DATA610 – Introduction to Data Mining
Group Project 1 – Classification

Prepared For:

Dr. Rasitha Jayasekare

Prepared By - Group 6:

Oliver O’Keefe
Anthony Orso
Walter Miller

Date:

April 16, 2022

Table of Contents

I. Introduction and summary of project

The teams within the DATA610 class have been presented two sets of data by the professor. The goal is to use the practices that the students are learning in the class to analyze the datasets and complete the goals outlined below.

The first set of data is labeled credit card data indicating those that have defaulted on the loans, along with several predictor variables that are to be used in the exercise. With this dataset, the team is asked to use data analytics to create a classification model that can be used to predict which borrowers are likely to default on a loan. The output will be a binomial response prediction that indicates whether a given borrower will default or not default.

The second set of data is labeled data about wine. It includes many measurable attributes about specific wines, along with a response variable that indicates the quality of the wine. The goal of the exercise is to create a model that can predict the quality of a wine given the measurable attributes of the wine. The quality response is a multi-class response variable that indicates the quality of the wine.

II. Approach:

In this report, the team has worked with two datasets described above and applied many of the steps of the Data Analytics Lifecycle in the process. The structure of this document is broken into several sections listed below culminating in the final interpretation and discussion.

1. Data Descriptions:

   In this phase we are getting to know the data that we will be working with. Understanding the predictors is key to this section. You will see the data analyzed both numerically and visually to help build this understanding.

2. Preprocessing of data:

   In this phase of the project, we are preparing the data for work in our models. We decide on the types of the variables we will use, and which ones are categorical and numerical. Outliers are identified and removed from the datasets, and then we perform Principal Component Analysis (PCA) to reduce the dimensions of the data.

3. Applying Classification techniques:

   In this phase of the project, we are applying classification techniques to each of the datasets. We will apply many classifications as appropriately to the datasets. The classifications that we will be considering are: Logistic Regression, Decision Trees, Bagging, Random Forests, and Boosting.

4. Perform and Interpret Predictions:

   In this phase, we will interpret the results of the classification techniques. The group will report the predictions and what these predictions mean, as well as compare the results from different classification techniques.

5. Interpret and Discuss Results:

   In the last phase of the project, the team will provide overall conclusions of the data analysis. The comparison of the results from each dataset, and the classifications used on these datasets will be discussed.

III. Data Descriptions

A. CreditCardDefaultData

Dataset source:

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#

Dataset Variables:

The first step in any data project is to understand the data that you are working with. The flowing list represents the set of variables in the CreditCardDefaultData dataset.

```
[1] "LIMIT_BAL"      "SEX"          "EDUCATION"      "MARRIAGE"
[5] "AGE"            "PAY_0"        "PAY_2"          "PAY_3"
[9] "PAY_4"          "PAY_5"        "PAY_6"          "BILL_AMT1"
[13] "BILL_AMT2"     "BILL_AMT3"    "BILL_AMT4"      "BILL_AMT5"
[17] "BILL_AMT6"     "PAY_AMT1"     "PAY_AMT2"       "PAY_AMT3"
[21] "PAY_AMT4"      "PAY_AMT5"     "PAY_AMT6"       "default.payment.next.month"
```

Determine the types of the variables:

In addition to understanding the available variables, it is critical to determine the types of these variables. In many cases, the actual types of the variables are not correct when imported and automatically recognized by many tools. In these cases, it is important to convert the types appropriately to continue to use these variables.

In the CreditCardDefaultData dataset all of the variables were imported as integers, but, analyzing the data, it is clear that some of these variables are categorical. In particular, the following fields needed to be identified as categorical (factors) in the data.

- SEX
- EDUCATION
- MARRIAGE
- default.Payment.Next.Month

Final Structure of data:

Once all of the variables are defined correctly in the dataset, the following output from our tool confirms our conversions have all been done correctly.

```
'data.frame':      5000 obs. of  24 variables:
 $ LIMIT_BAL   : int  50000 80000 320000 150000 170000 350000 280000 30000 20000 410000 ...
 $ SEX         : Factor w/ 2 levels "1","2": 1 2 2 2 2 2 2 2 2 2 ...
 $ EDUCATION   : Factor w/ 7 levels "0","1","2","3",..: 2 2 2 3 3 4 3 3 3 3 ...
 $ MARRIAGE    : Factor w/ 4 levels "0","1","2","3": 3 3 2 2 3 2 2 3 2 2 ...
 $ AGE         : int  34 27 34 35 27 38 49 29 53 50 ...
 $ PAY_0       : int  0 -1 -1 0 0 -2 0 0 0 0 ...
 $ PAY_2       : int  0 -1 -1 0 0 -2 0 0 0 0 ...
 $ PAY_3       : int  0 2 -1 0 0 -2 0 0 0 0 ...
 $ PAY_4       : int  0 -1 -1 0 0 -2 0 0 0 0 ...
 $ PAY_5       : int  0 -1 -1 2 0 -2 0 0 0 0 ...
 $ PAY_6       : int  0 -1 -1 0 0 -2 0 0 0 0 ...
 $ BILL_AMT1   : int  20613 988 800 39883 64195 1316 66749 26953 19594 246659 ...
 $ BILL_AMT2   : int  28018 12922 18873 41045 66163 3033 67942 27412 19502 248934 ...
 $ BILL_AMT3   : int  22744 12261 -150 41701 67856 2168 67908 27835 19431 246011 ...
 $ BILL_AMT4   : int  18484 2393 150 40699 38803 7951 63990 28215 20290 240845 ...
 $ BILL_AMT5   : int  16378 407 21565 25218 40259 4986 64050 28345 20293 213767 ...
 $ BILL_AMT6   : int  13352 2888 4818 9340 41541 3641 64486 29055 19665 214410 ...
 $ PAY_AMT1    : int  10000 12961 18873 1806 3000 3061 3001 1741 2000 11000 ...
 $ PAY_AMT2    : int  10000 0 0 1462 3000 2181 2382 2000 1700 10000 ...
 $ PAY_AMT3    : int  1500 2403 500 2000 2308 8071 67867 1439 1500 7500 ...
 $ PAY_AMT4    : int  3000 407 21565 0 3000 5002 2420 959 700 8000 ...
 $ PAY_AMT5    : int  2803 3176 5000 2000 3000 3654 2338 1500 750 8000 ...
 $ PAY_AMT6    : int  5000 7336 168492 44 1800 5820 2326 1411 1000 7000 ...
 $ default.payment.next.month: Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 1 1 1 ...
```

Numerical Description Summary:

After initial variable discovery, and conversion of the variables to the appropriate types, the following table represents out variable analysis.

Variable	Type	Description
LIMIT_BAL	integer	Represents the amount of given credit in NT Dollars (New Taiwan Dollars)
SEX	factor	Represents the sex of the participant 1=male;2=female
EDUCATION	factor	Represents the education level of the participant 1=grad school;2=university;3=high school;4=others
MARRIAGE	factor	1=married;2=single;3=other
AGE	integer	Represents the age of the participant
PAY_0 – PAY_6	integer	This is the payment history for n months -1=paymentOntime;1-9=payment delay for n mos
BILL_AMT1 – BILL_AMT6	integer	This is the bill amounts for the same period as payment history
PAY_AMT1 – PAY_AMT6	integer	This is the amount of payments made for the same time period as PAY0-6
default.payment.next.month	factor	Represents a binary response variable, default payment. 1=Yes; 0=No

Numerical Analysis:

Once the variables have been understood and appropriately “typed”, it is time to move on to numerical analysis of the data to learn more about the data. Numerical analysis will focus on the quantitative and qualitative variables, working to understand the data itself. We will be seeking to understand numerical properties of the data, such as mean, standard deviation, max, min, frequency(categorical), and more. We will approach these two ways. First, we will focus on the numerical representations of the dataset, and then we will transition to a visual analysis of this dataset.

The following output from our tool gives a good overview of the numerical properties of the dataset.

```
   LIMIT_BAL      SEX      EDUCATION MARRIAGE      AGE           PAY_0            PAY_2
 Min.   : 10000   1:2001   0:   2    0:  10   Min.   :21.00   Min.   :-2.0000   Min.   :-2.0000
 1st Qu.: 50000   2:2999   1:1771    1:2322   1st Qu.:28.00   1st Qu.:-1.0000   1st Qu.:-1.0000
 Median :150000            2:2319    2:2617   Median :34.00   Median : 0.0000   Median : 0.0000
 Mean   :168969            3: 835    3:  51   Mean   :35.52   Mean   :-0.0024   Mean   :-0.1126
 3rd Qu.:240000            4:   9             3rd Qu.:41.00   3rd Qu.: 0.0000   3rd Qu.: 0.0000
 Max.   :800000            5:  54             Max.   :73.00   Max.   : 8.0000   Max.   : 8.0000
                           6:  10
     PAY_3             PAY_4             PAY_5             PAY_6           BILL_AMT1
 Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   : -9095
 1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:  3968
 Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 22894
 Mean   :-0.1498   Mean   :-0.2016   Mean   :-0.2546   Mean   :-0.2924   Mean   : 51476
 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 64290
 Max.   : 7.0000   Max.   : 7.0000   Max.   : 7.0000   Max.   : 7.0000   Max.   :653062

   BILL_AMT2        BILL_AMT3         BILL_AMT4        BILL_AMT5        BILL_AMT6
 Min.   :-30000   Min.   : -24702   Min.   :-22108   Min.   : -8074   Min.   :-45734
 1st Qu.:  3587   1st Qu.:   2832   1st Qu.:  2366   1st Qu.:  1798   1st Qu.:  1260
 Median : 22045   Median :  20120   Median : 19294   Median : 18330   Median : 17178
 Mean   : 49598   Mean   :  47189   Mean   : 43094   Mean   : 40198   Mean   : 38666
 3rd Qu.: 62595   3rd Qu.:  58602   3rd Qu.: 52319   3rd Qu.: 49062   3rd Qu.: 48683
 Max.   :671563   Max.   :1664089   Max.   :706864   Max.   :514114   Max.   :499100

    PAY_AMT1         PAY_AMT2            PAY_AMT3         PAY_AMT4         PAY_AMT5
 Min.   :     0   Min.   :      0.0   Min.   :     0   Min.   :     0   Min.   :      0.0
 1st Qu.:  1034   1st Qu.:    905.8   1st Qu.:   390   1st Qu.:   292   1st Qu.:    303.8
 Median :  2209   Median :   2016.0   Median :  1800   Median :  1500   Median :   1571.0
 Mean   :  5824   Mean   :   6242.2   Mean   :  5104   Mean   :  4802   Mean   :   4985.8
 3rd Qu.:  5341   3rd Qu.:   5002.2   3rd Qu.:  4389   3rd Qu.:  4022   3rd Qu.:   4100.0
 Max.   :260416   Max.   :1684259.0   Max.   :380478   Max.   :528897   Max.   :426529.0

    PAY_AMT6      default.payment.next.month
 Min.   :     0   0:3906
 1st Qu.:   150   1:1094
 Median :  1500
 Mean   :  5662
 3rd Qu.:  4100
 Max.   :403500

Standard Deviation mapping of the quantitative variables
 LIMIT_BAL     AGE      PAY_0      PAY_2      PAY_3      PAY_4       PAY_5      PAY_6      BILL_AMT1
 129692.931   9.167     1.125      1.199      1.191      1.176       1.132      1.143      74075.848

 BILL_AMT2   BILL_AMT3  BILL_AMT4  BILL_AMT5  BILL_AMT6   PAY_AMT1   PAY_AMT2   PAY_AMT3   PAY_AMT4
 71158.598   72554.019  64374.408  60669.954  59189.956  14809.958  29448.017  16081.922  16385.864

 PAY_AMT5   PAY_AMT6
 17034.080  19133.738
```

From this data, we can make a few observations:

- EDUCATION is supposed to consist of the numbers 1,2,3,4 representing different levels of education. There are 2 records that contain a 0 for EDUCATION. We will have to determine if that is significant and may have to remove or repair this data.
- MARRIAGE is supposed to consist of the numbers 1,2,3 representing married, single, and other respectively. There are 10 records that contain a 0 for MARRIAGE. We will have to determine if that is significant and may have to remove or repair this data.
- BILL_AMT1-BILL_AMT6 has minimum values that are all negative. We will have to either determine what this indicates or remove these records as invalid data elements. A negative bill does not seem like a reasonable value.
- BILL_AMT1-BILL_AMT6 has a very large difference between all of the 3rd quartile values and the max value. Indicating that there may be some data outliers on the higher end of the data ranges.

