Coursework2

KieranPatel

21/03/2017

# Preliminary and reading data

library(car)  
library(dplyr)

##   
## Attaching package: 'dplyr'

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

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
library(leaps)  
library(ElemStatLearn)  
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-5

train = read.table("Train.txt",header = T)  
test = read.table("Test.txt",header = T)  
train = filter( train, !is.na(Status), !is.na(Duration), !is.na(History), !is.na(Purpose), !is.na(Amount), !is.na(Savings), !is.na(Employment), !is.na(Disposable), !is.na(Personal), !is.na(OtherParties), !is.na(Residence), !is.na(Property), !is.na(Age), !is.na(Plans), !is.na(Housing), !is.na(Existing), !is.na(Job), !is.na(Dependants), !is.na(Telephone), !is.na(Foreign), !is.na(CreditScore))  
test = filter( test, !is.na(Status), !is.na(Duration), !is.na(History), !is.na(Purpose), !is.na(Amount), !is.na(Savings), !is.na(Employment), !is.na(Disposable), !is.na(Personal), !is.na(OtherParties), !is.na(Residence), !is.na(Property), !is.na(Age), !is.na(Plans), !is.na(Housing), !is.na(Existing), !is.na(Job), !is.na(Dependants), !is.na(Telephone), !is.na(Foreign), !is.na(CreditScore))  
str(train)

## 'data.frame': 800 obs. of 21 variables:  
## $ Status : Factor w/ 4 levels "Large","Negative",..: 2 4 3 2 3 3 4 4 4 2 ...  
## $ Duration : int 6 48 12 42 36 24 36 30 12 48 ...  
## $ History : Factor w/ 5 levels "A","B","C","D",..: 5 3 5 3 3 3 3 5 3 3 ...  
## $ Purpose : Factor w/ 10 levels "Business","Domestic",..: 8 8 3 4 3 4 10 5 5 1 ...  
## $ Amount : int 1169 5951 2096 7882 9055 2835 6948 5234 1295 4308 ...  
## $ Savings : Factor w/ 5 levels "Large","Low",..: 4 2 2 2 4 1 2 2 2 2 ...  
## $ Employment : Factor w/ 5 levels "Long","Medium",..: 5 2 1 1 2 5 2 4 3 3 ...  
## $ Disposable : int 4 2 2 2 2 3 2 4 3 3 ...  
## $ Personal : Factor w/ 4 levels "F:DivSepMar",..: 4 1 4 4 4 4 4 2 1 1 ...  
## $ OtherParties: Factor w/ 3 levels "Coapp","Guarantor",..: 3 3 3 2 3 3 3 3 3 3 ...  
## $ Residence : int 4 2 3 4 4 4 2 2 1 4 ...  
## $ Property : Factor w/ 4 levels "Car","House",..: 2 2 2 4 3 4 1 1 1 4 ...  
## $ Age : int 67 22 49 45 35 53 35 28 25 24 ...  
## $ Plans : Factor w/ 3 levels "Bank","None",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Housing : Factor w/ 3 levels "Own","Rent","RentFree": 1 1 1 3 3 1 2 1 2 2 ...  
## $ Existing : int 2 1 1 1 1 1 1 2 1 1 ...  
## $ Job : Factor w/ 4 levels "Management:Self",..: 2 2 4 2 4 2 1 1 2 2 ...  
## $ Dependants : int 1 1 2 2 2 1 1 1 1 1 ...  
## $ Telephone : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 2 1 1 1 ...  
## $ Foreign : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...  
## $ CreditScore : num 636 372 678 613 575 ...

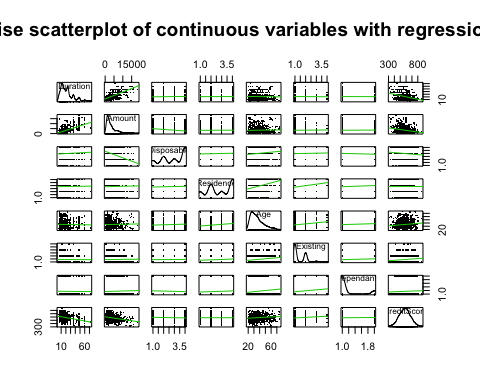
str(test)

## 'data.frame': 200 obs. of 21 variables:  
## $ Status : Factor w/ 4 levels "Large","Negative",..: 2 3 4 3 2 2 4 3 4 4 ...  
## $ Duration : int 24 12 12 9 6 10 18 48 9 27 ...  
## $ History : Factor w/ 5 levels "A","B","C","D",..: 4 3 3 5 3 5 3 5 3 4 ...  
## $ Purpose : Factor w/ 10 levels "Business","Domestic",..: 5 8 8 5 8 5 5 3 8 10 ...  
## $ Amount : int 4870 3059 1567 2134 2647 2241 5866 6110 458 5965 ...  
## $ Savings : Factor w/ 5 levels "Large","Low",..: 2 5 2 2 1 2 3 2 2 2 ...  
## $ Employment : Factor w/ 5 levels "Long","Medium",..: 2 1 2 2 2 3 2 2 2 5 ...  
## $ Disposable : int 3 2 1 4 2 1 2 1 4 1 ...  
## $ Personal : Factor w/ 4 levels "F:DivSepMar",..: 4 3 1 4 4 4 4 4 4 4 ...  
## $ OtherParties: Factor w/ 3 levels "Coapp","Guarantor",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ Residence : int 4 4 1 4 3 3 2 3 3 2 ...  
## $ Property : Factor w/ 4 levels "Car","House",..: 3 2 1 1 2 2 1 3 2 1 ...  
## $ Age : int 53 61 22 48 44 48 30 31 24 30 ...  
## $ Plans : Factor w/ 3 levels "Bank","None",..: 2 2 2 2 2 2 2 1 2 2 ...  
## $ Housing : Factor w/ 3 levels "Own","Rent","RentFree": 3 1 1 1 2 2 1 3 1 1 ...  
## $ Existing : int 2 1 1 3 1 2 2 1 1 2 ...  
## $ Job : Factor w/ 4 levels "Management:Self",..: 2 4 2 2 2 4 2 2 2 1 ...  
## $ Dependants : int 2 1 1 1 2 2 1 1 1 1 ...  
## $ Telephone : Factor w/ 2 levels "No","Yes": 1 1 2 2 1 1 2 2 1 2 ...  
## $ Foreign : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 2 2 ...  
## $ CreditScore : num 613 659 585 640 605 ...

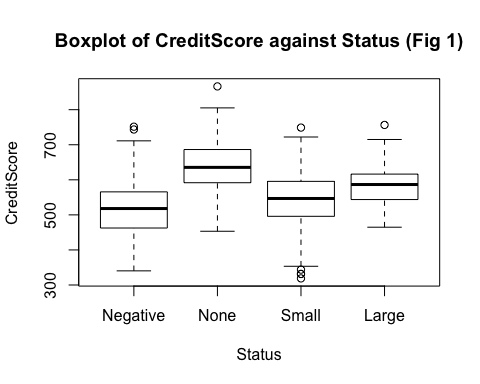
The read test and train data match the explanation of the variables which are illustrated in the coursework sheet.

## Q1a

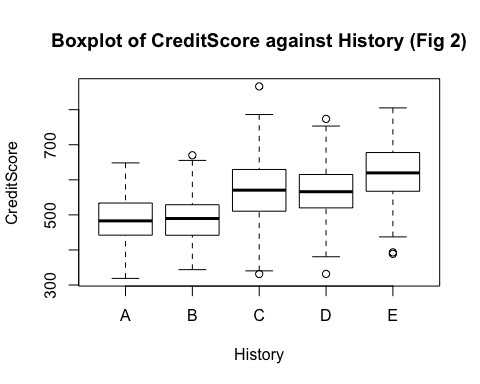
train1 <- train[c(2, 5, 8, 11, 13, 16, 18, 21)]   
scatterplotMatrix(train1[,1:8], pch = ".", smoother = FALSE, main = "Pairwise scatterplot of continuous variables with regression line")



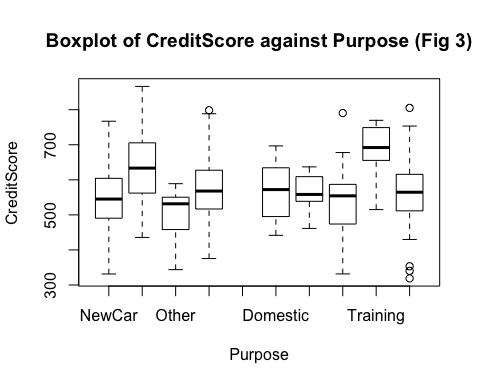
#Whilst observing the affect of the CreditScore individually against the variables, from the pairwise scatter plot we can see a obvious strong negative correlation between Duration and again with Amount. There is a weaker correlation when considering Existing and even weaker correlation when looking at Age, albeit both being a positive correlation. Disposable has an even weaker negative correlation to those mentioned so far. Variables named Residence and Dependants have very little correlation that it is not evident through the plot.  
  
