FE590. Assignment #4.

Enter Your Name Here, or “Anonymous” if you want to remain anonymous..

2019-11-26

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

By filling out the following fields, you are signing this pledge. No assignment will get credit without being pledged.

Name:Chunli Liu

CWID:10430963

Date:11/8/2019

Instructions

When you have completed the assignment, knit the document into a PDF file, and upload *both* the .pdf and .Rmd files to Canvas.

Note that you must have LaTeX installed in order to knit the equations below. If you do not have it installed, simply delete the questions below.

```
CWID = 10430963 #Place here your Campus wide ID number, this will personalize
#your results, but still maintain the reproduceable nature of using seeds.
#If you ever need to reset the seed in this assignment, use this as your seed
#Papers that use -1 as this CWID variable will earn 0's so make sure you change
#this value before you submit your work.
personal = CWID %% 10000
set.seed(personal)
```

Question 1:

In this assignment, you will be required to find a set of data to run regression on. This data set should be financial in nature, and of a type that will work with the models we have discussed this semester (hint: we didn’t look at time series) You may not use any of the data sets in the ISLR package that we have been looking at all semester. Your data set that you choose should have both qualitative and quantitative variables. (or has variables that you can transform)

Provide a description of the data below, where you obtained it, what the variable names are and what it is describing.

Question 2:

Pick a quantitative variable and fit at least four different models in order to predict that variable using the other predictors. Determine which of the models is the best fit. You will need to provide strong reasons as to why the particular model you chose is the best one. You will need to confirm the model you have selected provides the best fit and that you have obtained the best version of that particular model (i.e. subset selection or validation for example). You need to convince the grader that you have chosen the best model.

```
##Question 1
## the data set is named as German Credit Risk, origns from Kaggle, I downloaded the data set and load it fr
om my PC
## load the data

url <- "D:\\2019 fall\\FE 590\\assignment-4\\german_credit_data.csv"
## use the options as.is = TRUE, and na.strings="?". Remove the unavailable data
data1=read.table(url,header=T,na.strings="NA",as.is = TRUE,fill = TRUE,sep = ",")
data1<-na.omit(data1)
## there is no unavailable data
sum(is.na(data1))
```

```
## [1] 0
```

```
## number of row and column
dim(data1)
```

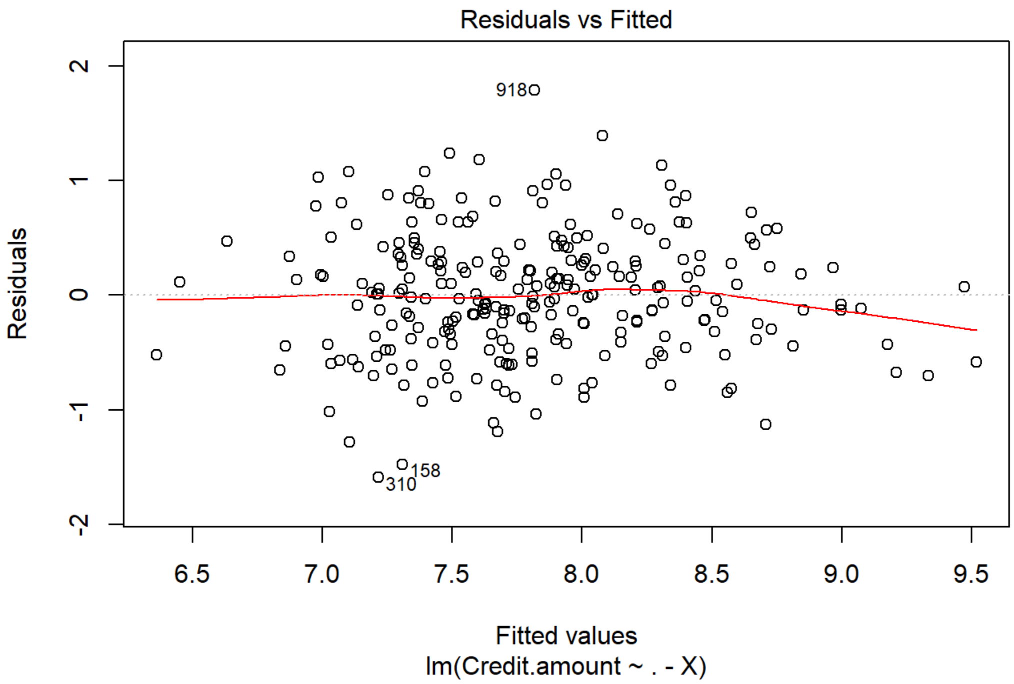
```
## [1] 522  10
```

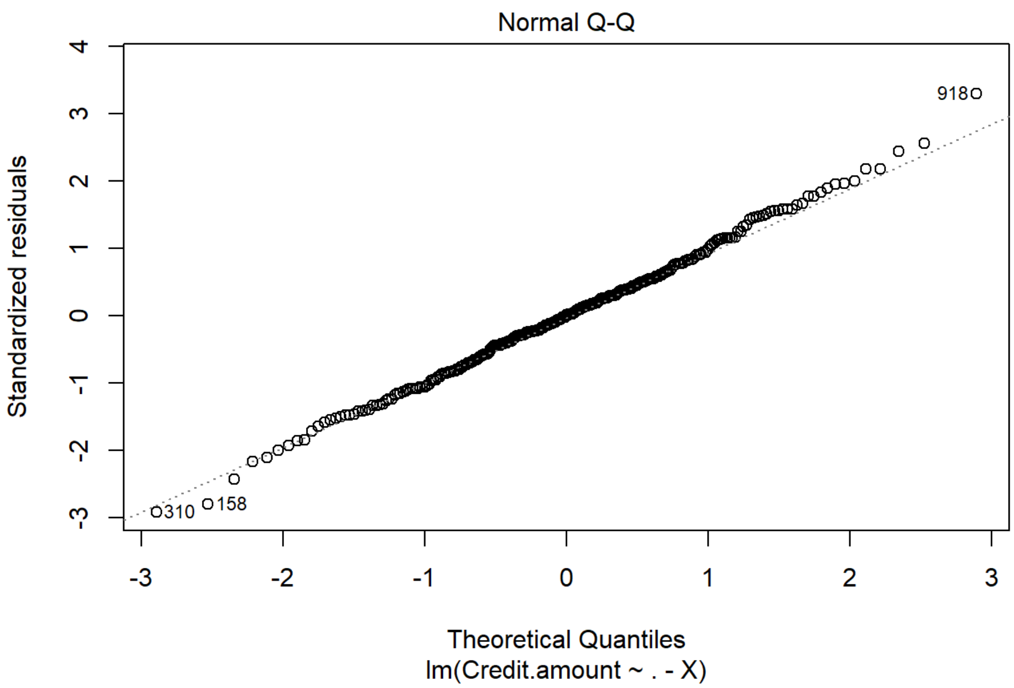
```
## take a look of the dataset
head(data1)
```

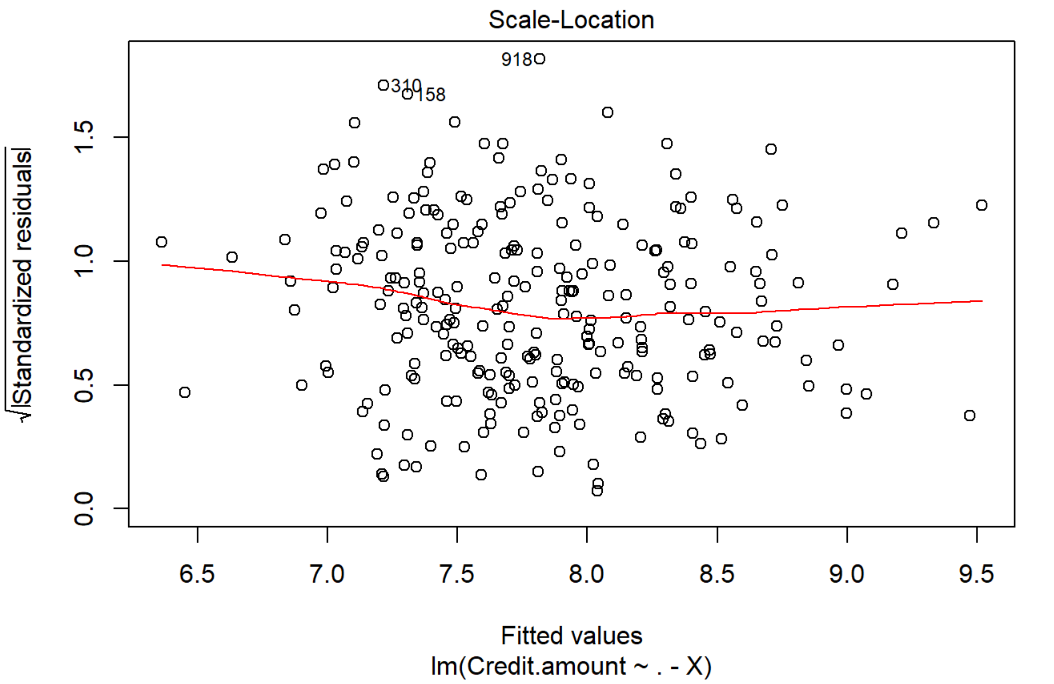
```
## we could see that the data set contains both quantative and qualitative variables as request
str(data1)
```

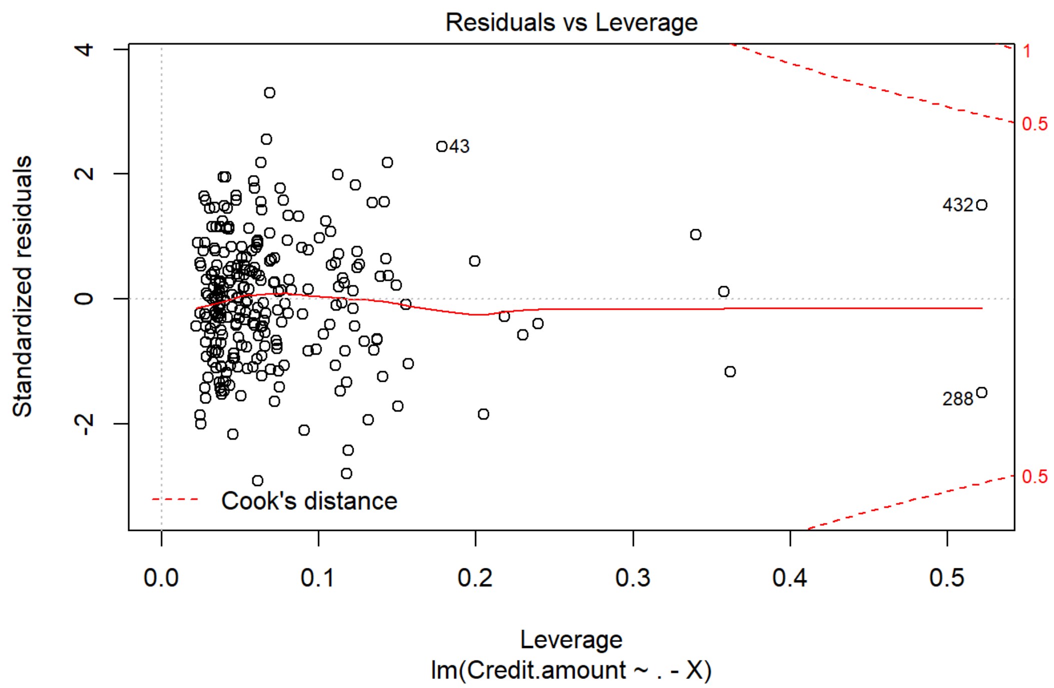
```
## 'data.frame':    522 obs. of  10 variables:
##  $ X               : int  1 3 4 7 9 10 11 12 13 14 ...
##  $ Age             : int  22 45 53 35 28 25 24 22 60 28 ...
##  $ Sex             : chr  "female" "male" "male" "male" ...
##  $ Job             : int  2 2 2 3 3 2 2 2 1 2 ...
##  $ Housing         : chr  "own" "free" "free" "rent" ...
##  $ Saving.accounts : chr  "little" "little" "little" "little" ...
##  $ Checking.account: chr  "moderate" "little" "little" "moderate" ...
##  $ Credit.amount   : int  5951 7882 4870 6948 5234 1295 4308 1567 1199 1403 ...
##  $ Duration        : int  48 42 24 36 30 12 48 12 24 15 ...
##  $ Purpose         : chr  "radio/TV" "furniture/equipment" "car" "car" ...
##  - attr(*, "na.action")= 'omit' Named int  1 3 6 7 9 17 18 20 21 25 ...
##   ..- attr(*, "names")= chr  "1" "3" "6" "7" ...
```

```
## the description of each variable

#1.Age (numeric): the age of observations
#2.Sex (text: male, female): gender of observations
#3.Job (numeric: 0 - unskilled and non-resident, 1 - unskilled and resident, 2 - skilled, 3 - highly skilled
): level of job of observations
#4.Housing (text: own, rent, or free)
#5.Saving accounts (text - little, moderate, quite rich, rich): conditions of Saving accounts
#6.Checking account (numeric, in DM - Deutsch Mark): conditions of Checking accounts
#7.Credit amount (numeric, in DM)
#8.Duration (numeric, in month): duration in month
#9.Purpose (text: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vac
ation/others): purpose of purchasing

##Question 2: Pick a quantitative variable and fit at least four different models

## log the variable which is pretty large
data1$Credit.amount = log(data1$Credit.amount)
attach(data1)

## split the data set
sample_size = floor(0.5*nrow(data1))
picked = sample(seq_len(nrow(data1)),size = sample_size)

## train data set
train =data1[picked,]

## test data set
test = data1[-picked,]


##fit linear regression we could see that the important predictors based on Credit.amount are Job, duation.
but the Purpose may also make sense because the p-value of Purposerepairs and Purposevacation/others is low.

fit.lm = lm(Credit.amount~.-X, data = train)
plot(fit.lm)
```


Residuals vs Fitted
918
158
310
Residuals
Fitted values
lm(Credit.amount ~ . - X)


Normal Q-Q
918
310
158
Standardized residuals
Theoretical Quantiles
lm(Credit.amount ~ . - X)





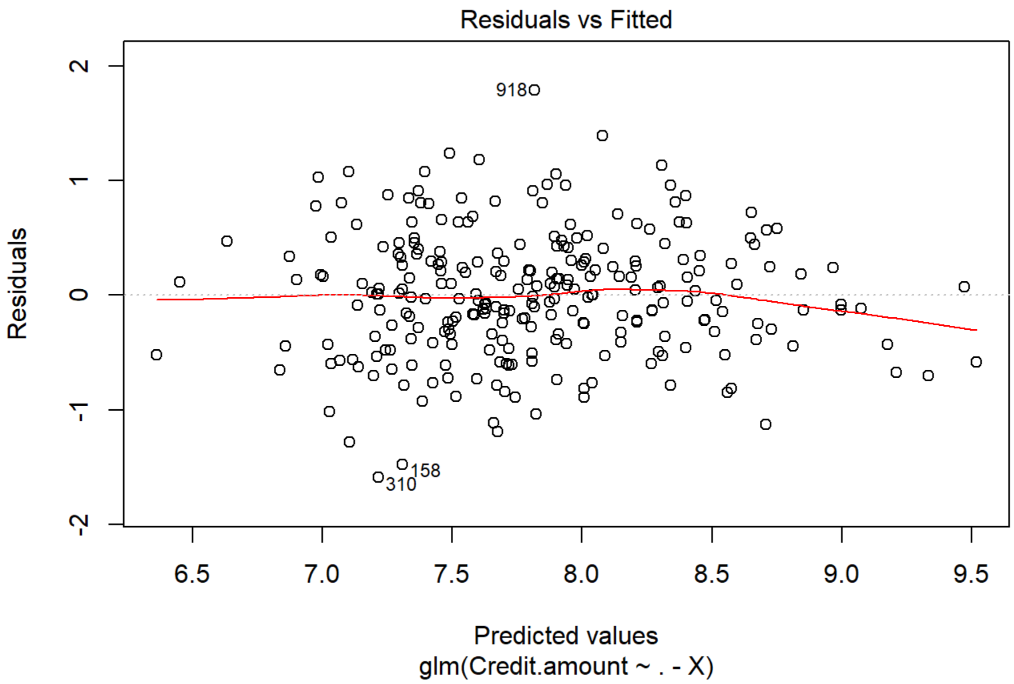

```
summary(fit.lm)
```

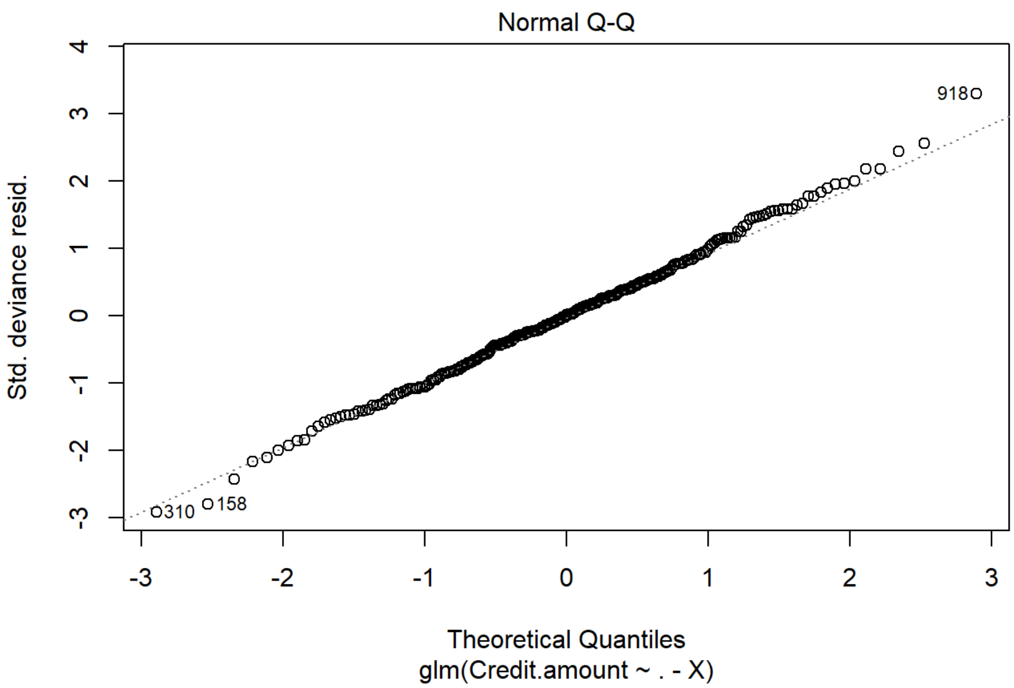
```
##
## Call:
## lm(formula = Credit.amount ~ . - X, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59340 -0.38289  0.00292  0.32972  1.79138
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.410334   0.268994  23.831  < 2e-16 ***
## Age                          0.007259   0.003358   2.162  0.03164 *
## Sexmale                     -0.089313   0.081475  -1.096  0.27408
## Job                          0.212462   0.050283   4.225 3.38e-05 ***
## Housingown                   0.058343   0.122432   0.477  0.63412
## Housingrent                  0.086514   0.144247   0.600  0.54923
## Saving.accountsmoderate     -0.046100   0.118411  -0.389  0.69738
## Saving.accountsquite rich   -0.096813   0.169043  -0.573  0.56737
## Saving.accountsrich         -0.202814   0.164478  -1.233  0.21874
## Checking.accountmoderate     0.019631   0.079329   0.247  0.80476
## Checking.accountrich        -0.229999   0.118976  -1.933  0.05438 .
## Duration                     0.035793   0.003108  11.515  < 2e-16 ***
## Purposecar                   0.092337   0.129076   0.715  0.47507
## Purposedomestic appliances -0.943633   0.350612  -2.691  0.00761 **
## Purposeeducation            -0.041456   0.206785  -0.200  0.84128
## Purposefurniture/equipment  0.148157   0.137474   1.078  0.28223
## Purposeradio/TV             -0.194891   0.133642  -1.458  0.14605
## Purposerepairs              -0.085397   0.267623  -0.319  0.74993
## Purposevacation/others       0.643795   0.419965   1.533  0.12659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5626 on 242 degrees of freedom
## Multiple R-squared:  0.5024, Adjusted R-squared:  0.4654
## F-statistic: 13.58 on 18 and 242 DF,  p-value: < 2.2e-16
```

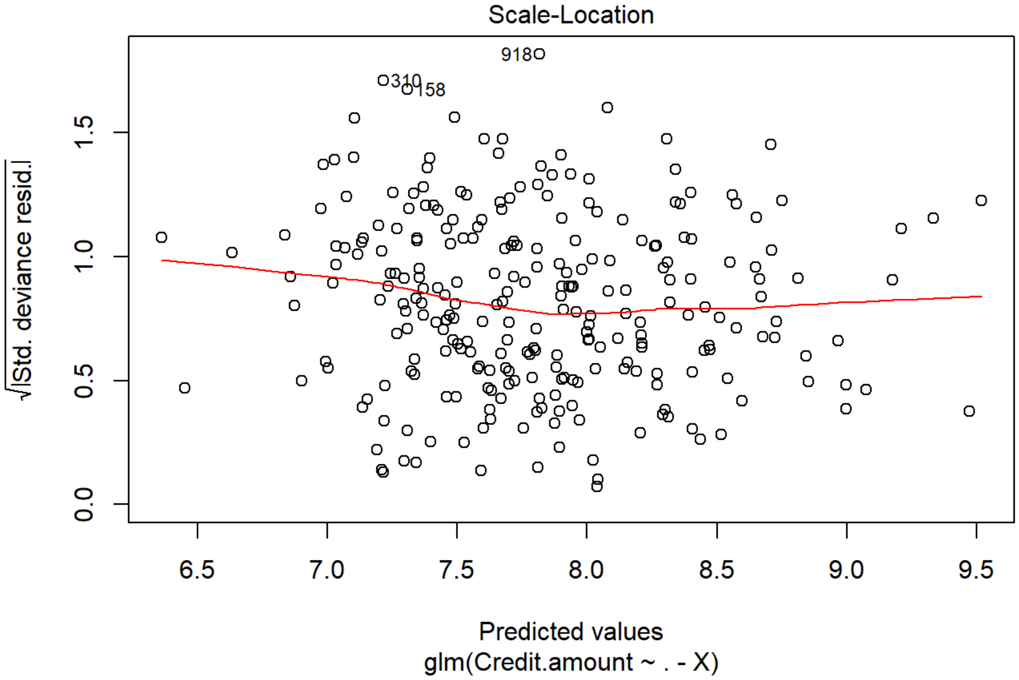
```
## now let's see the MSE of linear regression (0.3470)
fit.pred = predict(fit.lm, test)
lm.fit.err = mean((test$Credit.amount - fit.pred)^2)
lm.fit.err
```