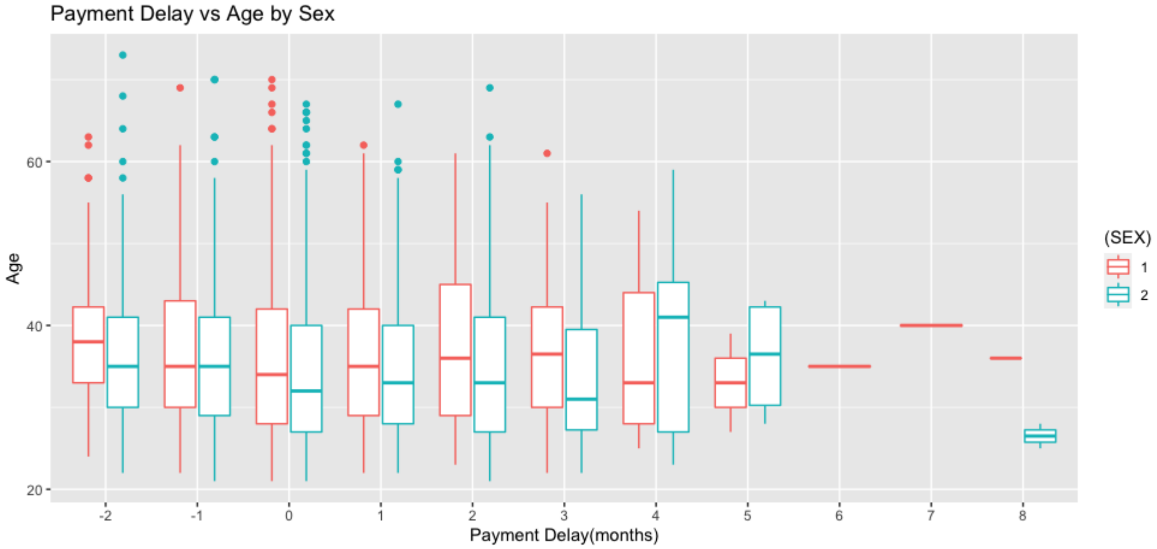
- PAY_AMT1-PAY_AMT6 has a very large difference between all of the 3rd quartile values and the max value. Indicating that there may be some data outliers on the higher end of the data ranges.

Visual Numerical Analysis:

Once basic numerical analysis is complete, it is important to move on to analyze the variables and their relationships visually.

Payment Delay vs. Age vs. Sex:

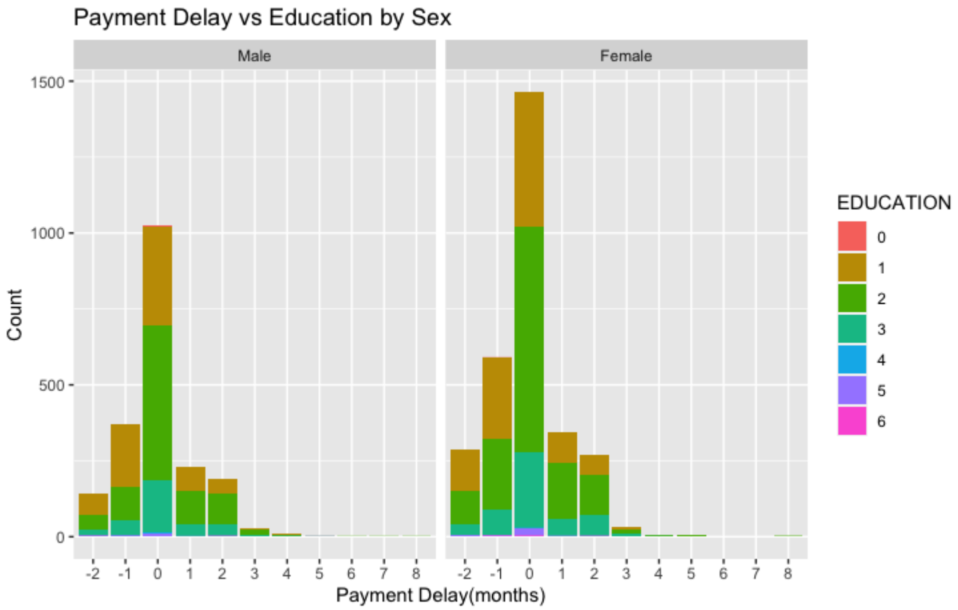
We want to understand if there is a visual correlation between being behind on the last month payment and the age (and sex) of the participant



Though not officially a factor variable, for this visualization we treated PAY_0 as a factor to get a plot for each month of delay. Keeping in mind that the "Payment Delay" represents the number of months that the last month payment was delayed, it appears that being behind on payments does not have a strong correlation to age. It does appear that, there might be a larger population of females behind on their payments as the number of months increases.

Payment Delay vs. Frequency vs. Education:

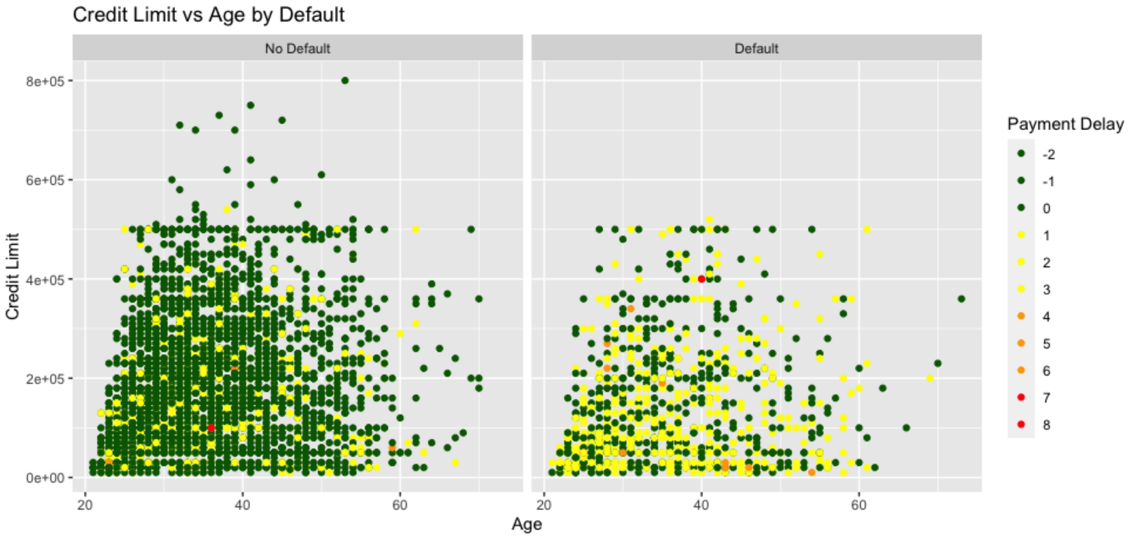
We want to understand if there is a visual correlation between being behind on payments and the education level of the borrower. By, again, treating payment delay as a factor (with 12 possible values), our third variable that contributes to this graph is the sex of the borrowers have been split across graphs.



In this visualization there are a couple of takeaways. First, it is clear that the majority of borrowers in the data are current on their payments (0 or less months behind). Also, the largest group of borrowers fall into the "university" education level. As the payment delay increases, the proportion of "university" education borrowers remains higher than the other.

Payment Delay vs. Frequency vs. Education:

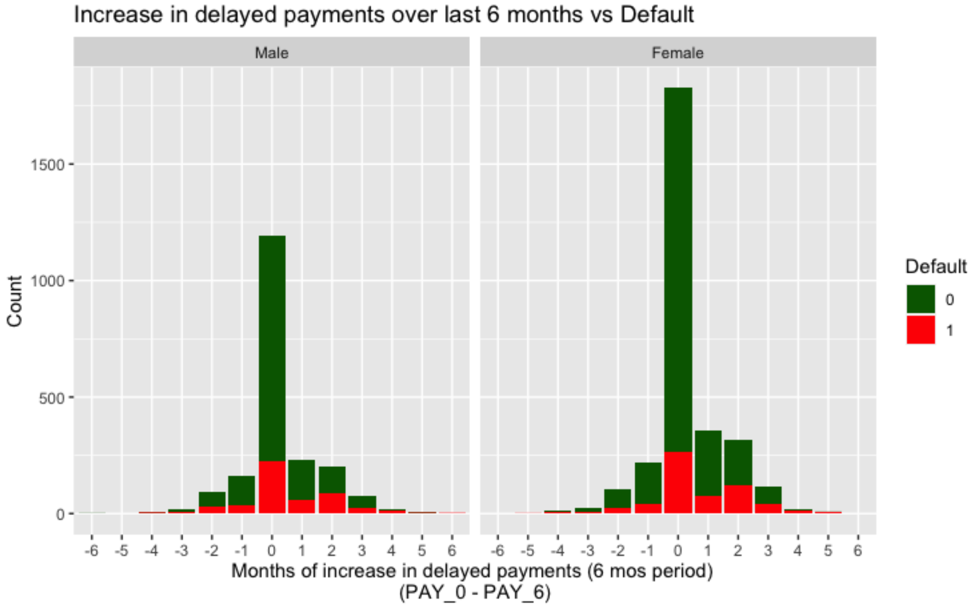
We wanted to compare the amount of given credit by age and then determine if it looks like it correlates to the response variable default.payment.next.month. To do this we will use a scatter plot of credit limit and age, and color the results by the response categorically variable. Adding in the predictor variable PAY_0, which indicates how "delayed" the most recent payment was (in months) allow us to visualize if being behind on payments could be a predictor of default.



From this plot it appears to indicate that borrowers that have a default in the next month are more likely to be 1-3 months behind on their payment (based on the payment delay color). Age, again, does not seem to be an influencer on the output variable. It interesting that the borrowers with the highest credit limits appear to not be in jeopardy of default or behind on their payments.