Status1 <- factor(train$Status, levels = c("Negative", "None", "Small", "Large"))  
History1 <- factor(train$History, levels = c("A", "B", "C", "D", "E"))  
Purpose1 <- factor(train$Purpose, levels = c("NewCar", "UsedCar", "Other", "Furniture", "Televison", "Domestic", "Repairs", "Education", "Training", "Business"))  
Savings1 <- factor(train$Savings, levels = c("Unknown", "Low", "Medium", "Large", "VeryLarge"))  
Employment1 <- factor(train$Employment, levels = c("Unemployed", "Short", "Mdeium", "Long", "VeryLong"))  
Personal1 <- factor(train$Personal, levels = c("M:DivSepMar", "F:DivSepMar", "M:Single", "F:Single"))  
OtherParties1 <- factor(train$OtherParties, levels = c("None", "Coapp", "Guarantor"))  
Property1 <- factor(train$Property, levels = c("House", "Savings", "Car", "None"))  
Plans1 <- factor(train$Plans, levels = c("Bank", "Stores", "None"))  
Housing1 <- factor(train$Housing, levels = c("Rent", "Own", "RentFree"))  
Telephone1 <- factor(train$Telephone, levels = c("Yes", "No"))  
Foreign1 <- factor(train$Foreign, levels = c("Yes", "No"))  
  
plot(Status1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Status (Fig 1)", xlab = "Status", ylab = "CreditScore")



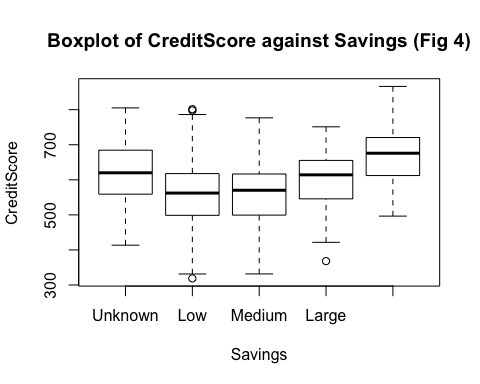
#Figure 1 shows that there is some slight positive correlation between CreditScore and Status, particularly if we temporarily ignore 'None'. This implies this factor should be considered in the model.  
  
plot(History1, train$CreditScore, data = train, main = "Boxplot of CreditScore against History (Fig 2)", xlab = "History", ylab = "CreditScore")



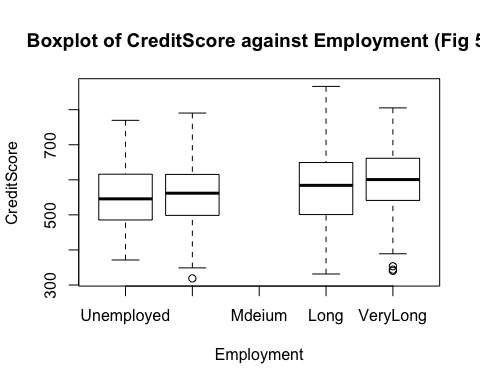
#Figure 2 shows that there is positive correlation between CreditScore and History. This implies this factor should be considered in the model.  
  
plot(Purpose1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Purpose (Fig 3)", xlab = "Purpose", ylab = "CreditScore")



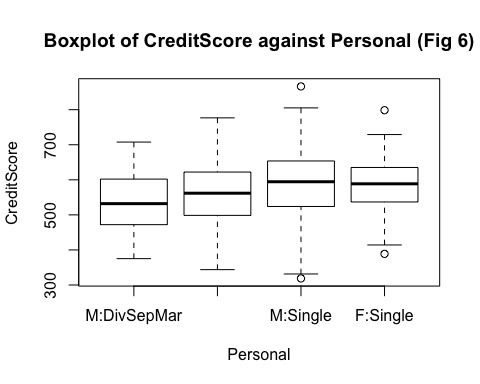
#Figure 3 shows that the data is sparsely distributed, but the levels are uncorrelated so we should consider this variable.  
  
plot(Savings1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Savings (Fig 4)", xlab = "Savings", ylab = "CreditScore")



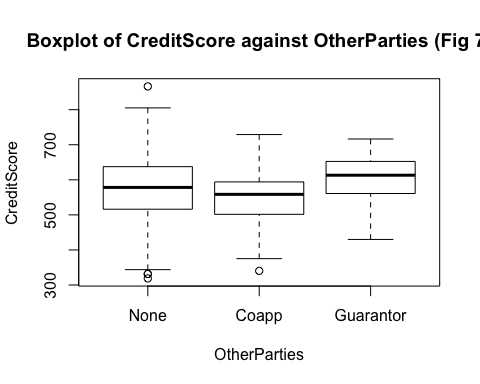
#There is a positive correlation between 'Medium', 'Large', and 'VeryLarge' so these three levels should be considered or some variation of them since there are three distinctive levels.  
  
plot(Employment1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Employment (Fig 5)", xlab = "Employment", ylab = "CreditScore")



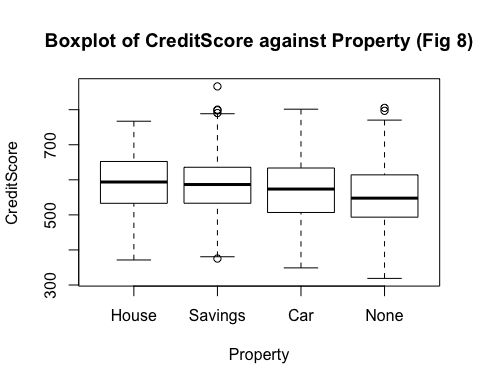
#Employement does not appaear to have a significant affect on CreditScore from the boxplot so we will investigate this variable.  
  
plot(Personal1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Personal (Fig 6)", xlab = "Personal", ylab = "CreditScore")



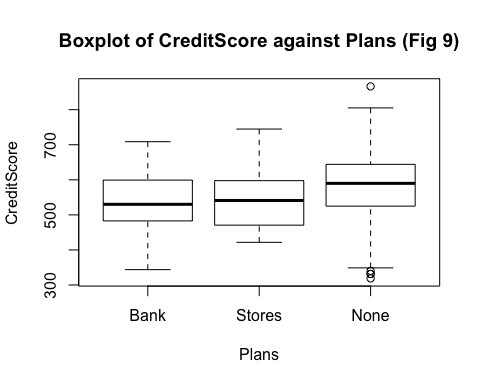
#There are three significant levels in this variable so this entire variable should be considered for the fonal model.  
  
plot(OtherParties1, train$CreditScore, data = train, main = "Boxplot of CreditScore against OtherParties (Fig 7)", xlab = "OtherParties", ylab = "CreditScore")



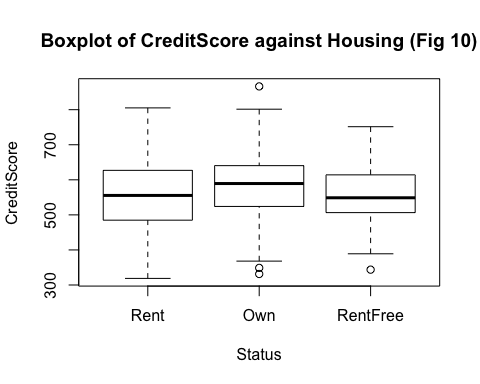
#OtherParties has very similar levels but the medium are slightly different so this variable will be looked at very carefully.  
  
plot(Property1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Property (Fig 8)", xlab = "Property", ylab = "CreditScore")



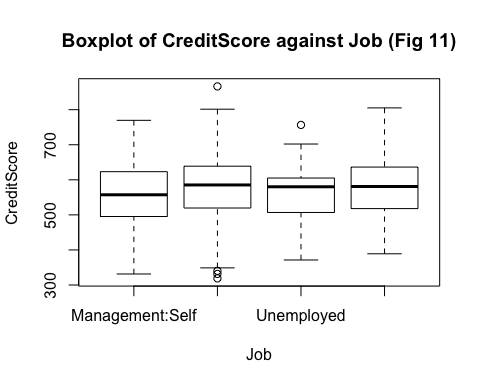
#The levels of Property are very similar and so we may be eble to exclude this variable.  
  
plot(Plans1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Plans (Fig 9)", xlab = "Plans", ylab = "CreditScore")



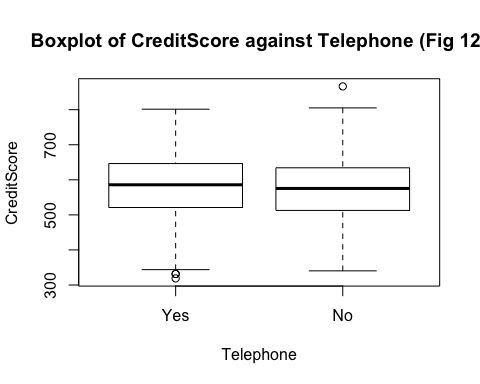
#Plans appears to have a significant affect with levels "Stores' and 'None' so this should be considered in the final model.  
  