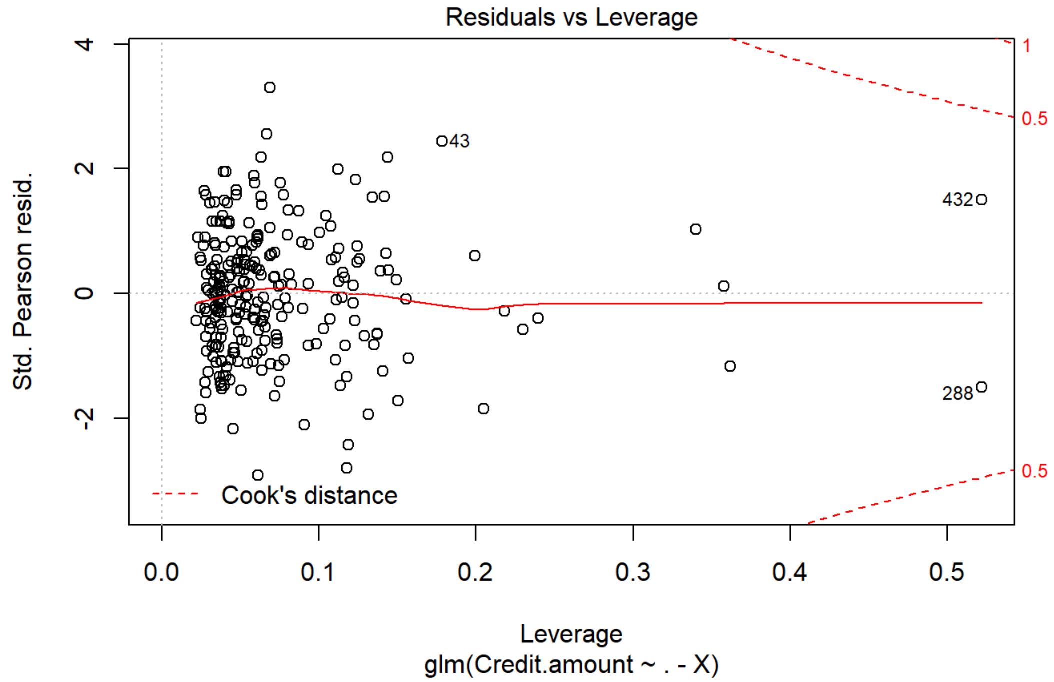
```
## [1] 0.3556043
```

```
## next I would like to use the logistic model to fit the data set
fit.glm = glm(Credit.amount~.-X, data = train)
plot(fit.glm)
```


Residuals vs Fitted
Residuals
Predicted values
glm(Credit.amount ~ . - X)
918
158
310
2
1
0
-1
-2
6.5
7.0
7.5
8.0
8.5
9.0
9.5


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(Credit.amount ~ . - X)
918
310
158
4
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3






```
summary(fit.glm)
```

```
##
## Call:
## glm(formula = Credit.amount ~ . - X, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.59340  -0.38289   0.00292   0.32972   1.79138
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.410334   0.268994  23.831  < 2e-16 ***
## Age                          0.007259   0.003358   2.162  0.03164 *
## Sexmale                     -0.089313   0.081475  -1.096  0.27408
## Job                          0.212462   0.050283   4.225 3.38e-05 ***
## Housingown                   0.058343   0.122432   0.477  0.63412
## Housingrent                  0.086514   0.144247   0.600  0.54923
## Saving.accountsmoderate     -0.046100   0.118411  -0.389  0.69738
## Saving.accountsquite rich   -0.096813   0.169043  -0.573  0.56737
## Saving.accountsrich         -0.202814   0.164478  -1.233  0.21874
## Checking.accountmoderate     0.019631   0.079329   0.247  0.80476
## Checking.accountrich        -0.229999   0.118976  -1.933  0.05438 .
## Duration                     0.035793   0.003108  11.515  < 2e-16 ***
## Purposecar                   0.092337   0.129076   0.715  0.47507
## Purposedomestic appliances -0.943633   0.350612  -2.691  0.00761 **
## Purposeeducation            -0.041456   0.206785  -0.200  0.84128
## Purposefurniture/equipment  0.148157   0.137474   1.078  0.28223
## Purposeradio/TV             -0.194891   0.133642  -1.458  0.14605
## Purposerepairs              -0.085397   0.267623  -0.319  0.74993
## Purposevacation/others       0.643795   0.419965   1.533  0.12659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3165251)
##
##     Null deviance: 153.949  on 260  degrees of freedom
## Residual deviance:  76.599  on 242  degrees of freedom
## AIC: 460.72
##
## Number of Fisher Scoring iterations: 2
```

```
## from the summary of the logistic regression model I learn that there are two important predictors accordi
ng to Credit.amount : Job, Duration and Purpose, just like linear regression model shows

## the MSE oflogistic regression, same as linear regression
fit.pred = predict(fit.glm, test)
glm.fit.err = mean((test$Credit.amount - fit.pred)^2)
glm.fit.err
```

```
## [1] 0.3556043
```

```
## fit the gam model
install.packages("gam",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'gam' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'gam'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\jolly\OneDrive\Documents\R\win-
## library\3.6\00LOCK\gam\libs\x64\gam.dll to C:
## \Users\jolly\OneDrive\Documents\R\win-library\3.6\gam\libs\x64\gam.dll:
## Permission denied
```

```
## Warning: restored 'gam'
```

```
##
## The downloaded binary packages are in
##  C:\Users\jolly\AppData\Local\Temp\RtmpkX3MWk\downloaded_packages
```

```
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.16.1
```

```
gam.fit = gam(Credit.amount~ns(Job,5)+ns(Duration, 5)+Purpose+Sex+Housing + Saving.accounts + Checking.accou
nt,data = train)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
## plotting the model
par(mfrow=c(1,3)) #to partition the Plotting Window
plot(gam.fit,se = TRUE,col = "red" )
```

```
## Warning in gplot.default(x = c("car", "car", "car", "radio/TV", "radio/
## TV", : The "x" component of "partial for Purpose" has class "character"; no
## gplot() methods available
```

```
## Warning in gplot.default(x = c("male", "female", "female", "female",
## "male", : The "x" component of "partial for Sex" has class "character"; no
## gplot() methods available
```

```
## Warning in gplot.default(x = c("own", "own", "rent", "own", "own", "own", :
## The "x" component of "partial for Housing" has class "character"; no
## gplot() methods available
```

```
## Warning in gplot.default(x = c("little", "moderate", "quite rich",
## "little", : The "x" component of "partial for Saving.accounts" has class
## "character"; no gplot() methods available
```

```
## Warning in gplot.default(x = c("little", "moderate", "little",
## "moderate", : The "x" component of "partial for Checking.account" has class
## "character"; no gplot() methods available
```

```
## (c)Evaluate the model obtained on the test set
gam.pred = predict(gam.fit, test)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
gam.err = mean((test$Credit.amount - gam.pred)^2)

gam.tss = mean((test$Credit.amount - mean(test$Credit.amount))^2)
test.rss = 1 - gam.err/gam.tss
test.rss
```

```
## [1] 0.5148802
```

```
## the test error rate (0.4641)
cat("the results produced a R squre value of", test.rss)
```

```
## the results produced a R squre value of 0.5148802
```

```
## gam model only agree that Duration is the most important feature and also admit Purpose may also need to be considered
summary(gam.fit)
```

```
##
## Call: gam(formula = Credit.amount ~ ns(Job, 5) + ns(Duration, 5) +
##     Purpose + Sex + Housing + Saving.accounts + Checking.account,
##     data = train)
## Deviance Residuals:
##       Min        1Q    Median        3Q       Max
## -1.684467 -0.337736 -0.002533  0.302347  1.887413
##
## (Dispersion Parameter for gaussian family taken to be 0.316)
##
##     Null Deviance: 153.9491 on 260 degrees of freedom
## Residual Deviance: 74.8811 on 237 degrees of freedom
## AIC: 464.7965
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## ns(Job, 5)         3 18.579  6.1928 19.6004 2.194e-11 ***
## ns(Duration, 5)    5 51.252 10.2503 32.4425 < 2.2e-16 ***
## Purpose            7  7.559  1.0798  3.4176  0.001685 **
## Sex                1  0.139  0.1394  0.4411  0.507236
## Housing            2  0.018  0.0091  0.0287  0.971702
## Saving.accounts    3  0.601  0.2003  0.6339  0.593841
## Checking.account   2  0.921  0.4603  1.4570  0.235018
## Residuals        237 74.881  0.3160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## conclusions: linear regression, glm model perform same including the accuracy rate and important predictors, gam model is worse than them in accuracy rate and have slight difference in selecting important predictors.

##  Now I decide to make dummy data to see if things go differently by using both qualititive and quantitative response because there are 5/9 of category features in my data set.

##quantitative response by performing tree model, linear regression,glm and gam model, plus hierarchical clustering.

a <- sub("free","1",data1$Housing)
b <- sub("own","2",a)
c <-sub("rent","3",b)
data1$Housing <- c
data1$Housing <- as.numeric(as.character(data1$Housing))
typeof(data1$Housing)
```

```
## [1] "double"
```

```
## convert variable names Credit.Savings

credit.savings.factor = factor(data1$Saving.accounts);
as.character(credit.savings.factor)
```

```
##   [1] "little"     "little"     "little"     "little"     "little"
##   [6] "little"     "little"     "little"     "little"     "little"
##  [11] "moderate"   "little"     "quite rich" "little"     "moderate"
##  [16] "little"     "rich"       "little"     "little"     "rich"
##  [21] "little"     "moderate"   "little"     "little"     "little"
##  [26] "little"     "little"     "quite rich" "little"     "moderate"
```

```
##  [26] "little"     "little"     "quite rich" "little"     "moderate"  
##  [31] "little"     "quite rich" "little"     "little"     "little"    
##  [36] "little"     "little"     "little"     "little"     "rich"      
##  [41] "little"     "little"     "little"     "little"     "little"    
##  [46] "little"     "little"     "little"     "little"     "moderate"  
##  [51] "moderate"   "little"     "little"     "rich"       "little"    
##  [56] "moderate"   "little"     "little"     "little"     "little"    
##  [61] "little"     "quite rich" "moderate"   "little"     "little"    
##  [66] "quite rich" "quite rich" "quite rich" "little"     "little"    
##  [71] "little"     "little"     "little"     "little"     "little"    
##  [76] "little"     "quite rich" "little"     "rich"       "little"    
##  [81] "little"     "little"     "moderate"   "little"     "little"    
##  [86] "little"     "rich"       "moderate"   "little"     "moderate"  
##  [91] "little"     "moderate"   "little"     "little"     "rich"      
##  [96] "little"     "little"     "little"     "little"     "little"    
## [101] "little"     "quite rich" "little"     "little"     "little"    
## [106] "little"     "little"     "moderate"   "little"     "moderate"  
## [111] "little"     "moderate"   "little"     "moderate"   "little"    
## [116] "little"     "little"     "little"     "little"     "little"    
## [121] "little"     "little"     "little"     "little"     "little"    
## [126] "little"     "rich"       "little"     "little"     "little"    
## [131] "little"     "little"     "moderate"   "little"     "little"    
## [136] "little"     "rich"       "little"     "little"     "little"    
## [141] "little"     "little"     "little"     "little"     "little"    
## [146] "little"     "little"     "moderate"   "little"     "little"    
## [151] "moderate"   "little"     "little"     "little"     "little"    
## [156] "little"     "little"     "rich"       "little"     "little"    
## [161] "moderate"   "little"     "little"     "little"     "little"    
## [166] "little"     "little"     "little"     "little"     "little"    
## [171] "little"     "rich"       "little"     "little"     "little"    
## [176] "little"     "moderate"   "little"     "little"     "little"    
## [181] "little"     "little"     "little"     "little"     "quite rich"
## [186] "little"     "little"     "little"     "quite rich" "rich"      
## [191] "little"     "little"     "little"     "little"     "little"    
## [196] "little"     "little"     "little"     "little"     "moderate"  
## [201] "little"     "little"     "little"     "little"     "little"    
## [206] "quite rich" "rich"       "little"     "little"     "moderate"  
## [211] "little"     "little"     "little"     "little"     "little"    
## [216] "quite rich" "little"     "little"     "little"     "little"    
## [221] "little"     "little"     "little"     "little"     "little"    
## [226] "little"     "little"     "little"     "little"     "little"    
## [231] "little"     "rich"       "little"     "little"     "little"    
## [236] "little"     "little"     "little"     "little"     "little"    
## [241] "little"     "moderate"   "little"     "little"     "little"    
## [246] "little"     "little"     "moderate"   "little"     "little"    
## [251] "little"     "rich"       "little"     "little"     "little"    
## [256] "moderate"   "little"     "moderate"   "little"     "little"    
## [261] "moderate"   "moderate"   "little"     "quite rich" "moderate"  
## [266] "little"     "little"     "little"     "little"     "little"    
## [271] "moderate"   "little"     "little"     "little"     "little"    
## [276] "little"     "little"     "little"     "moderate"   "little"    
## [281] "little"     "little"     "little"     "little"     "little"    
## [286] "little"     "little"     "little"     "moderate"   "little"    
## [291] "little"     "moderate"   "little"     "little"     "little"    
## [296] "little"     "moderate"   "little"     "little"     "little"    
## [301] "little"     "little"     "little"     "little"     "little"    
## [306] "little"     "little"     "little"     "little"     "little"    
## [311] "little"     "little"     "little"     "little"     "moderate"  
## [316] "little"     "little"     "little"     "little"     "little"    
## [321] "little"     "quite rich" "moderate"   "moderate"   "moderate"  
## [326] "rich"       "little"     "little"     "moderate"   "little"    
## [331] "little"     "little"     "little"     "moderate"   "little"    
## [336] "little"     "little"     "little"     "little"     "little"    
## [341] "little"     "little"     "little"     "little"     "little"    
## [346] "little"     "little"     "moderate"   "little"     "little"    
## [351] "little"     "little"     "little"     "little"     "little"    
## [356] "rich"       "little"     "little"     "moderate"   "little"    
## [361] "moderate"   "moderate"   "rich"       "little"     "moderate"  
## [366] "moderate"   "little"     "little"     "little"     "moderate"  
## [371] "moderate"   "little"     "moderate"   "little"     "little"    
## [376] "little"     "little"     "little"     "little"     "moderate"  
## [381] "little"     "rich"       "little"     "quite rich" "little"    
## [386] "little"     "rich"       "little"     "little"     "little"    
```

```
## [391] "little"     "moderate"   "moderate"   "moderate"   "little"
## [396] "quite rich" "little"     "little"     "little"     "little"
## [401] "little"     "moderate"   "little"     "little"     "little"
## [406] "little"     "little"     "little"     "little"     "little"
## [411] "quite rich" "little"     "little"     "little"     "little"
## [416] "moderate"   "rich"       "little"     "little"     "little"
## [421] "little"     "little"     "little"     "little"     "little"
## [426] "little"     "little"     "little"     "little"     "little"
## [431] "little"     "little"     "little"     "little"     "little"
## [436] "little"     "little"     "little"     "little"     "little"
## [441] "little"     "little"     "little"     "little"     "little"
## [446] "little"     "little"     "little"     "little"     "little"
## [451] "little"     "little"     "little"     "little"     "little"
## [456] "moderate"   "little"     "rich"       "little"     "little"
## [461] "little"     "little"     "little"     "little"     "little"
## [466] "little"     "little"     "little"     "little"     "little"
## [471] "little"     "little"     "little"     "little"     "moderate"
## [476] "little"     "little"     "little"     "little"     "little"
## [481] "little"     "little"     "little"     "little"     "little"
## [486] "little"     "moderate"   "little"     "little"     "little"
## [491] "little"     "quite rich" "quite rich" "little"     "little"
## [496] "quite rich" "little"     "rich"       "little"     "little"
## [501] "moderate"   "little"     "little"     "quite rich" "little"
## [506] "moderate"   "little"     "little"     "quite rich" "little"
## [511] "moderate"   "little"     "moderate"   "little"     "little"
## [516] "little"     "little"     "little"     "little"     "little"
## [521] "little"     "moderate"
```

```
data1$Saving.accounts = as.numeric(credit.savings.factor)
typeof(data1$Saving.accounts)
```

```
## [1] "double"
```

```
## convert variable names Credit.checkings

d <- sub("little","1",data1$Checking.account)
e <- sub("moderate","2",d)
f <-sub("rich","3",e)
data1$Checking.account <- f
data1$Checking.account <- as.numeric(as.character(data1$Checking.account))
typeof(data1$Checking.account)
```

```
## [1] "double"
```

```
## convert variable names Purpose
table(data1$Purpose)
```

```
##
##            business                 car domestic appliances
##                  53                 173                   6
##           education furniture/equipment            radio/TV
##                  28                 107                 132
##             repairs     vacation/others
##                  14                   9
```

```
purpose.factor = factor(data1$Purpose);
as.character(purpose.factor)
```

```
##   [1] "radio/TV"            "furniture/equipment" "car"
##   [4] "car"                 "car"                 "car"
##   [7] "business"            "radio/TV"            "car"
##  [10] "car"                 "radio/TV"            "car"
##  [13] "radio/TV"            "car"                 "car"
##  [16] "furniture/equipment" "radio/TV"            "radio/TV"
##  [19] "business"            "business"            "furniture/equipment"
##  [22] "car"                 "furniture/equipment" "radio/TV"
##  [25] "radio/TV"            "domestic appliances" "radio/TV"
##  [28] "radio/TV"            "repairs"             "car"
```

```
##  [28] "radio/TV"            "repairs"             "car"
##  [31] "car"                 "car"                 "car"
##  [34] "car"                 "car"                 "furniture/equipment"
##  [37] "business"            "business"            "business"
##  [40] "car"                 "vacation/others"     "business"
##  [43] "car"                 "radio/TV"            "radio/TV"
##  [46] "furniture/equipment" "vacation/others"     "radio/TV"
##  [49] "furniture/equipment" "education"           "car"
##  [52] "repairs"             "car"                 "car"
##  [55] "business"            "business"            "radio/TV"
##  [58] "radio/TV"            "furniture/equipment" "vacation/others"
##  [61] "car"                 "business"            "business"
##  [64] "education"           "car"                 "radio/TV"
##  [67] "furniture/equipment" "car"                 "radio/TV"
##  [70] "car"                 "car"                 "radio/TV"
##  [73] "repairs"             "car"                 "car"
##  [76] "education"           "radio/TV"            "radio/TV"
##  [79] "car"                 "radio/TV"            "radio/TV"
##  [82] "furniture/equipment" "business"            "car"
##  [85] "furniture/equipment" "radio/TV"            "car"
##  [88] "business"            "furniture/equipment" "repairs"
##  [91] "education"           "car"                 "car"
##  [94] "furniture/equipment" "furniture/equipment" "business"
##  [97] "car"                 "furniture/equipment" "radio/TV"
## [100] "education"           "furniture/equipment" "radio/TV"
## [103] "car"                 "business"            "car"
## [106] "car"                 "car"                 "radio/TV"
## [109] "furniture/equipment" "business"            "business"
## [112] "radio/TV"            "education"           "furniture/equipment"
## [115] "furniture/equipment" "car"                 "education"
## [118] "car"                 "domestic appliances" "business"
## [121] "business"            "business"            "business"
## [124] "radio/TV"            "furniture/equipment" "radio/TV"
## [127] "radio/TV"            "furniture/equipment" "furniture/equipment"
## [130] "radio/TV"            "radio/TV"            "radio/TV"
## [133] "business"            "radio/TV"            "car"
## [136] "furniture/equipment" "car"                 "furniture/equipment"
## [139] "car"                 "radio/TV"            "furniture/equipment"
## [142] "radio/TV"            "car"                 "radio/TV"
## [145] "car"                 "radio/TV"            "repairs"
## [148] "car"                 "car"                 "car"
## [151] "vacation/others"     "radio/TV"            "radio/TV"
## [154] "car"                 "car"                 "car"
## [157] "furniture/equipment" "furniture/equipment" "radio/TV"
## [160] "car"                 "radio/TV"            "furniture/equipment"
## [163] "car"                 "furniture/equipment" "car"
## [166] "furniture/equipment" "furniture/equipment" "furniture/equipment"
## [169] "car"                 "radio/TV"            "car"
## [172] "furniture/equipment" "car"                 "radio/TV"
## [175] "radio/TV"            "car"                 "car"
## [178] "furniture/equipment" "furniture/equipment" "radio/TV"
## [181] "furniture/equipment" "furniture/equipment" "education"
## [184] "furniture/equipment" "radio/TV"            "business"
## [187] "car"                 "radio/TV"            "radio/TV"
## [190] "education"           "furniture/equipment" "radio/TV"
## [193] "car"                 "furniture/equipment" "car"
## [196] "furniture/equipment" "furniture/equipment" "furniture/equipment"
## [199] "furniture/equipment" "vacation/others"     "business"
## [202] "car"                 "car"                 "car"
## [205] "furniture/equipment" "business"            "furniture/equipment"
## [208] "car"                 "radio/TV"            "education"
## [211] "furniture/equipment" "furniture/equipment" "car"
## [214] "radio/TV"            "radio/TV"            "car"
## [217] "radio/TV"            "car"                 "car"
## [220] "car"                 "repairs"             "vacation/others"
## [223] "furniture/equipment" "furniture/equipment" "repairs"
## [226] "business"            "furniture/equipment" "vacation/others"
## [229] "furniture/equipment" "car"                 "radio/TV"
## [232] "repairs"             "car"                 "car"
## [235] "car"                 "domestic appliances" "furniture/equipment"
## [238] "car"                 "furniture/equipment" "car"
## [241] "radio/TV"            "radio/TV"            "education"
## [244] "car"                 "radio/TV"            "furniture/equipment"
```

```
## [247] "radio/TV"            "business"            "furniture/equipment"
## [250] "radio/TV"            "car"                 "furniture/equipment"
## [253] "car"                 "business"            "car"
## [256] "furniture/equipment" "radio/TV"            "car"
## [259] "car"                 "car"                 "radio/TV"
## [262] "radio/TV"            "car"                 "car"
## [265] "car"                 "car"                 "business"
## [268] "radio/TV"            "car"                 "car"
## [271] "car"                 "radio/TV"            "furniture/equipment"
## [274] "radio/TV"            "car"                 "radio/TV"
## [277] "car"                 "education"           "car"
## [280] "education"           "furniture/equipment" "car"
## [283] "furniture/equipment" "radio/TV"            "furniture/equipment"
## [286] "car"                 "radio/TV"            "radio/TV"
## [289] "car"                 "education"           "radio/TV"
## [292] "car"                 "furniture/equipment" "furniture/equipment"
## [295] "radio/TV"            "radio/TV"            "radio/TV"
## [298] "car"                 "radio/TV"            "furniture/equipment"
## [301] "business"            "radio/TV"            "radio/TV"
## [304] "car"                 "car"                 "car"
## [307] "repairs"             "furniture/equipment" "car"
## [310] "furniture/equipment" "domestic appliances" "furniture/equipment"
## [313] "radio/TV"            "car"                 "car"
## [316] "car"                 "business"            "radio/TV"
## [319] "furniture/equipment" "education"           "furniture/equipment"
## [322] "furniture/equipment" "radio/TV"            "domestic appliances"
## [325] "car"                 "radio/TV"            "car"
## [328] "car"                 "furniture/equipment" "business"
## [331] "furniture/equipment" "radio/TV"            "furniture/equipment"
## [334] "car"                 "radio/TV"            "furniture/equipment"
## [337] "car"                 "radio/TV"            "education"
## [340] "repairs"             "radio/TV"            "furniture/equipment"
## [343] "car"                 "education"           "education"
## [346] "furniture/equipment" "car"                 "car"
## [349] "car"                 "car"                 "business"
## [352] "furniture/equipment" "radio/TV"            "furniture/equipment"
## [355] "education"           "furniture/equipment" "car"
## [358] "radio/TV"            "radio/TV"            "radio/TV"
## [361] "business"            "car"                 "car"
## [364] "furniture/equipment" "repairs"             "radio/TV"
## [367] "radio/TV"            "education"           "car"
## [370] "radio/TV"            "business"            "business"
## [373] "car"                 "furniture/equipment" "radio/TV"
## [376] "radio/TV"            "education"           "radio/TV"
## [379] "car"                 "car"                 "radio/TV"
## [382] "education"           "car"                 "radio/TV"
## [385] "radio/TV"            "business"            "business"
## [388] "radio/TV"            "radio/TV"            "radio/TV"
## [391] "car"                 "car"                 "business"
## [394] "car"                 "radio/TV"            "furniture/equipment"
## [397] "business"            "car"                 "car"
## [400] "furniture/equipment" "car"                 "business"
## [403] "car"                 "car"                 "furniture/equipment"
## [406] "radio/TV"            "radio/TV"            "furniture/equipment"
## [409] "furniture/equipment" "education"           "car"
## [412] "radio/TV"            "repairs"             "radio/TV"
## [415] "education"           "car"                 "business"
## [418] "education"           "education"           "business"
## [421] "furniture/equipment" "radio/TV"            "furniture/equipment"
## [424] "car"                 "radio/TV"            "car"
## [427] "car"                 "business"            "radio/TV"
## [430] "car"                 "domestic appliances" "car"
## [433] "car"                 "vacation/others"     "radio/TV"
## [436] "radio/TV"            "furniture/equipment" "car"
## [439] "car"                 "car"                 "car"
## [442] "business"            "radio/TV"            "car"
## [445] "car"                 "furniture/equipment" "radio/TV"
## [448] "radio/TV"            "car"                 "car"
## [451] "car"                 "radio/TV"            "radio/TV"
## [454] "furniture/equipment" "car"                 "business"
## [457] "furniture/equipment" "car"                 "radio/TV"
## [460] "car"                 "furniture/equipment" "education"
## [463] "business"            "business"            "car"
```

```
## [466] "car"                 "radio/TV"            "furniture/equipment"
## [469] "car"                 "radio/TV"            "furniture/equipment"
## [472] "business"            "vacation/others"     "car"
## [475] "furniture/equipment" "furniture/equipment" "radio/TV"
## [478] "car"                 "furniture/equipment" "car"
## [481] "furniture/equipment" "car"                 "car"
## [484] "furniture/equipment" "radio/TV"            "radio/TV"
## [487] "radio/TV"            "radio/TV"            "radio/TV"
## [490] "education"           "furniture/equipment" "car"
## [493] "furniture/equipment" "business"            "business"
## [496] "car"                 "car"                 "radio/TV"
## [499] "radio/TV"            "car"                 "furniture/equipment"
## [502] "car"                 "repairs"             "radio/TV"
## [505] "car"                 "repairs"             "car"
## [508] "business"            "radio/TV"            "radio/TV"
## [511] "car"                 "furniture/equipment" "car"
## [514] "car"                 "furniture/equipment" "business"
## [517] "car"                 "radio/TV"            "furniture/equipment"
## [520] "car"                 "radio/TV"            "car"
```

```
data1$Purpose = as.numeric(purpose.factor)
typeof(data1$Purpose)
```

```
## [1] "double"
```

```
## convert the Sex
##data1<-na.omit(data1)
table(data1$Sex)
```

```
##
## female   male
##    168    354
```

```
sex.factor = factor(data1$Sex)
as.character(sex.factor)
```

```
##   [1] "female" "male"   "male"   "male"   "male"   "female" "female"
##   [8] "female" "male"   "female" "female" "female" "male"   "male"
##  [15] "male"   "male"   "female" "male"   "male"   "male"   "male"
##  [22] "male"   "female" "male"   "male"   "male"   "male"   "male"
##  [29] "male"   "male"   "female" "female" "male"   "male"   "female"
##  [36] "female" "male"   "male"   "male"   "male"   "male"   "female"
##  [43] "male"   "male"   "male"   "male"   "female" "male"   "female"
##  [50] "male"   "male"   "male"   "male"   "male"   "male"   "male"
##  [57] "male"   "male"   "male"   "male"   "male"   "male"   "male"
##  [64] "female" "male"   "male"   "female" "male"   "female" "male"
##  [71] "male"   "male"   "male"   "male"   "female" "male"   "male"
##  [78] "female" "male"   "female" "male"   "male"   "male"   "female"
##  [85] "male"   "male"   "female" "male"   "female" "male"   "male"
##  [92] "male"   "male"   "female" "female" "male"   "male"   "female"
##  [99] "male"   "male"   "male"   "male"   "male"   "male"   "male"
## [106] "female" "male"   "male"   "female" "male"   "male"   "male"
## [113] "female" "female" "male"   "male"   "male"   "male"   "male"
## [120] "male"   "male"   "male"   "male"   "male"   "male"   "male"
## [127] "male"   "male"   "male"   "male"   "male"   "male"   "male"
## [134] "male"   "male"   "male"   "female" "female" "female" "male"
## [141] "male"   "female" "male"   "male"   "male"   "male"   "male"
## [148] "male"   "female" "male"   "male"   "female" "male"   "male"
## [155] "female" "male"   "female" "male"   "female" "male"   "male"
## [162] "female" "male"   "female" "male"   "male"   "male"   "female"
## [169] "male"   "male"   "male"   "male"   "male"   "male"   "male"
## [176] "male"   "female" "male"   "male"   "female" "male"   "male"
## [183] "female" "female" "male"   "male"   "male"   "male"   "female"
## [190] "male"   "female" "male"   "male"   "female" "female" "male"
## [197] "female" "male"   "female" "female" "female" "male"   "female"
## [204] "male"   "male"   "male"   "female" "male"   "female" "male"
## [211] "female" "male"   "male"   "male"   "male"   "male"   "female"
## [218] "male"   "male"   "male"   "female" "male"   "male"   "male"
## [225] "male"   "female" "female" "male"   "female" "female" "male"
```

```
## [225] "male"   "female" "female" "male"   "female" "female" "male"
## [232] "male"   "male"   "male"   "male"   "female" "male"   "male"
## [239] "female" "male"   "female" "male"   "female" "male"   "male"
## [246] "female" "male"   "male"   "male"   "female" "male"   "female"
## [253] "male"   "female" "male"   "male"   "male"   "male"   "female"
## [260] "male"   "male"   "male"   "female" "male"   "male"   "male"
## [267] "male"   "male"   "female" "male"   "male"   "female" "male"
## [274] "female" "male"   "male"   "male"   "male"   "female" "male"
## [281] "female" "male"   "female" "male"   "male"   "male"   "female"
## [288] "male"   "male"   "female" "male"   "female" "male"   "male"
## [295] "male"   "female" "female" "male"   "female" "female" "female"
## [302] "male"   "female" "male"   "male"   "male"   "male"   "female"
## [309] "male"   "male"   "male"   "male"   "female" "female" "female"
## [316] "female" "male"   "female" "female" "female" "female" "male"
## [323] "female" "female" "female" "female" "female" "male"   "female"
## [330] "male"   "female" "male"   "male"   "male"   "female" "male"
## [337] "female" "male"   "female" "male"   "male"   "male"   "male"
## [344] "male"   "male"   "male"   "male"   "male"   "male"   "male"
## [351] "female" "male"   "male"   "male"   "female" "male"   "male"
## [358] "male"   "male"   "male"   "male"   "male"   "male"   "male"
## [365] "female" "male"   "male"   "male"   "male"   "female" "male"
## [372] "female" "male"   "female" "female" "male"   "female" "male"
## [379] "male"   "male"   "male"   "female" "female" "female" "female"
## [386] "female" "male"   "male"   "male"   "female" "female" "male"
## [393] "female" "male"   "male"   "male"   "male"   "female" "female"
## [400] "female" "female" "female" "male"   "male"   "female" "male"
## [407] "male"   "male"   "male"   "female" "male"   "female" "female"
## [414] "male"   "male"   "female" "male"   "male"   "male"   "female"
## [421] "male"   "female" "female" "male"   "male"   "male"   "female"
## [428] "male"   "male"   "male"   "male"   "male"   "female" "male"
## [435] "male"   "male"   "male"   "male"   "male"   "female" "female"
## [442] "male"   "female" "male"   "male"   "male"   "male"   "male"
## [449] "male"   "male"   "female" "female" "female" "female" "male"
## [456] "male"   "male"   "female" "male"   "male"   "male"   "female"
## [463] "male"   "male"   "male"   "male"   "female" "male"   "male"
## [470] "male"   "female" "male"   "female" "male"   "male"   "male"
## [477] "female" "male"   "male"   "male"   "female" "male"   "male"
## [484] "male"   "female" "female" "male"   "female" "male"   "male"
## [491] "female" "female" "male"   "male"   "male"   "female" "female"
## [498] "female" "male"   "male"   "male"   "male"   "male"   "male"
## [505] "male"   "male"   "female" "male"   "female" "female" "male"
## [512] "male"   "female" "male"   "female" "male"   "male"   "male"
## [519] "male"   "male"   "male"   "male"
```

```
data1$Sex = as.numeric(sex.factor)
typeof(data1$Sex)
```

```
## [1] "double"
```

```
## (1) performing perform tree model
attach(data1)
```

```
## The following objects are masked from data1 (pos = 6):
##
##     Age, Checking.account, Credit.amount, Duration, Housing, Job,
##     Purpose, Saving.accounts, Sex, X
```

```
install.packages("tree",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'tree' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'tree'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\jolly\OneDrive\Documents\R\win-
## library\3.6\00LOCK\tree\libs\x64\tree.dll to C:
## \Users\jolly\OneDrive\Documents\R\win-library\3.6\tree\libs\x64\tree.dll:
## Permission denied
```