Payment Delay Increase vs. Sex vs. Default:

We want to understand if there is a relationship in the increase of payment delay over time to the default predictor. In this graph, we subtracted the delay in the most recent month from the delay 6 months ago to indicate if the participant has fallen further behind on payments over that time.



In looking at the bar graph it seems clear that the proportion of default is higher if the borrower has fallen at least one more month behind in the last 6 months. However, it is surprising that there are still defaults for those that have actually reduced their number of months of delay. It appears that the chance of default is proportionately low for those borrowers that have neither reduced or increased the delay of payments over this time.

Numerical Analysis Summary:

After initial analysis of the data, we have a much better understanding of the data. We have identified the data types, the numerical profiles of the quantitative variables, and the frequencies of the qualitative variables. We have also visualized the relationship between many of the variables and started to understand how some of these variables might act as predictors of the response variable. Moving on, we will start "Data Preparation" that will eventually feed our further analysis and model application.

B. WineQualityData

Dataset source:

https://archive.ics.uci.edu/ml/datasets/wine+quality

Dataset Variables:

To understand our data, we need to first identify our variables. The following represents the list of variables that define our dataset.

```
[1] "alcohol"          "chlorides"         "citricAcid"        "density"          "fixedAcidity"
[6] "freeSulfurDioxide"  "pH"              "quality"          "residualSugar"     "sulphates"
[11] "totalSulfurDioxide" "volatileAcidity"
```

Determine the types of variables:

Data is a tool that we use to ask questions and solve problems. Any craftsman will tell you that a tool is only useful if you know how to use it. A carpenter who does not know the difference between a hammer and saw is a carpenter that will go hungry.

Of our 12 variables, 11 of them are integers and 1 of them is a factor variable. That factor variable is "quality," which is the response variable. To account for this, we need to manually convert it from an integer to a factor in r.

Final Structure of data:

After converting our variables our dataset looks something like this.

```
alcohol :  num [1:4898] 8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
chlorides :  num [1:4898] 0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
citricAcid :  num [1:4898] 0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
density :  num [1:4898] 1.001 0.994 0.995 0.996 0.996 ...
fixedAcidity :  num [1:4898] 7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
freeSulfurDioxide :  num [1:4898] 45 14 30 47 47 30 30 45 14 28 ...
pH :  num [1:4898] 3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
```

quality : int [1:4898] 6 6 6 6 6 6 6 6 6 6 ...
residualSugar : num [1:4898] 20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
sulphates : num [1:4898] 0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
totalSulfurDioxide : num [1:4898] 170 132 97 186 186 97 136 170 132 129 ...
volatileAcidity : num [1:4898] 0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 …

Numerical Description Summary:

Variable	Type	Description
Alcohol	integer	A measure of the alcohol content. (% by volume)
Chlorides	integer	A measure of the salts in a wine. (g / dm^3)
citricAcid	integer	A measure of the citric acid in wine, often said to bring about a 'freshness' to the wine. (g / dm^3)
density	integer	A measure of the density of a wine. (g / cm^3)
fixedAcidity	integer	A measure of the fixed acidity in a wine. (g / dm^3)
volatileAcidity	integer	A measure of the volatile acids in a wine. (g / dm^3)
totalSulfurDioxide	integer	A measure of free and bound SO2s in a wine (mg / dm^3)
freeSulfurDioxide	integer	A measure of free SO2s in equilibrium between molecular SO2 and bisulfite ion (mg / dm^3)
residualSugar	integer	A measure of the remaining sugar after fermentation. (g / dm^3)
sulphates	integer	A measure of sulphates, an addative in wine. (g / dm^3)
pH	integer	A measure of a wine's pH balance
quality	Factor	A measure of 3-9 measuring the quality of a wine.

Numerical Analysis:

Below is a summary of all the variables and the values/frequencies of their ranges.

```
  fixedAcidity    volatileAcidity    citricAcid     residualSugar      chlorides     
 Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600   Min.   :0.00900  
 1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700   1st Qu.:0.03600  
 Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200   Median :0.04300  
 Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391   Mean   :0.04577  
 3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900   3rd Qu.:0.05000  
 Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800   Max.   :0.34600  

 freeSulfurDioxide totalSulfurDioxide    density              pH          sulphates     
 Min.   :  2.00    Min.   :  9.0      Min.   :0.9871   Min.   :2.720   Min.   :0.2200  
 1st Qu.: 23.00    1st Qu.:108.0      1st Qu.:0.9917   1st Qu.:3.090   1st Qu.:0.4100  
 Median : 34.00    Median :134.0      Median :0.9937   Median :3.180   Median :0.4700  
 Mean   : 35.31    Mean   :138.4      Mean   :0.9940   Mean   :3.188   Mean   :0.4898  
 3rd Qu.: 46.00    3rd Qu.:167.0      3rd Qu.:0.9961   3rd Qu.:3.280   3rd Qu.:0.5500  
 Max.   :289.00    Max.   :440.0      Max.   :1.0390   Max.   :3.820   Max.   :1.0800  

    alcohol      quality 
 Min.   : 8.00   3:  20  
 1st Qu.: 9.50   4: 163  
 Median :10.40   5:1457  
 Mean   :10.51   6:2198  
 3rd Qu.:11.40   7: 880  
 Max.   :14.20   8: 175  
                  9:   5  
```

The disparity between the minimum and maximum values of all our variables is way too large to pull any meaningful observations out of without conducting outlier analysis and removal. The outlier removal process will be discussed in detail in the next section, but below are the summary statistics of the trimmed-down data set.

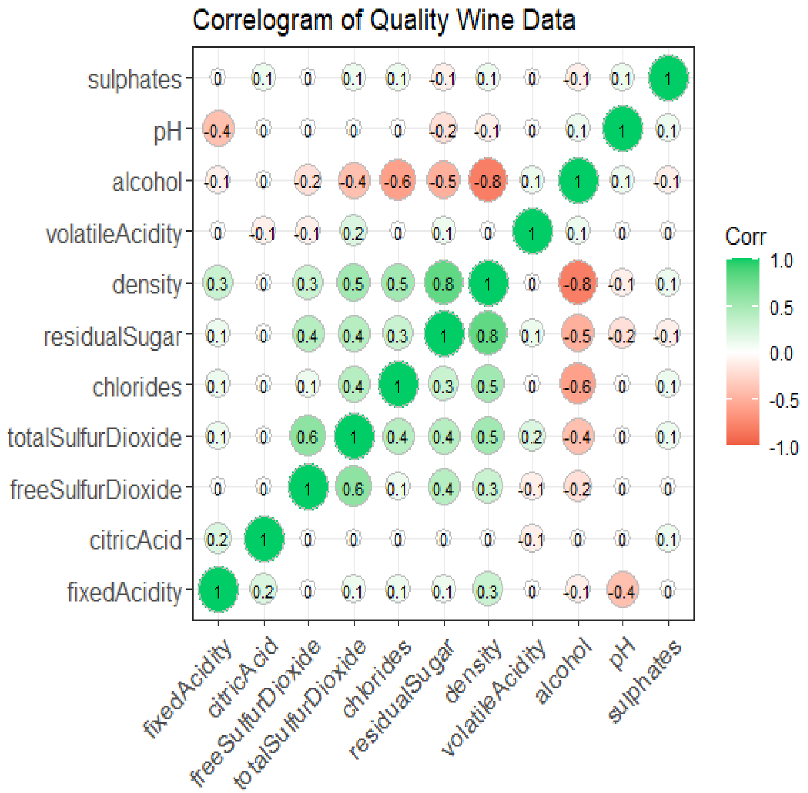
```
  fixedAcidity   volatileAcidity    citricAcid    residualSugar      chlorides      
 Min.   :4.800   Min.   :0.0800   Min.   :0.120   Min.   : 0.600   Min.   :0.01400  
 1st Qu.:6.300   1st Qu.:0.2100   1st Qu.:0.270   1st Qu.: 1.800   1st Qu.:0.03400  
 Median :6.800   Median :0.2600   Median :0.310   Median : 5.000   Median :0.04100  
 Mean   :6.792   Mean   :0.2641   Mean   :0.321   Mean   : 6.161   Mean   :0.04168  
 3rd Qu.:7.300   3rd Qu.:0.3100   3rd Qu.:0.370   3rd Qu.: 9.500   3rd Qu.:0.04900  
 Max.   :8.800   Max.   :0.4850   Max.   :0.540   Max.   :20.800   Max.   :0.07200  
 freeSulfurDioxide totalSulfurDioxide    density             pH          sulphates     
 Min.   : 3.00     Min.   : 24.0      Min.   :0.9871   Min.   :2.820   Min.   :0.2500  
 1st Qu.:24.00     1st Qu.:106.0      1st Qu.:0.9915   1st Qu.:3.090   1st Qu.:0.3900  
 Median :33.00     Median :132.0      Median :0.9933   Median :3.180   Median :0.4500  
 Mean   :34.76     Mean   :135.6      Mean   :0.9937   Mean   :3.191   Mean   :0.4765  
 3rd Qu.:45.00     3rd Qu.:164.0      3rd Qu.:0.9958   3rd Qu.:3.290   3rd Qu.:0.5500  
 Max.   :79.00     Max.   :252.0      Max.   :1.0010   Max.   :3.580   Max.   :0.7600  
    alcohol      quality
 Min.   : 8.50   3:  3  
 1st Qu.: 9.60   4: 29  
 Median :10.50   5:380  
 Mean   :10.64   6:610  
 3rd Qu.:11.50   7:300  
 Max.   :13.90   8: 59  
                 9:  2  
```

Our ranges are now much tighter and more meaningful. Through cleansing our data, we are able to also provide more insightful data visualizations

Visual Numerical Analysis:

Due to the extensive number of outliers that were detected in the upcoming step of data preprocessing, please note that these visuals are based off a reduced data set to produce more visually meaningful representations of our data. Before conducting outlier analysis, all graphs produced from ggplot2 lacked nuance and distinct patterns. However, the removal of over 600 outliers allowed us to return to this step and produce more nuanced graphs.