plot(Housing1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Housing (Fig 10)", xlab = "Status", ylab = "CreditScore")



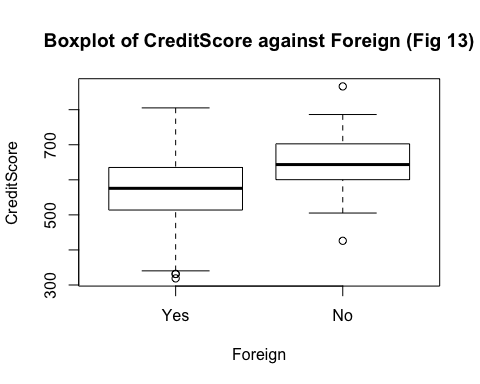
#Housing also appears to have a steady affect throught the levels, implying little correlation between the two variables.  
  
plot(train$Job, train$CreditScore, data = train, main = "Boxplot of CreditScore against Job (Fig 11)", xlab = "Job", ylab = "CreditScore")



#Job also does not appear to have an impat on CreditScore and so, we will investigate whether to include this.  
  
plot(Telephone1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Telephone (Fig 12)", xlab = "Telephone", ylab = "CreditScore")



#Telephone does not appear to have a significant impact on CreditScore so we will investigate this.  
  
plot(Foreign1, train$CreditScore, data = train, main = "Boxplot of CreditScore against Foreign (Fig 13)", xlab = "Foreign", ylab = "CreditScore")



#Foreign appears to have a significant affect since there is a difference between the levels.

To observe the affect of CreditScore against individual variables from the pairwise scatter plot we can see a obvious strong negative correlation between Duration and again with Amount. There is a weaker correlation when considering Existing and even weaker correlation when looking at Age, albeit both being a positive correlation. Disposable has an even weaker negative correlation to those mentioned so far. Variables named Residence and Dependants have very little correlation that it is not evident through the plot. The plot also shows the distribution of CreditScore which follows a Normal Distribution. This will be of use and relevance later when choosing models to see if the distribution has remained in tact.

# FULL MODEL

Full\_Model <- lm(CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Existing + Job + Dependants + Telephone + Foreign, data = train)  
  
predictions\_FULL <- predict(Full\_Model, newdata=select(test, -CreditScore))  
MSE\_Full\_Model <- mean((predictions\_FULL - select(test, CreditScore))^2)  
  
MSE\_Full\_Model

## [1] 2460.575

AIC(Full\_Model)

## [1] 8542.856

### SUMMARY

### Model\_Full <- lm(CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Existing + Job + Dependants + Telephone + Foreign)

has MSE = 2460.575 has AIC = 8542.856

# STEPWISE REGRESSION

# Beginning with Full Model

fit\_Full <- lm(CreditScore ~ . , data = train)  
fit\_step\_Full <- step(fit\_Full)

The output is hidden since there is a lot. ##Model\_Stepwise\_FullModel <- lm(CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign)

# STEPWISE REGRESSION

# Beginning with NullModel

fit\_Null <- lm(CreditScore ~ 1 , data = train)  
fit\_step\_Null <- step(fit\_Null, scope = CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Existing + Job + Dependants + Telephone + Foreign)

The output is hidden since there is a lot. ##Model\_Stepwise\_NullModel <- lm(CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age)

# MSE of Model\_Stepwise\_FullModel and Model\_Stepwise\_NullModel

predictions\_Full <- predict(fit\_step\_Full, newdata=select(test, -CreditScore))  
mse\_step\_Full <- mean((predictions\_Full - select(test, CreditScore))^2)  
  
predictions\_Null <- predict(fit\_step\_Null, newdata=select(test, -CreditScore))  
mse\_step\_Null <- mean((predictions\_Full - select(test, CreditScore))^2)  
  
mse\_step\_Full

## [1] 2431.625

mse\_step\_Null

## [1] 2431.625

Either starting from the Null Model or the Full Model, both conclude to the same model and not surprisingly, give the same mse of 2431.625. We will call this model Model\_Stepwise and directly below is an anova against Full\_Model to find the F-statistic which is defined earlier.

# Anova between Model\_Stepwise and Full\_Model

Model\_Stepwise <- lm (CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age, data = train)  
AIC(Model\_Stepwise)

## [1] 8527.637

anova(Model\_Stepwise, Full\_Model)

## Analysis of Variance Table  
##   
## Model 1: CreditScore ~ Status + Duration + Purpose + History + Savings +   
## Personal + Foreign + Plans + OtherParties + Disposable +   
## Amount + Housing + Telephone + Age  
## Model 2: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Employment + Disposable + Personal + OtherParties +   
## Residence + Property + Age + Plans + Housing + Existing +   
## Job + Dependants + Telephone + Foreign  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 764 1818970   
## 2 751 1794621 13 24349 0.7838 0.6779

### SUMMARY

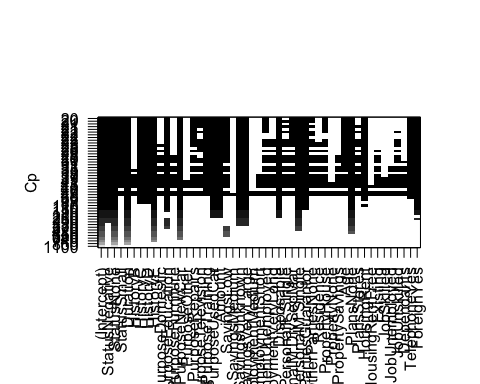
### Model\_Stepwise <- lm (CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age)

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

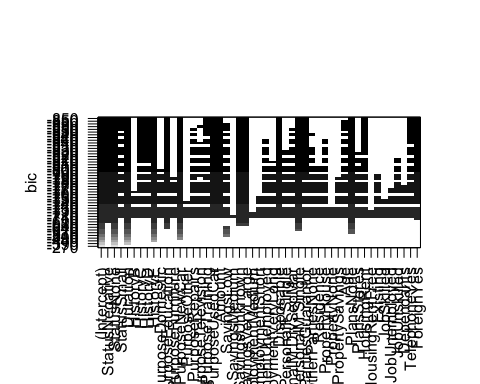
# BEST SUBSETS REGRESSION

# Cp and BIC

a <- regsubsets(train$CreditScore ~ . , data = train, nvmax = 48)  
summary.out <- summary(a)  
summary.out  
summary.out$cp  
plot( a , scale = 'Cp')



plot( a , scale = 'bic')



# Cp Models

The two models from this method are directly below with a further explanation.

Model\_BestSubsetRegression\_Cp\_1: CreditScore ~ Status (Negative, None, Small) + Duration + History (C, D, E) + Purpose (Domestic, Education, NewCar, Repairs, Television, Training, UsedCar) + Amount + Savings (Low, Unknown, VeryLarge) + Employment (Unemployed, VeryLong) + Disposable + Personal (F:Single, M:DivSepMar, M:Single) + OtherParties (Guarantor, None) + Residence + Property (House) + Age + Plans (None, Stores) + Housing (Rent) + Job (Unskilled) + Telephone (Yes) + Foreign (Yes)

Model\_BestSubsetRegression\_Cp\_2: CreditScore ~ Status (Negative, None, Small) + Duration + History (C, D, E) + Purpose (Education, NewCar, Television, Training, UsedCar) + Amount + Savings (Low, Unknown, VeryLarge) + Disposable + Personal (M:Single) + OtherParties (Guarantor) + Age + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

With levels ignored

## Model\_BestSubsetRegression\_Cp\_1.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Job + Telephone + Foreign

## Model\_BestSubsetRegression\_Cp\_2.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign

The best reduced model is when Cp = 24.96151 since this is the first stage in which Cp is approximately p, where p = 25 in this case. At this p value, the test shows 2 models. The two models which achieve this criteria are Model\_BestSubsetRegression\_Cp\_1 and Model\_BestSubsetRegression\_Cp\_2. With the levels being ignored, it is clear that the two models are different so we will determine which of the two to use using four types of statistics which are calculated below.

