2 3 2 3 4 3 4 ...
##  $ num_major_vessels                   : int  0 1 2 0 1 1 0 3 0 1 ...
##  $ fasting_blood_sugar_gt_120_mg_per_dl: int  0 0 1 0 0 0 0 1 0 0 ...
##  $ resting_ekg_results                 : int  0 0 0 2 0 2 0 0 0 0 ...
##  $ serum_cholesterol_mg_per_dl         : int  177 263 255 308 227 281 275
294 219 255 ...
##  $ oldpeak_eq_st_depression            : num  0.4 0.2 0 0 0 1.4 0.2 1.9
1.6 0 ...
##  $ sex                                 : int  1 1 1 1 1 1 0 0 0 1 ...
```

```
##  $ age                                : int  65 64 48 45 51 62 48 62 50
52 ...
##  $ max_heart_rate_achieved            : int  140 105 175 170 154 103 139
106 158 161 ...
##  $ exercise_induced_angina            : int  0 1 0 0 1 0 0 0 0 1 ...
##  $ heart_disease_present              : int  0 0 0 0 0 1 0 1 0 1 ...
```

```
logistic_model_0 <- glm(heart_disease_present ~ .,
data=heartData_train,family=binomial)

#printing the summary
summary(logistic_model_0)
```

```
##
## Call:
## glm(formula = heart_disease_present ~ ., family = binomial, data =
heartData_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4585  -0.5349  -0.1798   0.3832   2.4437
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -9.713528   4.829499  -2.011  0.04429
## slope_of_peak_exercise_st_segment    0.339614   0.598443   0.567  0.57038
## thal                                 1.576680   0.578897   2.724  0.00646
## resting_blood_pressure               0.013053   0.020034   0.652  0.51471
## chest_pain_type                      0.959271   0.330745   2.900  0.00373
## num_major_vessels                    1.122969   0.371006   3.027  0.00247
## fasting_blood_sugar_gt_120_mg_per_dl -1.106201   0.894763  -1.236  0.21634
## resting_ekg_results                  0.255216   0.302814   0.843  0.39933
## serum_cholesterol_mg_per_dl          0.004056   0.005110   0.794  0.42743
## oldpeak_eq_st_depression             0.490264   0.425826   1.151  0.24960
## sex                                  1.674191   0.793016   2.111  0.03476
## age                                 -0.018670   0.037566  -0.497  0.61919
## max_heart_rate_achieved             -0.015123   0.017009  -0.889  0.37393
## exercise_induced_angina              0.589713   0.666948   0.884  0.37659
##
## (Intercept)                         *
## slope_of_peak_exercise_st_segment
## thal                                **
## resting_blood_pressure
## chest_pain_type                     **
## num_major_vessels                   **
## fasting_blood_sugar_gt_120_mg_per_dl
## resting_ekg_results
## serum_cholesterol_mg_per_dl
## oldpeak_eq_st_depression
## sex                                 *
```

```
## age
## max_heart_rate_achieved
## exercise_induced_angina
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 173.114  on 125  degrees of freedom
## Residual deviance:  86.615  on 112  degrees of freedom
## AIC: 114.62
##
## Number of Fisher Scoring iterations: 6
```

Lets perform stepwise regression to find out which variables does the model choose

```
#Performing Stepwise Model on logistic_model_0
steplogistic <- stepAIC(logistic_model_0, trace=FALSE)

#step anova to see which was the final model chosen
steplogistic$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## heart_disease_present ~ slope_of_peak_exercise_st_segment + thal +
##     resting_blood_pressure + chest_pain_type + num_major_vessels +
##     fasting_blood_sugar_gt_120_mg_per_dl + resting_ekg_results +
##     serum_cholesterol_mg_per_dl + oldpeak_eq_st_depression +
##     sex + age + max_heart_rate_achieved + exercise_induced_angina
##
## Final Model:
## heart_disease_present ~ thal + chest_pain_type + num_major_vessels +
##     oldpeak_eq_st_depression + sex
##
##
##                                      Step Df  Deviance Resid. Df Resid. Dev
## 1                                                            112   86.61542
## 2                                    - age  1 0.2490258       113   86.86445
## 3    - slope_of_peak_exercise_st_segment  1 0.3250046       114   87.18945
## 4               - resting_blood_pressure  1 0.2986945       115   87.48815
## 5          - serum_cholesterol_mg_per_dl  1 0.4869047       116   87.97505
## 6              - max_heart_rate_achieved  1 0.5641504       117   88.53920
## 7              - exercise_induced_angina  1 1.0376054       118   89.57681
## 8                  - resting_ekg_results  1 1.3283992       119   90.90521
## 9 - fasting_blood_sugar_gt_120_mg_per_dl  1 1.8354394       120   92.74065
##        AIC
## 1 114.6154
## 2 112.8644
```

```
## 3 111.1895
## 4 109.4881
## 5 107.9751
## 6 106.5392
## 7 105.5768
## 8 104.9052
## 9 104.7406
```

To answer part(e) about the covariates which seem to be affecting heart disease are: presence of type of thalssemia type of chest pain num_major_vessels oldpeak_eq_st_depression sex

(f) Use an all subsets model selection procedure (note that this is slightly different from stepwise selection: helpful reference) to obtain a "best" fit model for your training data. Is the model different from the full model you fit in part (e)? Which variables are included in the "best" fit model? (You might find the bestglm() function available in the bestglm package helpful.)

```
#preparing data for bestglm by making the response variable as last variable
heartData_new <- heartData_train %>%
  dplyr::select(-heart_disease_present, everything())
str(heartData_new)
```

```
## 'data.frame':    126 obs. of  14 variables:
##  $ slope_of_peak_exercise_st_segment   : int  1 2 1 1 1 2 1 2 2 1 ...
##  $ thal                                : num  3 3 2 2 3 3 2 2 2 3 ...
##  $ resting_blood_pressure              : int  120 128 124 128 94 120 130
138 120 128 ...
##  $ chest_pain_type                     : int  4 4 3 2 3 2 3 4 3 4 ...
##  $ num_major_vessels                   : int  0 1 2 0 1 1 0 3 0 1 ...
##  $ fasting_blood_sugar_gt_120_mg_per_dl: int  0 0 1 0 0 0 0 1 0 0 ...
##  $ resting_ekg_results                 : int  0 0 0 2 0 2 0 0 0 0 ...
##  $ serum_cholesterol_mg_per_dl         : int  177 263 255 308 227 281 275
294 219 255 ...
##  $ oldpeak_eq_st_depression            : num  0.4 0.2 0 0 0 1.4 0.2 1.9
1.6 0 ...
##  $ sex                                 : int  1 1 1 1 1 1 0 0 0 1 ...
##  $ age                                 : int  65 64 48 45 51 62 48 62 50
52 ...
##  $ max_heart_rate_achieved             : int  140 105 175 170 154 103 139
106 158 161 ...
##  $ exercise_induced_angina             : int  0 1 0 0 1 0 0 0 0 1 ...
##  $ heart_disease_present               : int  0 0 0 0 0 1 0 1 0 1 ...
```

```
#Dropping column 1 from the heart training set
heartData_train_copy <- dplyr::select(heartData_train, -c(1))
str(heartData_train_copy)
```

```
## 'data.frame':    126 obs. of  13 variables:
##  $ thal                                : num  3 3 2 2 3 3 2 2 2 3 ...
##  $ resting_blood_pressure              : int  120 128 124 128 94 120 130
```

```
138 120 128 ...
##  $ chest_pain_type                    : int  4 4 3 2 3 2 3 4 3 4 ...
##  $ num_major_vessels                  : int  0 1 2 0 1 1 0 3 0 1 ...
##  $ fasting_blood_sugar_gt_120_mg_per_dl: int  0 0 1 0 0 0 0 1 0 0 ...
##  $ resting_ekg_results                : int  0 0 0 2 0 2 0 0 0 0 ...
##  $ serum_cholesterol_mg_per_dl        : int  177 263 255 308 227 281 275
294 219 255 ...
##  $ oldpeak_eq_st_depression           : num  0.4 0.2 0 0 0 1.4 0.2 1.9
1.6 0 ...
##  $ sex                                : int  1 1 1 1 1 1 0 0 0 1 ...
##  $ age                                : int  65 64 48 45 51 62 48 62 50
52 ...
##  $ max_heart_rate_achieved            : int  140 105 175 170 154 103 139
106 158 161 ...
##  $ exercise_induced_angina            : int  0 1 0 0 1 0 0 0 0 1 ...
##  $ heart_disease_present              : int  0 0 0 0 0 1 0 1 0 1 ...

#Preparing the input for bestglm
heartData_train_copy_bestglm <- within(heartData_train_copy, {
      y  <- heart_disease_present
    heart_disease_present  <- NULL
})
str(heartData_train_copy_bestglm)

## 'data.frame':    126 obs. of  13 variables:
##  $ thal                               : num  3 3 2 2 3 3 2 2 2 3 ...
##  $ resting_blood_pressure             : int  120 128 124 128 94 120 130
138 120 128 ...
##  $ chest_pain_type                    : int  4 4 3 2 3 2 3 4 3 4 ...
##  $ num_major_vessels                  : int  0 1 2 0 1 1 0 3 0 1 ...
##  $ fasting_blood_sugar_gt_120_mg_per_dl: int  0 0 1 0 0 0 0 1 0 0 ...
##  $ resting_ekg_results                : int  0 0 0 2 0 2 0 0 0 0 ...
##  $ serum_cholesterol_mg_per_dl        : int  177 263 255 308 227 281 275
294 219 255 ...
##  $ oldpeak_eq_st_depression           : num  0.4 0.2 0 0 0 1.4 0.2 1.9
1.6 0 ...
##  $ sex                                : int  1 1 1 1 1 1 0 0 0 1 ...
##  $ age                                : int  65 64 48 45 51 62 48 62 50
52 ...
##  $ max_heart_rate_achieved            : int  140 105 175 170 154 103 139
106 158 161 ...
##  $ exercise_induced_angina            : int  0 1 0 0 1 0 0 0 0 1 ...
##  $ y                                  : int  0 0 0 0 0 1 0 1 0 1 ...

#Performing all-subset regression based on AIC
heartData_bestglm <- bestglm(Xy = heartData_train_copy_bestglm, family =
binomial, IC = "AIC", method = "exhaustive")

## Morgan-Tatar search since family is non-gaussian.

names(heartData_bestglm)
```

```
## [1] "BestModel"   "BestModels"  "Bestq"       "qTable"      "Subsets"
## [6] "Title"       "ModelReport"

#looking at the variables chosen by BestModel
bestglm_model<-heartData_bestglm$BestModel

#finding out the variables chosen by best fit model
bestglm_model<-heartData_bestglm$BestModel
bestglm_model

##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
##              (Intercept)                      thal
##                  -10.5907                    1.7659
##          chest_pain_type         num_major_vessels
##                   1.0973                    1.0181
## oldpeak_eq_st_depression                       sex
##                   0.8578                    1.4195
##
## Degrees of Freedom: 125 Total (i.e. Null);  120 Residual
## Null Deviance:       173.1
## Residual Deviance: 92.74     AIC: 104.7
```

The variables chosen by best fit model are: thal, chest_pain_type, num_major_vessels, oldpeak_eq_st_depression, sex

The variables picked by both the models are same.

(g) Interpret the model parameters of your model from part (f).