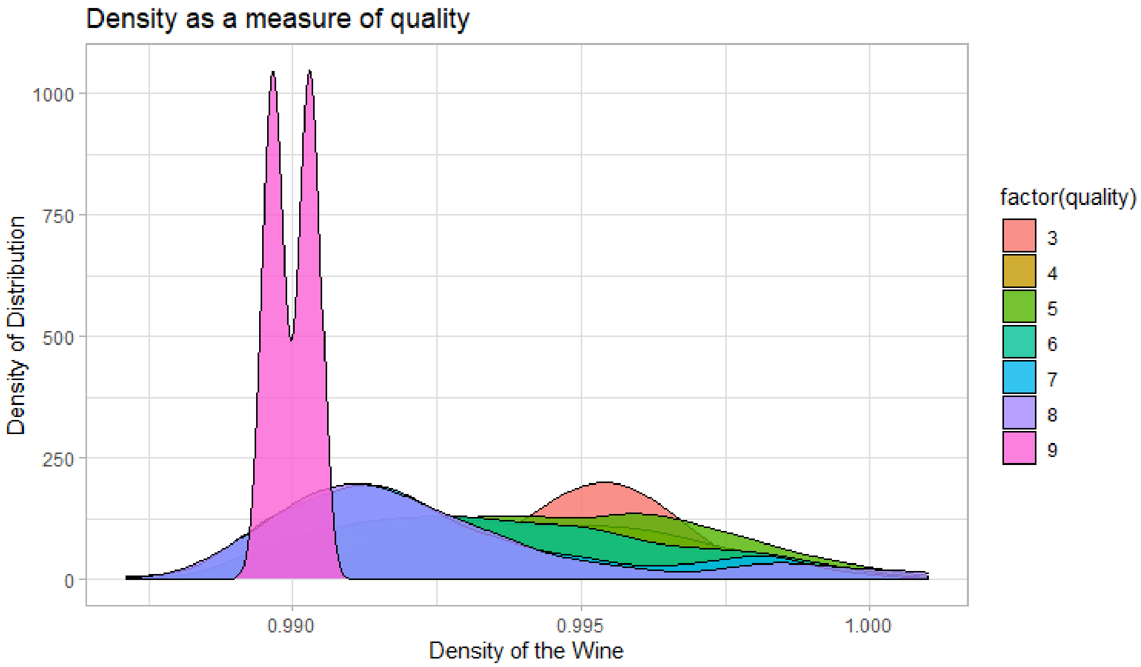
Sometimes when working with data this large, it is best to get a visual summary of how all your data interacts with each other. For this we can use a correlogram, which will plot the correlation of every noncategorical variable we have against one another. This can give us new insights into our data.



This reveals a lot about our data. Because we are studying wine, we need to keep in mind the finite nature of our subject matter expertise. A sommelier would have been crucial in allowing us to dive deeper into the various relationships within the data set to further direct our analysis and keep an eye out for multicollinearity. However, we can see that residual sugars are correlated positively to density. As wine ferments during the aging process, it increasingly loses more of these residual sugars. We may conclude that a denser wine is younger, which can help us to get an idea of the age of individual wines, which in turn gives us a new avenue of observation.

Measuring wine density vs quality.

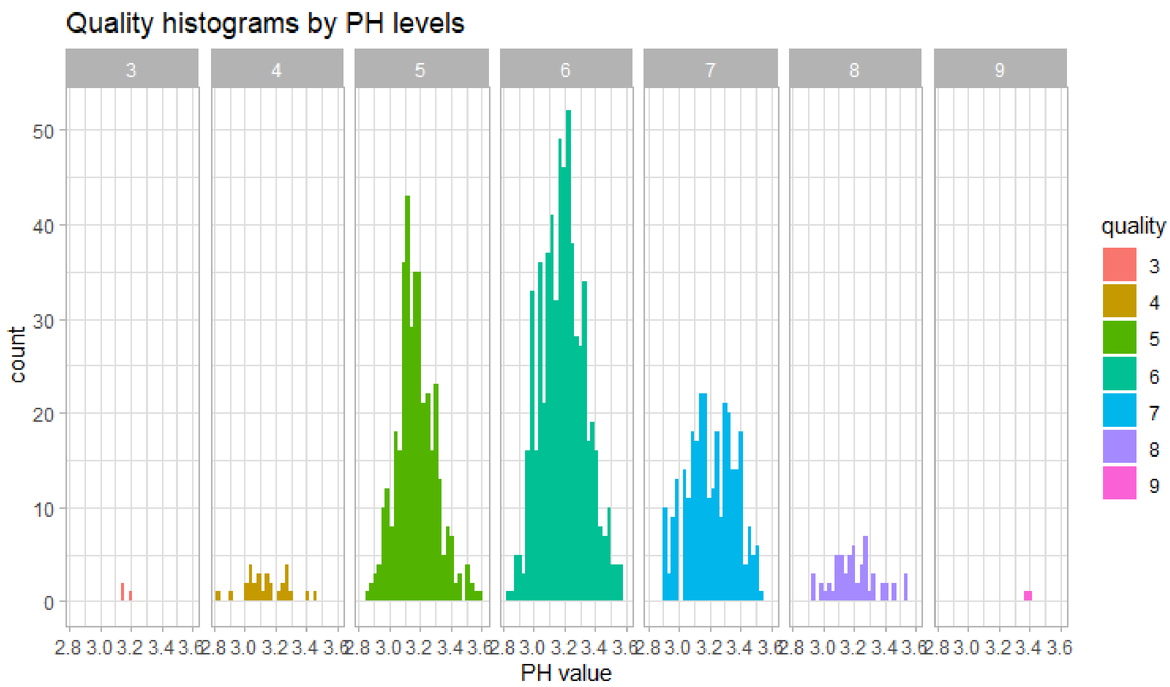
Density is more than just residual sugars; it is every tangible part of the wine. When we measure density, we do so by comparing it to water, which, for the purposes of the following graph, has a density value of 1. There is an old idiom that states “Too many cooks spoil the broth” so in that vein we can use a density plot to see if, “Too many molecules spoil the wine.”



When we observe the above plot, we see two immediately interesting things. The first being the overwhelming distribution of quality favoring lower density wines. With most wines falling off rapidly when passing that 1 density mark. The second being an unexpected dip in quality 7 and 8 wines between 0.995 and 1. If I had to put myself in the shoes of an amateur sommelier, I would assume that this is a dividing point between red and white wines. As established earlier, density is most correlated to residual sugars, that disappear as the wine ages. Red wines are usually severed later than white wines, which leads to them being less sweet and less dense.

PH vs Quality:

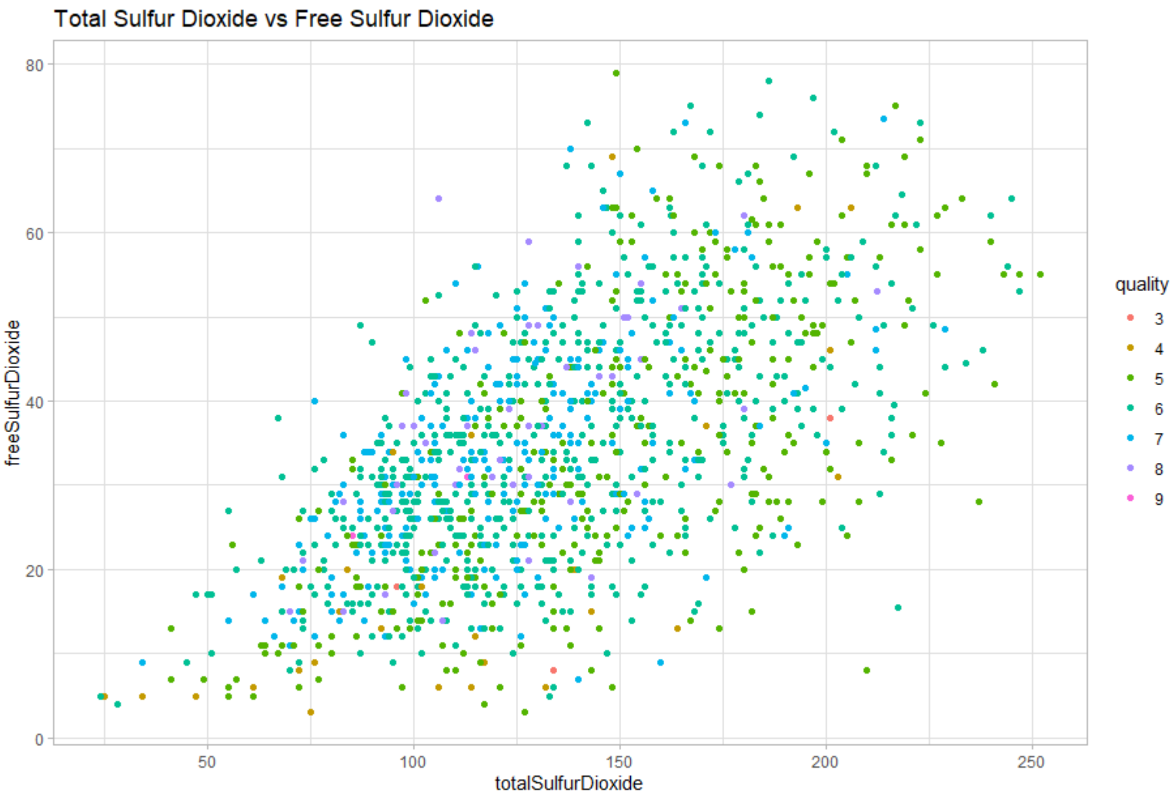
Much like how the density value is the sum of several physical parts, PH is the sum of several chemical parts. Things with a high PH value are faster to take to oxidation, which can be a detriment to wine. A higher PH can also tell us where a wine comes from, as wines with higher acidity are more likely to come from grapes in colder climates.



By looking at this histogram, we learn that a higher PH is preferred amongst those who tested and classified our wine. This in turn can inform us about the makeup of salts and sulfates that were used as additives into the wine to manipulate its PH values.

Sulfur dioxide and its effect on Quality:

Sulfur Dioxide is the widest in terms of range of all our variables observed. In both total and free sulfur dioxide, it has the largest range of values and thus a large potential to affect our quality. This is important to consider during the scaling process for classification and principal component analysis.



Looking at the graph, we can see that our highest quality wines favor a more balanced approach and are condensed in the center of the scatterplot.

Numerical Analysis summary:

When we review our findings, we can see that the PH and density of a wine have a measurable effect on the overall quality. They are a balancing act within themselves and against each other. We will keep these observations in mind when undergoing the dimension reduction process to ensure the variables selected for modeling have the greatest impact on variance in the response variable.

IV. Preprocessing of Data

A. CreditCardDefaultData

Check and remove rows of missing values.

After using our tool analyzing the 5000 samples in our CreditCardDefaultData set, we have determined that there are no missing values. We have noticed that there are a small number of factor variables that fall outside the described labels that we will have to determine whether to remove.

Check for outliers:

To remove outliers, we will start with the variable with the largest number of outliers and work our way through the list. Analysis of the outliers in the dataset result in the following sorted list of counts per variable.

```
PAY_2      PAY_3      PAY_4      PAY_0      PAY_6      PAY_5   PAY_AMT5 BILL_AMT5  PAY_AMT4  PAY_AMT2
 746        715        601        550        510        505        489       481       481       480

BILL_AMT4  PAY_AMT1  PAY_AMT6 BILL_AMT1 BILL_AMT6 BILL_AMT3 BILL_AMT2  PAY_AMT3   LIM_BAL     AGE
   479       472       455       452       450       447       436       435        20        20
```

With outlier counts of this level it likely requires that we do more analysis of the variable and not blindly listen to the tool.

PAY_x:

The PAY_x variables have a range from -2 to 9 in the data descriptions above. Because a significant number of the borrowers pay the proper amount and pay on-time, these values (-1 and 0) overwhelm the rest of the data. The tool believes that these values are outliers, buy in reality they are not, and could be important.

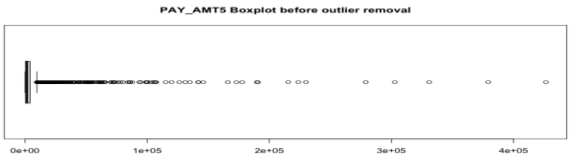
Decision: We will not perform outlier removal on the PAY_x variables.