# Cp Model

Model\_BestSubsetRegression\_Cp\_1 <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Job + Telephone + Foreign, data = train)  
  
Model\_BestSubsetRegression\_Cp\_2 <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign, data = train)  
  
predictions\_Model\_BestSubsetRegression\_Cp\_1 <- predict(Model\_BestSubsetRegression\_Cp\_1, newdata=select(test, -CreditScore))  
mse\_Model\_BestSubsetRegression\_Cp\_1 <- mean((predictions\_Model\_BestSubsetRegression\_Cp\_1 - select(test, CreditScore))^2)  
  
predictions\_Model\_BestSubsetRegression\_Cp\_2 <- predict(Model\_BestSubsetRegression\_Cp\_2, newdata=select(test, -CreditScore))  
mse\_Model\_BestSubsetRegression\_Cp\_2 <- mean((predictions\_Model\_BestSubsetRegression\_Cp\_2 - select(test, CreditScore))^2)  
  
mse\_Model\_BestSubsetRegression\_Cp\_1

## [1] 2449.475

AIC(Model\_BestSubsetRegression\_Cp\_1)

## [1] 8539.947

anova(Model\_BestSubsetRegression\_Cp\_1, Full\_Model)

## Analysis of Variance Table  
##   
## Model 1: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Employment + Disposable + Personal + OtherParties +   
## Residence + Property + Age + Plans + Housing + Job + Telephone +   
## Foreign  
## Model 2: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Employment + Disposable + Personal + OtherParties +   
## Residence + Property + Age + Plans + Housing + Existing +   
## Job + Dependants + Telephone + Foreign  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 753 1797071   
## 2 751 1794621 2 2450.2 0.5127 0.5991

mse\_Model\_BestSubsetRegression\_Cp\_2

## [1] 2431.625

AIC(Model\_BestSubsetRegression\_Cp\_2)

## [1] 8527.637

anova(Model\_BestSubsetRegression\_Cp\_2, Full\_Model)

## Analysis of Variance Table  
##   
## Model 1: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Disposable + Personal + OtherParties + Age + Plans +   
## Housing + Telephone + Foreign  
## Model 2: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Employment + Disposable + Personal + OtherParties +   
## Residence + Property + Age + Plans + Housing + Existing +   
## Job + Dependants + Telephone + Foreign  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 764 1818970   
## 2 751 1794621 13 24349 0.7838 0.6779

## Model\_BestSubsetRegression\_Cp\_1.1 <- CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Job + Telephone + Foreign

has MSE 2449.475 has AIC = 8539.947 has F-statistic against Full\_Model = 0.5127

## Model\_BestSubsetRegression\_Cp\_2.1 <- CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

Since Model\_BestSubsetRegression\_Cp\_2 is nested within Model\_BestSubsetRegression\_Cp\_1 we can determine which model to use using the F-statistic. The F-statistic for Model\_BestSubsetRegression\_Cp\_1.1 is 0.5127 while the value for Model\_BestSubsetRegression\_Cp\_2.1 is 0.7838, which is closer to 1 than the former of the two models. This would suggest to keep Model\_BestSubsetRegression\_Cp\_2. The AIC for the first model is also slightly larger by approximately 12, providing more evidence to accept the second model. So both pieces of data suggest to use the second model. This concludes the following model for the Best Subsets Regression Method for Cp

### SUMMARY

### Model\_Cp <- CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

# BIC Models

We want to minimise the value of BIC so from the BIC plot there are 6 models each holding the smallest BIC value of -850.

Model\_BestSubsetsRegression\_BIC\_1.0: CreditScore ~ Status (Negative, None, Small) + Duration + History (C ,D, E) + Purpose (Education, NewCar, Training, UsedCar) + Amount + Savings (Unknown, VeryLarge) + Disposable + Personal (M:Single) + OtherParties (Guarantor) + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

Model\_BestSubsetsRegression\_BIC\_2.0: CreditScore ~ Status (Negative, None, Small) + Duration + History (C ,D, E) + Purpose (Education, NewCar, Training, UsedCar) + Amount + Savings (Unknown, VeryLarge) + Disposable + Personal (M:Single) + OtherParties (Guarantor) + Age + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

Model\_BestSubsetsRegression\_BIC\_3.0: CreditScore ~ Status (Negative, None, Small) + Duration + History (C ,D, E) + Purpose (Education, NewCar, Training, UsedCar) + Amount + Savings (Low, Unknown, VeryLarge) + Disposable + Personal (M:Single) + OtherParties (Guarantor)+ Age + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

Model\_BestSubsetsRegression\_BIC\_4.0: CreditScore ~ Status (Negative, None, Small) + Duration + History (C ,D, E) + Purpose (Education, NewCar, Television, Training, UsedCar) + Amount + Savings (Low, Unknown, VeryLarge) + Disposable + Personal (M:Single) + OtherParties (Guarantor) + Age + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

Model\_BestSubsetsRegression\_BIC\_5.0: CreditScore ~ Status (Negative, None, Small) + Duration + History (C ,D, E) + Purpose (Education, NewCar, Repairs, Training, UsedCar) + Amount + Savings (Low, Unknown, VeryLarge) + Disposable + Personal (M:SepDivMar, M:Single) + OtherParties (Guarantor) + Age + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

Model\_BestSubsetsRegression\_BIC\_6.0: CreditScore ~ Status (Negative, None) + Duration + History (C ,D, E) + Purpose (Education, NewCar, Training, UsedCar) + Amount + Savings (Unknown, VeryLarge) + Disposable + Personal (M:Single) + OtherParties (Guarantor) + Plans (None) + Housing (Rent) + Telephone (Yes) + Foreign (Yes)

With levels ignored ##Model\_BestSubsetsRegression\_BIC\_1.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Plans + Housing + Telephone + Foreign

## Model\_BestSubsetsRegression\_BIC\_2.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign

Model\_BestSubsetsRegression\_BIC\_3.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties Age + Plans + Housing + Telephone + Foreign

Model\_BestSubsetsRegression\_BIC\_4.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign

Model\_BestSubsetsRegression\_BIC\_5.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign

Model\_BestSubsetsRegression\_BIC\_6.1: CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Plans + Housing + Telephone + Foreign

As can be seen by the models above where the levels are omitted, the set of models Model\_BestSubsetsRegression\_BIC\_2.1, Model\_BestSubsetsRegression\_BIC\_3.1, Model\_BestSubsetsRegression\_BIC\_4.1, and Model\_BestSubsetsRegression\_BIC\_5.1 and the second set of models Model\_BestSubsetsRegression\_BIC\_1.1 and Model\_BestSubsetsRegression\_BIC\_6.1 are the only the same when the different levels are ignored and this is how we are viewing the variables because for simplicity, we are not omitting levels from a variable. Therefore, there are only 2 different models which are between Model\_BestSubsetsRegression\_BIC\_1.1 and Model\_BestSubsetsRegression\_BIC\_2.1 and the only difference between the two models is that the model named Model\_BestSubsetsRegression\_BIC\_2.1 includes the variable Age. We will now find the MSE to see whether we should include Age in our model for this method.

# BIC Model with mse of Model\_BestSubsetsRegression\_BIC\_1 and Model\_BestSubsetsRegression\_BIC\_2

Model\_BestSubsetRegression\_BIC\_1 <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Plans + Housing + Telephone + Foreign, data = train)  
  
Model\_BestSubsetRegression\_BIC\_2 <- lm(CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign, data = train)  
  
predictions <- predict(Model\_BestSubsetRegression\_BIC\_1, newdata=select(test, -CreditScore))  
mse\_Model\_BestSubsetRegression\_BIC\_1 <- mean((predictions - select(test, CreditScore))^2)  
  
predictions <- predict(Model\_BestSubsetRegression\_BIC\_2, newdata=select(test, -CreditScore))  
mse\_Model\_BestSubsetRegression\_BIC\_2 <- mean((predictions - select(test, CreditScore))^2)  
  
mse\_Model\_BestSubsetRegression\_BIC\_1

## [1] 2426.406

AIC(Model\_BestSubsetRegression\_BIC\_1)

## [1] 8533.136

mse\_Model\_BestSubsetRegression\_BIC\_2

## [1] 2431.625

AIC(Model\_BestSubsetRegression\_BIC\_2)

## [1] 8527.637

anova(Model\_BestSubsetRegression\_BIC\_1, Model\_BestSubsetRegression\_BIC\_2)

## Analysis of Variance Table  
##   
## Model 1: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Disposable + Personal + OtherParties + Plans +   
## Housing + Telephone + Foreign  
## Model 2: CreditScore ~ Status + Duration + History + Purpose + Amount +   
## Savings + Disposable + Personal + OtherParties + Age + Plans +   
## Housing + Telephone + Foreign  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 765 1836101   
## 2 764 1818970 1 17132 7.1956 0.007466 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Model\_BestSubsetRegression\_BIC\_1 <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Plans + Housing + Telephone + Foreign)

has MSE 2426.406 has AIC = 8533.136 has F-statistic against Full\_Model = 1.2399

## Model\_BestSubsetRegression\_BIC\_2 <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign)

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

p-value between the two BIC models 0.007466

The significance between the two model is high so this means that Age is a significant variable which should be include. This would suggest using the following model also because the mse and AIC value is smaller, although just slightly, but this is a strong criteria. The F-statistics of the latter model is closer to 1 than that of the former This concludes the following model for the Best Subsets Regression Method for BIC.