```
#looking at the bestglm model's summary for model parameters
summary(heartData_bestglm$BestModel)

##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9351  -0.4994  -0.2148   0.5047   2.3395
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -10.5907     1.8743  -5.650  1.6e-08 ***
## thal                       1.7659     0.5049   3.498 0.000469 ***
## chest_pain_type            1.0973     0.3194   3.435 0.000592 ***
## num_major_vessels          1.0181     0.3100   3.284 0.001023 **
## oldpeak_eq_st_depression   0.8578     0.3095   2.772 0.005580 **
## sex                        1.4195     0.6671   2.128 0.033346 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 173.114  on 125  degrees of freedom
## Residual deviance:  92.741  on 120  degrees of freedom
## AIC: 104.74
##
## Number of Fisher Scoring iterations: 5
```

The AIC value with bestglm is smaller than with the logistic regression. Since the AIC value is smaller in best glm than we got in glm, it indicates that this one is a better fit.

(h) Use your test dataset and the predict function to obtain predicted probabilities of having heart disease for each case in the test data. Which model did you use for prediction and why? Interpret your results and use a visualization to support your interpretation. Using logistic regression:

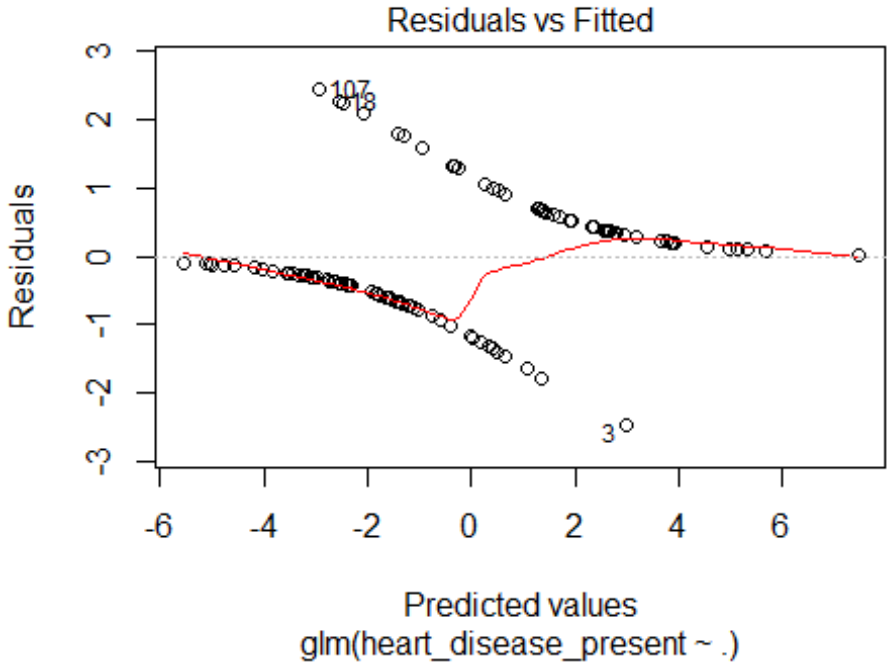
```
heart_disease_predictions <- predict(logistic_model_0, heartData_test,
type="response")

head(heartData_test$heart_disease_present)

## [1] 0 0 0 0 1 1

heart_disease_predictions = as.numeric(heart_disease_predictions)
table(heartData_test$heart_disease_present, heart_disease_predictions>0.5)

##
##     FALSE TRUE
##   0    25    5
##   1     5   19

#calculating the accuracy
(25+19)/(25+5+5+19)

## [1] 0.8148148
```

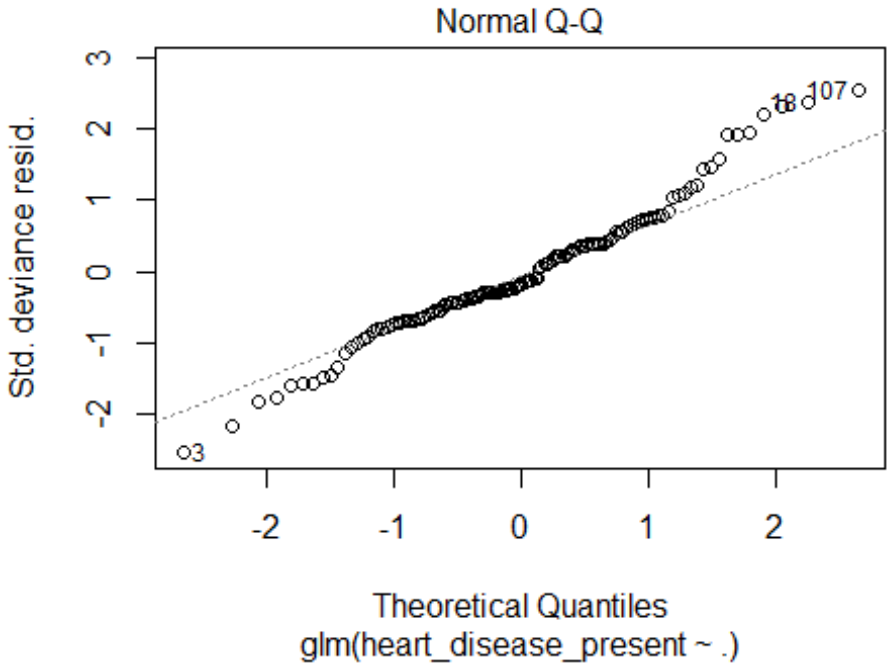
We get 81% accuracy with logistic model

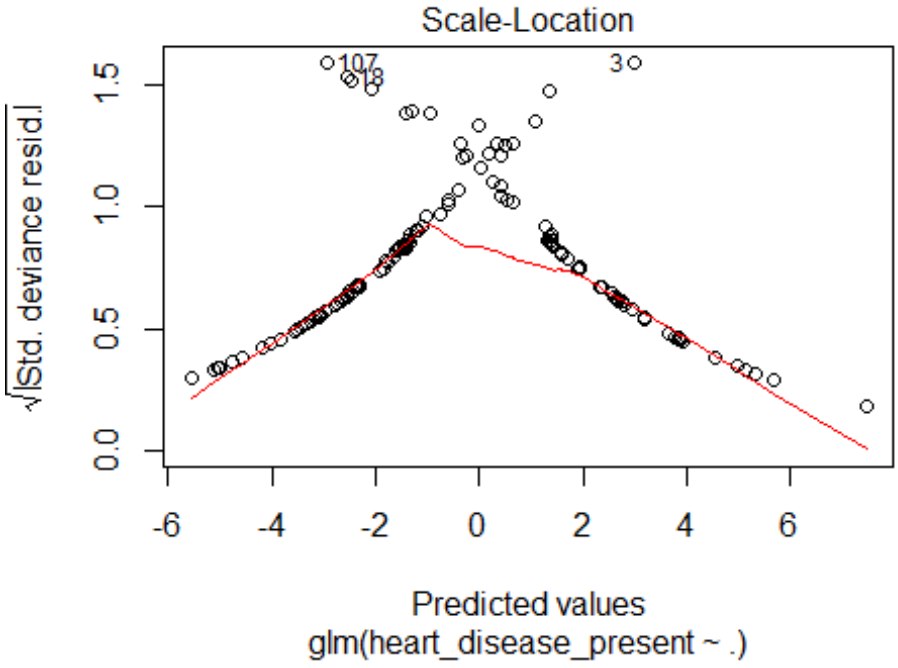
```
#making prediction with the help of bestglm model
heart_disease_predictions_bestglm <- predict(bestglm_model, heartData_test,
type="response")

table(heartData_test$heart_disease_present, heart_disease_predictions>0.5)

##
##     FALSE TRUE
##   0    25    5
##   1     5   19
```

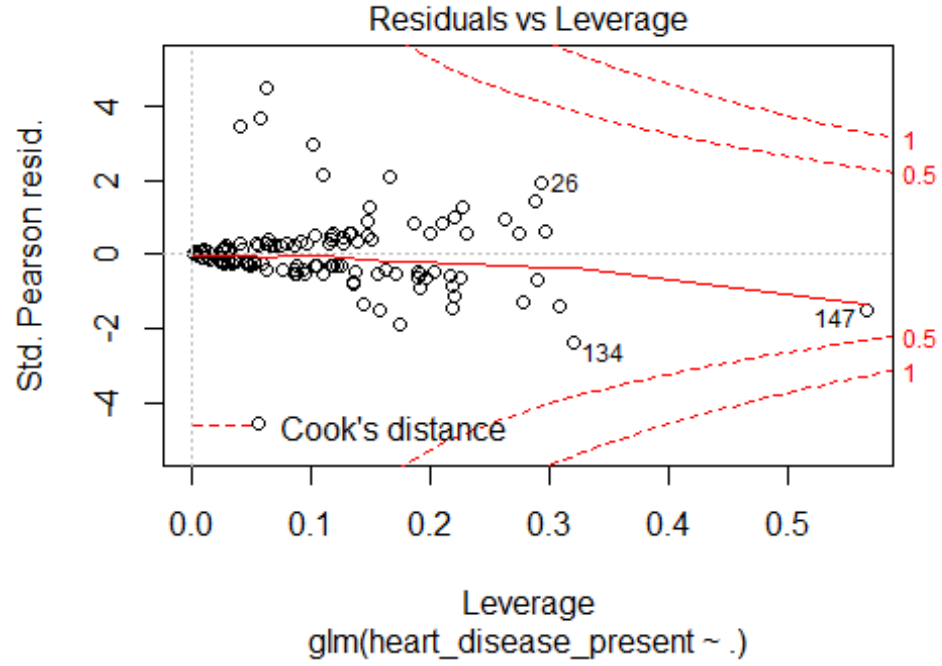
The numbers come out to be same here. Both the models have 81% accuracy.