After deciding how to handle the above variables, the list our outliers is:

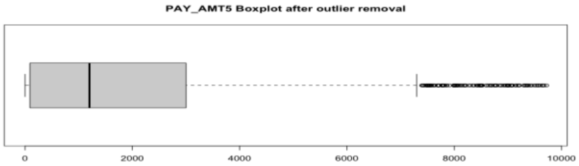
```
PAY_AMT5 BILL_AMT5  PAY_AMT4  PAY_AMT2 BILL_AMT4  PAY_AMT1  PAY_AMT6 BILL_AMT1 BILL_AMT6 BILL_AMT3
  489       481       481       480       479       472       455       452       450       447
BILL_AMT2  PAY_AMT3   LIM_BAL       AGE
   436       435        20        20
```

Remove outliers (in order from largest count to fewest):

```
#PAY_AMT5
         There are 5000 rows in the overall dataset before removal of PAY_AMT5 outliers
```

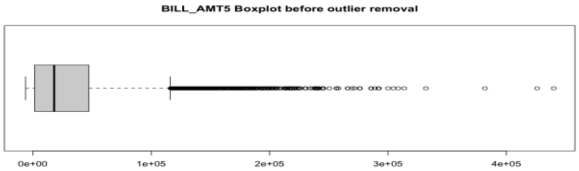


```
         Removing 489 outliers from PAY_AMT5.
```

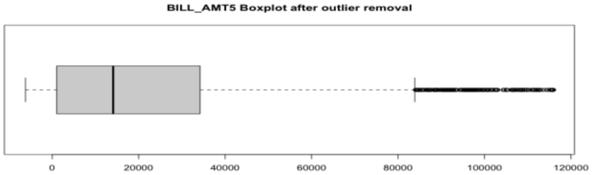


```
        There are 4511 rows in the overall dataset before removal of PAY_AMT5 outliers
```

```
#BILL_AMT5
        There are 4511 rows in the overall dataset before removal of BILL_AMT5 outliers
```

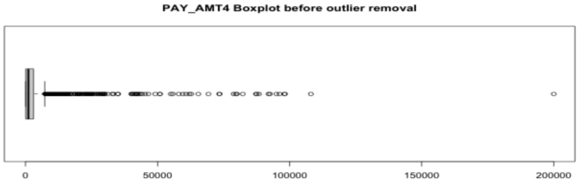


```
        Removing 377 outliers from BILL_AMT5
```

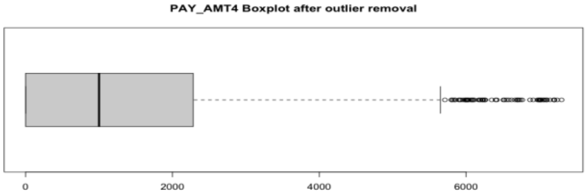


```
        There are 4134 rows in the overall dataset before removal of BILL_AMT5 outliers
```

```
#PAY_AMT4
        There are 4134 rows in the overall dataset before removal of PAY_AMT5 outliers
```

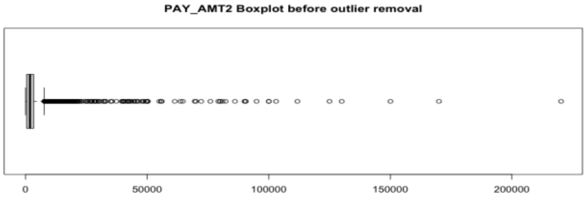


```
        Removing 289 outliers from PAY_AMT4.
```

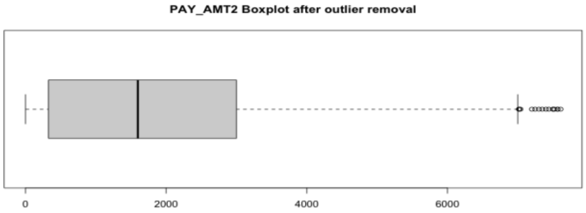


```
        There are 3845 rows in the overall dataset before removal of PAY_AMT4 outliers
```

```
#PAY_AMT2
        There are 3845 rows in the overall dataset before removal of PAY_AMT2 outliers
```



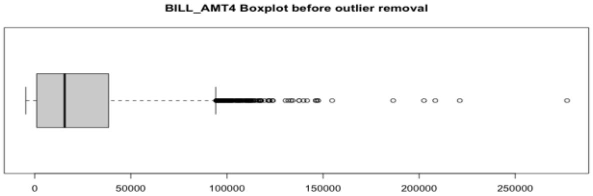
```
        Removing 308 outliers from PAY_AMT2.
```



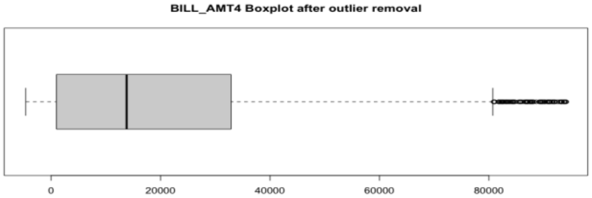
```
There are 3537 rows in the overall dataset before removal of PAY_AMT2 outliers
```

#BILL_AMT4

```
There are 3537 rows in the overall dataset before removal of BILL_AMT4 outliers
```



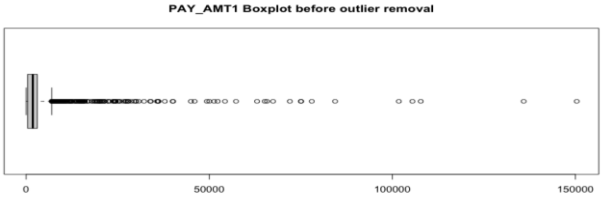
```
Removing 144 outliers from BILL_AMT4
```



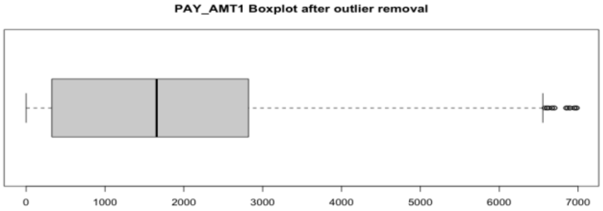
```
There are 3393 rows in the overall dataset before removal of BILL_AMT4 outliers
```

#PAY_AMT1

```
There are 3393 rows in the overall dataset before removal of PAY_AMT1 outliers
```



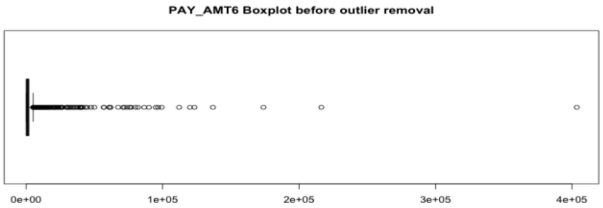
```
Removing 242 outliers from PAY_AMT1.
```



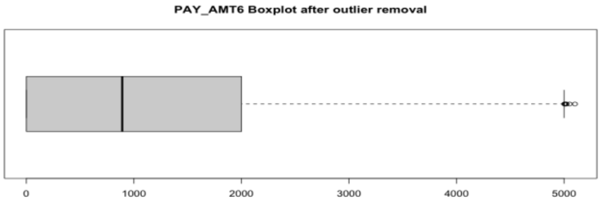
```
There are 3151 rows in the overall dataset before removal of PAY_AMT1 outliers
```

#PAY_AMT6

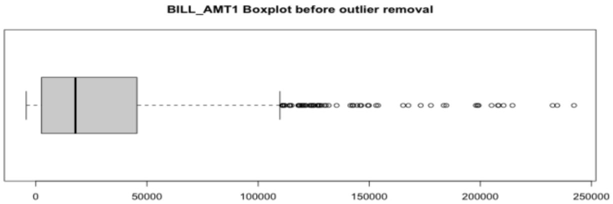
```
There are 3151 rows in the overall dataset before removal of PAY_AMT6 outliers
```



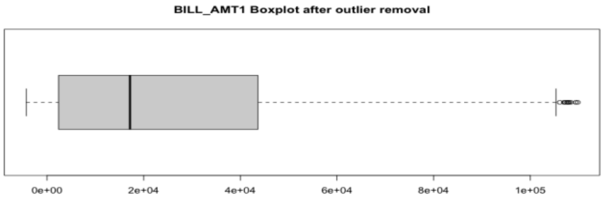
```
        Removing 216 outliers from PAY_AMT6.
```



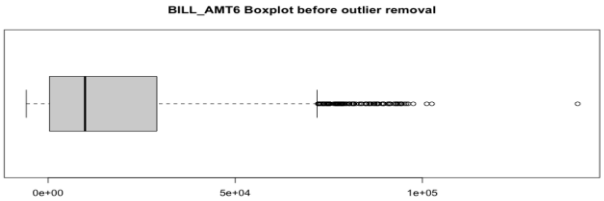
```
        There are 2935 rows in the overall dataset before removal of PAY_AMT6 outliers


#BILL AMT1
        There are 2935 rows in the overall dataset before removal of BILL_AMT1 outliers
```



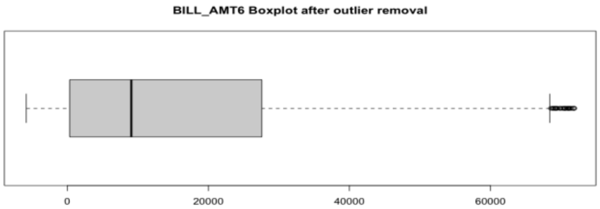
```
        Removing 64 outliers from BILL_AMT1
```



```
        There are 2871 rows in the overall dataset before removal of BILL_AMT1 outliers

#BILL AMT6
        There are 2871 rows in the overall dataset before removal of BILL_AMT6 outliers
```



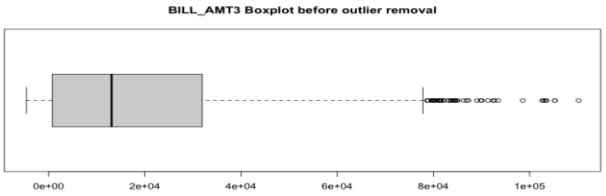
```
        Removing 106 outliers from BILL_AMT6
```



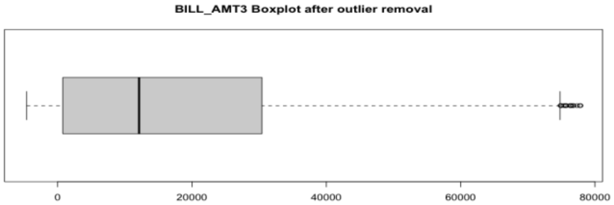
```
        There are 2765 rows in the overall dataset before removal of BILL_AMT6 outliers
```

#BILL_AMT3

```
        There are 2765 rows in the overall dataset before removal of BILL_AMT3 outliers
```



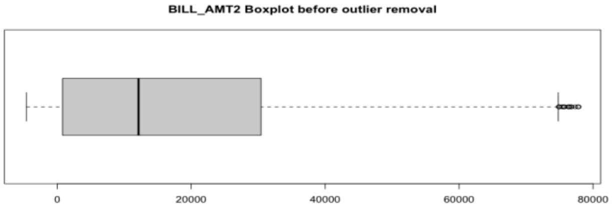
```
        Removing 49 outliers from BILL_AMT3
```