### SUMMARY

### Model\_BIC <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign)

has MSE 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

# RIDGE REGRESSION

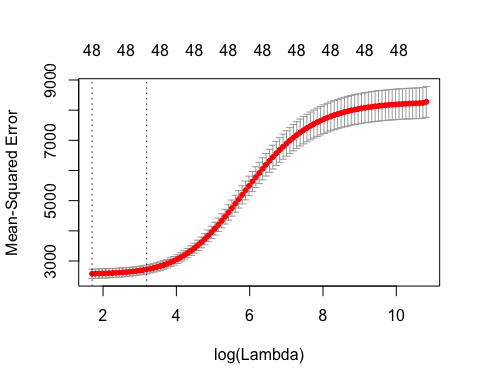
fit.1 = lm(CreditScore ~ ., data = train)  
x.1 = model.matrix(fit.1)  
y.1 = select(train, CreditScore)  
y.1 = as.matrix(y.1)  
ridge = glmnet(x.1, y.1, alpha = 0)  
ridge.cv = cv.glmnet(x.1, y.1, alpha = 0)  
ridge.cv$lambda.min

## [1] 5.477582

ridge.cv$lambda.1se

## [1] 24.26909

plot(ridge.cv)



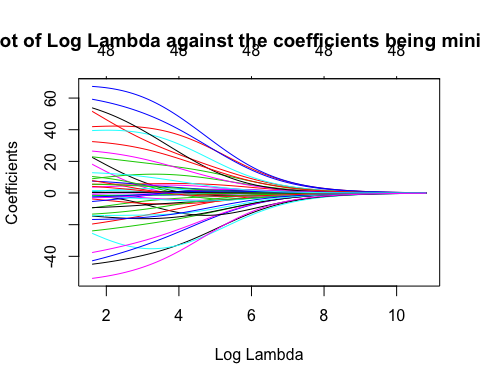
fit.xnew = lm(CreditScore ~ ., data = test)  
fit.xnew = model.matrix(fit.xnew)  
  
ridge.prediction1. = predict(ridge.cv, fit.xnew, s = "lambda.1se")  
ridge.prediction2. = predict(ridge.cv, fit.xnew, s = "lambda.min")  
mse\_ridge1 <- mean((ridge.prediction1. - select(test, CreditScore))^2)  
mse\_ridge2 <- mean((ridge.prediction2. - select(test, CreditScore))^2)  
  
mse\_ridge1

## [1] 2699.884

mse\_ridge2

## [1] 2531.103

plot(ridge, xvar = 'lambda', main = "Plot of Log Lambda against the coefficients being minimised")

 From the plot, it is evident that as lambda increases, the mean square error increases. The largest value of lambda that gives a cross-validation error within 1 standard deviation of the minimum is 24.26909 and the value of lambda that minimises the cross-validation error is 5.477582.

### SUMMARY

# The Ridge Regression model suggests using lambda.min = 5.477582 as our lambda value since this value has MSE 2531.103

In Summary,

### Model\_Full <- lm(CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Employment + Disposable + Personal + OtherParties + Residence + Property + Age + Plans + Housing + Existing + Job + Dependants + Telephone + Foreign)

has MSE = 2460.575 has AIC = 8542.856

### Model\_Stepwise <- lm (CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age)

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

### Model\_Cp <- CreditScore ~ lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign)

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

### Model\_BIC <- lm (CreditScore ~ Status + Duration + History + Purpose + Amount + Savings + Disposable + Personal + OtherParties + Age + Plans + Housing + Telephone + Foreign)

has MSE = 2431.625 has AIC = 8527.637 has F-statistic against Full\_Model = 0.7838

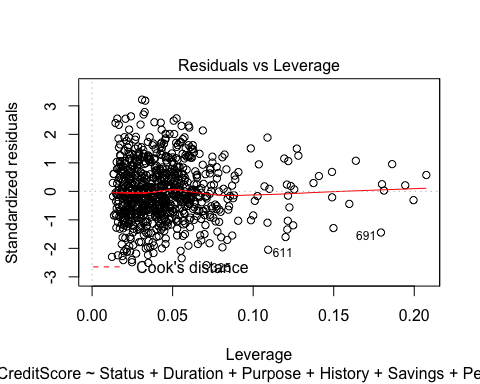
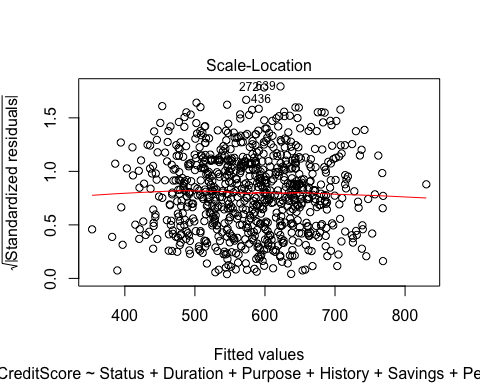
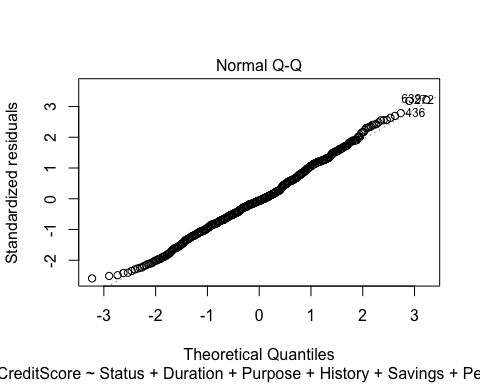
### Model\_Ridge

The Ridge Regression model suggests using lambda.min = 5.477582 as our lambda value. has MSE 2531.103

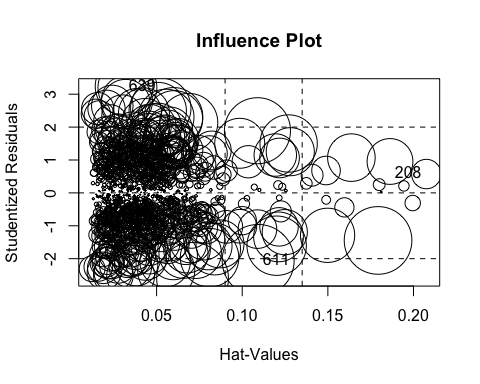
Using the MSE values we can clearly see that the Ridge Regression Model does not give us a good model since it has the highest mse of all the models so we can reject this model, especially since as this method does not remove insignificant factor/s, but instead, it adjusts the coefficients for all the variables in the model to compensate for which variables have the greatest affect on CreditScore and which do not. Model\_BIC, Model\_Cp, and Model\_Stepwise are all the same model so all three methods give the same model which is not necessarily common, we will call this Model\_step\_BIC\_Cp. Now, we are left to compare Model\_Full and Model\_step\_BIC\_Cp and we can see clearly that the mse and AIC are lower in Model\_step\_BIC\_Cp, which is ideal so we can conclude that Model\_step\_BIC\_Cp is the best less complex model. While accepting this model, we accept the limitations of the the Stepwise Regression method and the Best Subsets Regression. The Stepwise Regression method does not exhaust all possible models and they include variables which just happen to explain the data by chance and this decreases its effectiveness with predictions. A notable point to mention about not using the model from the Ridge Regression is that while the Best Subsets Method produces a method which can be easily interpreted and has a lower prediction error, there tends to be a higher variance as a result of keeping the entirety of certain variables whereas for Ridge Regression the variance does not suffer as much due to its continuous nature of the method and its shrinking technique.

The Best Subsets Regression method also has some limitations, which is that it is not supposed to be used for many variables and the running time of this method increases exponentially with a larger number of variables, however, since the stepwise can deal with many variables and we are given the same model from all methods, this in return, strengthens the argument to use this model despite the limitations of the methods discussed. This justifies and concludes the reasons to use the parsimonious model that is ####Model\_Best <- lm (CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age)

Model\_Best <- lm (CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age, data = train)  
  
plot(Model\_Best)

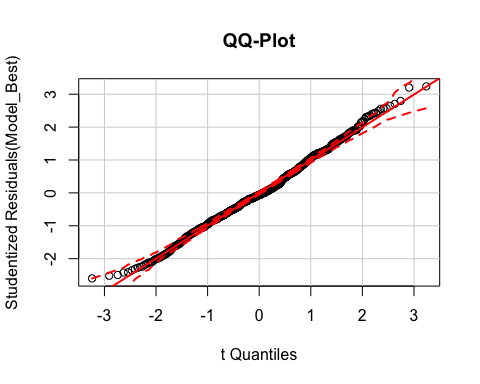


influencePlot(Model\_Best, main = "Influence Plot")