```r
plot(logistic_model_0)
```

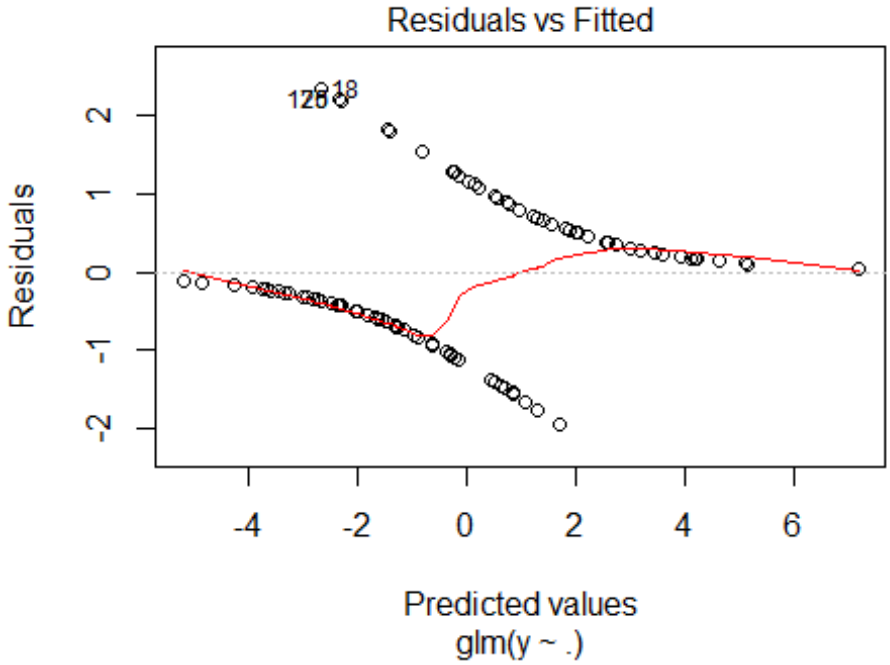


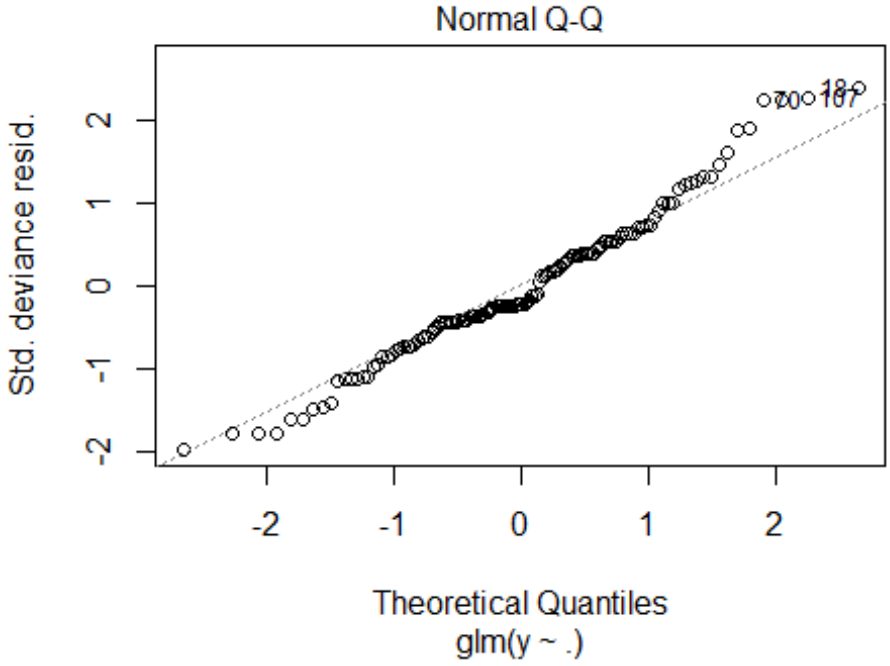


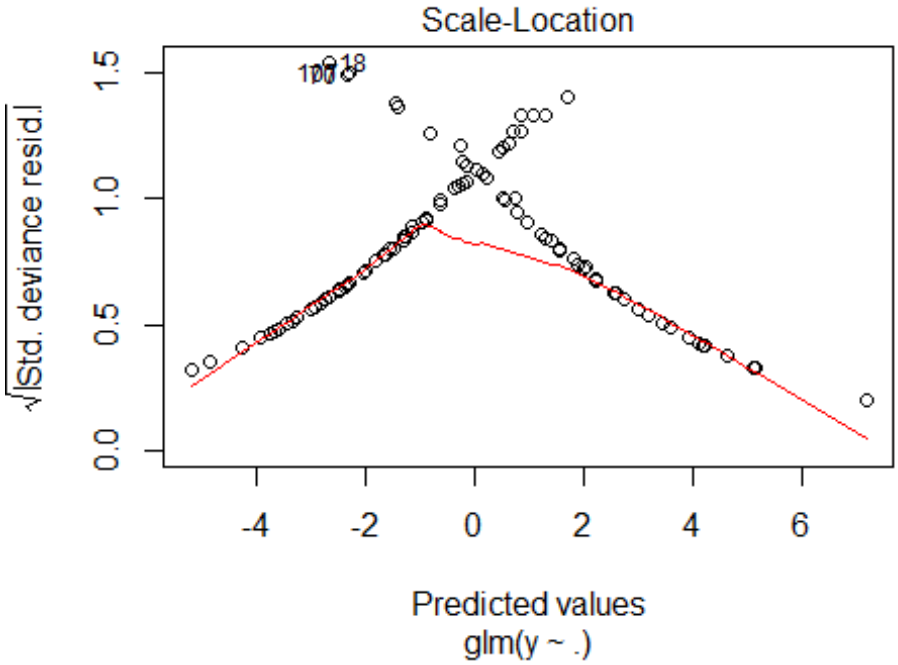


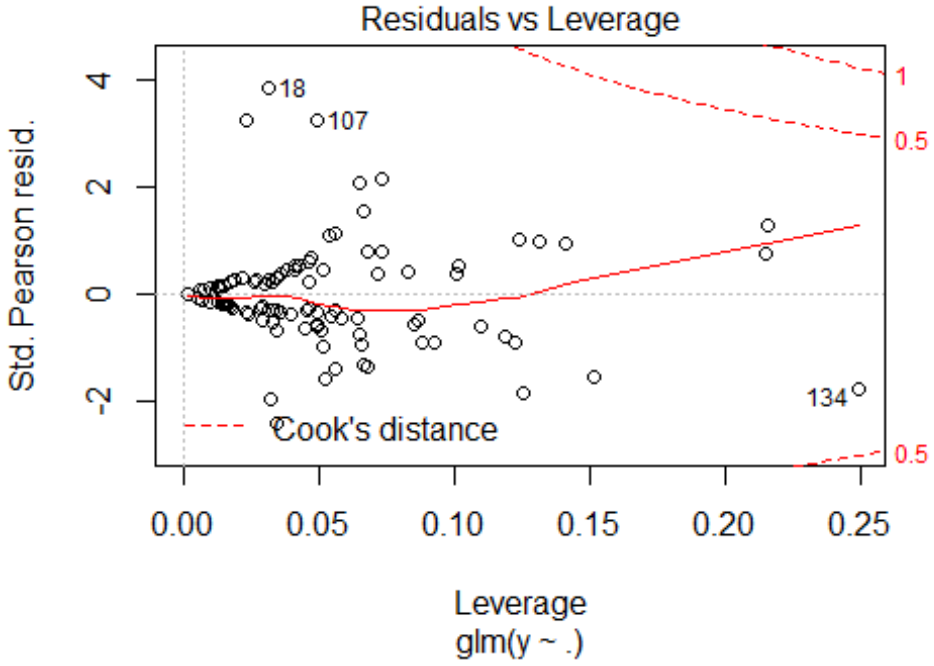



```
plot(bestglm_model)
```


Residuals vs Fitted
Residuals
Predicted values
glm(y ~ .)


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(y ~ .)





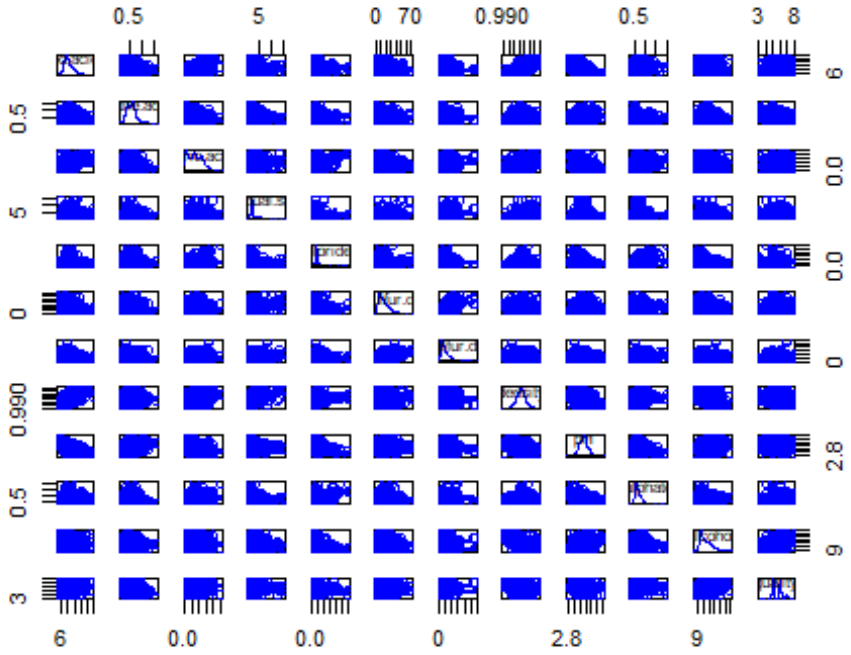
Problem 2: Suppose you want to explore the relationship between wine quality and other characteristics of the wine. Follow the questions below to perform this analysis.

```r
#loading the data from csv file provided
wineQuality <-read.csv("winequality-red-commas.csv")

#looking at the variables, data-type and some values
str(wineQuality)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58
0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069
0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36
3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57
0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

(a) Examine the bivariate relationships present in the data. Briefly discuss notable results. You might find the scatterplotMatrix() function available in the car package helpful.

```r
#creating scatterplot Matrix from Car package
scatterplotMatrix(wineQuality)
```



(b) Fit a multiple linear regression model. How much variance in the wine quality do the predictor variables explain

```
# single variable linear regression model to find out the significant
variables
wine_model<- lm(quality ~ . , data=wineQuality)

#printing the summary to find out measures that would help me present my
thoughts about variance
summary(wine_model)
```

```
##
## Call:
## lm(formula = quality ~ ., data = wineQuality)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.197e+01  2.119e+01   1.036   0.3002
## fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
## volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
## citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
## residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
## chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
```

```
## free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
## total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
## density              -1.788e+01  2.163e+01  -0.827   0.4086
## pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
## sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
## alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

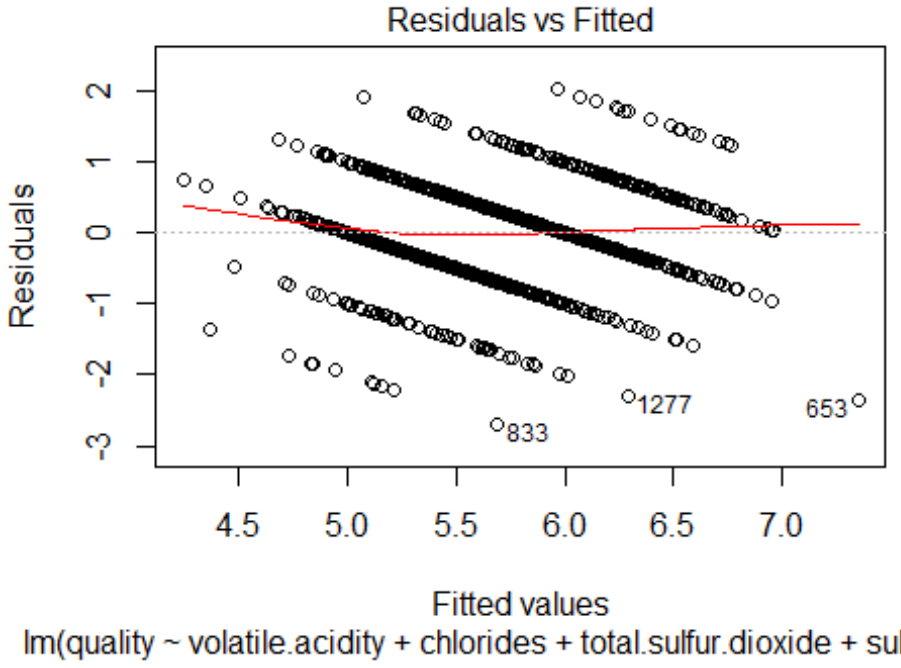
The predictor variables explain about 35% of variation in the data. Variables volatile.acidity, chlorides, total.sulfur.dioxide, sulphates and alcohol are statistically significant.

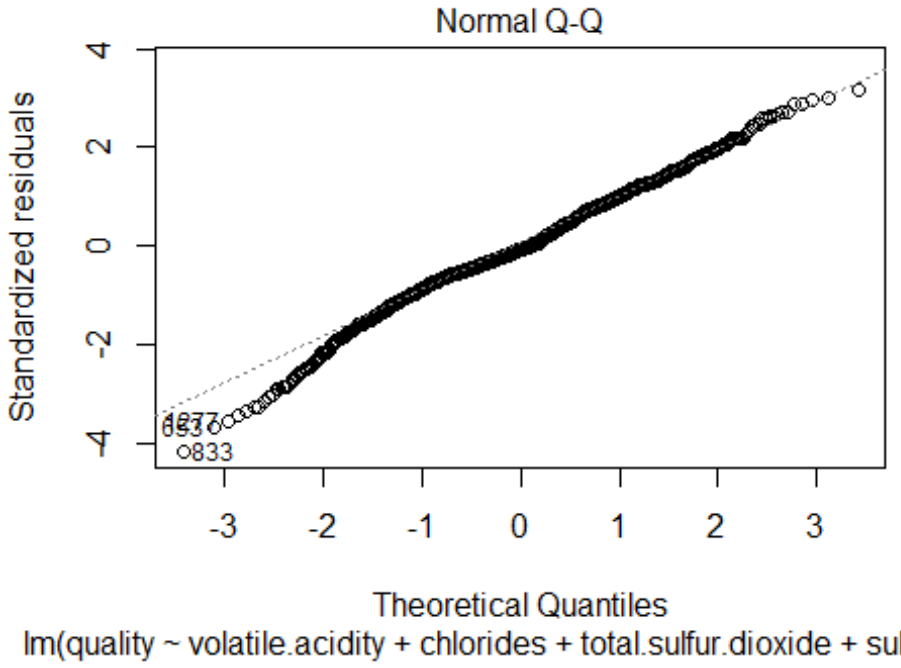
```
#performing multiple linear regression with these variables
wine_model_multiplelinear <- lm(quality ~
volatile.acidity+chlorides+total.sulfur.dioxide+sulphates+alcohol+pH+free.sul
fur.dioxide, data =wineQuality)