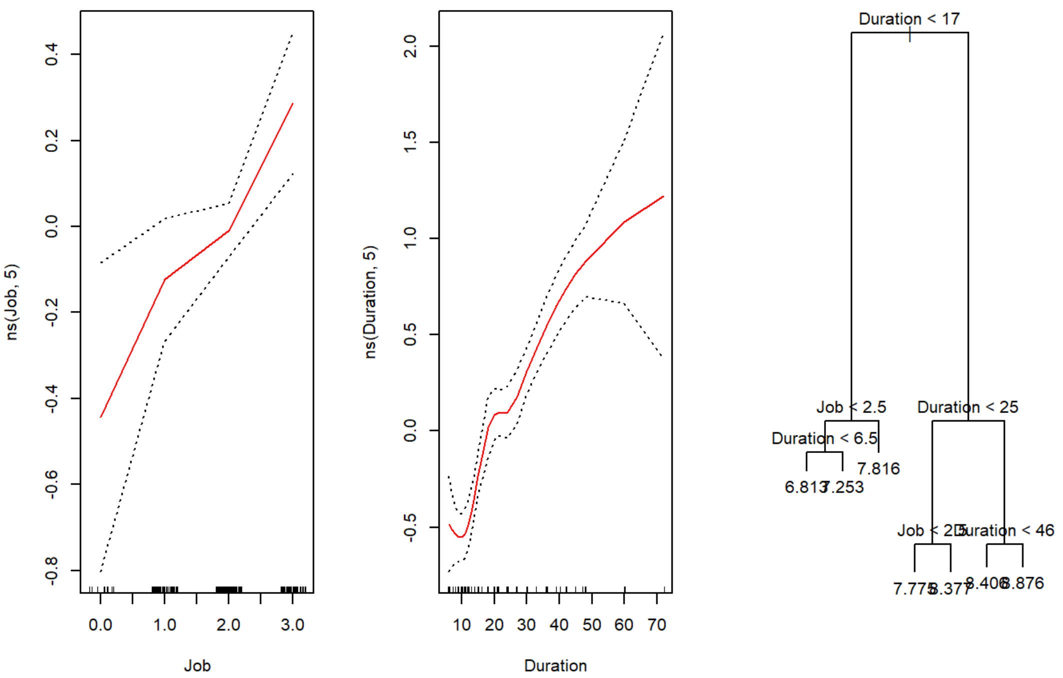
```
## Warning: restored 'tree'
```

```
##
## The downloaded binary packages are in
##  C:\Users\jolly\AppData\Local\Temp\RtmpkX3MWk\downloaded_packages
```

```
library(tree)
tree.data = tree(Credit.amount~.-X,data1)
summary(tree.data)
```

```
##
## Regression tree:
## tree(formula = Credit.amount ~ . - X, data = data1)
## Variables actually used in tree construction:
## [1] "Duration" "Job"
## Number of terminal nodes:  7
## Residual mean deviance:  0.3311 = 170.5 / 515
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.6330 -0.4328 -0.0232  0.0000  0.4065  1.7930
```

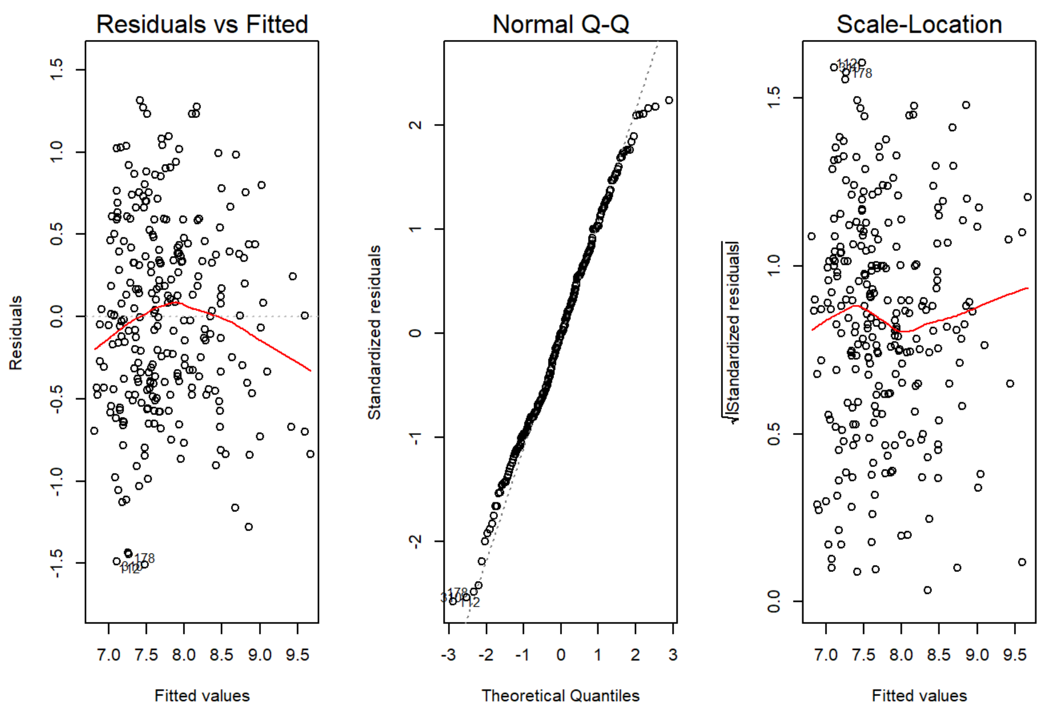
```
## as I could see through the summary that the variables actually used in tree is Duration and Job
## plot the decision tree
plot(tree.data )
text(tree.data ,pretty =0)
```



```r
## obtain the MSE: 0.3218

sample_size = floor(0.5*nrow(data1))
picked = sample(seq_len(nrow(data1)),size = sample_size)

## train data set
train1 =data1[picked,]

## test data set
test1 = data1[-picked,]

pred.tree = predict(tree.data, newdata = test1)
MSE = mean((pred.tree - test1$Credit.amount)^2)
cat("the MSE of decision tree is: " , MSE)
```

```
## the MSE of decision tree is:  0.2906518
```

```r
## (2) performing linear regression : important features are Job,Duration and Purpose, the MSE: 0.3636
fit.lm2 = lm(Credit.amount~.-X, data = train1)
plot(fit.lm2)
```



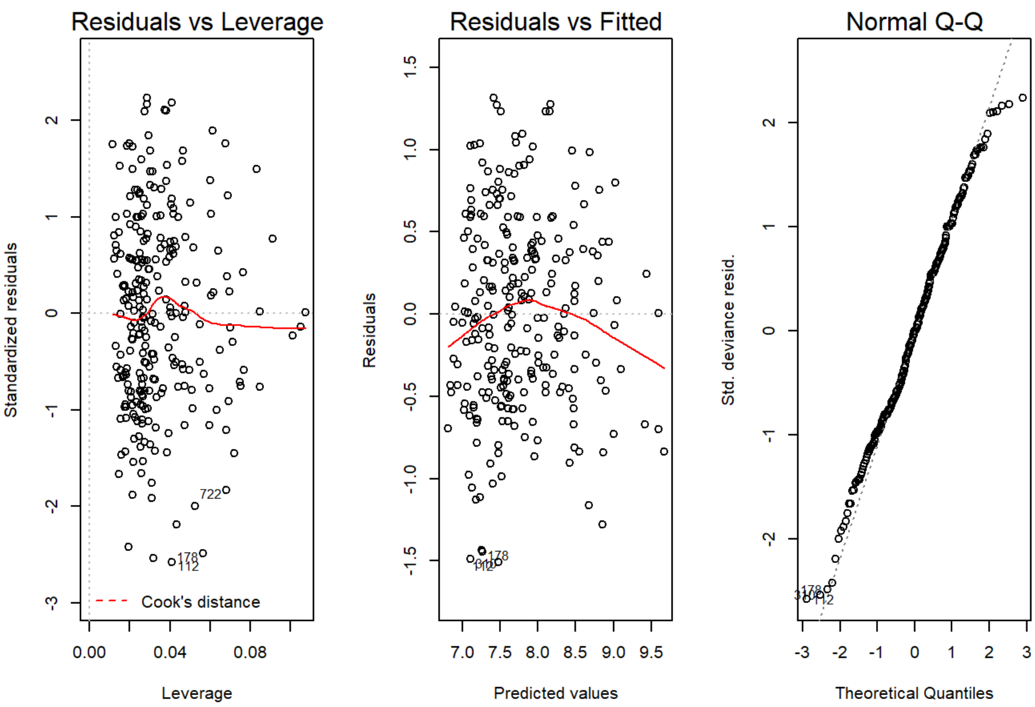

```r
summary(fit.lm2)
```

```
##
## Call:
## lm(formula = Credit.amount ~ . - X, data = train1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50736 -0.43562 -0.00468  0.41715  1.31598
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.091439   0.316552  19.243  < 2e-16 ***
## Age                0.004838   0.003389   1.427    0.155
## Sex                0.089069   0.080141   1.111    0.267
## Job                0.267541   0.054480   4.911 1.63e-06 ***
## Housing            0.059152   0.067220   0.880    0.380
## Saving.accounts   -0.001280   0.050634  -0.025    0.980
## Checking.account  -0.018057   0.057818  -0.312    0.755
## Duration           0.041333   0.003222  12.829  < 2e-16 ***
## Purpose           -0.022556   0.018935  -1.191    0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5973 on 252 degrees of freedom
## Multiple R-squared:  0.4961, Adjusted R-squared:  0.4801
## F-statistic: 31.01 on 8 and 252 DF,  p-value: < 2.2e-16
```

```
## the MSE of linear regression is 0.3636
fit.pred2 = predict(fit.lm2, test1)
lm.fit.err2 = mean((test1$Credit.amount - fit.pred2)^2)
lm.fit.err2
```

```
## [1] 0.3269206
```

```
## (3)performing logistic model: important features are Job,Duration and Purpos
## the MSE: 0.3636
fit.glm2 = glm(Credit.amount~.-X, data = train1)
plot(fit.glm2)
```




```
summary(fit.glm2)
```

```
##
## Call:
## glm(formula = Credit.amount ~ . - X, data = train1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.50736  -0.43562  -0.00468   0.41715   1.31598
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.091439   0.316552  19.243  < 2e-16 ***
## Age               0.004838   0.003389   1.427    0.155
## Sex               0.089069   0.080141   1.111    0.267
## Job               0.267541   0.054480   4.911 1.63e-06 ***
## Housing           0.059152   0.067220   0.880    0.380
## Saving.accounts  -0.001280   0.050634  -0.025    0.980
## Checking.account -0.018057   0.057818  -0.312    0.755
## Duration          0.041333   0.003222  12.829  < 2e-16 ***
## Purpose          -0.022556   0.018935  -1.191    0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3567679)
##
##     Null deviance: 178.425  on 260  degrees of freedom
## Residual deviance:  89.906  on 252  degrees of freedom
## AIC: 482.52
##
## Number of Fisher Scoring iterations: 2
```

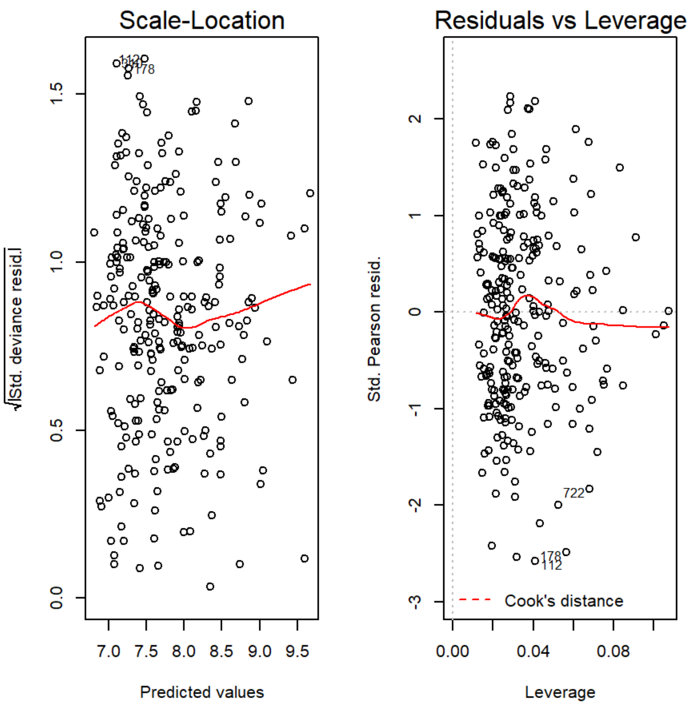
```
fit.pred3 = predict(fit.glm2, test1)
glm.fit.err3 = mean((test1$Credit.amount - fit.pred3)^2)
glm.fit.err3
```

```
## [1] 0.3269206
```