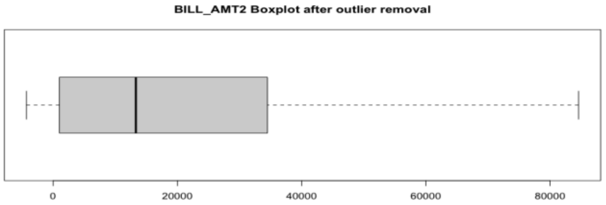
```
        There are 2716 rows in the overall dataset before removal of BILL_AMT3 outliers
```

#BILL_AMT2

```
        There are 2716 rows in the overall dataset before removal of BILL_AMT2 outliers
```



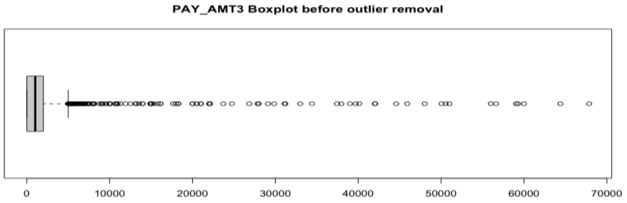
```
        Removing 17 outliers from BILL_AMT2
```



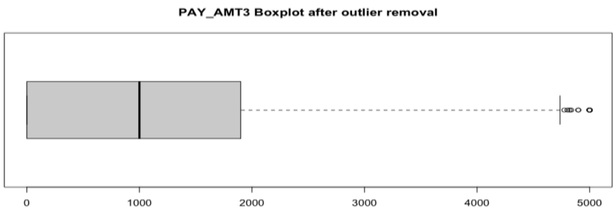
```
        There are 2699 rows in the overall dataset before removal of BILL_AMT2 outliers
```

#PAY_AMT3

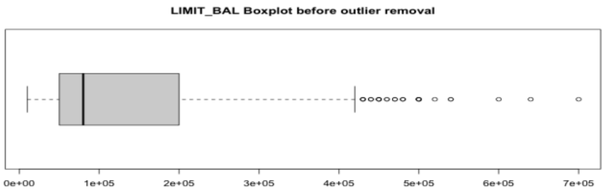
```
        There are 2699 rows in the overall dataset before removal of PAY_AMT3 outliers
```



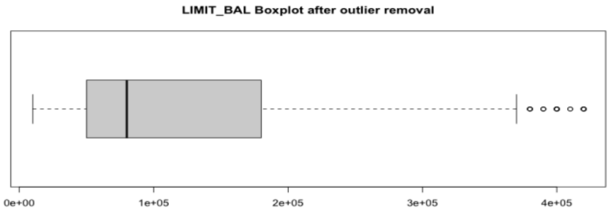
```
        Removing 216 outliers from PAY_AMT3.
```



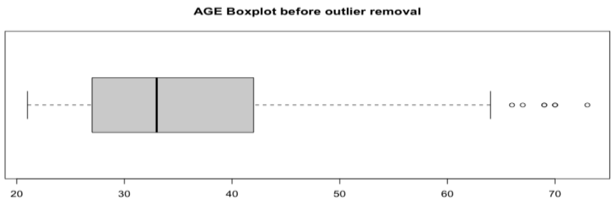
```
        There are 2550 rows in the overall dataset before removal of PAY_AMT3 outliers

#LIMIT_BAL
        There are 2550 rows in the overall dataset before removal of LIMIT_BAL outliers
```



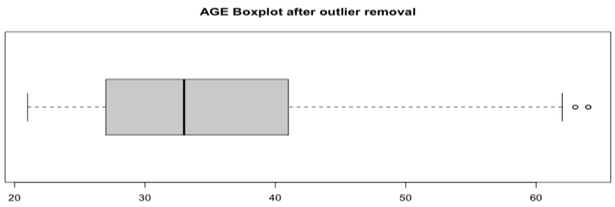
```
        Removing 58 outliers from LIMIT_BAL
```



```
        There are 2492 rows in the overall dataset before removal of LIMIT_BAL outliers

#AGE
        There are 2492 rows in the overall dataset before removal of AGE outliers
```



```
        Removing 7 outliers from AGE
```



```
        There are 2485 rows in the overall dataset before removal of AGE outliers
```

Outlier Summary:

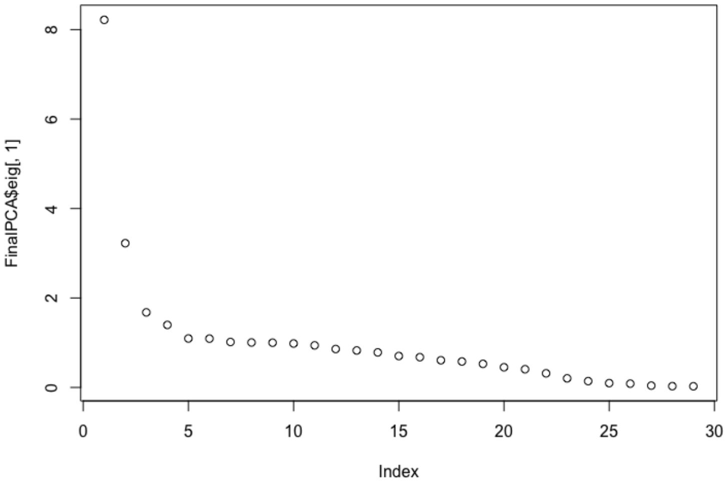
After outlier removal, there is 2485 samples left in the dataset for further analysis.

Create the training and testing datasets

The next step is to separate the cleaned dataset into a training dataset and a testing dataset. It is important to have both datasets and keep them separate so that testing is not performed with the same dataset that was used to train a model.
In this instance, we will perform an 80/20 split of the data where 80% of the data is used as the training dataset, and the remaining 20% of the data is used as the testing dataset. Since we had 2485 samples after outlier removal, we will have 1988 samples in the training dataset and 497 samples in the testing dataset. The datasets will be saved as creditCardDefaultTrainingData.txt and creditCardDefaultTestingData.txt.

Apply PCA to the training dataset to reduce the dimensions.
There are currently 20 quantitative variables and 3 qualitative variables in the dataset (and one categorical response variable). Our goal is to understand which of these are most significant and remove the insignificant ones. We will attempt to explain 80% of the variance in the data, and use a correlation coefficient of |r| > 0.59 (r^2=0.3481).

Running the mixed Principle Component Analysis in the tool returns the following screeplot. Visually, we determined that we need to include ~12-13 PC's in our analysis. Examining at the eigen values shows us that if we include 13 PC's the model will explain over 80.45% of the variance. We will select PC1-PC13 as our principle components.



```
       Eigenvalue  Proportion    Cumulative
dim 1  8.21862166  28.34007469   28.34007
dim 2  3.22444789  11.11878582   39.45886
dim 3  1.67752621   5.78457314   45.24343
dim 4  1.39749421   4.81894556   50.06238
dim 5  1.09423102   3.77321041   53.83559
dim 6  1.09158894   3.76409979   57.59969
dim 7  1.01592321   3.50318349   61.10287
dim 8  1.00447288   3.46369959   64.56657
dim 9  0.99997638   3.44819441   68.01477
dim 10 0.98177204   3.38542082   71.40019
dim 11 0.93995208   3.24121407   74.64140
dim 12 0.85865099   2.96086550   77.60227
dim 13 0.82629330   2.84928724   80.45155
dim 14 0.78336072   2.70124387   83.15280
…
```

Once we have selected our principle components, we need to analyze the correlation coefficients to determine which of our original variables are actually significant.

```
Squared loadings :
          dim 1 dim 2 dim 3 dim 4 dim 5 dim 6 dim 7 dim 8 dim 9 dim 10 dim 11 dim 12 dim 13 ~~dim 14~~
LIMIT_BAL  0.17  0.16  0.01  0.09  0.00  0.01  0.00  0.00  0.01   0.01   0.17   0.05   0.01   ~~0.00~~
AGE        0.00  0.00  0.70  0.01  0.01  0.02  0.00  0.00  0.00   0.00   0.01   0.00   0.00   ~~0.00~~
PAY_0      0.20  0.32  0.00  0.01  0.01  0.00  0.00  0.00  0.00   0.00   0.02   0.02   0.04   ~~0.02~~
PAY_2      0.35  0.35  0.00  0.00  0.00  0.00  0.00  0.00  0.00   0.00   0.01   0.00   0.02   ~~0.01~~
PAY_3      0.39  0.34  0.00  0.01  0.00  0.00  0.00  0.00  0.00   0.00   0.00   0.01   0.00   ~~0.07~~
PAY_4      0.40  0.36  0.00  0.06  0.00  0.00  0.00  0.00  0.00   0.00   0.00   0.01   0.02   ~~0.02~~
PAY_5      0.42  0.31  0.00  0.07  0.00  0.00  0.00  0.00  0.00   0.00   0.00   0.01   0.01   ~~0.02~~
PAY_6      0.44  0.24  0.00  0.06  0.00  0.00  0.00  0.00  0.00   0.00   0.00   0.00   0.00   ~~0.00~~
BILL_AMT1  0.63  0.09  0.00  0.13  0.00  0.00  0.00  0.00  0.00   0.00   0.00   0.00   0.00   ~~0.00~~
BILL_AMT2  0.74  0.08  0.00  0.10  0.00  0.00  0.00  0.00  0.00   0.00   0.00   0.00   0.00   ~~0.00~~
BILL_AMT3  0.81  0.07  0.00  0.06  0.00  0.00  0.00  0.00  0.00   0.01   0.00   0.00   0.00   ~~0.00~~
BILL_AMT4  0.83  0.06  0.00  0.03  0.00  0.00  0.00  0.00  0.00   0.01   0.01   0.00   0.00   ~~0.00~~
BILL_AMT5  0.81  0.05  0.00  0.02  0.00  0.00  0.00  0.00  0.00   0.00   0.01   0.00   0.01   ~~0.00~~
BILL_AMT6  0.79  0.05  0.00  0.01  0.00  0.00  0.00  0.00  0.00   0.00   0.01   0.00   0.02   ~~0.00~~
PAY_AMT1   0.21  0.11  0.00  0.01  0.01  0.02  0.00  0.01  0.00   0.01   0.07   0.04   0.22   ~~0.17~~
PAY_AMT2   0.21  0.09  0.00  0.09  0.00  0.00  0.00  0.00  0.01   0.00   0.01   0.00   0.07   ~~0.36~~
PAY_AMT3   0.25  0.13  0.00  0.06  0.00  0.00  0.00  0.00  0.00   0.00   0.02   0.00   0.00   ~~0.04~~
PAY_AMT4   0.20  0.12  0.00  0.13  0.00  0.00  0.00  0.00  0.00   0.01   0.01   0.00   0.00   ~~0.02~~
PAY_AMT5   0.12  0.13  0.00  0.20  0.01  0.00  0.01  0.00  0.00   0.00   0.01   0.01   0.10   ~~0.00~~
PAY_AMT6   0.19  0.12  0.00  0.15  0.00  0.00  0.01  0.00  0.00   0.03   0.00   0.02   0.04   ~~0.01~~
SEX        0.00  0.01  0.00  0.00  0.22  0.39  0.00  0.00  0.01   0.05   0.14   0.02   0.12   ~~0.00~~
EDUCATION  0.05  0.04  0.31  0.08  0.57  0.18  0.45  0.92  0.71   0.66   0.38   0.34   0.08   ~~0.02~~
MARRIAGE   0.00  0.00  0.64  0.02  0.26  0.45  0.53  0.05  0.25   0.17   0.04   0.31   0.05   ~~0.00~~
```

PCA Conclusions:

The PCA results indicate that we should carry forward 14 of the original 23 variables into our model planning and data analytics. The list of variables being carried forward are:

Variable	Type
EDUCATION	factor
MARRIAGE	factor
AGE	integer
PAY_2	integer
PAY_3	integer
PAY_4	integer
PAY_5	integer
PAY_6	integer
BILL_AMT1	integer
BILL_AMT2	integer
BILL_AMT3	integer
BILL_AMT4	integer
BILL_AMT5	integer
BILL_AMT6	integer

We will move forward with this dataset, and the datasets will be saved as creditCardDefaultTrainingDataReduced.txt and creditCardDefaultTestingDataReduced.txt.