#looking at the summary variables
summary(wine_model_multiplelinear)

##
## Call:
## lm(formula = quality ~ volatile.acidity + chlorides + total.sulfur.dioxide
+
##     sulphates + alcohol + pH + free.sulfur.dioxide, data = wineQuality)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68918 -0.36757 -0.04653  0.46081  2.02954
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.4300987  0.4029168  10.995  < 2e-16 ***
## volatile.acidity     -1.0127527  0.1008429 -10.043  < 2e-16 ***
## chlorides            -2.0178138  0.3975417  -5.076 4.31e-07 ***
## total.sulfur.dioxide -0.0034822  0.0006868  -5.070 4.43e-07 ***
## sulphates             0.8826651  0.1099084   8.031 1.86e-15 ***
## alcohol               0.2893028  0.0167958  17.225  < 2e-16 ***
## pH                   -0.4826614  0.1175581  -4.106 4.23e-05 ***
## free.sulfur.dioxide   0.0050774  0.0021255   2.389    0.017 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6477 on 1591 degrees of freedom
```

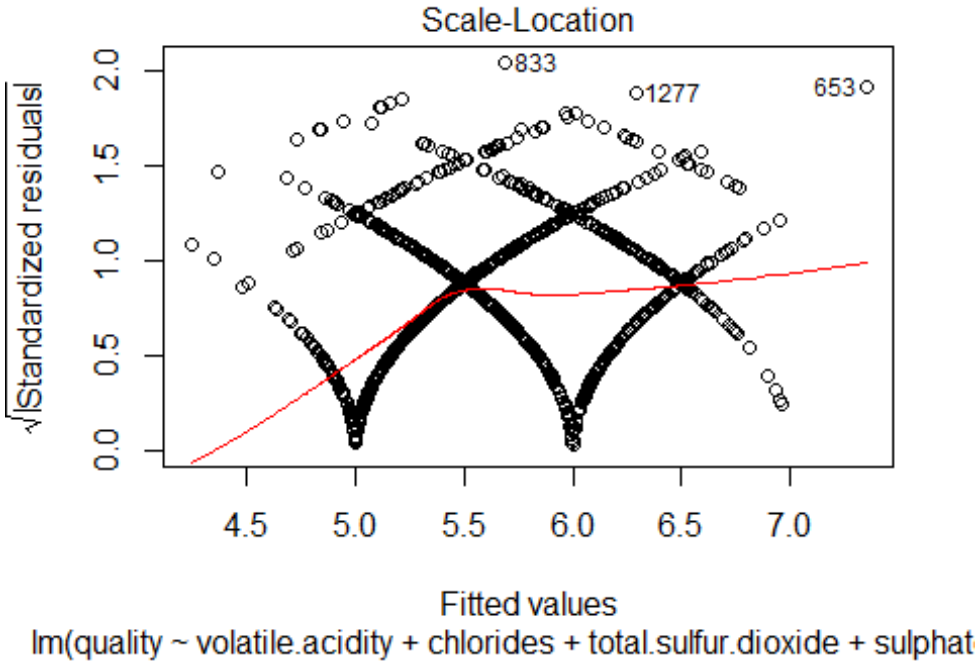
```
## Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
## F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16
```

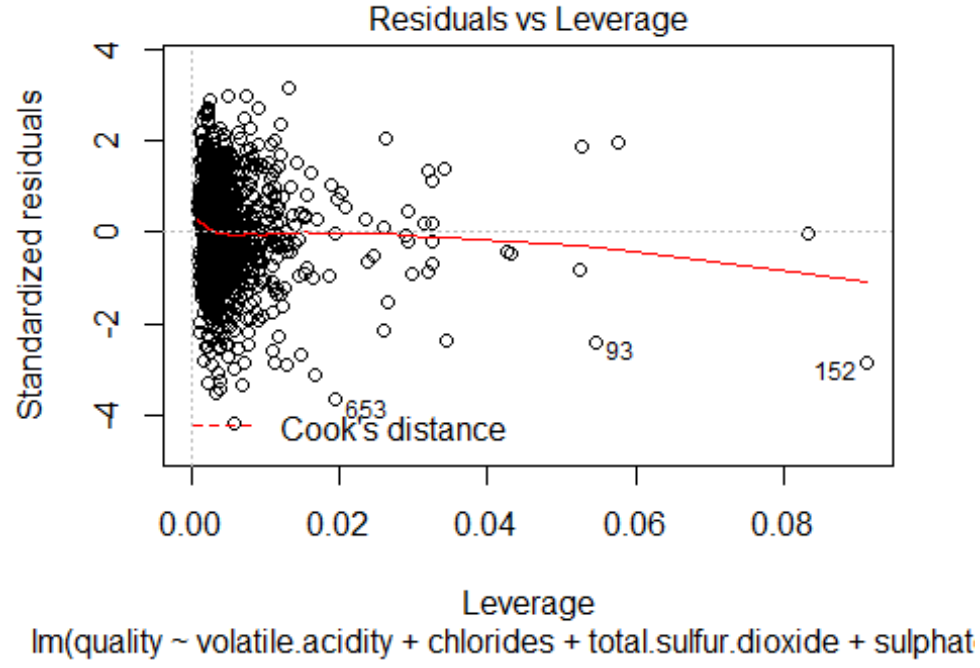
(c) Evaluate the statistical assumptions in your regression analysis from part (b) by performing a basic analysis of model residuals and any unusual observations. Discuss any concerns you have about your model.

```
#plotting the residuals to verify the statistical assumptions
plot(wine_model_multiplelinear)
```


Residuals vs Fitted
Residuals
Fitted values
lm(quality ~ volatile.acidity + chlorides + total.sulfur.dioxide + sulphat
1277
653
833

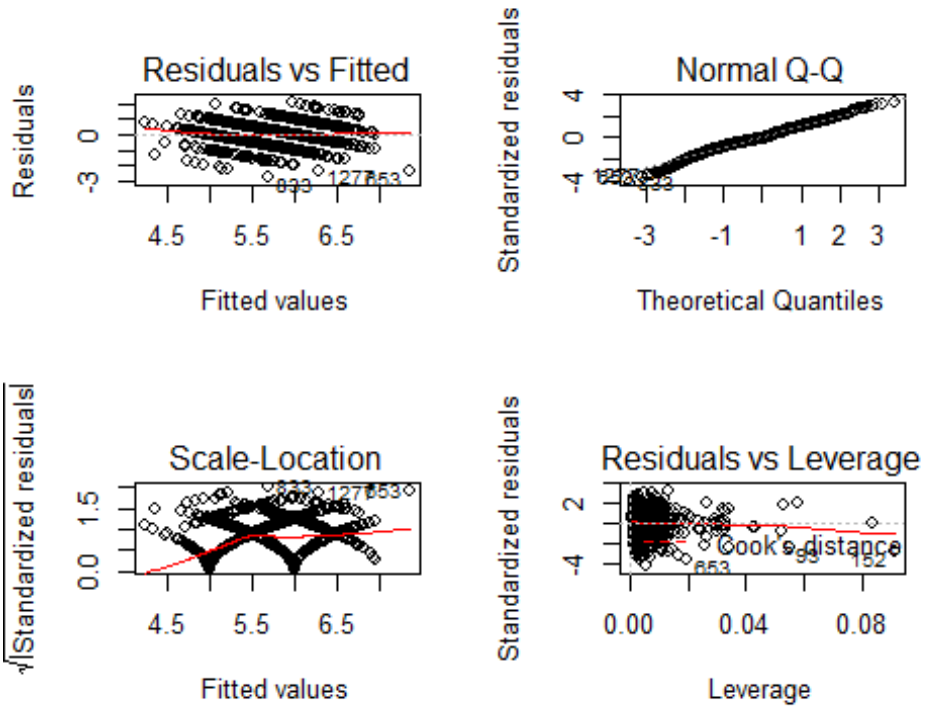

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(quality ~ volatile.acidity + chlorides + total.sulfur.dioxide + sulphat
833





Assumption: The mean of the residuals is zero In our case it is close to zero but not completely

```
par(mfrow=c(2,2))  # set 2 rows and 2 column plot layout
plot(wine_model_multiplelinear)
```



The top-left and bottom-left plots shows how the residuals vary as the fitted values increase.Using above we can see the assumption Homoscedasticity of residuals or equal variance.

Next assumption, the number of observations must be greater than number of Xs This can be directly observed by looking at the data.

(d) Use a stepwise model selection procedure of your choice to obtain a "best" fit model. Is the model different from the full model you fit in part (b)? If yes, how so?