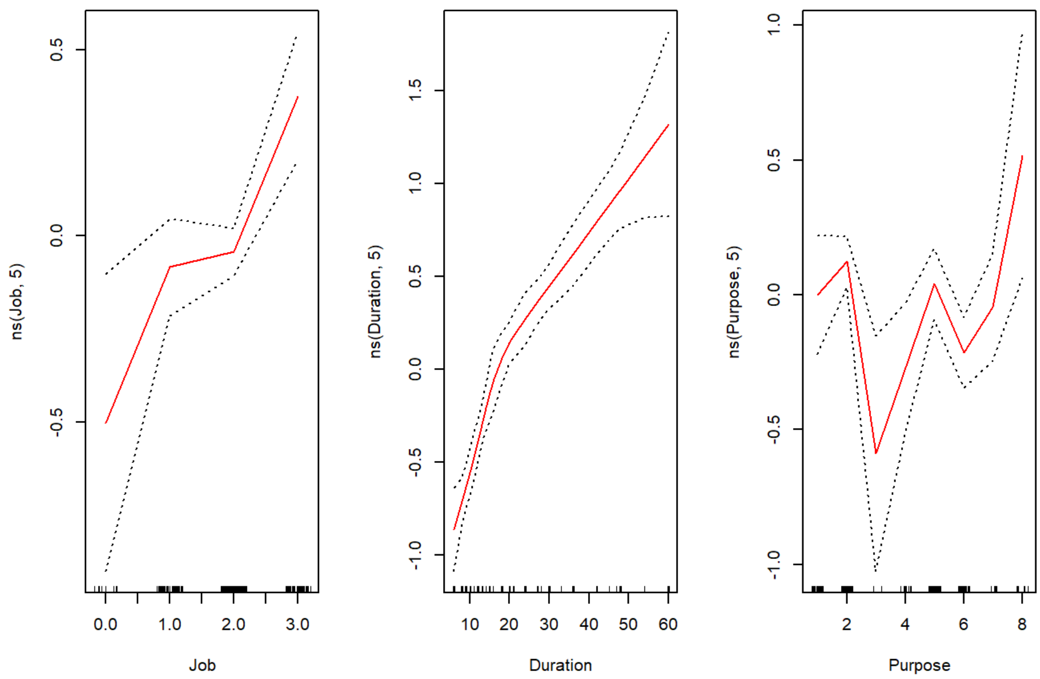
```
## conslusion: the MSE of linear regression and logistic regression is same, also the important features are same.

## (4)performing a gam model by using features select before
gam.fit2 = gam(Credit.amount~ ns(Job,5)+ns(Duration, 5)+ns(Purpose,5),data = train1)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
## plotting the model
par(mfrow=c(1,3)) #to partition the Plotting Window
```




```
plot(gam.fit2,se = TRUE,col = "red" )
```




```
## gam model test error rate: 0.4347, comparing to the former data set, the GAMS model performs worse on dum
my data
gam.pred1 = predict(gam.fit2, test1)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
gam.err1 = mean((test1$Credit.amount - gam.pred1)^2)

gam.tss1 = mean((test1$Credit.amount - mean(test1$Credit.amount))^2)
test.rss1 = 1 - gam.err1/gam.tss1
test.rss1
```

```
## [1] 0.4600827
```

```
cat("the results produced a R squre value of", test.rss1)
```

```
## the results produced a R squre value of 0.4600827
```

```
## after modifying the data set, the performance of linear regression and logistic regression model doesn't improved significantly, gam model perform a little better on dummy data set, improved from 0.46 to 0.43, but it may also be worse after running for couple of times.so far, the decision tree perform best, accuracy is 68%, let's keep moving with more models.

## (5)perform random forest

install.packages("randomForest",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'randomForest' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'randomForest'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\jolly\OneDrive\Documents\R\win-
## library\3.6\00LOCK\randomForest\libs\x64\randomForest.dll
## to C:\Users\jolly\OneDrive\Documents\R\win-
## library\3.6\randomForest\libs\x64\randomForest.dll: Permission denied
```

```
## Warning: restored 'randomForest'
```

```
##
## The downloaded binary packages are in
##  C:\Users\jolly\AppData\Local\Temp\RtmpkX3MWk\downloaded_packages
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
## build 500 of trees, the mean of squared residual is  0.33 and 45.13% var explained
bag.credit <- randomForest(Credit.amount ~ ., data = train1,mtry = 9,importance = TRUE)
bag.credit
```

```
##
## Call:
##  randomForest(formula = Credit.amount ~ ., data = train1, mtry = 9,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##           Mean of squared residuals: 0.3692696
##                     % Var explained: 45.98
```

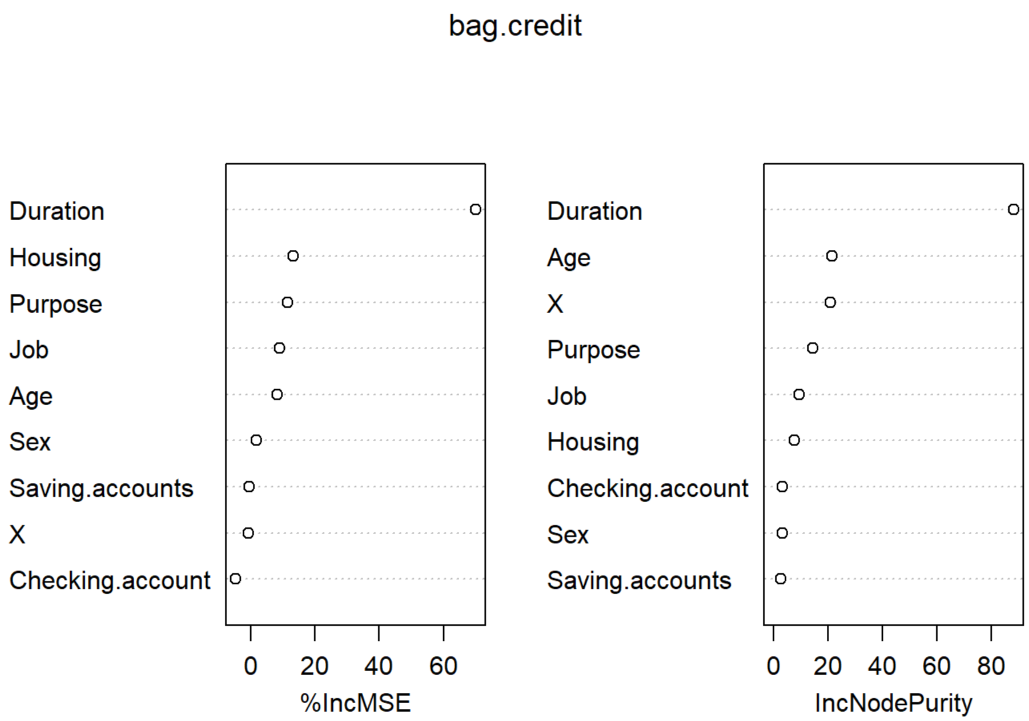
```
## the MSE of randomforest, is 0.4034, worse than decison tree,linear regression and regression regression, but better than GAMs
predict.bag <- predict(bag.credit, newdata = test1)
mean((predict.bag - test1$Credit.amount)^2)
```

```
## [1] 0.3878451
```

```
## let's see the importance of each feature: Duration is the most important one, then is the Job and purpose
, consistent with all the other models so far.
importance(bag.credit)
```

```
##                      %IncMSE IncNodePurity
## X                 -0.8726552     20.788092
## Age                8.1439538     21.454951
## Sex                1.7524814      3.078432
## Job                8.7884900      9.256487
## Housing           13.1354195      7.630938
## Saving.accounts   -0.5514635      2.541971
## Checking.account  -4.8098697      3.130570
## Duration          70.1389933     88.143189
## Purpose           11.3841345     14.230444
```

```
varImpPlot(bag.credit)
```




```
## (6)perform hierarchical clustering: it doesn't need a response as required, but I am want to try this mod
el.

hc.complete <- hclust(dist(data1), method = "complete")

## now plot the dendrograms obtained using the usual plot() function, The numbers at the bottom of the plot
identify each observation.
plot(hc.complete)

##determine the cluster labels for each observation associated with a given cut of the dendrogram by cutree(
) function:
cutree(hc.complete, 3)
```

```
##    2    4    5    8   10   11   12   13   14   15   16   19   22   23   24 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##   26   28   29   30   31   32   33   35   36   38   39   40   42   43   44 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##   45   48   52   55   59   60   61   63   64   68   73   74   76   77   78 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##   80   84   85   87   88   89   90   92   95   96   98   99  102  104  106 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  108  110  111  112  113  115  119  120  121  124  126  127  128  129  130 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  132  138  140  141  142  143  144  146  147  149  153  154  155  156  157 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  158  159  164  167  168  170  171  173  174  175  177  178  180  182  185 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  187  188  189  190  192  193  195  196  198  200  202  204  206  208  209 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  213  214  217  218  219  221  227  228  230  231  234  236  238  240  243 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  249  251  252  253  258  261  262  263  266  269  274  275  285  286  287 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  288  289  290  292  293  294  296  300  302  304  308  309  310  313  314 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  316  317  320  321  322  323  324  326  329  330  331  333  335  336  337 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  339  340  341  342  343  344  345  347  348  350  352  354  356  360  363 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  365  368  369  370  375  376  379  382  384  388  389  392  393  394  396 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1 
##  397  398  399  406  408  410  411  417  423  426  430  432  433  435  439 
##    1    1    1    2    2    2    2    2    2    2    2    2    2    2    2 
##  440  442  443  445  447  448  450  455  457  458  459  461  462  463  466 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  467  471  472  473  475  476  478  479  480  481  482  483  486  492  495 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  497  499  500  501  502  503  504  505  507  508  511  513  514  516  517 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  519  522  523  525  526  529  530  531  532  536  538  539  540  541  544 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  546  549  553  554  555  556  557  559  560  562  563  566  567  570  571 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  574  575  577  579  581  582  584  586  587  588  589  590  591  594  596 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  597  598  601  602  603  605  606  608  611  612  613  614  618  619  621 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  624  625  627  628  631  632  635  640  641  642  645  647  649  650  651 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  652  653  654  656  657  659  660  661  664  665  667  669  670  678  679 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  685  688  690  691  692  693  697  700  702  703  704  705  707  708  709 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  710  712  714  715  720  721  722  723  724  728  729  730  731  732  733 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  737  738  740  741  742  744  746  747  748  751  752  753  757  760  762 
##    2    2    2    2    2    2    2    2    2    2    2    2    3    3    3 
##  763  766  767  769  772  775  778  780  781  783  784  786  789  790  791 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  794  802  803  806  807  809  810  811  812  813  814  815  816  819  820 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  822  823  824  826  827  832  833  835  836  839  841  849  850  851  854 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  859  863  867  870  872  873  875  876  877  879  885  886  888  891  893 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  894  897  900  901  906  912  915  916  918  919  920  923  924  925  926 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  927  928  930  931  932  935  936  937  938  939  945  946  947  951  952 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  953  955  956  958  959  960  962  965  967  970  971  973  974  976  977 
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3 
##  980  981  983  984  986  987  989  990  994  997  999 1000 
##    3    3    3    3    3    3    3    3    3    3    3    3 
```

```r
sd.data <- scale(data1)
hc.complete.sd <- hclust(dist(sd.data), method = "complete")

plot(hc.complete.sd)
cutree(hc.complete.sd, 3)
```

```
##    2    4    5    8   10   11   12   13   14   15   16   19   22   23   24
##    1    1    2    1    2    2    1    2    2    2    2    2    3    2    2
##   26   28   29   30   31   32   33   35   36   38   39   40   42   43   44
##    2    3    2    1    3    2    2    2    1    2    2    2    3    1    2
##   45   48   52   55   59   60   61   63   64   68   73   74   76   77   78
##    1    3    2    2    2    1    2    2    1    3    2    1    2    1    1
##   80   84   85   87   88   89   90   92   95   96   98   99  102  104  106
##    1    2    2    2    1    2    2    2    3    1    2    1    1    2    1
##  108  110  111  112  113  115  119  120  121  124  126  127  128  129  130
##    2    3    2    2    2    3    3    3    2    2    2    2    2    2    2
##  132  138  140  141  142  143  144  146  147  149  153  154  155  156  157
##    1    3    2    3    1    2    2    1    2    1    1    3    1    2    2
##  158  159  164  167  168  170  171  173  174  175  177  178  180  182  185
##    2    2    2    2    3    2    2    2    2    2    2    3    2    2    2
##  187  188  189  190  192  193  195  196  198  200  202  204  206  208  209
##    2    2    2    2    1    2    1    2    2    1    2    2    2    2    2
##  213  214  217  218  219  221  227  228  230  231  234  236  238  240  243
##    2    2    2    1    1    2    1    2    2    1    2    2    2    2    1
##  249  251  252  253  258  261  262  263  266  269  274  275  285  286  287
##    2    3    2    2    2    2    2    2    2    2    1    1    2    1    1
##  288  289  290  292  293  294  296  300  302  304  308  309  310  313  314
##    1    2    2    1    2    2    1    3    1    2    2    2    2    2    2
##  316  317  320  321  322  323  324  326  329  330  331  333  335  336  337
##    2    2    2    2    2    2    3    2    1    2    2    1    2    2    2
##  339  340  341  342  343  344  345  347  348  350  352  354  356  360  363
##    2    2    2    2    3    2    2    2    3    3    2    2    2    2    2
##  365  368  369  370  375  376  379  382  384  388  389  392  393  394  396
##    2    2    1    2    1    1    1    2    2    1    3    3    2    2    1
##  397  398  399  406  408  410  411  417  423  426  430  432  433  435  439
##    2    1    2    2    2    3    2    2    2    2    2    1    2    2    2
##  440  442  443  445  447  448  450  455  457  458  459  461  462  463  466
##    2    2    2    1    2    2    3    2    2    2    2    1    2    2    2
##  467  471  472  473  475  476  478  479  480  481  482  483  486  492  495
##    2    3    2    2    2    2    1    2    2    2    2    3    2    2    2
##  497  499  500  501  502  503  504  505  507  508  511  513  514  516  517
##    1    2    2    2    1    2    2    2    3    1    2    2    2    2    2
##  519  522  523  525  526  529  530  531  532  536  538  539  540  541  544
##    2    2    1    2    1    1    2    2    2    2    2    1    2    2    2
##  546  549  553  554  555  556  557  559  560  562  563  566  567  570  571
##    2    2    1    2    2    2    2    2    2    1    2    2    2    1    2
##  574  575  577  579  581  582  584  586  587  588  589  590  591  594  596
##    2    2    2    2    2    2    1    2    2    2    2    2    2    2    2
##  597  598  601  602  603  605  606  608  611  612  613  614  618  619  621
##    2    2    2    2    2    2    2    2    2    2    3    2    2    2    1
##  624  625  627  628  631  632  635  640  641  642  645  647  649  650  651
##    2    2    2    2    2    2    2    1    2    2    2    2    2    2    1
##  652  653  654  656  657  659  660  661  664  665  667  669  670  678  679
##    2    2    1    2    2    2    1    2    2    2    3    2    2    1    2
##  685  688  690  691  692  693  697  700  702  703  704  705  707  708  709
##    1    1    3    2    2    2    2    2    1    2    1    2    1    2    2
##  710  712  714  715  720  721  722  723  724  728  729  730  731  732  733
##    2    2    2    1    1    2    3    2    2    2    1    3    1    1    2
##  737  738  740  741  742  744  746  747  748  751  752  753  757  760  762
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  763  766  767  769  772  775  778  780  781  783  784  786  789  790  791
##    2    2    2    2    1    2    2    2    1    2    2    3    1    1    2
##  794  802  803  806  807  809  810  811  812  813  814  815  816  819  820
##    2    2    2    1    2    1    2    2    2    1    1    1    1    1    2
##  822  823  824  826  827  832  833  835  836  839  841  849  850  851  854
##    2    1    2    2    2    2    1    2    2    2    1    2    2    2    2
##  859  863  867  870  872  873  875  876  877  879  885  886  888  891  893
##    2    2    2    2    2    2    2    3    2    2    1    2    1    1    2
##  894  897  900  901  906  912  915  916  918  919  920  923  924  925  926
##    1    2    1    2    2    2    2    1    2    2    1    2    2    1    2
##  927  928  930  931  932  935  936  937  938  939  945  946  947  951  952
##    2    1    2    2    2    2    2    2    2    1    2    1    2    2    2
##  953  955  956  958  959  960  962  965  967  970  971  973  974  976  977
##    2    2    3    2    2    2    1    2    2    2    2    2    1    2    2
##  980  981  983  984  986  987  989  990  994  997  999 1000
##    2    1    2    1    2    1    2    2    1    1    2    1
```

```
table(cutree(hc.complete, 3), cutree(hc.complete.sd, 3))
```

```
##
##       1   2   3
##   1  50 139  24
##   2  31 148  10
##   3  33  84   3
```

```
table(cutree(hc.complete, 2), cutree(hc.complete.sd, 2))
```

```
##
##       1   2
##   1  50 163
##   2  64 245
```

```
##(7) performing subset selection model
install.packages("leaps",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'leaps' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\jolly\AppData\Local\Temp\RtmpkX3MWk\downloaded_packages
```

```
library(leaps)
## output indicates that the best two-variable model contains only Duration and job
regfit.full=regsubsets (Credit.amount~.,data1 )
summary (regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(Credit.amount ~ ., data1)
## 9 Variables  (and intercept)
##                  Forced in Forced out
## X                    FALSE      FALSE
## Age                  FALSE      FALSE
## Sex                  FALSE      FALSE
## Job                  FALSE      FALSE
## Housing              FALSE      FALSE
## Saving.accounts      FALSE      FALSE
## Checking.account     FALSE      FALSE
## Duration             FALSE      FALSE
## Purpose              FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          X   Age Sex Job Housing Saving.accounts Checking.account Duration
## 1  ( 1 ) " " " " " " " " " "     " "             " "              "*"
## 2  ( 1 ) " " " " " " "*" " "     " "             " "              "*"
## 3  ( 1 ) " " " " " " "*" " "     " "             " "              "*"
## 4  ( 1 ) " " " " " " "*" " "     "*"             " "              "*"
## 5  ( 1 ) " " " " "*" "*" " "     "*"             " "              "*"
## 6  ( 1 ) " " "*" "*" "*" " "     "*"             " "              "*"
## 7  ( 1 ) " " "*" "*" "*" "*"     "*"             " "              "*"
## 8  ( 1 ) "*" "*" "*" "*" "*"     "*"             " "              "*"
##          Purpose
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) "*"
## 4  ( 1 ) "*"
## 5  ( 1 ) "*"
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
## 8  ( 1 ) "*"
```

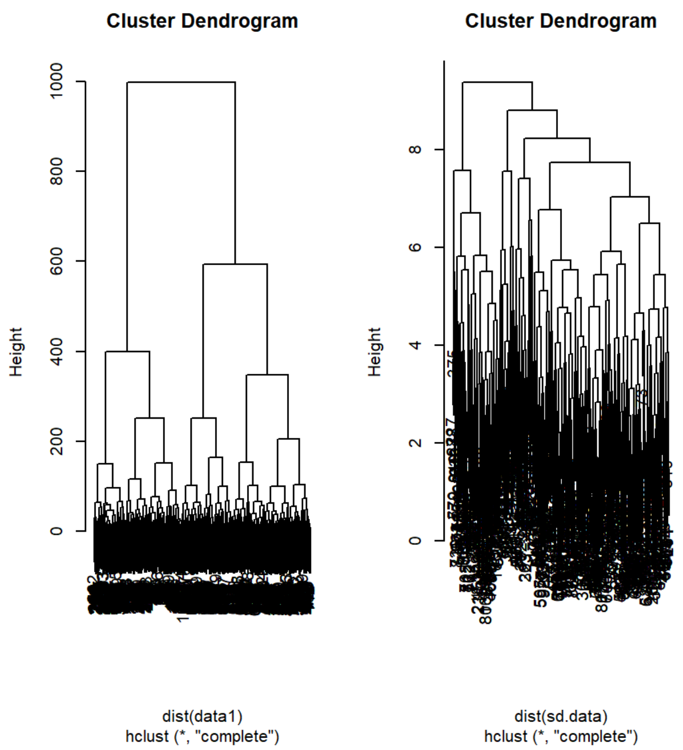
```
## fit up to a 9-variable model.
regfit.full2=regsubsets (Credit.amount~.,data1,nvmax = 19)
reg.summary = summary (regfit.full2)

names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

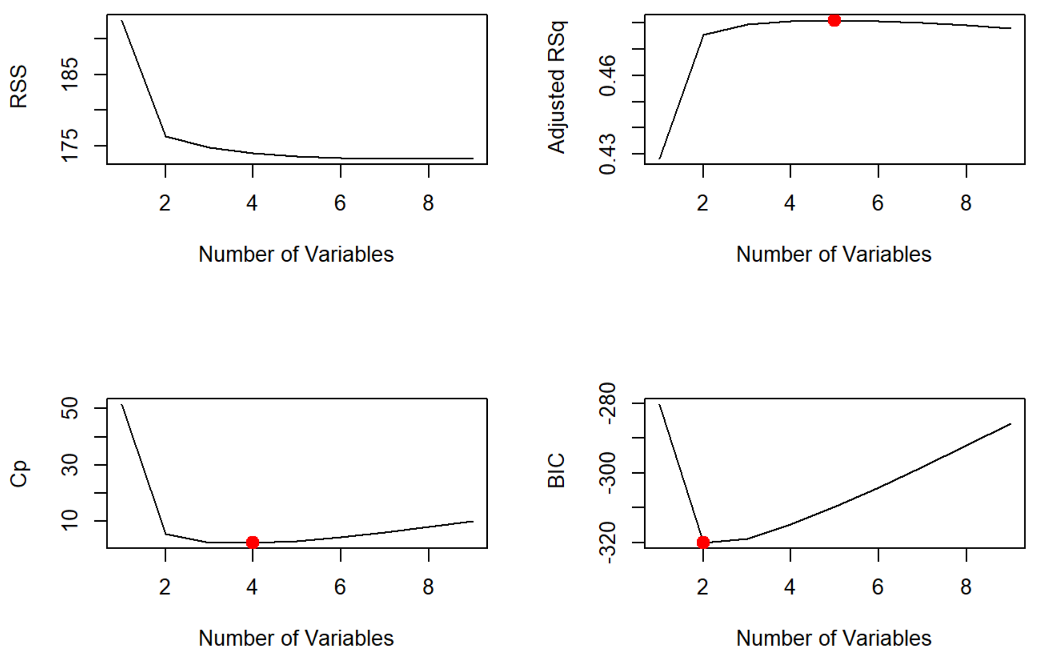
```
## we see that the R2 statistic increases from 42%, when only one variable is included in the model, to almost 48 %, when all variables are included. As expected, the R2 statistic increases monotonically as more variables are included.
reg.summary$rsq
```

```
## [1] 0.4293735 0.4774788 0.4824575 0.4845673 0.4858597 0.4865326 0.4869207
## [8] 0.4869645 0.4869654
```

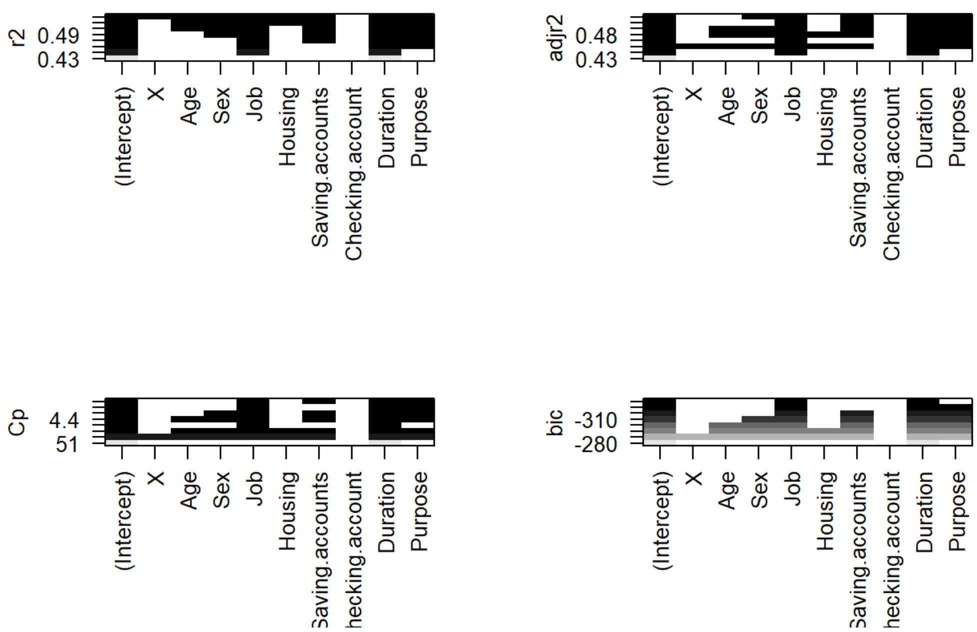
```
## Plotting RSS, adjusted R2, Cp, and BIC for all of the models at once will help me decide which model to select
par(mfrow =c(2,2))
```




```
plot(reg.summary$rss ,xlab=" Number of Variables ",ylab=" RSS",
type="l")
plot(reg.summary$adjr2 ,xlab =" Number of Variables ",
ylab=" Adjusted RSq",type="l")

max1 = which.max (reg.summary$adjr2)

## We will now plot a red dot to indicate the model with the largest adjusted R2 statistic.
points (max1, reg.summary$adjr2[max1], col ="red",cex =2, pch =20)

## In a similar fashion we can plot the Cp and BIC statistics, and indicate the models with the smallest statistic using which.min().
plot(reg.summary$cp ,xlab =" Number of Variables ",ylab="Cp",
type='l')
min1 = which.min (reg.summary$cp )
points (min1, reg.summary$cp [min1], col ="red",cex =2, pch =20)

min2 = which.min (reg.summary$bic )
plot(reg.summary$bic ,xlab=" Number of Variables ",ylab=" BIC",
type='l')
points (min2, reg.summary$bic [min2], col =" red",cex =2, pch =20)
```




```
plot(regfit.full,scale ="r2")
plot(regfit.full,scale ="adjr2")
plot(regfit.full,scale ="Cp")
plot(regfit.full,scale ="bic")
```




```
## the model with the lowest BIC is the two-variable model that contains only Duration and Job
## coefficient estimates associated with this model.
coef(regfit.full,2)
```

```
## (Intercept)         Job    Duration
##   6.43581260  0.26393728  0.03938316
```

Question 3:

Do the same approach as in question 2, but this time for a qualitative variable.