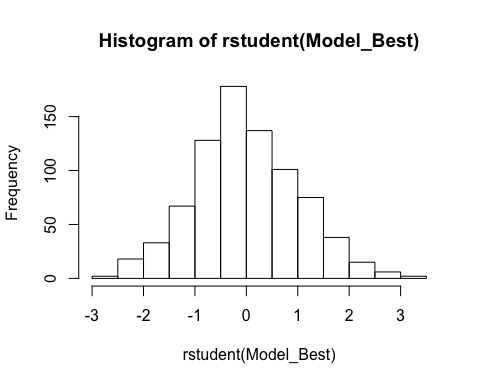


## StudRes Hat CookD  
## 208 0.572889 0.20747923 0.002388826  
## 611 -2.054525 0.10930796 0.014329049  
## 639 3.237357 0.03102072 0.009205773

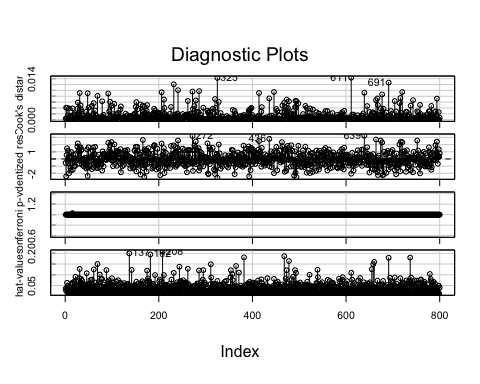
qqPlot(Model\_Best, main = "QQ-Plot")



hist(rstudent(Model\_Best))



influenceIndexPlot(Model\_Best, id.n = 3)



rstandard(Model\_Best)