```
# Question d - stepwise regression for best fit model
result1 <-stepAIC(wine_model, trace=FALSE)
summary(result1)

##
## Call:
## lm(formula = quality ~ volatile.acidity + chlorides + free.sulfur.dioxide
+
##     total.sulfur.dioxide + pH + sulphates + alcohol, data = wineQuality)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68918 -0.36757 -0.04653  0.46081  2.02954
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

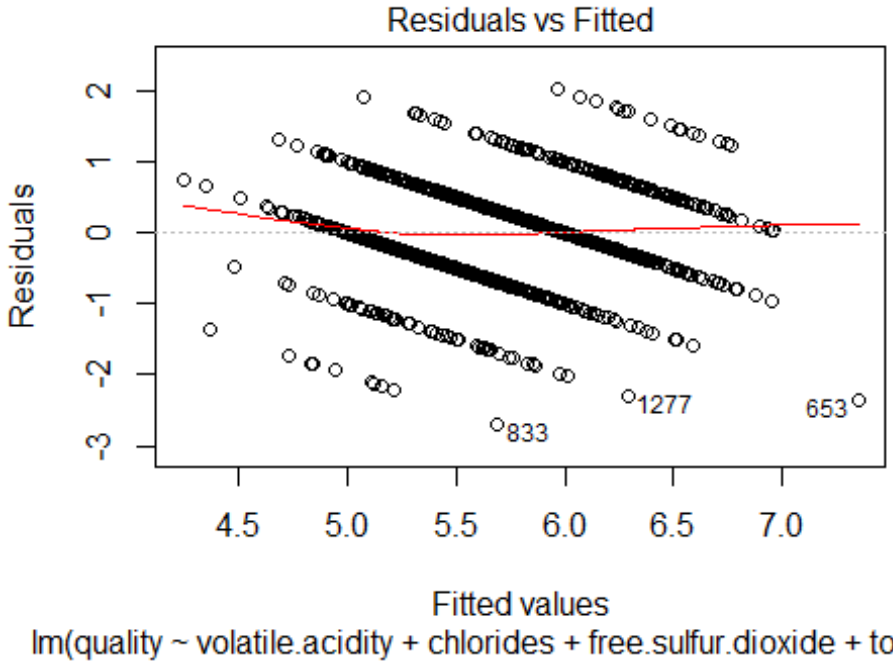
```
## (Intercept)           4.4300987  0.4029168  10.995  < 2e-16 ***
## volatile.acidity     -1.0127527  0.1008429 -10.043  < 2e-16 ***
## chlorides            -2.0178138  0.3975417  -5.076 4.31e-07 ***
## free.sulfur.dioxide   0.0050774  0.0021255   2.389    0.017 *
## total.sulfur.dioxide -0.0034822  0.0006868  -5.070 4.43e-07 ***
## pH                   -0.4826614  0.1175581  -4.106 4.23e-05 ***
## sulphates             0.8826651  0.1099084   8.031 1.86e-15 ***
## alcohol               0.2893028  0.0167958  17.225  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6477 on 1591 degrees of freedom
## Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
## F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16
```

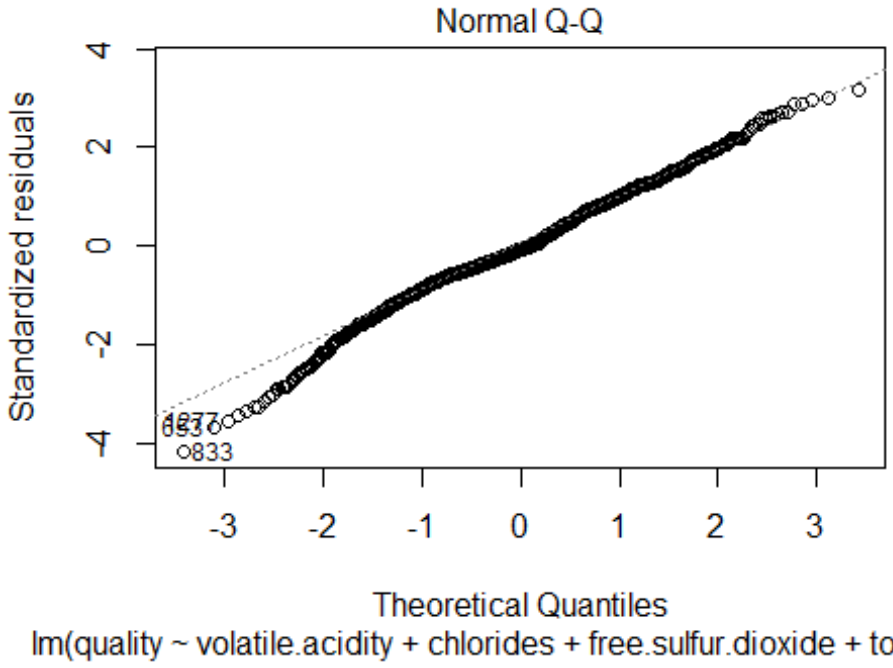
```
#using anova helps us to see the final model selected and other some measurer
result1$anova
```

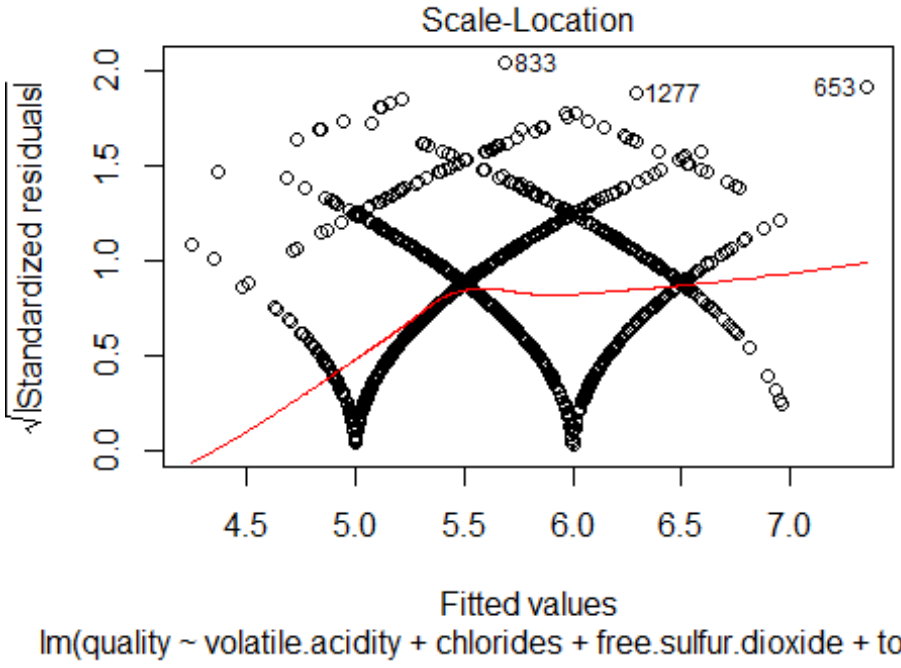
```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar
+
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
## Final Model:
## quality ~ volatile.acidity + chlorides + free.sulfur.dioxide +
##     total.sulfur.dioxide + pH + sulphates + alcohol
##
##
##               Step Df  Deviance Resid. Df Resid. Dev       AIC
## 1                                    1587   666.4107 -1375.489
## 2        - density  1 0.2868924      1588   666.6976 -1376.801
## 3  - fixed.acidity  1 0.1079824      1589   666.8056 -1378.542
## 4 - residual.sugar  1 0.2566805      1590   667.0623 -1379.926
## 5    - citric.acid  1 0.4748034      1591   667.5371 -1380.789
```

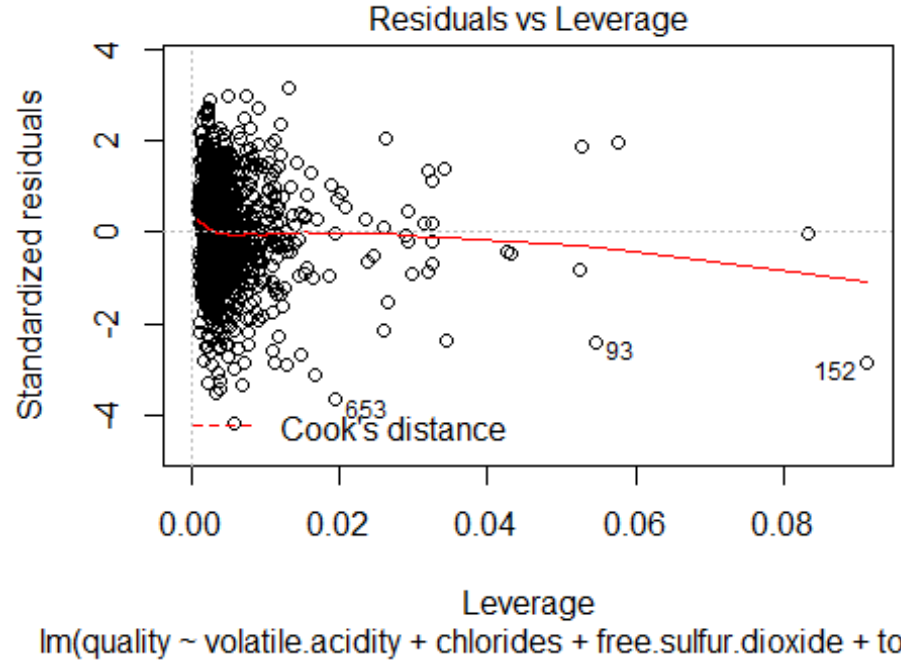
It comes out to be the same model.

```
#seeing the residuals for this model
plot(result1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(quality ~ volatile.acidity + chlorides + free.sulfur.dioxide + total.su
1277
653
833


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(quality ~ volatile.acidity + chlorides + free.sulfur.dioxide + total.su
833





(e) Assess the generalizability of the model (from part (d)). Perform a 10-fold cross validation to estimate model performance. Report the results

```
# code citation: http://www.sthda.com/english/articles/38-regression-model-
validation/157-cross-validation-essentials-in-r/#k-fold-cross-validation
```

```r
# Define training control
set.seed(123)
train.control <- trainControl(method = "cv", number = 10, verboseIter = TRUE)
# Train the model
wine_model_kfold <- train(quality ~., data = wineQuality, method = "lm",
               trControl = train.control)
```

```
## + Fold01: intercept=TRUE
## - Fold01: intercept=TRUE
## + Fold02: intercept=TRUE
## - Fold02: intercept=TRUE
## + Fold03: intercept=TRUE
## - Fold03: intercept=TRUE
## + Fold04: intercept=TRUE
## - Fold04: intercept=TRUE
## + Fold05: intercept=TRUE
## - Fold05: intercept=TRUE
## + Fold06: intercept=TRUE
## - Fold06: intercept=TRUE
## + Fold07: intercept=TRUE
## - Fold07: intercept=TRUE
## + Fold08: intercept=TRUE
## - Fold08: intercept=TRUE
## + Fold09: intercept=TRUE
## - Fold09: intercept=TRUE
## + Fold10: intercept=TRUE
## - Fold10: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```r
# Summarize the results
print(wine_model_kfold)
```

```
## Linear Regression
##
## 1599 samples
##   11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1438, 1439, 1440, 1438, 1439, 1439, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.6513858  0.3547876  0.50493
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
names(wine_model_kfold)
```

```
##  [1] "method"       "modelInfo"    "modelType"    "results"
##  [5] "pred"         "bestTune"     "call"         "dots"
##  [9] "metric"       "control"      "finalModel"   "preProcess"
## [13] "trainingData" "resample"     "resampledCM"  "perfNames"
## [17] "maximize"     "yLimits"      "times"        "levels"
## [21] "terms"        "coefnames"    "xlevels"

#REPORT THE RESULTS
wine_model_kfold$results

##   intercept      RMSE  Rsquared     MAE    RMSESD RsquaredSD      MAESD
## 1      TRUE 0.6513858 0.3547876 0.50493 0.0396182   0.056247 0.03324951
```

(f) Fit a regression tree using the same covariates in your "best" fit model from part (d). Use cross validation to select the "best" tree.