```r
## For qualitative variable, I would like to create a qualitative variable based on Credit.amount named High
("yes","no") which I think is proficient to predict the Credit risk

## Performing tree model, hierarchical clustering, LDA,QDA and Resampling method(K-Fold across-validation)

## (1) perform decision tree
High=ifelse (Credit.amount <=7," No"," Yes ")
data1 = data.frame(data1,High)
attach(data1)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     High
```

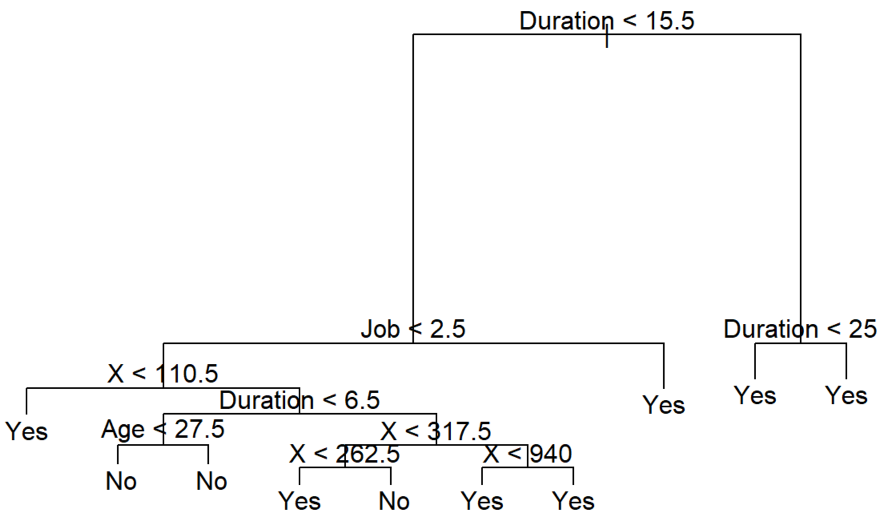
```
## The following objects are masked from data1 (pos = 6):
##
##     Age, Checking.account, Credit.amount, Duration, Housing, Job,
##     Purpose, Saving.accounts, Sex, X
```

```
## The following objects are masked from data1 (pos = 10):
##
##     Age, Checking.account, Credit.amount, Duration, Housing, Job,
##     Purpose, Saving.accounts, Sex, X
```

```r
tree.credit =tree(High~.-Credit.amount,data1)
summary (tree.credit)
```

```
##
## Classification tree:
## tree(formula = High ~ . - Credit.amount, data = data1)
## Variables actually used in tree construction:
## [1] "Duration" "Job"      "X"        "Age"
## Number of terminal nodes:  10
## Residual mean deviance:  0.586 = 300 / 512
## Misclassification error rate: 0.1303 = 68 / 522
```

```r
## through the result of summary, I learnt that the training error rate is 13% and the variables actually us
ed in tree construction is Duration, Job and Age

## plot the tree
plot(tree.credit)
text(tree.credit ,pretty =0)
```



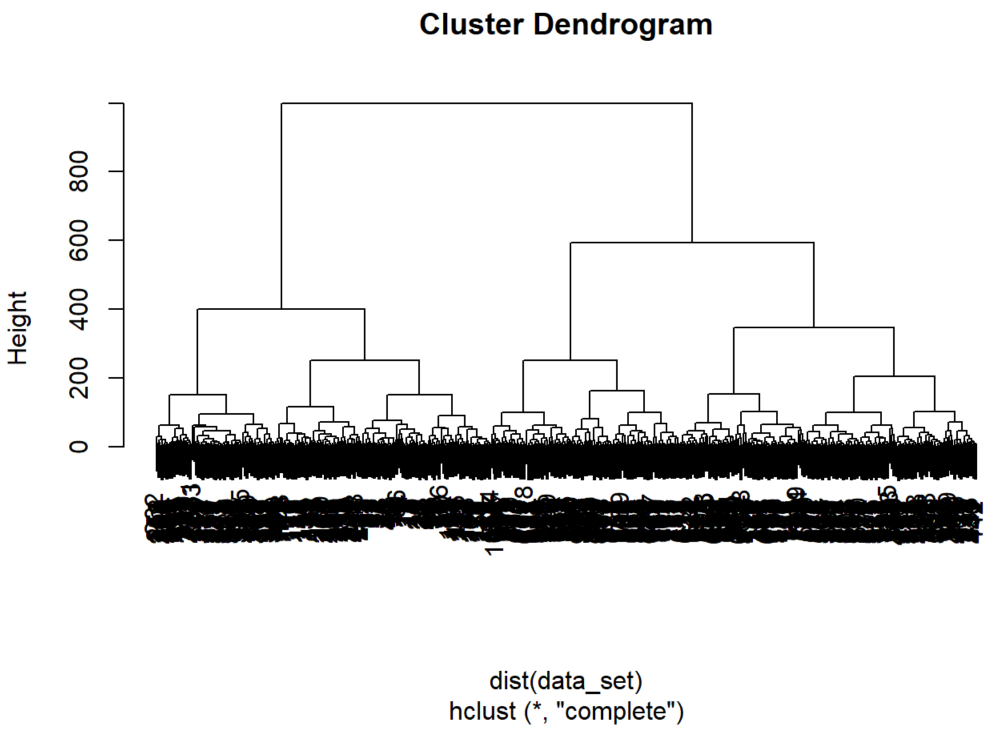

```
tree.credit
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 522 457.300  Yes  ( 0.15900 0.84100 )
##    2) Duration < 15.5 223 283.400  Yes  ( 0.33184 0.66816 )
##      4) Job < 2.5 199 261.600  Yes  ( 0.36683 0.63317 )
##        8) X < 110.5 24  18.080  Yes  ( 0.12500 0.87500 ) *
##        9) X > 110.5 175 235.600  Yes  ( 0.40000 0.60000 )
##         18) Duration < 6.5 27  34.370  No ( 0.66667 0.33333 )
##           36) Age < 27.5 6   0.000  No ( 1.00000 0.00000 ) *
##           37) Age > 27.5 21  28.680  No ( 0.57143 0.42857 ) *
##         19) Duration > 6.5 148 191.900  Yes  ( 0.35135 0.64865 )
##           38) X < 317.5 33  45.470  No ( 0.54545 0.45455 )
##             76) X < 262.5 25  34.300  Yes  ( 0.44000 0.56000 ) *
##             77) X > 262.5 8   6.028  No ( 0.87500 0.12500 ) *
##           39) X > 317.5 115 139.600  Yes  ( 0.29565 0.70435 )
##             78) X < 940 108 134.500  Yes  ( 0.31481 0.68519 ) *
##             79) X > 940 7   0.000  Yes  ( 0.00000 1.00000 ) *
##      5) Job > 2.5 24   8.314  Yes  ( 0.04167 0.95833 ) *
##    3) Duration > 15.5 299  80.780  Yes  ( 0.03010 0.96990 )
##      6) Duration < 25 167  70.080  Yes  ( 0.05389 0.94611 ) *
##      7) Duration > 25 132   0.000  Yes  ( 0.00000 1.00000 ) *
```

```
## compute the test error rate
train3=sample (1: nrow(data1), 200)
data1.test=data1[-train3,]
High.test=High[-train3 ]

tree.credit =tree(High~.-Credit.amount,data1 ,subset =train3 )
tree.credit.pred=predict (tree.credit ,data1.test,type ="class")
table.credit = table(tree.credit.pred,High.test)

## according to the table, This approach leads to incorrect predictions below, is 18.3%
1-sum(diag(table.credit))/sum(table.credit)
```

```
## [1] 0.1863354
```

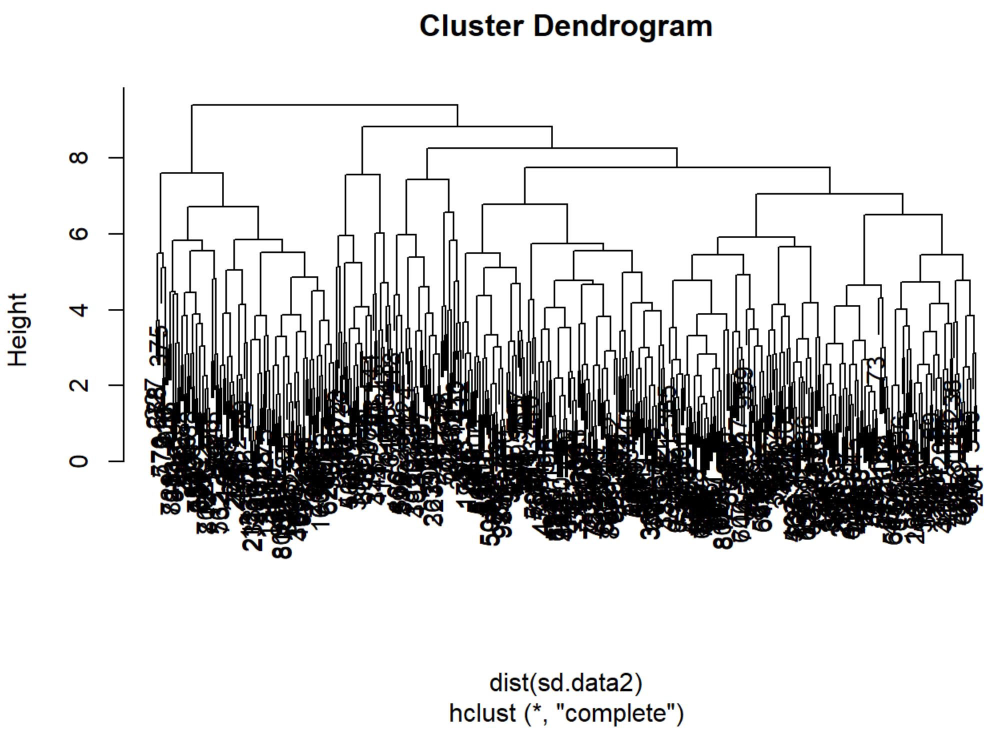
```
## (2)implementing hierarchical clustering
data_set <- subset(data1, select = -c(High))
hc.complete2 <- hclust(dist(data_set), method = "complete")

## now plot the dendrograms obtained using the usual plot() function, The numbers at the bottom of the plot identify each observation.
plot(hc.complete2)
```




```
##determine the cluster labels for each observation associated with a given cut of the dendrogram by cutree() function:
cutree(hc.complete2, 3)
```

```
##    2    4    5    8   10   11   12   13   14   15   16   19   22   23   24
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##   26   28   29   30   31   32   33   35   36   38   39   40   42   43   44
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##   45   48   52   55   59   60   61   63   64   68   73   74   76   77   78
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##   80   84   85   87   88   89   90   92   95   96   98   99  102  104  106
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  108  110  111  112  113  115  119  120  121  124  126  127  128  129  130
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  132  138  140  141  142  143  144  146  147  149  153  154  155  156  157
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  158  159  164  167  168  170  171  173  174  175  177  178  180  182  185
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  187  188  189  190  192  193  195  196  198  200  202  204  206  208  209
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  213  214  217  218  219  221  227  228  230  231  234  236  238  240  243
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  249  251  252  253  258  261  262  263  266  269  274  275  285  286  287
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  288  289  290  292  293  294  296  300  302  304  308  309  310  313  314
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  316  317  320  321  322  323  324  326  329  330  331  333  335  336  337
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  339  340  341  342  343  344  345  347  348  350  352  354  356  360  363
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  365  368  369  370  375  376  379  382  384  388  389  392  393  394  396
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  397  398  399  406  408  410  411  417  423  426  430  432  433  435  439
##    1    1    1    2    2    2    2    2    2    2    2    2    2    2    2
##  440  442  443  445  447  448  450  455  457  458  459  461  462  463  466
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  467  471  472  473  475  476  478  479  480  481  482  483  486  492  495
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  497  499  500  501  502  503  504  505  507  508  511  513  514  516  517
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  519  522  523  525  526  529  530  531  532  536  538  539  540  541  544
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  546  549  553  554  555  556  557  559  560  562  563  566  567  570  571
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  574  575  577  579  581  582  584  586  587  588  589  590  591  594  596
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  597  598  601  602  603  605  606  608  611  612  613  614  618  619  621
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  624  625  627  628  631  632  635  640  641  642  645  647  649  650  651
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  652  653  654  656  657  659  660  661  664  665  667  669  670  678  679
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  685  688  690  691  692  693  697  700  702  703  704  705  707  708  709
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  710  712  714  715  720  721  722  723  724  728  729  730  731  732  733
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
##  737  738  740  741  742  744  746  747  748  751  752  753  757  760  762
##    2    2    2    2    2    2    2    2    2    2    2    2    3    3    3
##  763  766  767  769  772  775  778  780  781  783  784  786  789  790  791
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  794  802  803  806  807  809  810  811  812  813  814  815  816  819  820
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  822  823  824  826  827  832  833  835  836  839  841  849  850  851  854
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  859  863  867  870  872  873  875  876  877  879  885  886  888  891  893
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  894  897  900  901  906  912  915  916  918  919  920  923  924  925  926
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  927  928  930  931  932  935  936  937  938  939  945  946  947  951  952
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  953  955  956  958  959  960  962  965  967  970  971  973  974  976  977
##    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
##  980  981  983  984  986  987  989  990  994  997  999 1000
##    3    3    3    3    3    3    3    3    3    3    3    3
```

```
sd.data2 <- scale(data_set)
hc.complete.sd2 <- hclust(dist(sd.data2), method = "complete")

plot(hc.complete.sd2)
```




```
cutree(hc.complete.sd2, 3)
```

```
##    2    4    5    8   10   11   12   13   14   15   16   19   22   23   24 
##    1    1    2    1    2    2    1    2    2    2    2    2    3    2    2 
##   26   28   29   30   31   32   33   35   36   38   39   40   42   43   44 
##    2    3    2    1    3    2    2    2    1    2    2    2    3    1    2 
##   45   48   52   55   59   60   61   63   64   68   73   74   76   77   78 
##    1    3    2    2    2    1    2    2    1    3    2    1    2    1    1 
##   80   84   85   87   88   89   90   92   95   96   98   99  102  104  106 
##    1    2    2    2    1    2    2    2    3    1    2    1    1    2    1 
##  108  110  111  112  113  115  119  120  121  124  126  127  128  129  130 
##    2    3    2    2    2    3    3    3    2    2    2    2    2    2    2 
##  132  138  140  141  142  143  144  146  147  149  153  154  155  156  157 
##    1    3    2    3    1    2    2    1    2    1    1    3    1    2    2 
##  158  159  164  167  168  170  171  173  174  175  177  178  180  182  185 
##    2    2    2    2    3    2    2    2    2    2    2    3    2    2    2 
##  187  188  189  190  192  193  195  196  198  200  202  204  206  208  209 
##    2    2    2    2    1    2    1    2    2    1    2    2    2    2    2 
##  213  214  217  218  219  221  227  228  230  231  234  236  238  240  243 
##    2    2    2    1    1    2    1    2    2    1    2    2    2    2    1 
##  249  251  252  253  258  261  262  263  266  269  274  275  285  286  287 
##    2    3    2    2    2    2    2    2    2    2    1    1    2    1    1 
##  288  289  290  292  293  294  296  300  302  304  308  309  310  313  314 
##    1    2    2    1    2    2    1    3    1    2    2    2    2    2    2 
##  316  317  320  321  322  323  324  326  329  330  331  333  335  336  337 
##    2    2    2    2    2    2    3    2    1    2    2    1    2    2    2 
##  339  340  341  342  343  344  345  347  348  350  352  354  356  360  363 
##    2    2    2    2    3    2    2    2    3    3    2    2    2    2    2 
##  365  368  369  370  375  376  379  382  384  388  389  392  393  394  396 
##    2    2    1    2    1    1    1    2    2    1    3    3    2    2    1 
##  397  398  399  406  408  410  411  417  423  426  430  432  433  435  439 
##    2    1    2    2    2    3    2    2    2    2    2    1    2    2    2 
##  440  442  443  445  447  448  450  455  457  458  459  461  462  463  466 
##    2    2    2    1    2    2    3    2    2    2    2    1    2    2    2 
##  467  471  472  473  475  476  478  479  480  481  482  483  486  492  495 
##    2    3    2    2    2    2    1    2    2    2    2    3    2    2    2 
##  497  499  500  501  502  503  504  505  507  508  511  513  514  516  517 
##    1    2    2    2    1    2    2    2    3    1    2    2    2    2    2 
##  519  522  523  525  526  529  530  531  532  536  538  539  540  541  544 
##    2    2    1    2    1    1    2    2    2    2    2    1    2    2    2 
##  546  549  553  554  555  556  557  559  560  562  563  566  567  570  571 
##    2    2    1    2    2    2    2    2    2    1    2    2    2    1    2 
##  574  575  577  579  581  582  584  586  587  588  589  590  591  594  596 
##    2    2    2    2    2    2    1    2    2    2    2    2    2    2    2 
##  597  598  601  602  603  605  606  608  611  612  613  614  618  619  621 
##    2    2    2    2    2    2    2    2    2    2    3    2    2    2    1 
##  624  625  627  628  631  632  635  640  641  642  645  647  649  650  651 
##    2    2    2    2    2    2    2    1    2    2    2    2    2    2    1 
##  652  653  654  656  657  659  660  661  664  665  667  669  670  678  679 
##    2    2    1    2    2    2    1    2    2    2    3    2    2    1    2 
##  685  688  690  691  692  693  697  700  702  703  704  705  707  708  709 
##    1    1    3    2    2    2    2    2    1    2    1    2    1    2    2 
##  710  712  714  715  720  721  722  723  724  728  729  730  731  732  733 
##    2    2    2    1    1    2    3    2    2    2    1    3    1    1    2 
##  737  738  740  741  742  744  746  747  748  751  752  753  757  760  762 
##    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2 
##  763  766  767  769  772  775  778  780  781  783  784  786  789  790  791 
##    2    2    2    2    1    2    2    2    1    2    2    3    1    1    2 
##  794  802  803  806  807  809  810  811  812  813  814  815  816  819  820 
##    2    2    2    1    2    1    2    2    2    1    1    1    1    1    2 
##  822  823  824  826  827  832  833  835  836  839  841  849  850  851  854 
##    2    1    2    2    2    2    1    2    2    2    1    2    2    2    2 
##  859  863  867  870  872  873  875  876  877  879  885  886  888  891  893 
##    2    2    2    2    2    2    2    3    2    2    1    2    1    1    2 
##  894  897  900  901  906  912  915  916  918  919  920  923  924  925  926 
##    1    2    1    2    2    2    2    1    2    2    1    2    2    1    2 
##  927  928  930  931  932  935  936  937  938  939  945  946  947  951  952 
##    2    1    2    2    2    2    2    2    2    1    2    1    2    2    2 
##  953  955  956  958  959  960  962  965  967  970  971  973  974  976  977 
##    2    2    3    2    2    2    1    2    2    2    2    2    1    2    2 
##  980  981  983  984  986  987  989  990  994  997  999 1000 
##    2    1    2    1    2    1    2    2    1    1    2    1
```

```r
## ## if dividing into two clusters, the rate of each cluster is 9% and 47%
## if dividing innto three clusters, the rate of each cluster 9%, 28% and 0.5%
## I think cluster = 2 is good, for cluster = 3, the rate of label 3 is only 0.5 % but the rate of label 2 i
s much lower than label 2 than cluster = 2, it means high error rate exist
table(cutree(hc.complete2, 3), cutree(hc.complete.sd2, 3))
```

```
##
##        1   2   3
##   1   50 139  24
##   2   31 148  10
##   3   33  84   3
```