## 1 2 3 4 5   
## -0.552415794 -2.406171351 0.551215600 1.474725643 -0.187688871   
## 6 7 8 9 10   
## 1.151429864 -0.579805133 -2.004191778 0.178606701 0.268775413   
## 11 12 13 14 15   
## -1.230231791 -1.616017575 0.956645847 0.245311838 -0.616921500   
## 16 17 18 19 20   
## -1.501621507 0.605244734 0.013186561 -0.968291145 -0.317339890   
## 21 22 23 24 25   
## -0.995267521 0.675588674 -0.182458214 -0.136193140 -0.160707684   
## 26 27 28 29 30   
## 1.714302015 2.163544983 1.085557875 1.259506205 -0.643167006   
## 31 32 33 34 35   
## 1.500004212 -0.286778975 1.621775181 1.195529164 1.266716045   
## 36 37 38 39 40   
## -0.824293300 -0.070772909 -0.069148333 -0.505691274 -1.051342253   
## 41 42 43 44 45   
## -0.272158276 0.176819186 -0.850972877 -0.164813934 -1.313706773   
## 46 47 48 49 50   
## 0.174956608 1.326734892 -2.348012171 -1.205382412 1.624150364   
## 51 52 53 54 55   
## -0.855390997 0.164472466 -0.883572754 0.238592973 -0.281543531   
## 56 57 58 59 60   
## 0.294551143 -1.412500477 0.531539306 1.142211235 1.574236845   
## 61 62 63 64 65   
## -1.192910922 -0.195035719 0.688641047 1.679634487 0.041279945   
## 66 67 68 69 70   
## -0.560068656 -1.221076429 -0.600150379 -1.288924202 -0.790446466   
## 71 72 73 74 75   
## -0.310350592 0.584715169 -0.286062478 -0.333746114 0.043472706   
## 76 77 78 79 80   
## -0.721411788 -0.639577123 0.337221570 -1.611692363 1.058150008   
## 81 82 83 84 85   
## 0.697746009 -1.029437003 0.705696627 0.576182287 0.606469563   
## 86 87 88 89 90   
## 0.026629424 -0.049837702 -0.151470159 0.358375070 0.619794761   
## 91 92 93 94 95   
## 0.067518856 2.238114762 0.006947323 1.260524859 0.457896799   
## 96 97 98 99 100   
## 1.896378657 1.012152180 0.826873584 2.342919945 -0.264878938   
## 101 102 103 104 105   
## -0.154851508 -1.001227431 -0.246601625 0.534181037 2.015660827   
## 106 107 108 109 110   
## 0.010324051 0.692989509 0.642394068 0.933752802 -0.950612809   
## 111 112 113 114 115   
## 1.163293869 0.986525884 1.010417384 0.061161450 -0.261772114   
## 116 117 118 119 120   
## 0.122922986 -0.597317279 1.361672685 -1.846638526 0.580551091   
## 121 122 123 124 125   
## 0.835338481 -1.023764536 -0.347316892 -0.141419056 -0.841993877   
## 126 127 128 129 130   
## -0.289830564 1.139308367 0.524413096 1.167015497 0.042978533   
## 131 132 133 134 135   
## 0.443859157 0.616935770 1.228064300 -0.260798999 -0.685871896   
## 136 137 138 139 140   
## 0.368794078 -0.308958068 -0.471192096 -1.316879024 0.222446208   
## 141 142 143 144 145   
## -0.003925318 -0.562529314 -0.946963141 -0.006404659 -1.312880889   
## 146 147 148 149 150   
## -1.090608593 -0.033754154 1.191128663 1.105145574 -0.094388575   
## 151 152 153 154 155   
## 1.703057509 -1.723620734 -0.727975109 1.669790789 1.131849105   
## 156 157 158 159 160   
## -0.779524341 -0.428214133 0.196345421 -0.949776064 0.609650477   
## 161 162 163 164 165   
## -2.164754428 0.891188918 -0.313522947 -0.111834505 -0.836032231   
## 166 167 168 169 170   
## 2.631971453 -1.257217055 0.361861974 -2.487159878 1.337411915   
## 171 172 173 174 175   
## -0.879391858 1.200308819 -0.110165982 -1.741646794 1.151228242   
## 176 177 178 179 180   
## -0.689665855 0.772599629 1.501470930 0.822390458 -0.013863948   
## 181 182 183 184 185   
## -0.518759921 0.211291551 0.096635435 -0.204302061 -0.528259159   
## 186 187 188 189 190   
## -1.335165563 0.706067961 -1.126447015 1.774516558 1.899236653   
## 191 192 193 194 195   
## 1.892847962 -1.013605997 0.369013873 -0.126406658 -1.457507011   
## 196 197 198 199 200   
## -0.054502696 1.412023288 -1.135539853 -0.077271976 -1.322268617   
## 201 202 203 204 205   
## 0.264597877 -0.624329924 -2.243926210 -0.468055979 -0.666785902   
## 206 207 208 209 210   
## 2.135763324 0.005574239 0.573141081 -0.271754529 1.030157863   
## 211 212 213 214 215   
## 1.506743115 -1.959283046 -1.201357454 0.125883769 0.848470902   
## 216 217 218 219 220   
## -0.776763889 -1.261382813 -0.620670971 -0.179239846 -0.505311183   
## 221 222 223 224 225   
## 0.120746617 0.001587260 0.469987703 -1.268540290 -0.708574381   
## 226 227 228 229 230   
## 1.809491731 -0.522114903 0.550072398 0.677248977 0.069474182   
## 231 232 233 234 235   
## -0.837880313 1.886383891 0.768976719 0.857001065 1.630867842   
## 236 237 238 239 240   
## -1.088183510 0.594059574 -0.748352497 -0.872132886 1.548519525   
## 241 242 243 244 245   
## 2.561073382 1.215701544 -0.608258292 0.290011482 0.703220497   
## 246 247 248 249 250   
## -0.191611055 -0.381785269 0.008940869 -0.203741815 0.677232282   
## 251 252 253 254 255   
## 1.451331895 -0.601527903 -0.780883336 1.233605862 0.642048593   
## 256 257 258 259 260   
## -0.997569104 -0.245744150 0.240615981 -0.921556167 -1.854759322   
## 261 262 263 264 265   
## -0.052629662 1.251507222 -0.758110973 1.192332177 1.263504650   
## 266 267 268 269 270   
## 0.135961271 1.208741311 -0.483307106 -0.415842226 -0.501483669   
## 271 272 273 274 275   
## -0.661425708 3.182458554 -1.653662340 -0.642826519 0.119065746   
## 276 277 278 279 280   
## 1.861615847 1.212514594 1.645354978 2.425193866 -0.730604607   
## 281 282 283 284 285   
## 0.098698366 1.233660061 -0.283098020 -0.565170388 1.494992811   
## 286 287 288 289 290   
## -1.603559738 1.062084635 -1.222981400 0.121793441 0.558636259   
## 291 292 293 294 295   
## 0.823273529 -0.492399966 0.216449824 0.450967249 -1.032314077   
## 296 297 298 299 300   
## -0.386381188 -0.429930685 0.244761588 -0.040894579 0.048924480   
## 301 302 303 304 305   
## 0.269783204 -0.418288287 -0.522658650 2.167739603 -2.021676736   
## 306 307 308 309 310   
## -2.106070910 0.153862913 1.547053178 -0.718742631 1.316704406   
## 311 312 313 314 315   
## -0.210078420 -0.240600435 0.001747181 -0.308929597 1.558269459   
## 316 317 318 319 320   
## -0.019515544 -0.245044675 -1.457464874 0.792315266 -2.087347298   
## 321 322 323 324 325   
## -1.564451942 -0.134326635 0.211951417 0.719895478 -2.589868329   
## 326 327 328 329 330   
## 0.462916632 -0.326419022 2.011614566 -0.478724038 0.783260821   
## 331 332 333 334 335   
## -0.222430624 -1.349073031 -0.799687917 -0.060008831 -1.055306992   
## 336 337 338 339 340   
## -0.245920535 -0.007972775 -0.154053037 -0.065082539 -0.097408017   
## 341 342 343 344 345   
## -0.383494589 0.527674790 -0.442729510 -0.160105056 -0.967477072   
## 346 347 348 349 350   
## -0.422608425 -0.069790182 -1.194267688 0.467143417 -0.787260141   
## 351 352 353 354 355   
## -0.550420310 0.148825479 -0.137458689 0.237318342 0.422740637   
## 356 357 358 359 360   
## 1.269045876 -0.163755042 1.217066520 -0.571731362 -0.021655799   
## 361 362 363 364 365   
## 0.575604343 -2.110354133 -0.477664161 -1.163806386 0.537617130   
## 366 367 368 369 370   
## -0.043582328 -0.692137187 -0.272697210 0.900919250 0.231135542   
## 371 372 373 374 375   
## 0.087935900 -0.785411481 0.273681517 -0.676273777 -0.941417499   
## 376 377 378 379 380   
## 0.725754187 0.584051855 0.130276736 -1.080517317 -0.831105161   
## 381 382 383 384 385   
## 1.851159361 0.026187470 0.912602546 0.146421048 -0.824637723   
## 386 387 388 389 390   
## -0.647404294 1.420527436 0.076761959 0.075610194 -0.368877556   
## 391 392 393 394 395   
## 1.153582280 1.733685995 -2.413857326 1.317110858 -0.160872525   
## 396 397 398 399 400   
## -1.549775915 1.535959810 1.482788931 -0.731217658 1.045020155   
## 401 402 403 404 405   
## -0.365708162 1.254112792 0.262662006 0.960246522 0.133572896   
## 406 407 408 409 410   
## -0.701732002 0.243762383 -0.181097873 -0.265323002 0.599753094   
## 411 412 413 414 415   
## 0.006654826 0.761265882 1.307315361 2.479335508 -0.313710975   
## 416 417 418 419 420   
## 1.022753783 0.666576277 -0.141226226 -1.072089236 1.540948884   
## 421 422 423 424 425   
## -0.046242219 0.125964947 -0.073365118 -0.451508629 -0.585970849   
## 426 427 428 429 430   
## -0.923490061 0.650409732 -0.569381806 -0.683934055 -0.408085047   
## 431 432 433 434 435   
## 0.472091843 -0.003304858 -0.366542761 -0.105680575 0.150259366   
## 436 437 438 439 440   
## 2.782699036 -1.033319062 0.330701954 0.164798759 1.063056558   
## 441 442 443 444 445   
## -1.134893905 -0.275976124 -0.125234857 0.356527906 0.090582175   
## 446 447 448 449 450   
## -1.966146390 1.073696599 0.222645408 0.066998758 0.050506391   
## 451 452 453 454 455   
## 1.674614216 0.985148646 -1.571729468 0.111991697 0.469361344   
## 456 457 458 459 460   
## 0.107265127 0.426672505 1.137947622 -0.175603902 -0.477378413   
## 461 462 463 464 465   
## -0.045631297 -0.237374374 2.332683165 1.729872800 -0.340837578   
## 466 467 468 469 470   
## 0.776744549 -0.065803794 0.954805682 -1.777406114 1.386210292   
## 471 472 473 474 475   
## 1.290761449 -1.665202684 -1.336944824 0.498436453 1.812575585   
## 476 477 478 479 480   
## 1.511466151 1.310717374 1.072340413 -0.189896092 0.015173013   
## 481 482 483 484 485   
## -0.112562235 -1.057676794 1.900652751 -0.005541142 1.135006595   
## 486 487 488 489 490   
## -0.090108888 -1.032187185 -0.183595023 -2.300079624 -0.469977618   
## 491 492 493 494 495   
## 1.061740532 -1.904771353 -1.101106536 -2.510576055 -0.295932971   
## 496 497 498 499 500   
## -0.373308480 0.054919632 1.832077736 0.790135068 -0.842132588   
## 501 502 503 504 505   
## 0.418069633 -0.440179546 -1.096313961 0.326083303 -1.079937808   
## 506 507 508 509 510   
## -0.574623249 0.558093809 1.195519035 0.532766823 -0.724914704   
## 511 512 513 514 515   
## -1.119497538 -0.410121584 1.712536037 -0.857423233 -0.706672011   
## 516 517 518 519 520   
## -1.897319298 -0.051055264 -0.108829840 -0.840988433 1.124229885   
## 521 522 523 524 525   
## 0.449590083 -0.844760326 -1.960085651 -0.066921496 -0.195224783   
## 526 527 528 529 530   
## 0.782849687 -1.009752083 0.838238763 -1.791081865 0.264457057   
## 531 532 533 534 535   
## -0.673841007 0.283771518 0.161954022 -0.547511525 -0.254048474   
## 536 537 538 539 540   
## -1.633484963 1.179328330 -0.131359688 0.187150873 0.550680269   
## 541 542 543 544 545   
## 0.654264605 -1.463128282 -1.070676985 1.141570657 -0.181270001   
## 546 547 548 549 550   
## -0.793139425 0.864655077 0.875464351 -0.684514262 -0.621996167   
## 551 552 553 554 555   
## 0.610934184 0.485994427 1.274576287 -0.739133964 0.011780694   
## 556 557 558 559 560   
## 1.528093559 -0.977181174 0.157400256 0.598306355 -1.596665847   
## 561 562 563 564 565   
## -0.464428281 0.919196253 1.316721981 0.773382846 -0.625230863   
## 566 567 568 569 570   
## -0.743389159 -1.276070298 -0.447392368 0.633958682 -0.452968829   
## 571 572 573 574 575   
## -1.160739384 0.065630407 0.916003036 0.485079153 -0.169293675   
## 576 577 578 579 580   
## 0.426562460 -0.095217245 -0.235808520 0.563755710 -1.131186694   
## 581 582 583 584 585   
## -1.148742260 -1.296805262 -0.045953827 0.506394370 0.089559787   
## 586 587 588 589 590   
## -0.694020193 1.049194641 -0.718637408 0.662620698 -1.241276013   
## 591 592 593 594 595   
## 1.381440195 -1.056180544 0.413768461 -0.526291607 -1.447932394   
## 596 597 598 599 600   
## 1.572340641 0.580903855 0.019296764 -0.458943774 0.327400812   
## 601 602 603 604 605   
## 0.413491164 0.601864443 -0.551727352 -0.124855316 0.248184178   
## 606 607 608 609 610   
## -0.930741102 -0.603880412 -0.291355316 -0.194891616 -0.033268094   
## 611 612 613 614 615   
## -2.050207543 -0.287734617 0.855805837 -0.817385404 0.062266471   
## 616 617 618 619 620   
## -1.600746877 0.041628234 -0.688520055 -0.458253902 0.122428962   
## 621 622 623 624 625   
## -0.846184834 0.243637564 0.662325865 -0.076880589 1.215157725   
## 626 627 628 629 630   
## -0.093447997 -0.518431380 0.088206839 -0.434753486 -0.523700995   
## 631 632 633 634 635   
## 0.061888856 0.773898276 1.284476261 -0.961345139 -0.143672507   
## 636 637 638 639 640   
## -0.066013179 1.318403704 -0.032113947 3.217456250 -0.242078709   
## 641 642 643 644 645   
## -0.611387932 0.284819677 0.428337670 -0.820938111 1.620043714   
## 646 647 648 649 650   
## 0.706904596 -0.513032538 -0.840874750 0.393269617 -1.891412918   
## 651 652 653 654 655   
## 0.488422767 0.565876349 -0.411386666 -0.666780420 0.068523690   
## 656 657 658 659 660   
## -0.076904747 -1.387343068 0.680930988 0.157678887 -0.440879952   
## 661 662 663 664 665   
## -1.526136343 -1.355610495 -0.139298739 2.694205986 -0.892345775   
## 666 667 668 669 670   
## -0.448747955 -1.820911948 -2.232909042 0.184383788 0.467166725   
## 671 672 673 674 675   
## -0.105694920 -0.793879769 -1.053234815 2.548164647 -0.450845147   
## 676 677 678 679 680   
## -1.367959018 2.297482289 -2.270642504 -0.249849062 -2.182221153   
## 681 682 683 684 685   
## 0.989928843 -0.323228394 -0.792243200 -0.365025258 0.626967281   
## 686 687 688 689 690   
## -1.924249777 -0.257105768 0.850207348 1.113962215 -0.346965370   
## 691 692 693 694 695   
## -1.444334864 -0.547691151 0.170677790 -0.162961796 -0.513407906   
## 696 697 698 699 700   
## -1.185909378 0.958575691 -0.745367861 1.240977649 -0.189509753   
## 701 702 703 704 705   
## -0.013346062 0.158092848 -0.497074218 -1.549120074 -0.085054908   
## 706 707 708 709 710   
## 0.209200722 -2.124739836 0.760435647 -1.170724634 -0.798084498   
## 711 712 713 714 715   
## -0.888083134 -0.369233089 -0.674558070 2.015127551 -0.773174598   
## 716 717 718 719 720   
## -1.705481177 2.398389467 0.863059005 -0.195671424 2.307782766   
## 721 722 723 724 725   
## -0.484328239 1.028891087 -0.252108228 -0.536614012 1.080214942   
## 726 727 728 729 730   
## 0.175840338 -0.594684135 -1.783038573 -0.521945627 1.217916120   
## 731 732 733 734 735   
## 0.471434039 -1.204506928 -0.265144758 0.019875975 -0.906629484   
## 736 737 738 739 740   
## 0.249356370 0.251758513 -1.545950692 0.200261094 0.006532900   
## 741 742 743 744 745   
## 0.124505682 -0.024044948 -1.847918134 0.989479818 -0.083680090   
## 746 747 748 749 750   
## 0.210922679 0.135402763 -0.817526949 -0.173093399 0.056668556   
## 751 752 753 754 755   
## 0.630648262 2.551962245 0.300238849 1.180996703 -0.366078575   
## 756 757 758 759 760   
## -1.220986454 -0.182735490 -2.001199678 -0.925464804 0.641478622   
## 761 762 763 764 765   
## -0.049149582 -0.438857031 0.485672464 -0.113286012 -0.817837803   
## 766 767 768 769 770   
## 0.984065046 -0.509886282 -0.675572895 -1.054128579 0.787836926   
## 771 772 773 774 775   
## -0.818091331 -0.680865231 -1.042046841 -0.529915205 -0.111451774   
## 776 777 778 779 780   
## -1.060033267 1.616852933 0.903284172 0.947042693 0.579779615   
## 781 782 783 784 785   
## -0.718153365 -1.498855755 -1.160008221 0.834284302 -0.229719085   
## 786 787 788 789 790   
## -0.180439801 -0.151841173 -0.088993970 0.578226130 1.216314671   
## 791 792 793 794 795   
## 1.912676251 -0.166022048 0.880650922 2.408738523 0.489760892   
## 796 797 798 799 800   
## -0.672212556 0.375862528 1.330299082 0.296054140 0.759922210