```
# creating regression tree using rpart library
wine_tree <-rpart(quality ~
volatile.acidity+chlorides+total.sulfur.dioxide+sulphates+alcohol+pH+free.sul
fur.dioxide, data =wineQuality)

#looking at the summary
summary(wine_tree)

## Call:
## rpart(formula = quality ~ volatile.acidity + chlorides +
total.sulfur.dioxide +
##     sulphates + alcohol + pH + free.sulfur.dioxide, data = wineQuality)
##   n= 1599
##
##           CP nsplit rel error    xerror       xstd
## 1 0.17822061      0 1.0000000 1.0013246 0.03787020
## 2 0.05358865      1 0.8217794 0.8313643 0.03612507
## 3 0.02974329      2 0.7681907 0.7940493 0.03343543
## 4 0.02888577      3 0.7384474 0.7900962 0.03288336
## 5 0.02234278      4 0.7095617 0.7755678 0.03226893
## 6 0.01927238      5 0.6872189 0.7568986 0.03107502
## 7 0.01511346      6 0.6679465 0.7342583 0.02962010
## 8 0.01015909      7 0.6528331 0.7180119 0.02895062
## 9 0.01000000      9 0.6325149 0.7109817 0.02911450
##
## Variable importance
##              alcohol     volatile.acidity            sulphates
##                   39                   21                   18
##            chlorides                   pH total.sulfur.dioxide
##                    8                    7                    5
##  free.sulfur.dioxide
##                    2
##
## Node number 1: 1599 observations,    complexity param=0.1782206
##   mean=5.636023, MSE=0.6517605
```

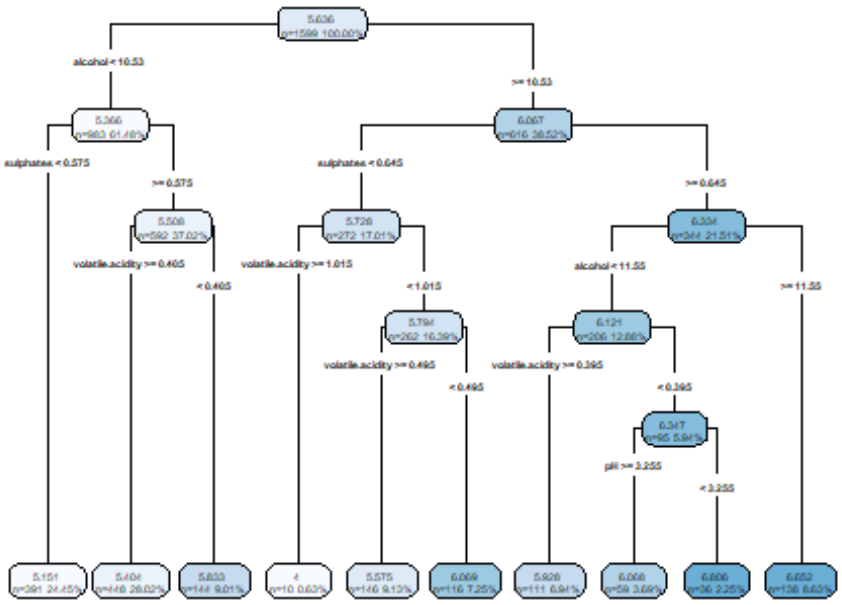
```
##   left son=2 (983 obs) right son=3 (616 obs)
##   Primary splits:
##       alcohol              < 10.525   to the left,  improve=0.1782206, (0
missing)
##       sulphates            < 0.645    to the left,  improve=0.1256516, (0
missing)
##       volatile.acidity     < 0.425    to the right, improve=0.1140062, (0
missing)
##       chlorides            < 0.0665   to the right, improve=0.0374106, (0
missing)
##       total.sulfur.dioxide < 59.5     to the right, improve=0.0360760, (0
missing)
##   Surrogate splits:
##       chlorides            < 0.0685   to the right, agree=0.690,
adj=0.195, (0 split)
##       volatile.acidity     < 0.3675   to the right, agree=0.662,
adj=0.123, (0 split)
##       total.sulfur.dioxide < 17.5     to the right, agree=0.641,
adj=0.068, (0 split)
##       sulphates            < 0.675    to the left,  agree=0.635,
adj=0.054, (0 split)
##       pH                   < 3.535    to the left,  agree=0.635,
adj=0.052, (0 split)
##
## Node number 2: 983 observations,    complexity param=0.02888577
##   mean=5.366226, MSE=0.4314941
##   left son=4 (391 obs) right son=5 (592 obs)
##   Primary splits:
##       sulphates            < 0.575    to the left,  improve=0.07097282, (0
missing)
##       volatile.acidity     < 0.335    to the right, improve=0.06388554, (0
missing)
##       alcohol              < 9.85     to the left,  improve=0.05212216, (0
missing)
##       total.sulfur.dioxide < 83.5     to the right, improve=0.02749674, (0
missing)
##       chlorides            < 0.0665   to the right, improve=0.01227854, (0
missing)
##   Surrogate splits:
##       volatile.acidity     < 0.6525   to the right, agree=0.636,
adj=0.084, (0 split)
##       total.sulfur.dioxide < 9.5      to the left,  agree=0.608,
adj=0.015, (0 split)
##       chlorides            < 0.0575   to the left,  agree=0.607,
adj=0.013, (0 split)
##       free.sulfur.dioxide  < 56       to the right, agree=0.607,
adj=0.013, (0 split)
##
## Node number 3: 616 observations,    complexity param=0.05358865
##   mean=6.066558, MSE=0.7017388
```

```
##   left son=6 (272 obs) right son=7 (344 obs)
##   Primary splits:
##       sulphates        < 0.645    to the left,  improve=0.12919720, (0
missing)
##       volatile.acidity < 0.87     to the right, improve=0.11482610, (0
missing)
##       alcohol          < 11.55    to the left,  improve=0.10309310, (0
missing)
##       pH               < 3.355    to the right, improve=0.07557599, (0
missing)
##       chlorides        < 0.0785   to the right, improve=0.01831590, (0
missing)
##   Surrogate splits:
##       volatile.acidity     < 0.5875   to the right, agree=0.653,
adj=0.213, (0 split)
##       pH                   < 3.405    to the right, agree=0.630,
adj=0.162, (0 split)
##       total.sulfur.dioxide < 14.5     to the left,  agree=0.625,
adj=0.151, (0 split)
##       free.sulfur.dioxide  < 5.5      to the left,  agree=0.597,
adj=0.088, (0 split)
##       chlorides            < 0.0595   to the left,  agree=0.575,
adj=0.037, (0 split)
##
## Node number 4: 391 observations
##   mean=5.150895, MSE=0.3276143
##
## Node number 5: 592 observations,    complexity param=0.01927238
##   mean=5.508446, MSE=0.449253
##   left son=10 (448 obs) right son=11 (144 obs)
##   Primary splits:
##       volatile.acidity     < 0.405    to the right, improve=0.07551952, (0
missing)
##       total.sulfur.dioxide < 81.5     to the right, improve=0.05845854, (0
missing)
##       alcohol              < 9.85     to the left,  improve=0.05386312, (0
missing)
##       chlorides            < 0.0975   to the right, improve=0.03262428, (0
missing)
##       pH                   < 3.535    to the right, improve=0.02710288, (0
missing)
##   Surrogate splits:
##       chlorides           < 0.0565   to the right, agree=0.765, adj=0.035,
(0 split)
##       free.sulfur.dioxide < 2.5      to the right, agree=0.764, adj=0.028,
(0 split)
##       alcohol             < 8.6      to the right, agree=0.758, adj=0.007,
(0 split)
##
## Node number 6: 272 observations,    complexity param=0.02974329
```

```
##   mean=5.727941, MSE=0.7053958
##   left son=12 (10 obs) right son=13 (262 obs)
##   Primary splits:
##       volatile.acidity    < 1.015    to the right, improve=0.16155630, (0
missing)
##       alcohol             < 11.45    to the left,  improve=0.11901850, (0
missing)
##       pH                  < 3.365    to the right, improve=0.09055459, (0
missing)
##       sulphates           < 0.585    to the left,  improve=0.04970438, (0
missing)
##       free.sulfur.dioxide < 28.5     to the left,  improve=0.03110483, (0
missing)
##
## Node number 7: 344 observations,    complexity param=0.02234278
##   mean=6.334302, MSE=0.5364978
##   left son=14 (206 obs) right son=15 (138 obs)
##   Primary splits:
##       alcohol              < 11.55    to the left,  improve=0.12616750, (0
missing)
##       chlorides            < 0.0785   to the right, improve=0.05765389, (0
missing)
##       total.sulfur.dioxide < 101.5    to the right, improve=0.05496021, (0
missing)
##       volatile.acidity     < 0.425    to the right, improve=0.04136603, (0
missing)
##       free.sulfur.dioxide  < 19.5     to the right, improve=0.03298003, (0
missing)
##   Surrogate splits:
##       chlorides            < 0.053    to the right, agree=0.651,
adj=0.130, (0 split)
##       pH                   < 3.565    to the left,  agree=0.619,
adj=0.051, (0 split)
##       volatile.acidity     < 0.14     to the right, agree=0.608,
adj=0.022, (0 split)
##       total.sulfur.dioxide < 15.5     to the right, agree=0.608,
adj=0.022, (0 split)
##       sulphates            < 1.12     to the left,  agree=0.602,
adj=0.007, (0 split)
##
## Node number 10: 448 observations
##   mean=5.404018, MSE=0.3925731
##
## Node number 11: 144 observations
##   mean=5.833333, MSE=0.4861111
##
## Node number 12: 10 observations
##   mean=4, MSE=0.6
##
## Node number 13: 262 observations,    complexity param=0.01511346
```

```
##   mean=5.793893, MSE=0.5911077
##   left son=26 (146 obs) right son=27 (116 obs)
##   Primary splits:
##       volatile.acidity    < 0.495    to the right, improve=0.10170270, (0
missing)
##       alcohol             < 11.45    to the left,  improve=0.09838534, (0
missing)
##       pH                  < 3.295    to the right, improve=0.07618253, (0
missing)
##       sulphates           < 0.585    to the left,  improve=0.04299293, (0
missing)
##       free.sulfur.dioxide < 31.5     to the left,  improve=0.03668719, (0
missing)
##   Surrogate splits:
##       pH                   < 3.305    to the right, agree=0.733,
adj=0.397, (0 split)
##       alcohol              < 11.85    to the left,  agree=0.641,
adj=0.190, (0 split)
##       total.sulfur.dioxide < 12.5     to the right, agree=0.637,
adj=0.181, (0 split)
##       free.sulfur.dioxide  < 4.5      to the right, agree=0.595,
adj=0.086, (0 split)
##       chlorides            < 0.0365   to the right, agree=0.569,
adj=0.026, (0 split)
##
## Node number 14: 206 observations,    complexity param=0.01015909
##   mean=6.121359, MSE=0.4949807
##   left son=28 (111 obs) right son=29 (95 obs)
##   Primary splits:
##       volatile.acidity     < 0.395    to the right, improve=0.08832113, (0
missing)
##       total.sulfur.dioxide < 49.5     to the right, improve=0.06808035, (0
missing)
##       chlorides            < 0.0945   to the right, improve=0.05079896, (0
missing)
##       free.sulfur.dioxide  < 25.5     to the right, improve=0.03611908, (0
missing)
##       pH                   < 3.255    to the right, improve=0.02835972, (0
missing)
##   Surrogate splits:
##       sulphates            < 0.765    to the left,  agree=0.655,
adj=0.253, (0 split)
##       chlorides            < 0.0675   to the right, agree=0.617,
adj=0.168, (0 split)
##       pH                   < 3.305    to the right, agree=0.583,
adj=0.095, (0 split)
##       total.sulfur.dioxide < 10.5     to the right, agree=0.568,
adj=0.063, (0 split)
##       alcohol              < 11.03333 to the right, agree=0.568,
adj=0.063, (0 split)
```