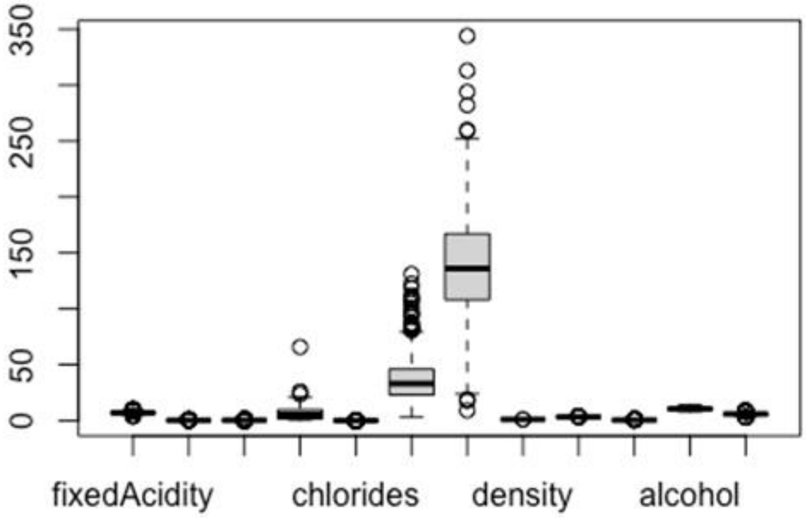
B. Wine Quality

Check and remove rows of missing values.

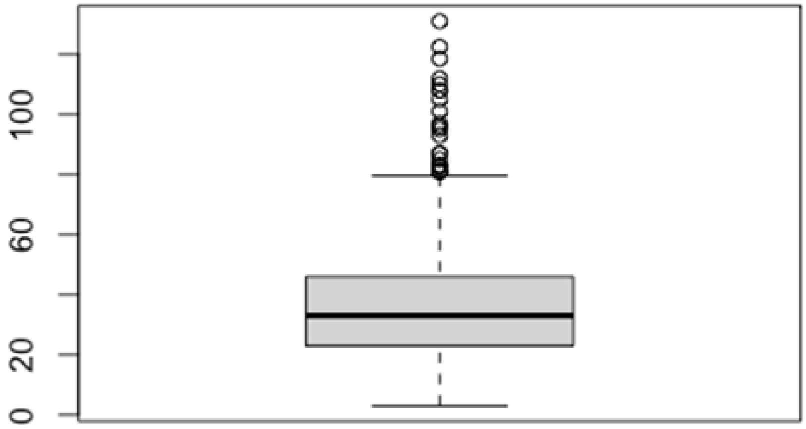
When executing *which()* commands, we learn there are no missing values for this data set. Therefore, we can move straight into outlier removal. To begin outlier removal, we must first use the *boxplot()* function to visualize where the greatest number of outliers exist.

Check for outliers:

A simple boxplot of all the variables next to each other helps us casually visualize which variables to appear to need cleaning first. To note, our methodology will only involve removing outliers once per variable. This is important because the removal of outliers during a first pass results in a change to the parameters of the data set, which may produce new outliers. To prevent endless iterations that narrowly thin out our data set, outliers will only be removed in one cleaning per variable.

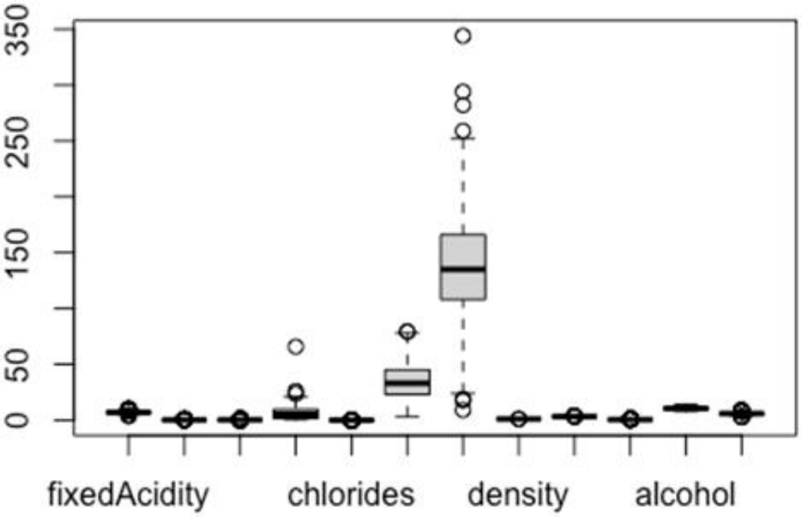


As we can see, all variables except "alcohol" have outliers. Because "freeSulfurDioxide" (one variable to the right of "chlorides") has the greatest number of visible outliers, we will start there.

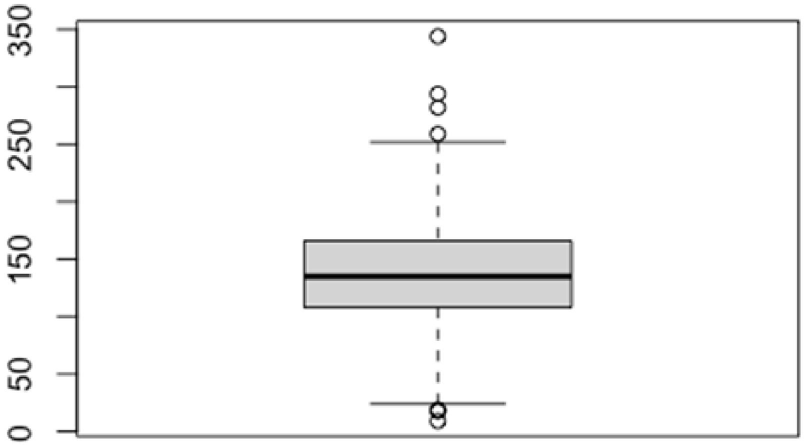


Once the outliers are removed, the reduced data set is renamed to wineQualitySubsample2. Below are the dimensions for the new data set and its boxplot.

```
## [1] 1977    12
```

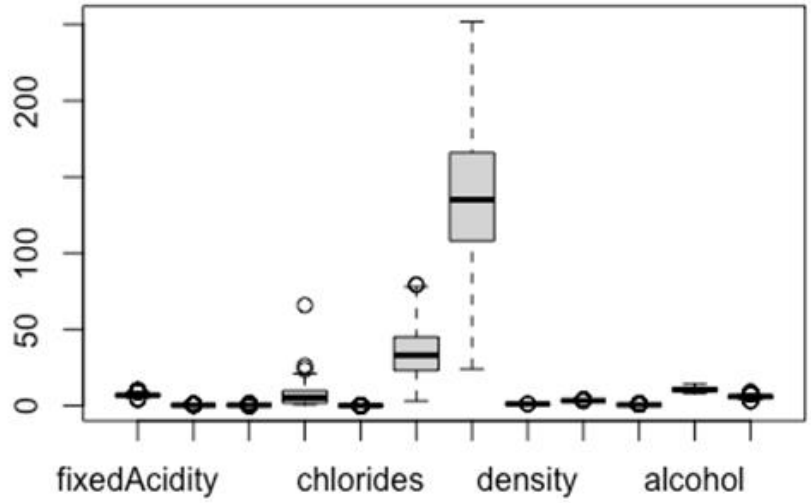


One can see there are a couple of new outliers in that variale, but there are far fewer now. As you can see, the removal of these outliers also resulted in a reduction of outliers in “totalSulfurDioxide” (the variable to the right of it) Thus, we will move on to cleaning that variable, which is visualized in the boxplot below.

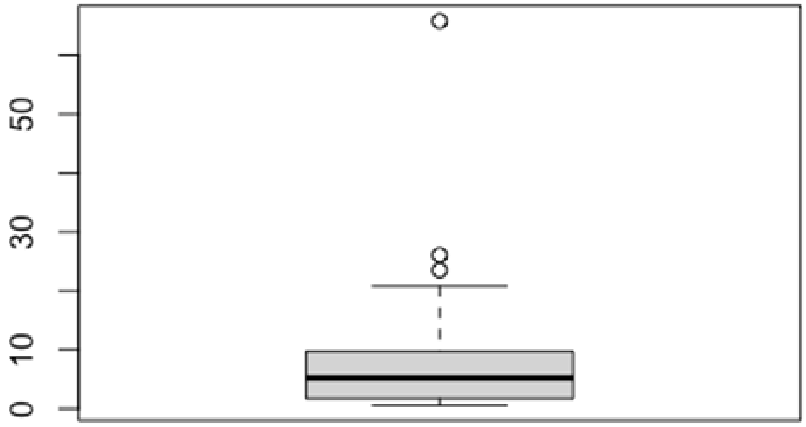


Once we remove these outliers, a new data set called wineQualitySubsample 3 is created, which has the following dimensions and boxplot.

```
## [1] 1969   12
```

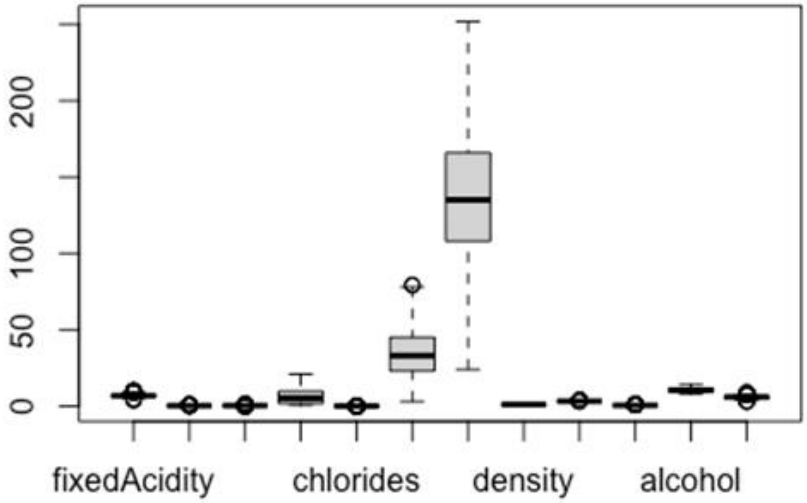


There are now no outliers in the “totalSulfurDioxide” variable. The rest of the variables are on much smaller scales, so the number of outliers in these variables can only be viewed when they are visualized individually or with other variables of similar ranges. We begin with “residualSugar”(column 4).