sort(coef(Model\_Best))

## ForeignYes PurposeEducation StatusNegative   
## -58.00264683 -49.91664512 -49.80452577   
## PurposeNewCar StatusSmall PurposeRepairs   
## -44.35569562 -28.57073603 -26.91521625   
## HousingRent PurposeDomestic PersonalM:DivSepMar   
## -18.00822494 -17.39756187 -15.25453924   
## Disposable SavingsLow PurposeOther   
## -14.50083348 -13.03031811 -4.15964574   
## HistoryB PurposeFurniture Duration   
## -3.90477037 -2.55218922 -1.46476796   
## HousingRentFree SavingsMedium Amount   
## -1.02545954 -0.79800347 -0.00622422   
## Age PurposeTelevision PersonalF:Single   
## 0.46131554 4.20604797 8.17187886   
## PlansStores TelephoneYes OtherPartiesNone   
## 10.51652296 12.79232219 13.75456886   
## SavingsUnknown PersonalM:Single PlansNone   
## 25.61133876 28.68487862 32.82941794   
## PurposeUsedCar StatusNone HistoryC   
## 36.25052527 38.39841213 44.02824943   
## HistoryD SavingsVeryLarge OtherPartiesGuarantor   
## 46.31359081 61.99917945 62.29696115   
## PurposeTraining HistoryE (Intercept)   
## 69.40856679 76.86358783 618.87560200

The Residual vs Fitted plot shows a horizontal line and so, there is no need to transform. Scale-Location shows a that the fitted values are very close to the observed data. The trend of the graphs are almost a complete horizontal line which is what we hope from a good model. The Normal QQ-plot shows that Model\_Best fits a Normal Distribution very well, which is our intention because from the scatter plot matrix from earlier we can see the distribution of CreditScore resembles a Normal Distribution and so, the new model has kept this distribution of the data in tact. The QQ-plot is also in sync with Normal QQ-plot but is more clear in demonstrating what the QQ-plot shows. This implies that the removed variables were indeed insignificant, further justifying our claim to remove them, moreover we can see from the histogram of the model that the new and improved model from the full model still retains this feature. The influence plot in itself diagrammatically demonstrates how parameter estimates would vary if individual points were excluded. From the Residual vs Leverage plot along with the influence plot, we can clearly see there are no leverage points which skew the data, suggesting that we need not remove any individual data points. In addition, specifically to the influence plot, the plot clearly shows us that very few points appear to have a high leverage when considering the whole data, further justifying the analysis that we need not omit data points. By using the other plots mentioned we can see that there are no leverage points which skew the data and the diagnostic plot backs this up by showing this since even the three points with the highest Cook's distance are still relatively low in the wider view of the data.

The 208th data point's Studentized Residuals is less than 2 so this is not a high leverage point. In addition, the hat values for the 611th and 639th data points are less than 0.12 and this value is derived by the rule of thumb that p\_ii > 2p / n where p = 48, n = 800. Furthermore, the three data points all have a Cook's Distance value less than 1 so all these pieces of information leads to the conclusion that we shall not remove data points.

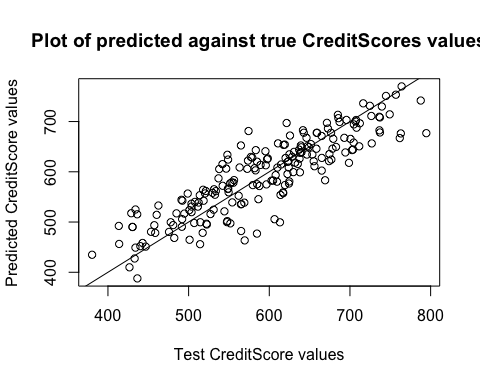
0.12 p = 48 n = 800, cook's < 1 Lastly, there is print out of the of the coefficients: - HistoryE has the strongest correlation which happens to be positive at 76.86358783, however, this is contradictory to logic. This is because if individuals have defaulted in the past this improves their credit rating. - StatusNone has a strong correlation of 69.40856679 meaning having no account or whether this is unknown means that this increases your credit rating. - OtherPartiesGuarantor and SavingsVeryLarge also have a positive and strong correlation with credit rating. - ForeignYes, PurposeEducation, StatusNegative, and PurposeNewCar all have a strong negative correlation and so decrease your credit rating. - Age, Amount, SavingsMedium are exhibit weak correlation and so, have little affect on the credit score.

## Q1b

# MSE

Within Q1a, the mean-square errors were necessary to determine which model would be the best and least complex model so determining the mse of the models was completed in Q1a and this is evident in the summary of the models.

Model\_Best <- lm (CreditScore ~ Status + Duration + Purpose + History + Savings + Personal + Foreign + Plans + OtherParties + Disposable + Amount + Housing + Telephone + Age, data = test)  
  
prediction\_Best <- predict(Model\_Best, newdata = select(test, - CreditScore))  
x.1 = select(test, CreditScore)  
x.1 = as.matrix(x.1)  
y.1 = as.matrix(prediction\_Best)  
plot(x.1, y.1, xlab = "Test CreditScore values", ylab = "Predicted CreditScore values", main = "Plot of predicted against true CreditScores values")  
abline(0, 1)

 The plot of the predicted against true CreditScore values demonstrates a positive correlation and so, implies the predicted and true values of CreditScore are directly proportional. This concludes that the predicted values are close to the true value and so, the model is suitable. Below is a numerical test providing further evidence of this observation.

t.test(x.1, y.1, paired = TRUE)

##   
## Paired t-test  
##   
## data: x.1 and y.1  
## t = 7.369e-14, df = 199, p-value = 1  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -6.008494 6.008494  
## sample estimates:  
## mean of the differences   
## 2.245307e-13

The t-test gives a p-value of 0.3241 which shows there is no significant difference between the predicted and true CreditScore values, showing that the model which we derived from Q1a using the train data is also suitable for test data.

## Q1c

predict\_data <- predict(Model\_Best, newdata = select(test, -CreditScore))  
true\_data <- select(test, CreditScore)  
risk\_predict\_data <- ifelse(predict\_data < 500, 1, 0)  
risk\_true\_data <- ifelse(true\_data < 500, 1, 0)  
sum(risk\_predict\_data == risk\_true\_data)

## [1] 176

This shows that 176/200 = 88% of the individuals from the test data were assigned the correct risk rating according to the classification given and using Model\_Best. This implies the predictions give a 88% accuracy.