```
##
## Node number 15: 138 observations
##   mean=6.652174, MSE=0.4297417
##
## Node number 26: 146 observations
##   mean=5.575342, MSE=0.5045975
##
## Node number 27: 116 observations
##   mean=6.068966, MSE=0.5642093
##
## Node number 28: 111 observations
##   mean=5.927928, MSE=0.337148
##
## Node number 29: 95 observations,    complexity param=0.01015909
##   mean=6.347368, MSE=0.5845983
##   left son=58 (59 obs) right son=59 (36 obs)
##   Primary splits:
##       pH                   < 3.255    to the right, improve=0.21911830, (0
missing)
##       total.sulfur.dioxide < 56.5     to the right, improve=0.18528400, (0
missing)
##       free.sulfur.dioxide  < 24.5     to the right, improve=0.11666000, (0
missing)
##       alcohol              < 10.75    to the left,  improve=0.05498168, (0
missing)
##       chlorides            < 0.086    to the right, improve=0.05160159, (0
missing)
##   Surrogate splits:
##       total.sulfur.dioxide < 28.5     to the right, agree=0.737,
adj=0.306, (0 split)
##       free.sulfur.dioxide  < 9.5      to the right, agree=0.716,
adj=0.250, (0 split)
##       chlorides            < 0.0635   to the right, agree=0.663,
adj=0.111, (0 split)
##       sulphates            < 0.935    to the left,  agree=0.663,
adj=0.111, (0 split)
##       volatile.acidity     < 0.245    to the right, agree=0.642,
adj=0.056, (0 split)
##
## Node number 58: 59 observations
##   mean=6.067797, MSE=0.5038782
##
## Node number 59: 36 observations
##   mean=6.805556, MSE=0.378858

#plotting the tree
rpart.plot(wine_tree, digits = 4, fallen.leaves = TRUE, type = 4, extra =
101)
```



Problem 3: TheWisconsinBreastCancerdatasetisavailableasacomma-delimitedtextfileontheUCIMachineLearning Repository http://archive.ics.uci.edu/ml. Our goal in this problem will be to predict whether observations (i.e. tumors) are malignant or benign. (a) Obtain the data, and load it into R by pulling it directly from the web. (Do not download it and import it from a CSV file.) Give a brief description of the data.

```
fileURL <- "http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"
myfile <- readLines(fileURL)
#head(myfile)
breastCancerData <- read.csv(fileURL, header = FALSE, sep = ",", quote = "\"'")

names(breastCancerData) <- c('sample_code_number', 'clump_thickness',
'uniformity_of_cell_size',
         'uniformity_of_cell_shape', 'marginal_adhesion',
'single_epithelial_cell_size',
         'bare_nuclei', 'bland_chromatin',
         'normal_nucleoli','mitoses',
         'class')

str(breastCancerData)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ sample_code_number         : int  1000025 1002945 1015425 1016277
1017023 1017122 1018099 1018561 1033078 1033078 ...
```

```
##  $ clump_thickness            : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ uniformity_of_cell_size    : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ uniformity_of_cell_shape   : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ marginal_adhesion          : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ single_epithelial_cell_size: int  2 7 2 3 2 7 2 2 2 2 ...
##  $ bare_nuclei                : Factor w/ 11 levels "?","1","10","2",..: 2
3 4 6 2 3 3 2 2 2 ...
##  $ bland_chromatin            : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ normal_nucleoli            : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ mitoses                    : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ class                      : int  2 2 2 2 2 4 2 2 2 2 ...
```

Brief Description of the data : the dataset consists of 699 observations and 11 variables during the diagnoses process, pathologists look at the following characteristics to come to a conclusion: clump_thickness,uniformity_of_cell_size, uniformity_of_cell_shape, marginal_adhesion, single_epithelial_cell_size, bare_nuclei, bland_chromatin, normal_nucleoli, mitoses

(b) Tidy the data, ensuring that each variable is properly named and cast as the correct data type. Discuss any missing data.

```
check <- sum(is.na(breastCancerData))
check #printing about the number of na's in the dataset so that we can remove
them if necessary

## [1] 0
```

There are no NAs in the dataset.

```
head(breastCancerData)

##   sample_code_number clump_thickness uniformity_of_cell_size
## 1            1000025               5                        1
## 2            1002945               5                        4
## 3            1015425               3                        1
## 4            1016277               6                        8
## 5            1017023               4                        1
## 6            1017122               8                       10
##   uniformity_of_cell_shape marginal_adhesion single_epithelial_cell_size
## 1                        1                 1                           2
## 2                        4                 5                           7
## 3                        1                 1                           2
## 4                        8                 1                           3
## 5                        1                 3                           2
## 6                       10                 8                           7
##   bare_nuclei bland_chromatin normal_nucleoli mitoses class
## 1           1               3               1       1     2
## 2          10               3               2       1     2
## 3           2               3               1       1     2
## 4           4               3               7       1     2
```

```
## 5           1               3               1       1     2
## 6          10               9               7       1     4
```

```
#lets drop column sample_code_number
breastCancerData <- dplyr::select(breastCancerData, -c(1))
head(breastCancerData)
```

```
##   clump_thickness uniformity_of_cell_size uniformity_of_cell_shape
## 1               5                       1                       1
## 2               5                       4                       4
## 3               3                       1                       1
## 4               6                       8                       8
## 5               4                       1                       1
## 6               8                      10                      10
##   marginal_adhesion single_epithelial_cell_size bare_nuclei
## 1                 1                           2           1
## 2                 5                           7          10
## 3                 1                           2           2
## 4                 1                           3           4
## 5                 3                           2           1
## 6                 8                           7          10
##   bland_chromatin normal_nucleoli mitoses class
## 1               3               1       1     2
## 2               3               2       1     2
## 3               3               1       1     2
## 4               3               7       1     2
## 5               3               1       1     2
## 6               9               7       1     4
```

```
breastCancerData$class <- ifelse(breastCancerData$class==2, 0,1)
```

Since 2 was for benign, 0 indicates false cases and 1 inidcated malignant

```
#breastCancerData$class
```

(c) Split the data into a training and test set such that a random 70% of the observations are in the training set.

```
# code adapted from https://rpubs.com/ID_Tech/S1 AND
https://stackoverflow.com/a/31634462

# Set seed for reproducibility
set.seed(112718)
# splits the data in the ratio mentioned in SplitRatio. After splitting marks
these rows as logical
# TRUE and the the remaining are marked as logical FALSE
sample = sample.split(breastCancerData$class, SplitRatio = .7)
# creates a training dataset named train with rows which are marked as TRUE
breastCancerData_train = subset(breastCancerData, sample == TRUE)
# creates a training dataset named test with rows which are marked as FALSE
str(breastCancerData_train)
```

```
## 'data.frame':    490 obs. of  10 variables:
##  $ clump_thickness            : int  5 3 8 1 2 2 4 1 5 1 ...
##  $ uniformity_of_cell_size    : int  4 1 10 1 1 1 2 1 3 1 ...
##  $ uniformity_of_cell_shape   : int  4 1 10 1 2 1 1 1 3 1 ...
##  $ marginal_adhesion          : int  5 1 8 1 1 1 1 1 3 1 ...
##  $ single_epithelial_cell_size: int  7 2 7 2 2 2 2 1 2 2 ...
##  $ bare_nuclei                : Factor w/ 11 levels "?","1","10","2",..: 3
4 3 3 2 2 2 2 5 5 ...
##  $ bland_chromatin            : int  3 3 9 3 3 1 2 3 4 3 ...
##  $ normal_nucleoli            : int  2 1 7 1 1 1 1 1 4 1 ...
##  $ mitoses                    : int  1 1 1 1 1 5 1 1 1 1 ...
##  $ class                      : num  0 0 1 0 0 0 0 0 1 0 ...

breastCancerData_test  = subset(breastCancerData, sample == FALSE)
nrow(breastCancerData_test)

## [1] 209
```

(d) Fit a regression model to predict whether tissue samples are malignant or benign. Classify cases in the validation set. Compute and discuss the resulting confusion matrix.

```
#fitting a regression model on the training data
breastcancer_glm1 <- glm(class ~ . ,family = binomial,
data=breastCancerData_train)
summary(breastcancer_glm1)

##
## Call:
## glm(formula = class ~ ., family = binomial, data = breastCancerData_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46229  -0.07176  -0.03793   0.01154   2.07400
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -15.53811    3.03114  -5.126 2.96e-07 ***
## clump_thickness                0.38022    0.17661   2.153 0.031330 *
## uniformity_of_cell_size        0.01846    0.26903   0.069 0.945287
## uniformity_of_cell_shape       0.79530    0.34879   2.280 0.022600 *
## marginal_adhesion              0.11411    0.15600   0.731 0.464508
## single_epithelial_cell_size    0.06728    0.19472   0.346 0.729705
## bare_nuclei1                   4.77569    2.24180   2.130 0.033148 *
## bare_nuclei10                  7.91833    2.10644   3.759 0.000171 ***
## bare_nuclei2                   4.62702    2.45039   1.888 0.058989 .
## bare_nuclei3                   6.93097    2.23861   3.096 0.001961 **
## bare_nuclei4                   7.47942    3.08216   2.427 0.015238 *
## bare_nuclei5                   6.25250    2.46024   2.541 0.011040 *
## bare_nuclei6                  25.03077 2907.15238   0.009 0.993130
## bare_nuclei7                   4.45926    2.39048   1.865 0.062123 .
```

```
## bare_nuclei8                    5.37996    2.00331   2.686 0.007241 **
## bare_nuclei9                   22.59326 2500.03375   0.009 0.992789
## bland_chromatin                 0.64766    0.25375   2.552 0.010700 *
## normal_nucleoli                 0.17990    0.17343   1.037 0.299591
## mitoses                         0.47577    0.32592   1.460 0.144352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 631.346  on 489  degrees of freedom
## Residual deviance:  60.918  on 471  degrees of freedom
## AIC: 98.918
##
## Number of Fisher Scoring iterations: 17
```

```
#performing predictions on the test data
predictions <- predict(breastcancer_glm1, newdata = breastCancerData_test)

#creating confusion matrix
cfmtrx<-table(predictions>0.5,breastCancerData_test$class)
cfmtrx
```

```
##
##           0   1
##   FALSE 134   8
##   TRUE    3  64
```

```
#lets calculate accuracy
# (2 for benign(i.e. FALSE), 4 for malignant(i.e. TRUE))
(134+64)/(134 +64+11)
```

```
## [1] 0.9473684
```

We get close to 94% accuracy for this model.

(e) Fit a random forest model to predict whether tissue samples are malignant or benign. Classify cases in the validation set. Compute and discuss the resulting confusion matrix.