```r
table(cutree(hc.complete2, 2), cutree(hc.complete.sd2, 2))
```

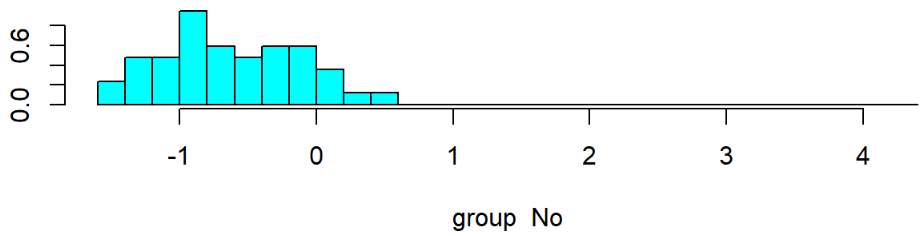
```
##
##        1   2
##   1   50 163
##   2   64 245
```

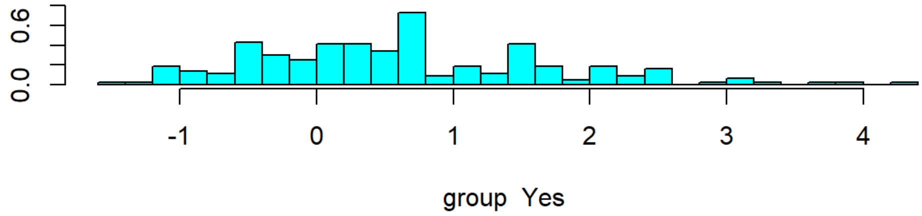
```r
## (3)fit the logistic model important predictors found in the last step: Duration,Job and Age
sample_size = floor(0.5*nrow(data1))
picked = sample(seq_len(nrow(data1)),size = sample_size)
train2 =data1[picked,]
test2 = data1[-picked,]

## (4)perform LDA on data set
install.packages("MASS",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'MASS' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\jolly\AppData\Local\Temp\RtmpkX3MWk\downloaded_packages
```

```r
library(MASS)
lda.fit=lda(High~Duration+Job+Age,data=train2)

## from the lda.fit I learnt that the coefficients of linear discrimnants for Duration, Job and Age is 0.079
, 0.49 annd 0.01. all of the three is relation to credit risk(high or low)
plot(lda.fit)
```






```
## compute the error rate of LDA on test data set, is 18.77%
lda.pred = predict(lda.fit, newdata=test2, type="response")
lda.class = lda.pred$class
tab1<-table(lda.class, test2$High)

print(paste0("the error rate is:",1 - sum(diag(tab1))/sum(tab1)))
```

```
## [1] "the error rate is:0.164750957854406"
```

```
## (5)repeat using QDA, the test error rate of QDA is 18%
qda.fit = qda(High~Duration+Job+Age, data= train2)
qda.fit
```

```
## Call:
## qda(High ~ Duration + Job + Age, data = train2)
##
## Prior probabilities of groups:
##        No       Yes
## 0.1609195 0.8390805
##
## Group means:
##       Duration      Job      Age
##  No   11.47619 1.52381 36.35714
##  Yes  24.52055 1.96347 34.88584
```

```
qda.pred = predict(qda.fit, newdata=test2, type="response")
qda.class = qda.pred$class
tab2<-table(qda.class, test2$High)

print(paste0("the error rate is:",1 - sum(diag(tab2))/sum(tab2)))
```

```
## [1] "the error rate is:0.210727969348659"
```

```
## (6) performing subset selection model
install.packages("leaps", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## Warning: package 'leaps' is in use and will not be installed
```

```
library (leaps)
## output indicates that the best two-variable model contains only Duration and job
regfit.full=regsubsets (High~.-Credit.amount,data1 )
summary (regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(High ~ . - Credit.amount, data1)
## 9 Variables  (and intercept)
##                  Forced in Forced out
## X                    FALSE      FALSE
## Age                  FALSE      FALSE
## Sex                  FALSE      FALSE
## Job                  FALSE      FALSE
## Housing              FALSE      FALSE
## Saving.accounts      FALSE      FALSE
## Checking.account     FALSE      FALSE
## Duration             FALSE      FALSE
## Purpose              FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          X   Age Sex Job Housing Saving.accounts Checking.account Duration
## 1  ( 1 ) " " " " " " " " " "     " "             " "              "*"
## 2  ( 1 ) " " " " " " "*" " "     " "             " "              "*"
## 3  ( 1 ) " " " " "*" "*" " "     " "             " "              "*"
## 4  ( 1 ) " " " " "*" "*" "*"     " "             " "              "*"
## 5  ( 1 ) " " " " "*" "*" "*"     "*"             " "              "*"
## 6  ( 1 ) "*" " " "*" "*" "*"     "*"             " "              "*"
## 7  ( 1 ) "*" "*" "*" "*" "*"     "*"             " "              "*"
## 8  ( 1 ) "*" "*" "*" "*" "*"     "*"             "*"              "*"
##          Purpose
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) " "
## 7  ( 1 ) " "
## 8  ( 1 ) " "
```

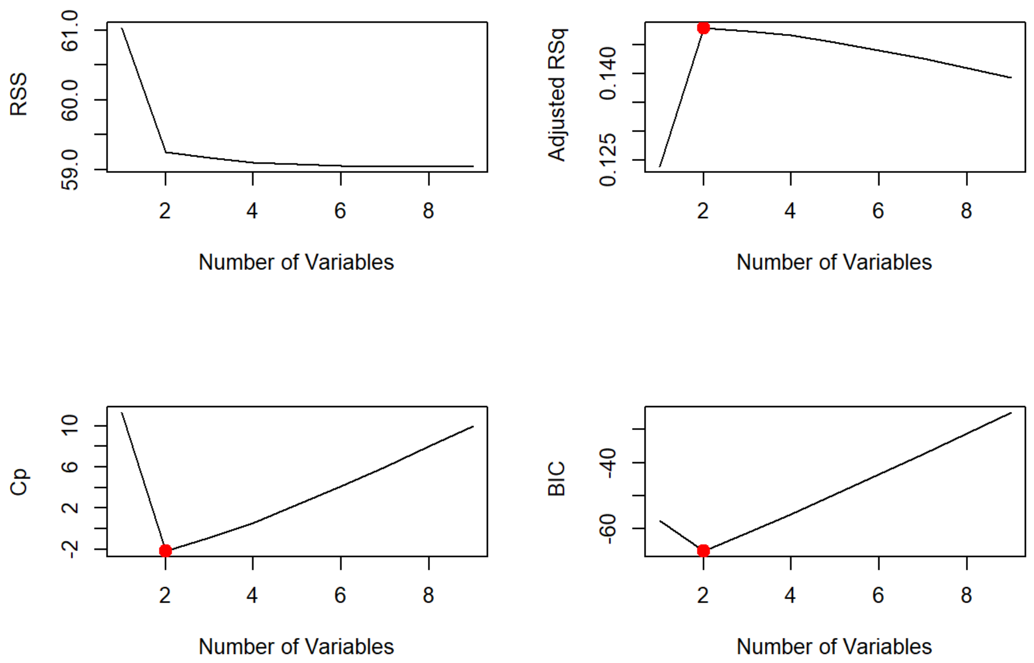
```
## fit up to a 9-variable model.
regfit.full=regsubsets (High~.-Credit.amount,data1,nvmax = 19)
reg.summary = summary (regfit.full)

names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

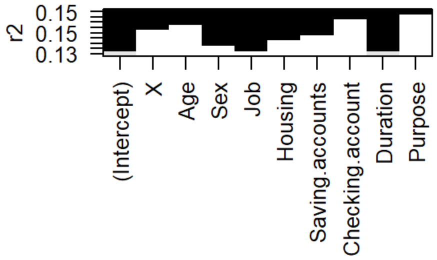
```
## we see that the R2 statistic increases from 12%, when only one variable is included in the model, to almo
st 15 %, when all variables are included. As expected, the R2 statistic increases monotonically as more vari
ables are included.
reg.summary$rsq
```

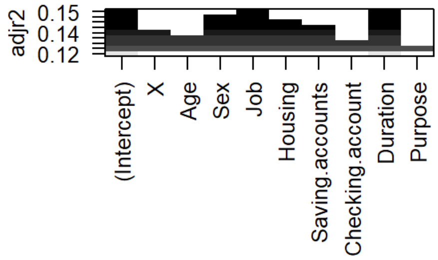
```
## [1] 0.1256251 0.1511946 0.1523181 0.1533029 0.1536603 0.1539660 0.1541382
## [8] 0.1541632 0.1541731
```

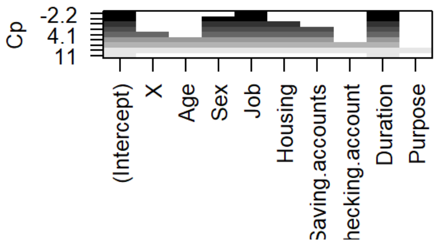
```r
## Plotting RSS, adjusted R2, Cp, and BIC for all of the models at once will help me decide which model to s
elect
par(mfrow =c(2,2))
plot(reg.summary$rss ,xlab=" Number of Variables ",ylab=" RSS",
type="l")
plot(reg.summary$adjr2 ,xlab =" Number of Variables ",
ylab=" Adjusted RSq",type="l")

max1 = which.max (reg.summary$adjr2)

## We will now plot a red dot to indicate the model with the largest adjusted R2 statistic.
points (max1, reg.summary$adjr2[max1], col ="red",cex =2, pch =20)

## In a similar fashion we can plot the Cp and BIC statistics, and indicate the models with the smallest sta
tistic using which.min().
plot(reg.summary$cp ,xlab =" Number of Variables ",ylab="Cp",
type='l')
min1 = which.min (reg.summary$cp )
points (min1, reg.summary$cp [min1], col ="red",cex =2, pch =20)

min2 = which.min (reg.summary$bic )
plot(reg.summary$bic ,xlab=" Number of Variables ",ylab=" BIC",
type='l')
points (min2, reg.summary$bic [min2], col =" red",cex =2, pch =20)
```

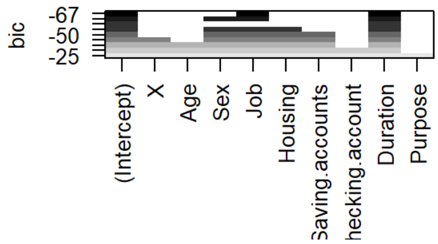



```r
plot(regfit.full ,scale ="r2")
plot(regfit.full ,scale ="adjr2")
plot(regfit.full ,scale ="Cp")
plot(regfit.full ,scale ="bic")
```










```
## the model with the lowest BIC is the two-variable model that contains only Duration and Job
## coefficient estimates associated with this model.
coef(regfit.full ,2)
```

```
## (Intercept)         Job    Duration
## 1.475478668 0.087501256 0.009438586
```

```
## final conclusion:
## to predict the Credit risk, I use the Credit amount as the quantitative predictor  because I think the hi
gher of credit amount, the high risk of credit. then I perform linear regression, logistic regression, GAMs,
tree model and subset selection model to prove that the most two features deciding credit amount are Duratio
n and Job, decisin tree performs best since the error rate is 32.18%, GAMs performs worst, linear regression
and logistic model ia the same. Then I use High as qualitative response since it is classified as(yes OR no)
to decide if the credit risk is high or not, I perform decision tree, logistic regression, LDA,QDA and subse
t selection model to prove that the most two features deciding High are Duration and Job, the decision QDA p
erforms best, then is the tree model and LDA.

## others to say, maybe thing will change when I use other variables pattern and split the data differently.

## the results I get is that the longer duration and more skilled tha job is, the credit risk is higher. The
variable named Purpose also make sense because more expensive thing they buy(for example: education, vacatio
n), the higher credit risk is.
```

Question 4:

(Based on ISLR Chapter 9 #7) In this problem, you will use support vector approaches in order to predict whether a given car gets high or low gas mileage based on the Auto data set.

(a)

Create a binary variable that takes on a 1 for cars with gas mileage above the median, and a 0 for cars with gas mileage below the median.

(b)

Fit a support vector classifier to the data with various values of cost, in order to predict whether a car gets high or low gas mileage. Report the cross-validation errors associated with different values of this parameter. Comment on your results.

(c)

Now repeat for (b), this time using SVMs with radial and polynomial basis kernels, with different values of gamma and degree and cost.

Comment on your results.

(d)

Make some plots to back up your assertions in (b) and (c). Hint: In the lab, we used the plot() function for svm objects only in cases with p=2 When p>2,you can use the plot() function to create plots displaying pairs of variables at a time. Essentially, instead of typing plot(svmfit , dat) where svmfit contains your fitted model and dat is a data frame containing your data, you can type plot(svmfit , dat, x1~x4) in order to plot just the first and fourth variables. However, you must replace x1 and x4 with the correct variable names. To find out more, type ?plot.svm.

```
##(a) Create a binary variable that takes on a 1 for cars with gas mileage above the median, and a 0 for car
s with gas mileage below the median.
##require(ISLR);

install.packages('e1071', dependencies=TRUE,repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/jolly/OneDrive/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
##
##   There is a binary version available but the source version is
##   later:
##       binary source needs_compilation
## e1071  1.7-2  1.7-3              TRUE
##
##   Binaries will be installed
## package 'e1071' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'e1071'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE):
## problem copying C:\Users\jolly\OneDrive\Documents\R\win-
## library\3.6\00LOCK\e1071\libs\x64\e1071.dll to C:
## \Users\jolly\OneDrive\Documents\R\win-library\3.6\e1071\libs\x64\e1071.dll:
## Permission denied
```

```
## Warning: restored 'e1071'
```

```
##
## The downloaded binary packages are in
##  C:\Users\jolly\AppData\Local\Temp\RtmpkX3MWk\downloaded_packages
```

```
library(e1071)
library(ISLR)
data(Auto)
var <- ifelse(Auto$mpg > median(Auto$mpg), 1, 0)
Auto$mpglevel <- as.factor(var)

## (b)Fit a support vector classifier to the data with various values of cost, in order to predict whether a
car gets high or low gas mileage. Report the cross-validation errors associated with different values of thi
s parameter. Comment on your results.

tune.out <- tune(svm, mpglevel ~ ., data = Auto, kernel = "linear", ranges = list(cost = c(0.01, 0.1, 1, 5,
10, 100, 1000)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     1
##
## - best performance: 0.01019231
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-02 0.07416667 0.02845982
## 2 1e-01 0.05108974 0.01720596
## 3 1e+00 0.01019231 0.01315951
## 4 5e+00 0.01775641 0.01700310
## 5 1e+01 0.02288462 0.02226748
## 6 1e+02 0.03826923 0.04052180
## 7 1e+03 0.03826923 0.04052180
```

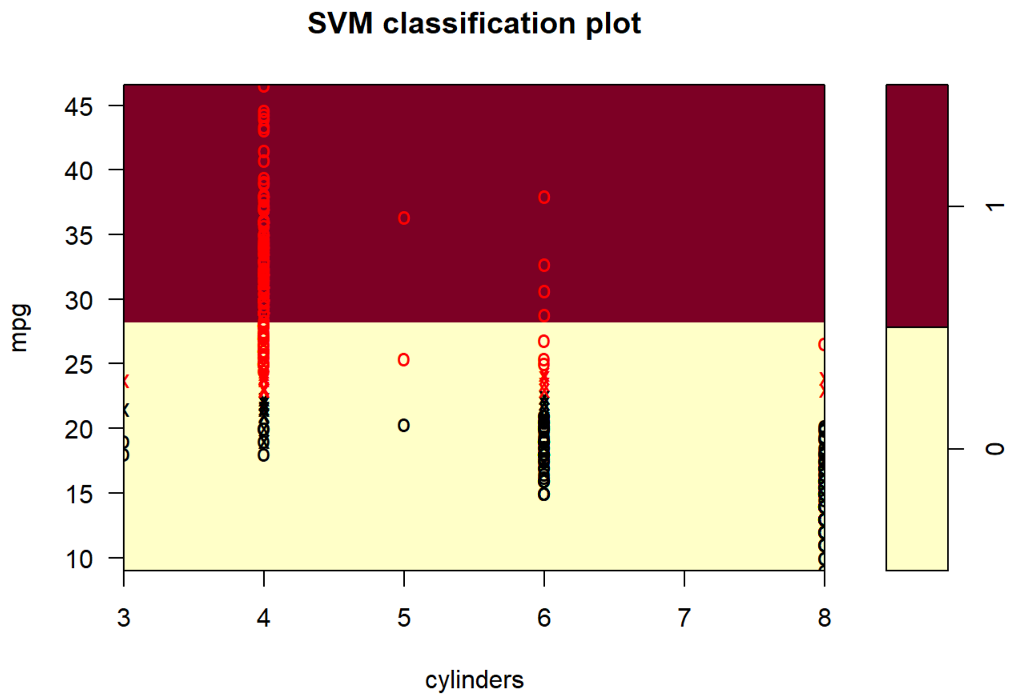
```
## A cost of 1 seems to perform best.

##(c) Now repeat for (b), this time using SVMs with radial and polynomial basis kernels, with different valu
es of gamma and degree and cost. Comment on your results.
tune.out <- tune(svm, mpglevel ~ ., data = Auto, kernel = "polynomial", ranges = list(cost = c(0.01, 0.1, 1,
5, 10, 100), degree = c(2, 3, 4)))
summary(tune.out)
```

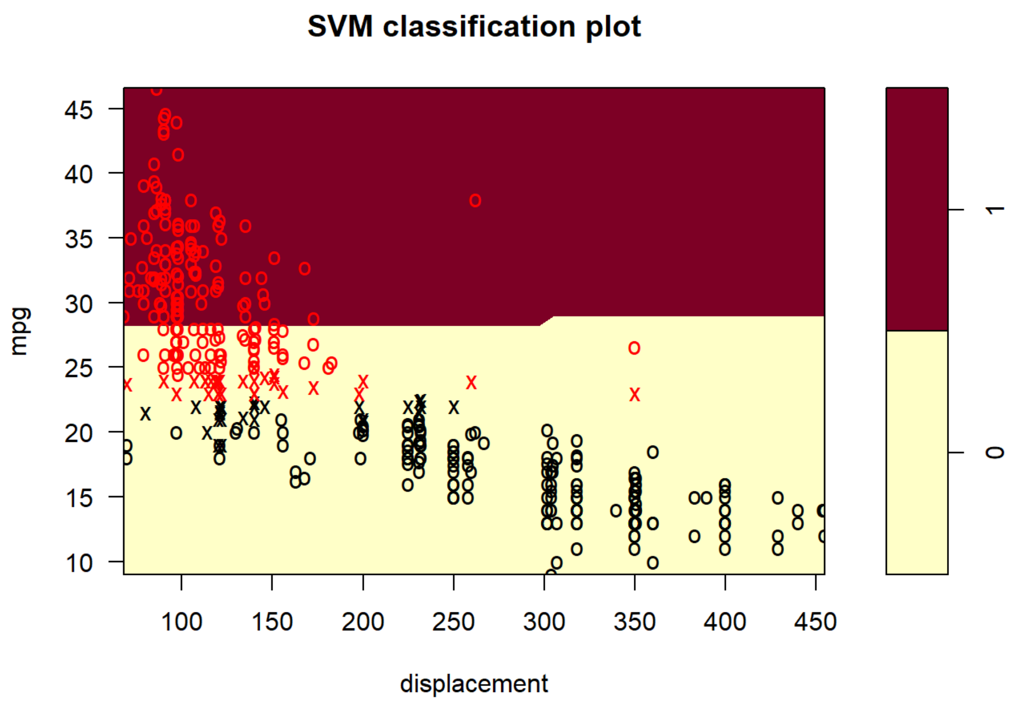
```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost degree
##   100      2
##
## - best performance: 0.2989103
##
## - Detailed performance results:
##      cost degree     error dispersion
## 1  1e-02      2 0.5533974 0.04391330
## 2  1e-01      2 0.5533974 0.04391330
## 3  1e+00      2 0.5533974 0.04391330
## 4  5e+00      2 0.5533974 0.04391330
## 5  1e+01      2 0.4944231 0.11634377
## 6  1e+02      2 0.2989103 0.09080269
## 7  1e-02      3 0.5533974 0.04391330
## 8  1e-01      3 0.5533974 0.04391330
## 9  1e+00      3 0.5533974 0.04391330
## 10 5e+00      3 0.5533974 0.04391330
## 11 1e+01      3 0.5533974 0.04391330
## 12 1e+02      3 0.3373077 0.10093471
## 13 1e-02      4 0.5533974 0.04391330
## 14 1e-01      4 0.5533974 0.04391330
## 15 1e+00      4 0.5533974 0.04391330
## 16 5e+00      4 0.5533974 0.04391330
## 17 1e+01      4 0.5533974 0.04391330
## 18 1e+02      4 0.5533974 0.04391330
```