```
# using random forest for training data
breastcancer_rf <- randomForest(class ~ . ,family = binomial,
data=breastCancerData_train)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
#looking at the summary of model
summary(breastcancer_rf)
```

```
##                 Length Class  Mode
## call               4    -none- call
```

```
## type              1    -none- character
## predicted       490    -none- numeric
## mse             500    -none- numeric
## rsq             500    -none- numeric
## oob.times       490    -none- numeric
## importance        9    -none- numeric
## importanceSD      0    -none- NULL
## localImportance   0    -none- NULL
## proximity         0    -none- NULL
## ntree             1    -none- numeric
## mtry              1    -none- numeric
## forest           11    -none- list
## coefs             0    -none- NULL
## y               490    -none- numeric
## test              0    -none- NULL
## inbag             0    -none- NULL
## terms             3    terms  call
```

```
#predicting using random forest model
predictions_rf <- predict(breastcancer_rf, newdata = breastCancerData_test)


#keeping the threshold as 0.5
cfmtrx<-table(predictions_rf>0.5,breastCancerData_test$class)
cfmtrx
```

```
##
##           0   1
##   FALSE 134   6
##   TRUE    3  66
```

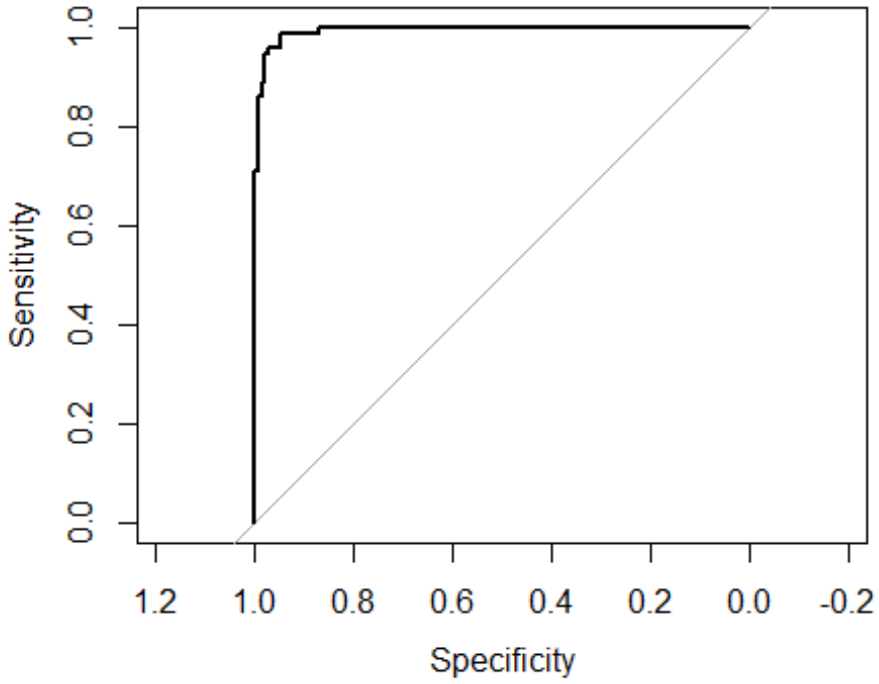
```
#calculating accuracy of the model
(134+65)/(134+65+7+3)
```

```
## [1] 0.9521531
```

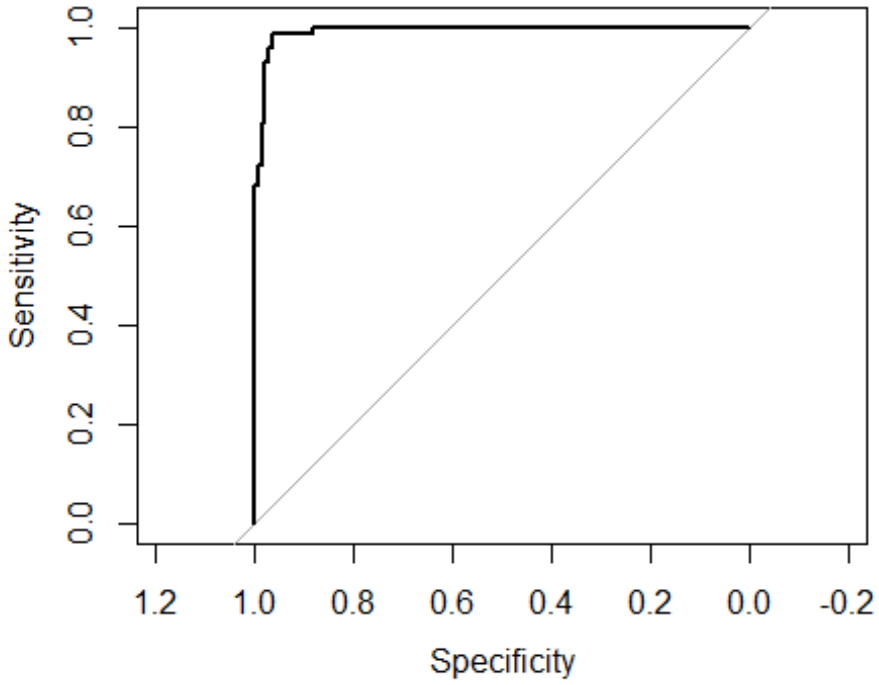
We get 95% accuracy with random forest model.

(f) Compare the models from part (d) and (e) using ROC curves. Which do you prefer? Be sure to justify your preference.

```
#lets plot ROC curve for linear regression
logistic_roc_forpartd <- roc(breastCancerData_test$class ~ predictions)
plot(logistic_roc_forpartd)
```




```r
#lets plot ROC for randomForest model
logistic_roc_forparte <- roc(breastCancerData_test$class ~ predictions_rf)
plot(logistic_roc_forparte)
```



Problem 4 (15 pts) Please answer the questions below by writing a short response.

(a) Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal in each application inference or prediction? Explain your answer.

Stastitical Classification consists of identifying to which set of categories does a new observation belong. Some of the real-life applications I can think about are: 1. Classifying a new mail arriving in the inbox into a spam or not a spam Response: email is a spam or not Predictors: subjectline, emailbodycontent, keywords, classification(prior classification in training set), sent_by, reply_to 2. At my firm where I am currently interning, a team built a classification engine for the user entered comments in Reviews section to classify them into positive and negative sentiments about the product. Response: Comment is negative or positive Predictors: Tokenized keywords from the review text entered by users 3. Biological classification : To annotate species of various kinds, we can have an algorithm which has been trained on samples and can classify a newly submitted picture into a category. Response: Name of the species, class Predictors: Name, class, subclass, image 4. Speech recognition is one area where classification can be used to identify which speaker is currently speaking by making patterns about his speech frequencies. Response:Identify the person who is speaking Predictors: Voice input, name of the person, device name from where the speaker is speaking, time of the day

The goal in all the above mentioned scenarios is Prediction.

(b) Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal in each application inference or predictions? Explain your answer.

1. Affect on housing prices on one country based on financial markets in another country The application here is inferential- since we are looking back on what must have happened and what variables were involved. I am considering the example of 2008 financial crisis and how it impacted the housing prices in other countries like Taiwan and China. Response: Housing Prices increased or decreased Predictors: country name, city name, location(East, west, central), housing prices before,housing prices after crisis, time of the year. Goal: Inference
2. Weather Prediction: Based on the past years trends in winds, temperature Response: Weather patterns - expected temperature and precipitation Predictors: Atleast 3 years of data or more detailing temperature, precipitation, humidity, air pressure, air density, wind speed, region, time of the year Goal: Prediction
3. Quantitative UX Research to test various hypotheses for example: How many users successfully configure the trial product and convert to paid customers after 30 days Response: conversion rate, %self-configuration Predictors: user ids, region, number of pages configured, %usage of product,%transactions done in the trial phase Goal: Inference or patterns in the findings will tell us more about the product

(c) What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible

approach be preferred? Answer: In general, the more flexible our model is, the less bias (in absolute value) and the more variance we will get when predicting on a test dataset. Our goal here is to minimize the sum of the squared bias and the variance, there are trade offs. In some scenarios flexible model will perform better than the inflexible one and in other we can see the scenario turning other way.

(i)For cases in which the sample size is large and the number of predictors is small A flexible model will perform better in general. Because of the large sample size, we're less likely to overfit even when using a more flexible model. Meanwhile, a more flexible model tends to reduce bias.

(ii)For cases in which the number of predictors is large and the sample size is small. An inflexible model will perform better in general. A flexible model will cause overfitting because of the small sample size. This usually means a bigger inflation in variance and a small reduction in bias.

(iii)for cases in which the relationship between the predictors and response is highly non-linear. A flexible model will perform better in general because it'll be necessary to use a flexible model to find the non-linear effect.

(iv)An inflexible model will perform better in general. Because a flexible model will capture too much of the noise in the data due to the large variance of the errors.

Problem 5 (10 pts) Suppose we have a dataset with five predictors, X1 = GPA, X2 = IQ, X3 = Degree (1 for B.A. degree holder, and 0 for B.S. degree holder), X4 = InteractionbetweenGPAandIQ, and X5 = InteractionbetweenGPAandDegree. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model and get $\hat{\beta}0 = 50$; $\hat{\beta}1 = 20$, $\hat{\beta}2 = 0.07$, $\hat{\beta}3 = 35$, $\hat{\beta}4 = 0.01$, and $\hat{\beta}5 = -10$.

(a) Which answer is correct and why?

i. For a fixed value of IQ and GPA, B.A. degree holders earn more on average than B.S. degree holders.

ii. For a fixed value of IQ and GPA, B.S. degree holders earn more on average than B.A. degree holders.

iii. For a fixed value of IQ and GPA, B.S. degree holders earn more on average than B.A. degree holders provided that the GPA is high enough.

iv. For a fixed value of IQ and GPA, B.A. degree holders earn more on average than B.S. degree holders provided that the GPA is high enough.

Writing the least square regression line equation from given information:

$\hat{y} = \hat{\beta}0 + \hat{\beta}1X1 + \hat{\beta}2X2 + \hat{\beta}3X3 + \hat{\beta}4X4 + \hat{\beta}5X5$ y_hat = 50 + 20GPA + 0.07IQ + 35Degree + 0.01GPA*IQ* -*10GPA*Degree

For BA holders, Degree =1

y_hat=50 + 20GPA + 0.07IQ + 35*1 + 0.01GPA*IQ -10GPA

this can be simplified as:

y_hat1 = 85 +10GPA +0.07IQ+0.01GPA*IQ

For BS degree holders: Degree = 0

y_hat2 = 50+20GPA+0.07IQ+0+0.1GPA*IQ-0

For a given GPA and IQ equating the two equations, we will get

85 +10GPA = 50+20GPA 35 = 10GPA GPA = 3.5

From this we can conclude that for a fixed GPA and IQ,, B.S. degree holders earn more on average than B.A. degree holders provided that the GPA is high enough. I would choose Option (iii)

(b) Predict the salary of a B.A. with IQ of 110 and a GPA of 4.0. For B.A., Degree =1 For IQ=110 and GPA = 4.0

y_hat=50 + 20GPA + 0.07IQ + 35*1 + 0.01GPA*IQ -10GPA y_hat = 50+20(4.0)+0.07*110 +35+0.01(4*110) - 10*4.0

The salary of a B.A in this case would be 137,100

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is little evidence of an interaction effect. Justify your answer.

Answer: We cannot conclude without running a model to test the null hyothesis for H0 = B4^ = 0 and finding the value of co-efficient and also finding its statistical significance using p-value, and f-statistic.

Statement of Compliance: Please copy and sign the following statement. I affirm that I have not collaborated on or asked questions about the content of this exam with any persons other than the instructor. Further, I certify that the attached work represents my own thinking. Any information, concepts, or words that originate from other sources are cited in accordance with University of Washington guidelines as published in the Academic Policies (available here: https://depts.washington.edu/infodocs/academic_policies/). I am aware of the serious consequences that result from improper discussions with others or from the improper citation of work that is not my own. Signed:Neelam Purswani Dated: 11 Decemeber 2018