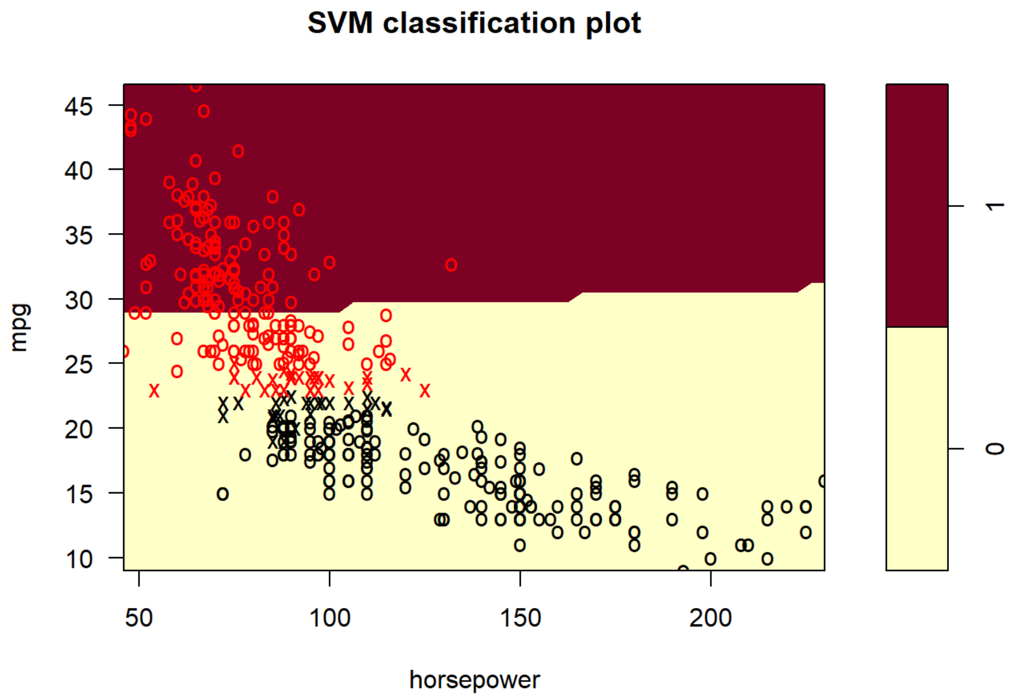
```
## For a polynomial kernel, the lowest cross-validation error is obtained for a degree of 2 and a cost of 10
0.

tune.out <- tune(svm, mpglevel ~ ., data = Auto, kernel = "radial", ranges = list(cost = c(0.01, 0.1, 1, 5,
10, 100), gamma = c(0.01, 0.1, 1, 5, 10, 100)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##   100  0.01
##
## - best performance: 0.01275641
##
## - Detailed performance results:
##     cost gamma      error dispersion
## 1  1e-02 1e-02 0.51115385 0.16112732
## 2  1e-01 1e-02 0.08423077 0.04195898
## 3  1e+00 1e-02 0.07147436 0.03789378
## 4  5e+00 1e-02 0.04333333 0.02409173
## 5  1e+01 1e-02 0.02294872 0.02534336
## 6  1e+02 1e-02 0.01275641 0.01808165
## 7  1e-02 1e-01 0.18397436 0.07167776
## 8  1e-01 1e-01 0.07653846 0.04179325
## 9  1e+00 1e-01 0.05352564 0.03286051
## 10 5e+00 1e-01 0.02301282 0.01891104
## 11 1e+01 1e-01 0.02807692 0.02810426
## 12 1e+02 1e-01 0.02801282 0.01875978
## 13 1e-02 1e+00 0.50865385 0.16866413
## 14 1e-01 1e+00 0.50865385 0.16866413
## 15 1e+00 1e+00 0.06121795 0.02154891
## 16 5e+00 1e+00 0.06115385 0.02451815
## 17 1e+01 1e+00 0.06371795 0.02155304
## 18 1e+02 1e+00 0.06371795 0.02155304
## 19 1e-02 5e+00 0.55365385 0.05321847
## 20 1e-01 5e+00 0.55365385 0.05321847
## 21 1e+00 5e+00 0.48717949 0.09386594
## 22 5e+00 5e+00 0.48717949 0.08989050
## 23 1e+01 5e+00 0.48717949 0.08989050
## 24 1e+02 5e+00 0.48717949 0.08989050
## 25 1e-02 1e+01 0.55615385 0.05095713
## 26 1e-01 1e+01 0.55615385 0.05095713
## 27 1e+00 1e+01 0.51012821 0.07349732
## 28 5e+00 1e+01 0.50756410 0.07675004
## 29 1e+01 1e+01 0.50756410 0.07675004
## 30 1e+02 1e+01 0.50756410 0.07675004
## 31 1e-02 1e+02 0.55365385 0.05321847
## 32 1e-01 1e+02 0.55365385 0.05321847
## 33 1e+00 1e+02 0.55365385 0.05321847
## 34 5e+00 1e+02 0.55365385 0.05321847
## 35 1e+01 1e+02 0.55365385 0.05321847
## 36 1e+02 1e+02 0.55365385 0.05321847
```

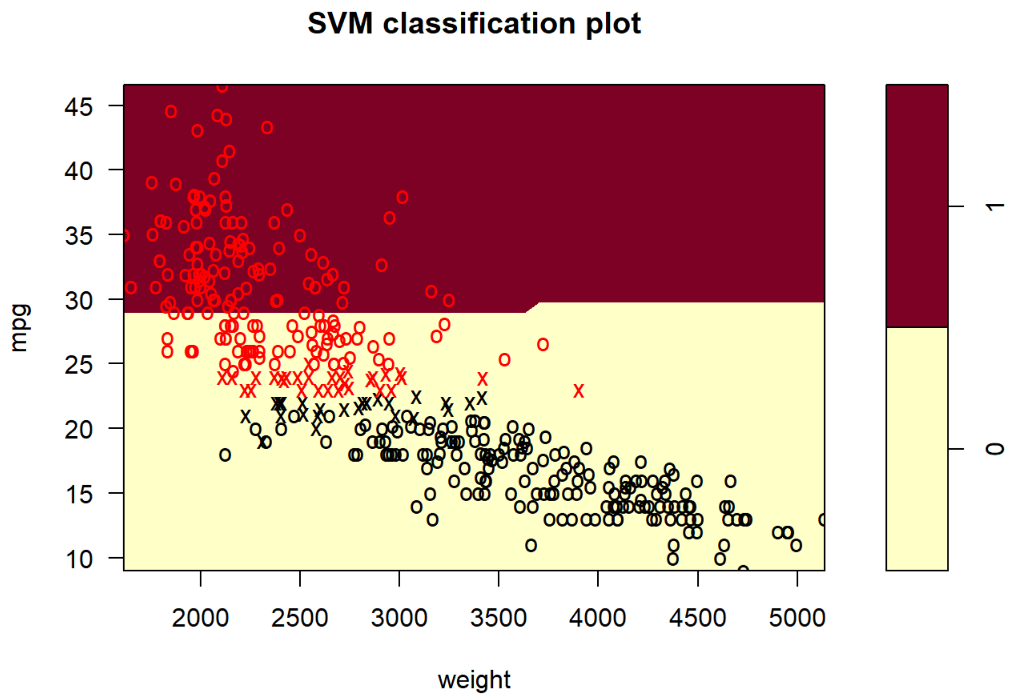
```
## For a radial kernel, the lowest cross-validation error is obtained for a gamma of 0.01 and a cost of 100.


##(d) Make some plots to back up your assertions in (b) and (c). Hint: In the lab, we used the plot() functi
on for svm objects only in cases with p=2 When p>2,you can use the plot() function to create plots displayin
g pairs of variables at a time. Essentially, instead of typing plot(svmfit , dat) where svmfit contains your
fitted model and dat is a data frame containing your data, you can type plot(svmfit , dat, x1~x4) in order t
o plot just the first and fourth variables. However, you must replace x1 and x4 with the correct variable na
mes. To find out more, type ?plot.svm.
svm.linear <- svm(mpglevel ~ ., data = Auto, kernel = "linear", cost = 1)
svm.poly <- svm(mpglevel ~ ., data = Auto, kernel = "polynomial", cost = 100, degree = 2)
svm.radial <- svm(mpglevel ~ ., data = Auto, kernel = "radial", cost = 100, gamma = 0.01)
plotpairs = function(fit) {
    for (name in names(Auto)[!(names(Auto) %in% c("mpg", "mpglevel", "name"))]) {
        plot(fit, Auto, as.formula(paste("mpg~", name, sep = "")))
    }
}
plotpairs(svm.linear)
```


SVM classification plot
mpg
45
40
35
30
25
20
15
10
3
4
5
6
7
8
cylinders
1
0

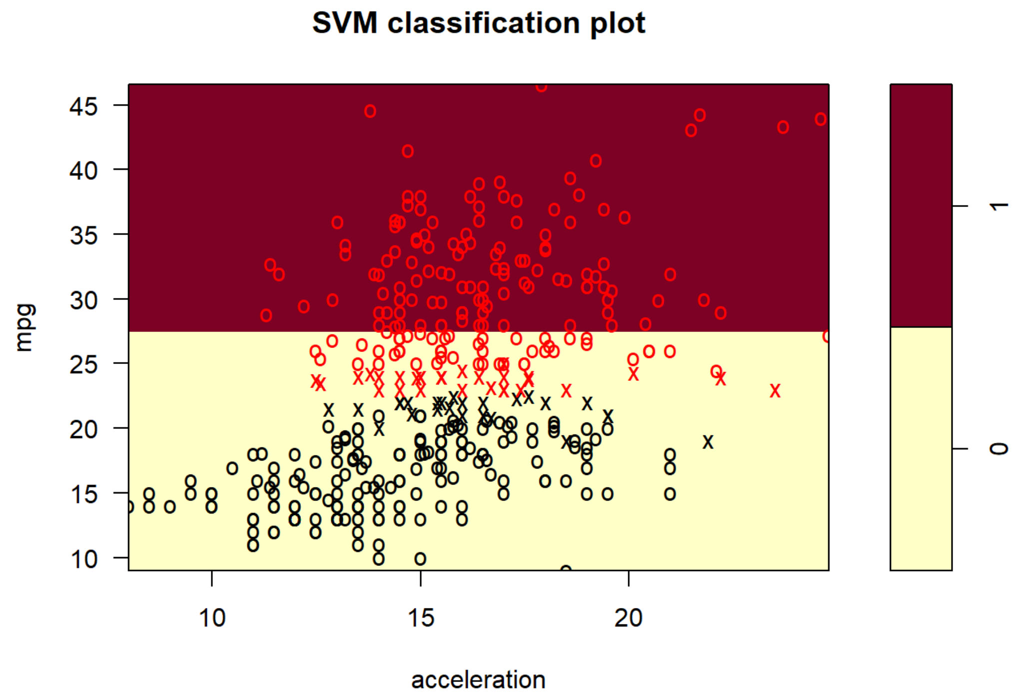

SVM classification plot
mpg
45
40
35
30
25
20
15
10
100
150
200
250
300
350
400
450
displacement
1
0

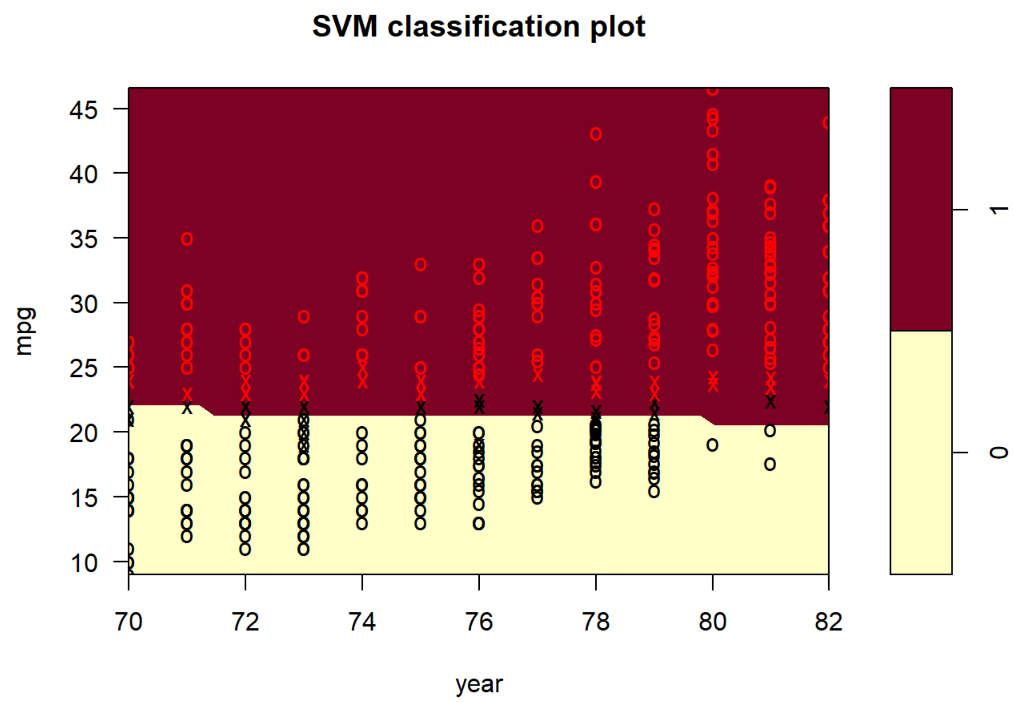


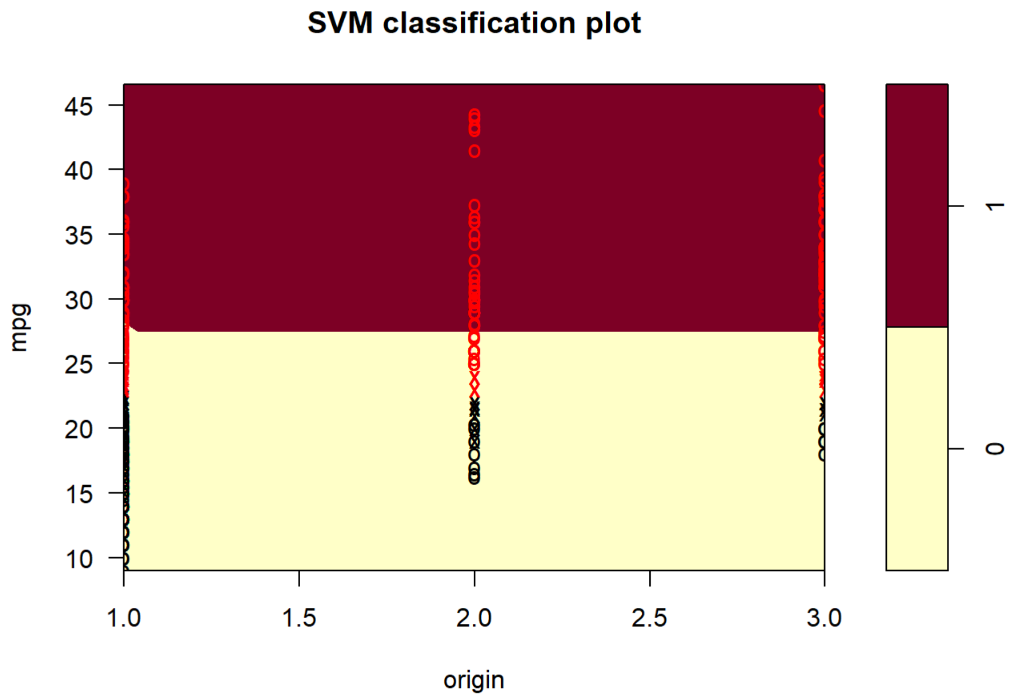
SVM classification plot
mpg
45
40
35
30
25
20
15
10
50
100
150
200
horsepower
1
0



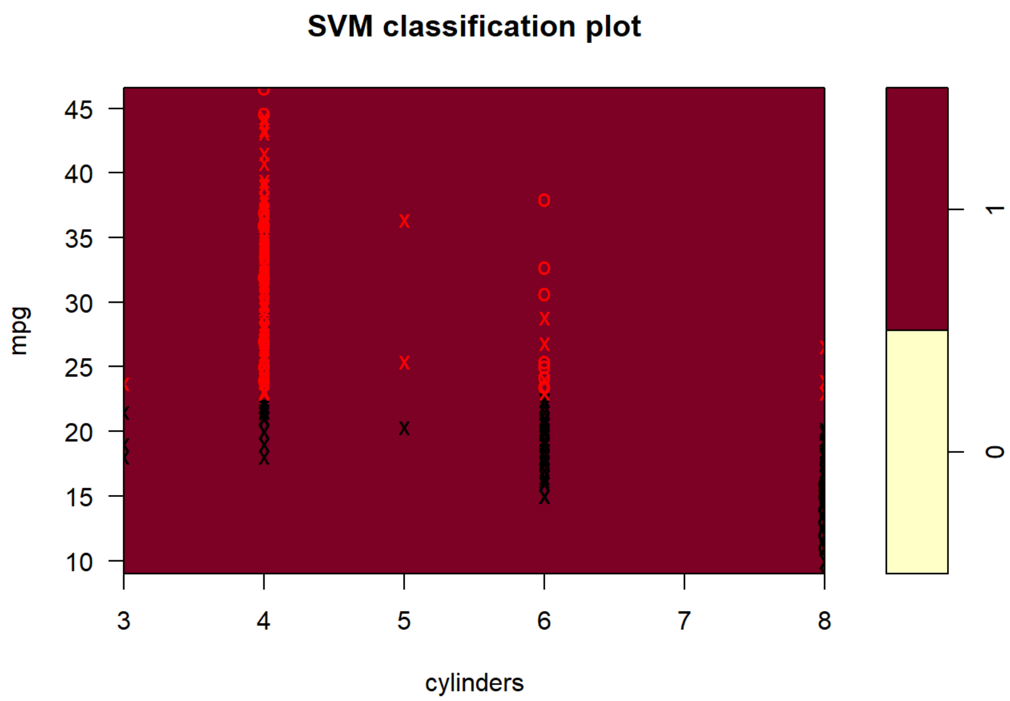
SVM classification plot
mpg
45
40
35
30
25
20
15
10
2000
2500
3000
3500
4000
4500
5000
weight
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
10
15
20
acceleration
1
0

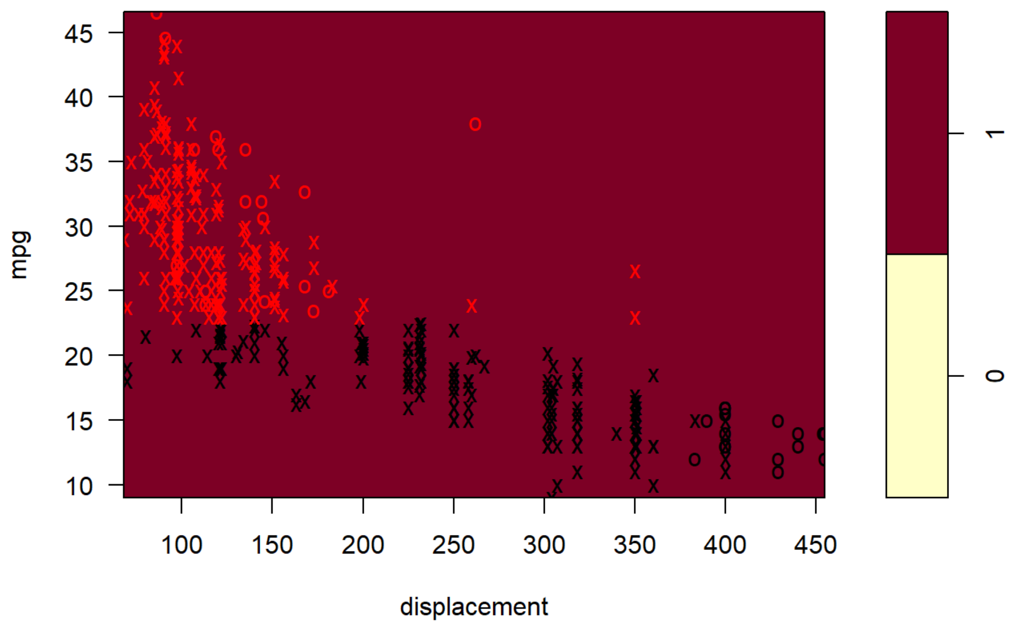

SVM classification plot
mpg
45
40
35
30
25
20
15
10
70
72
74
76
78
80
82
year
1
0



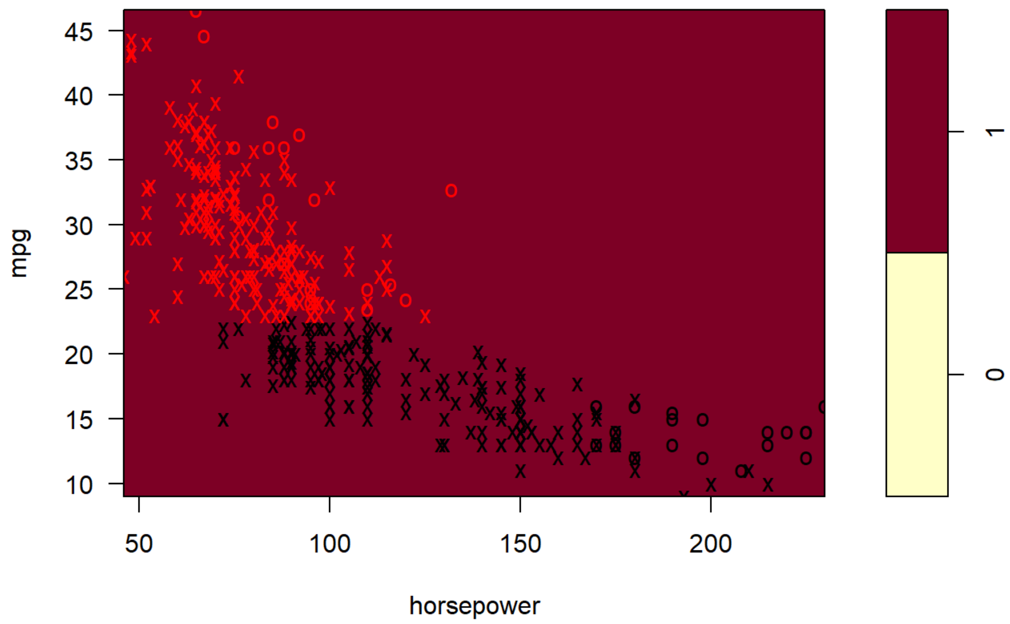

```
plotpairs(svm.poly)
```

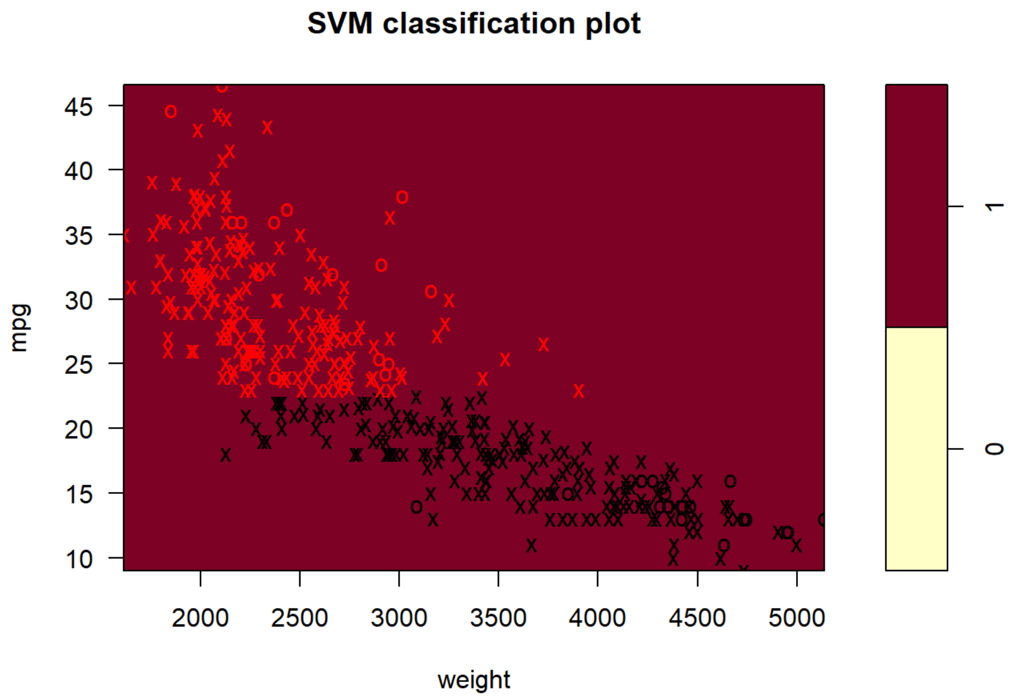


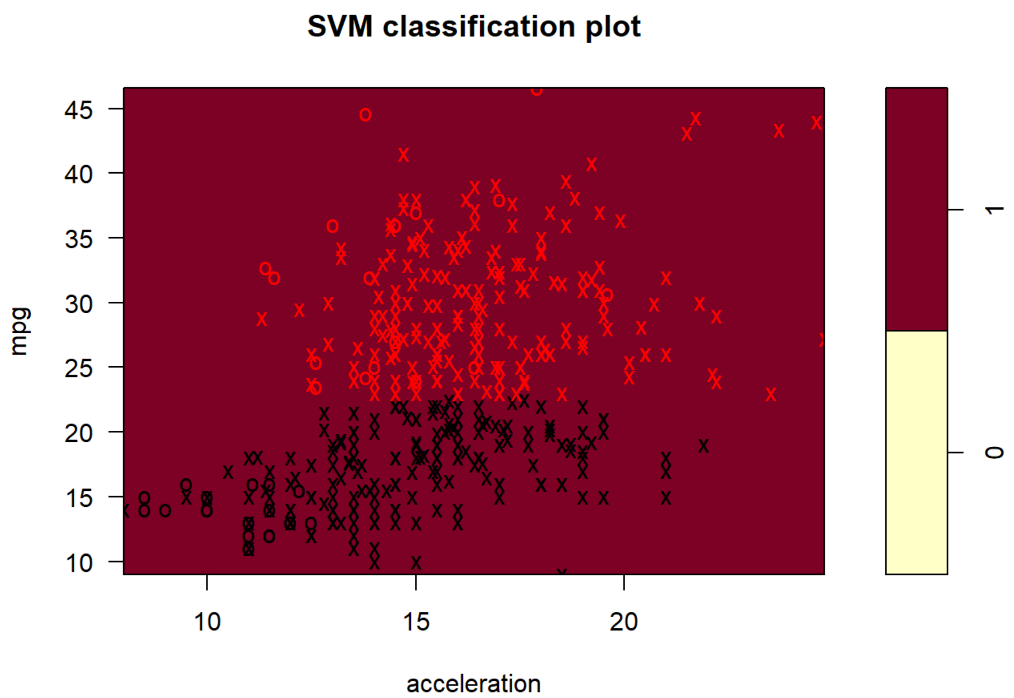
SVM classification plot

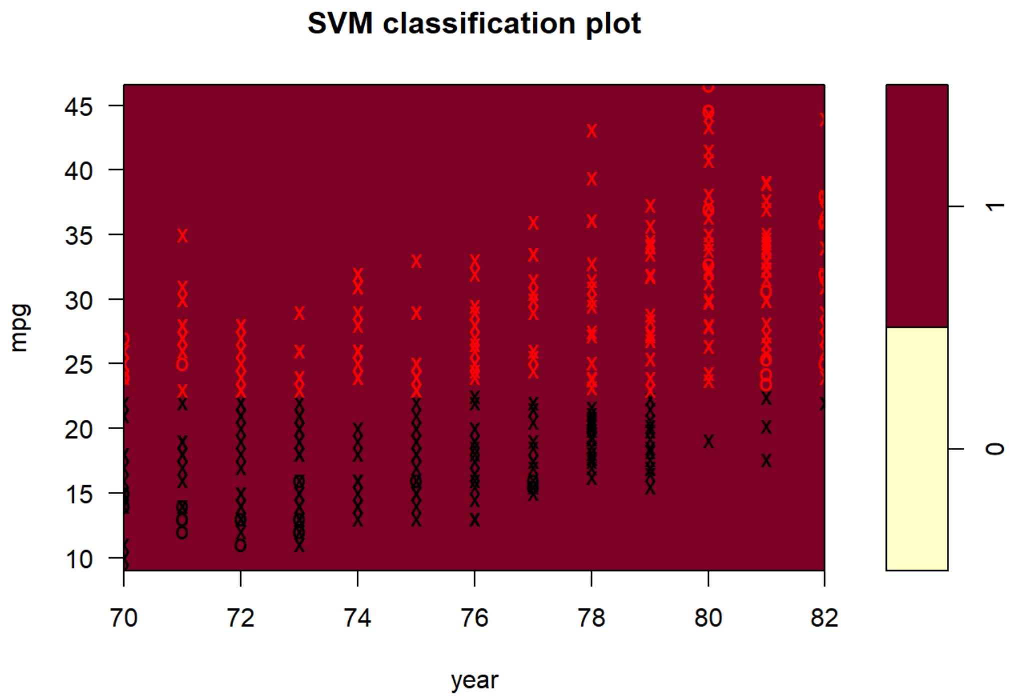


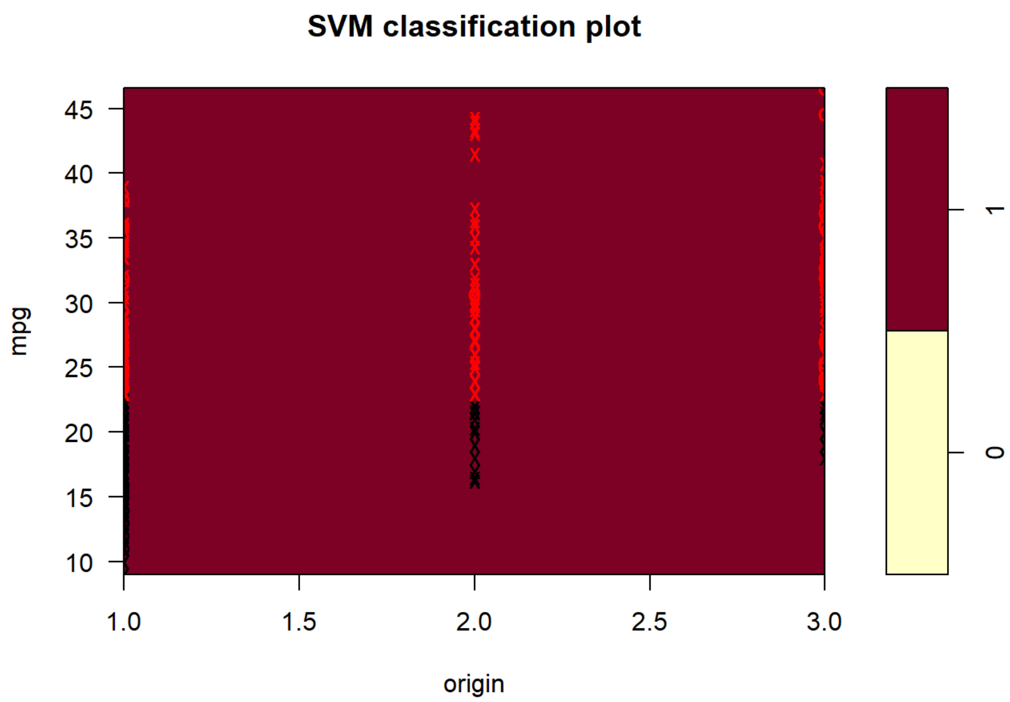
SVM classification plot



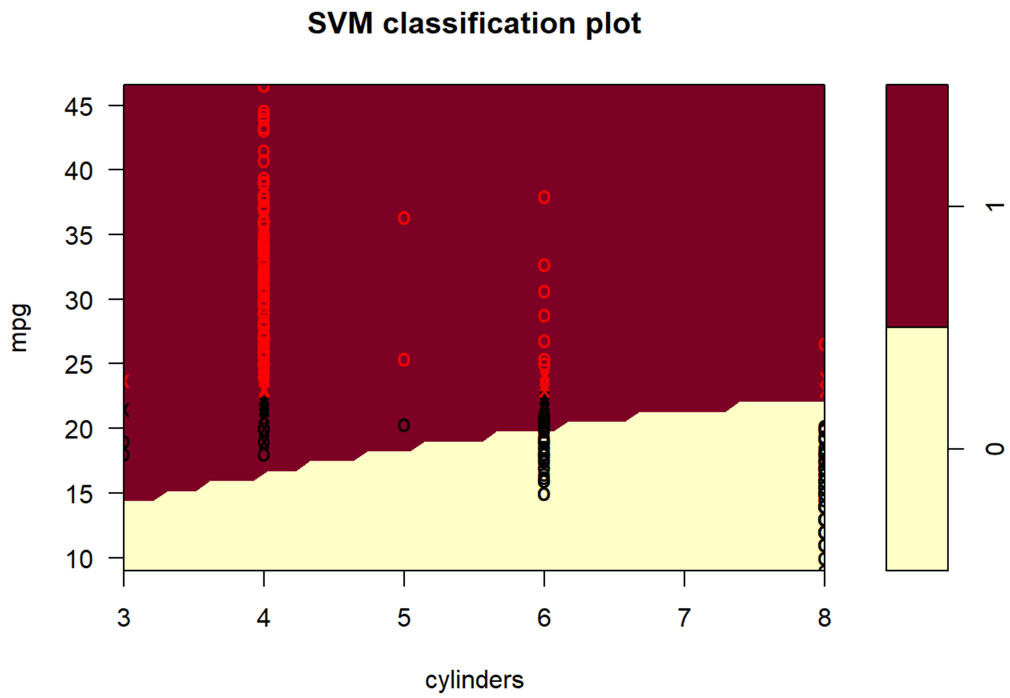

SVM classification plot
mpg
45
40
35
30
25
20
15
10
2000
2500
3000
3500
4000
4500
5000
weight
1
0

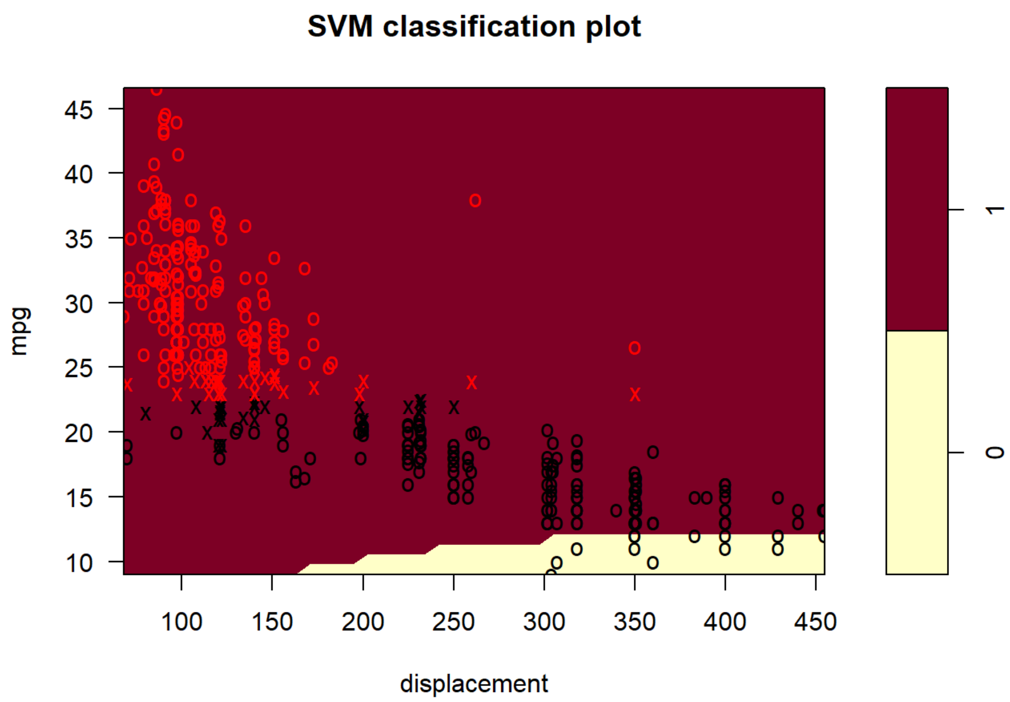

SVM classification plot
mpg
45
40
35
30
25
20
15
10
10
15
20
acceleration
1
0

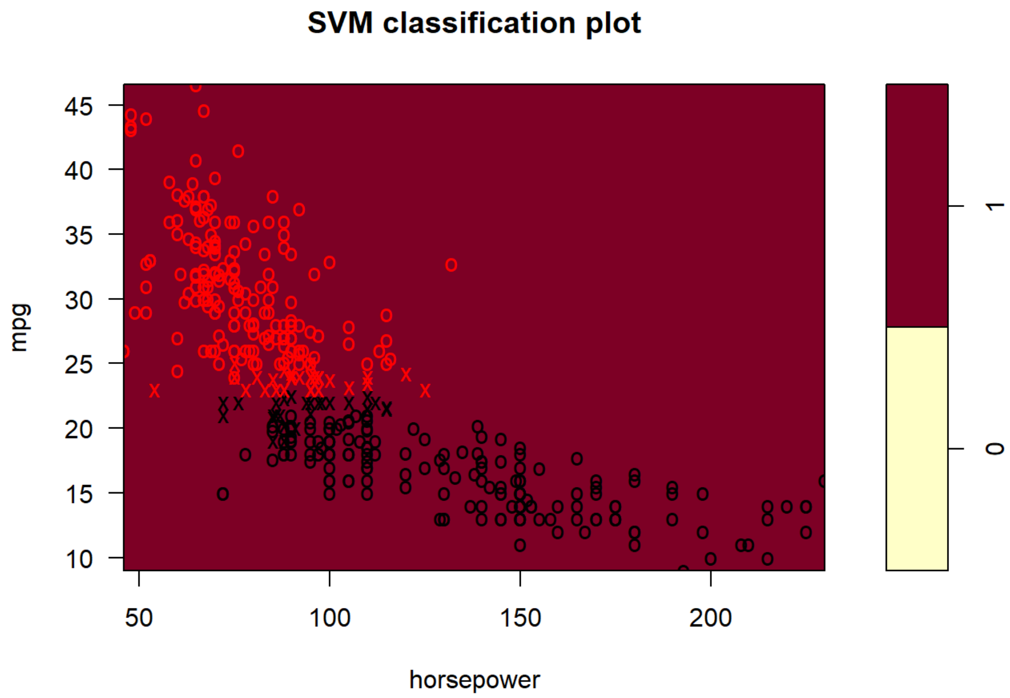


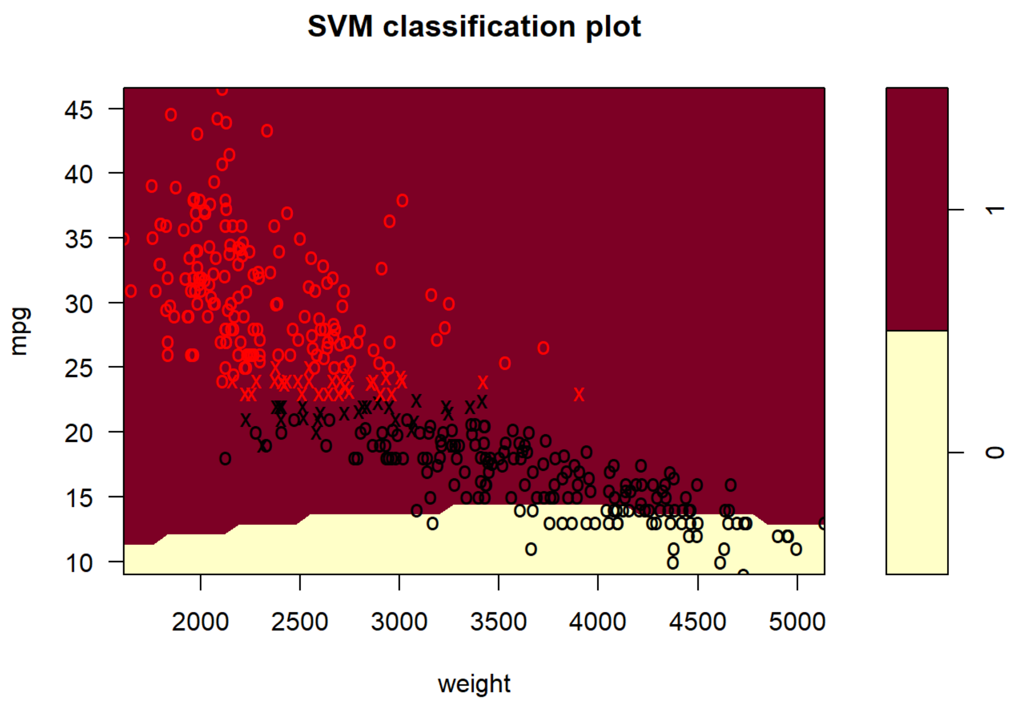


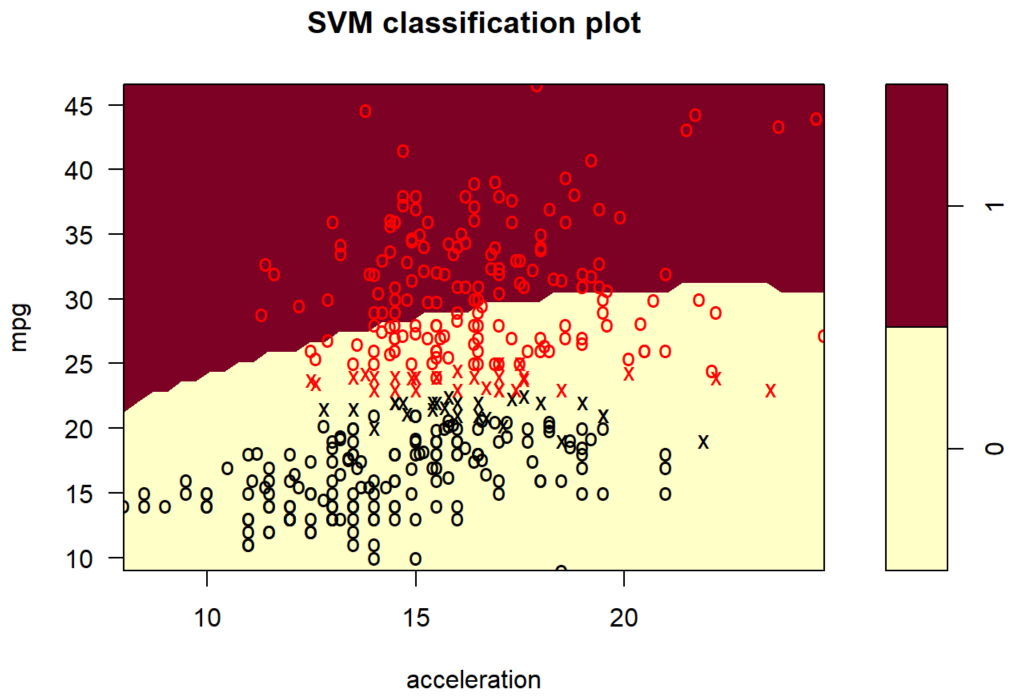

```
plotpairs(svm.radial)
```

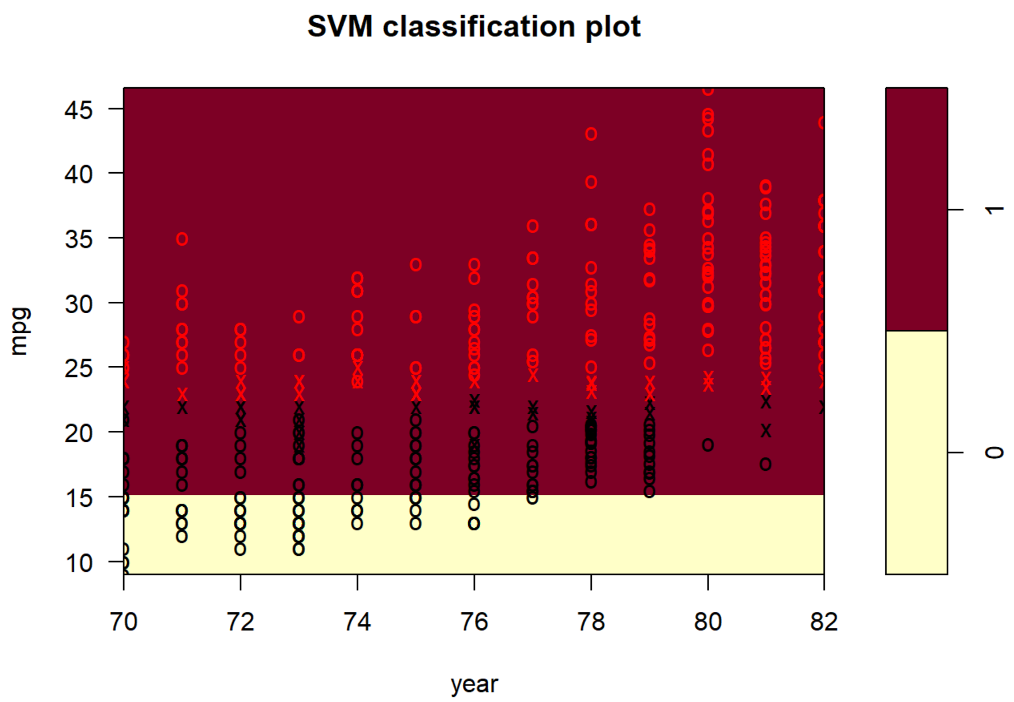

SVM classification plot
mpg
45
40
35
30
25
20
15
10
3
4
5
6
7
8
cylinders
1
0

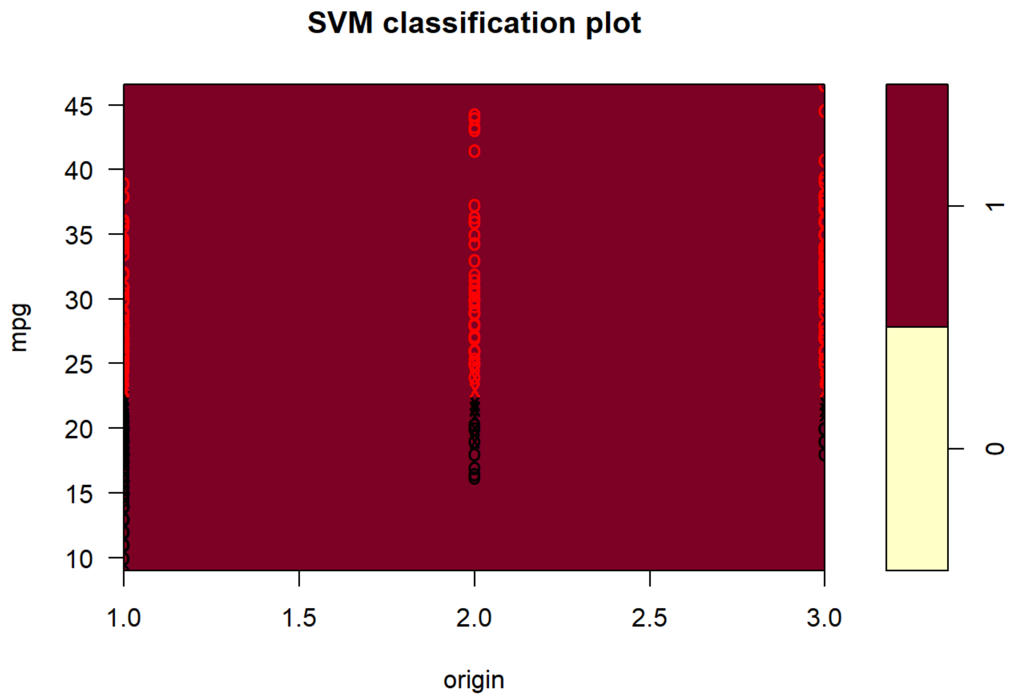

SVM classification plot
mpg
45
40
35
30
25
20
15
10
100
150
200
250
300
350
400
450
displacement
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
50
100
150
200
horsepower
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
2000
2500
3000
3500
4000
4500
5000
weight
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
10
15
20
acceleration
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
70
72
74
76
78
80
82
year
1
0


SVM classification plot
mpg
45
40
35
30
25
20
15
10
1.0
1.5
2.0
2.5
3.0
origin
1
0