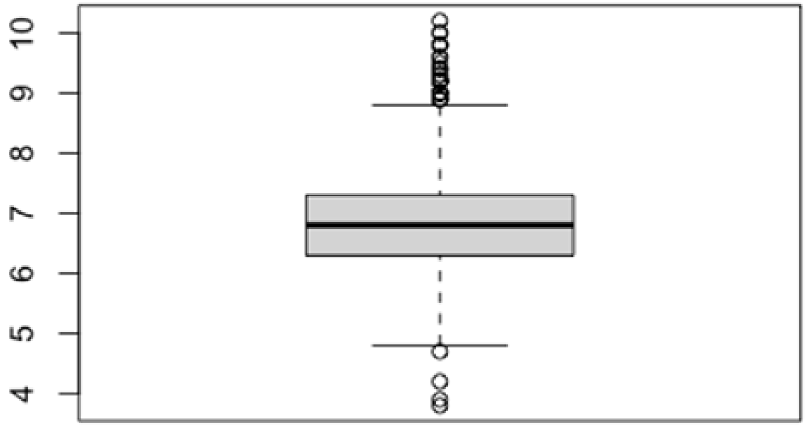


After removing the outliers from “residualSugar,” a new data set called wineQualitySubsample4 is created. The following dimensions and boxplot are observed.

```
## [1] 1966     12
```

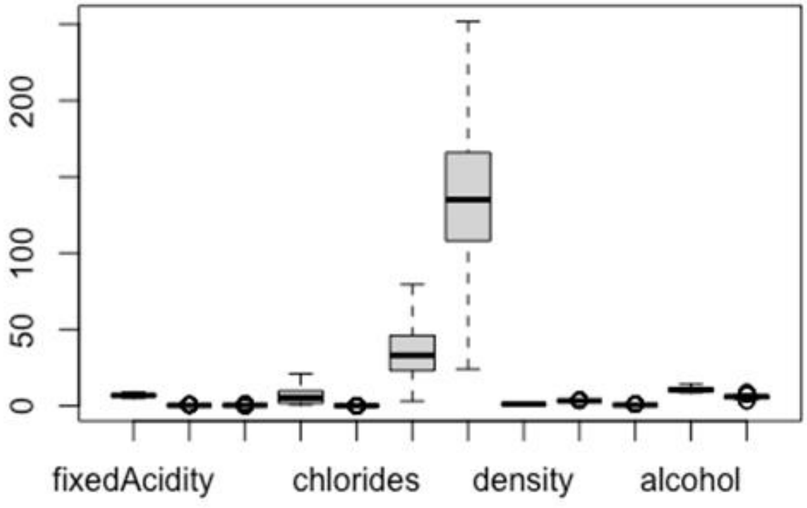


There are now zero outliers in the “residualSugar” column. The next step is to move onto the variables with the most miniscule ranges, starting with “fixedAcidity.” This variable has the following boxplot.

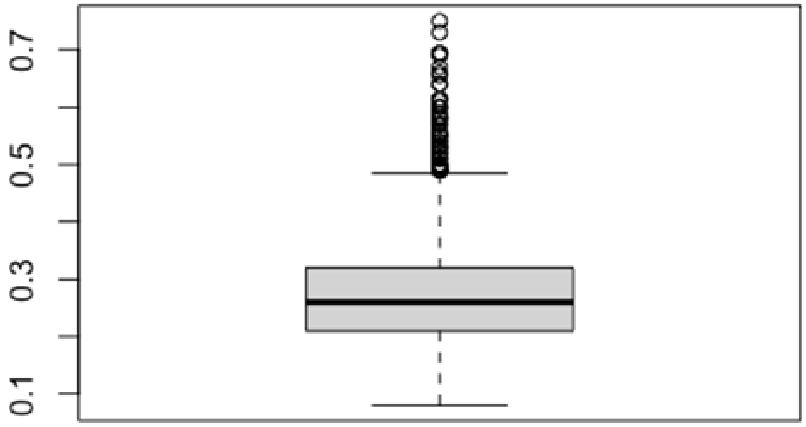


There are many outliers, and a new data set titled wineQualitySubsample5 is created after the outliers are removed. WineQualitySubsample5 has the following dimensions and boxplot.

```
## [1] 1923    12
```

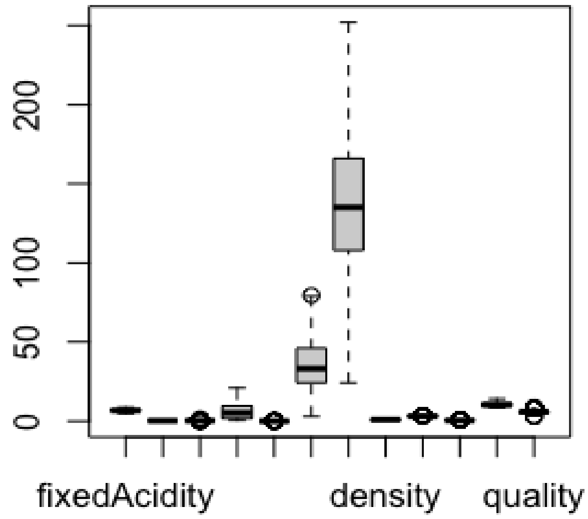


There are now no outliers for “fixedAcidity.” We will proceed to clean the next variable to its right called “volatileAcidity.”

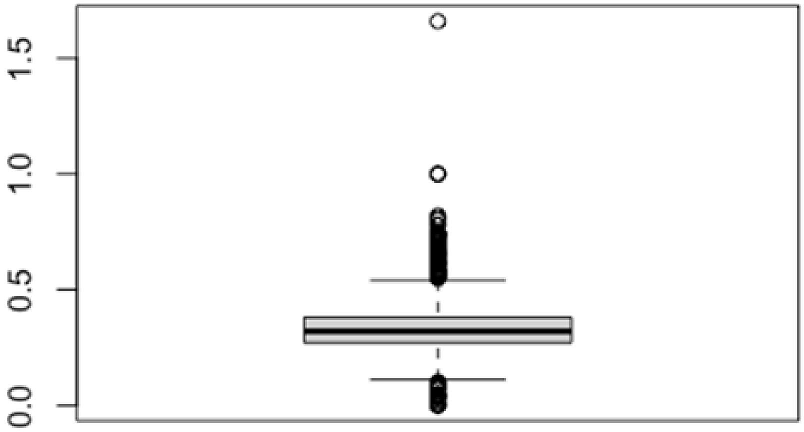


This variable clearly has many outliers, and a new data set called wineQualitySubsample6 is produced after they are removed. This further cleaned data set has the dimensions and boxplot below.

```
## [1] 1867    12
```

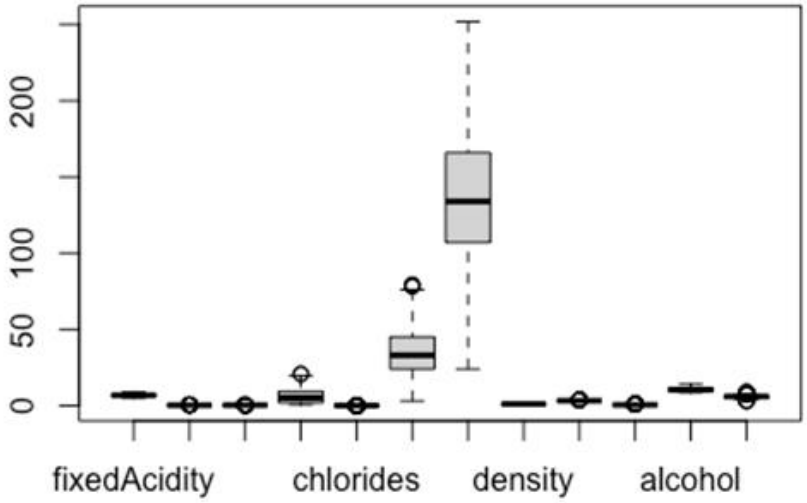


Now that the “volatileAcidity” variable has been cleaned, we must target the variable immediately to its right, “citricAcid.” This variable produces the boxplot below to help us continue data preprocessing.

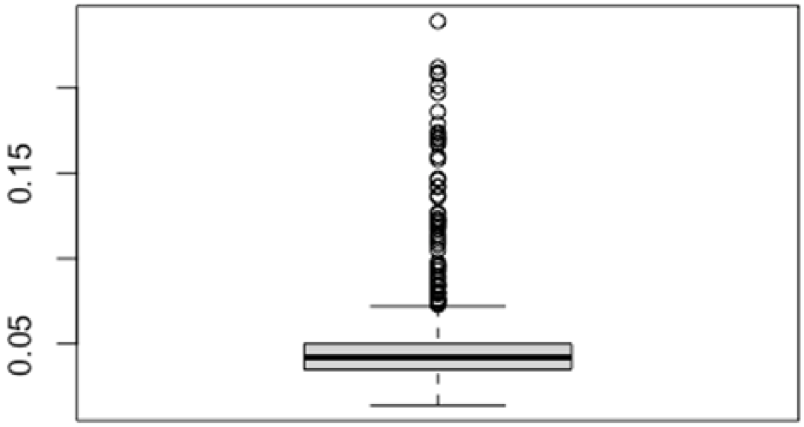


After removing the outliers, we saved the cleaned data set as wineQualitySubsample7. This data frame has the following dimensions and boxplot.

```
## [1] 1750   12
```

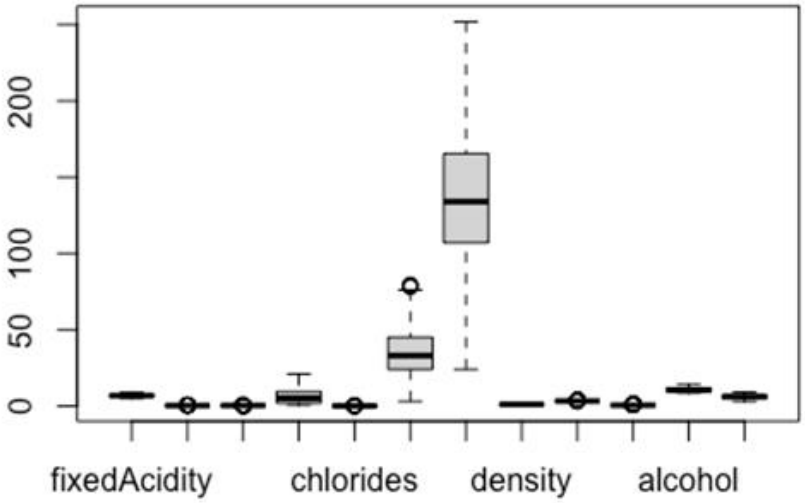


Although this new data frame resulted in new outliers for “citricAcid” and “volatileAcidity,” we will not do further outlier removal for these variables. Therefore, we will proceed with cleaning the “chlorides” variable. To begin this step, we will visualize the variable with a simple boxplot.

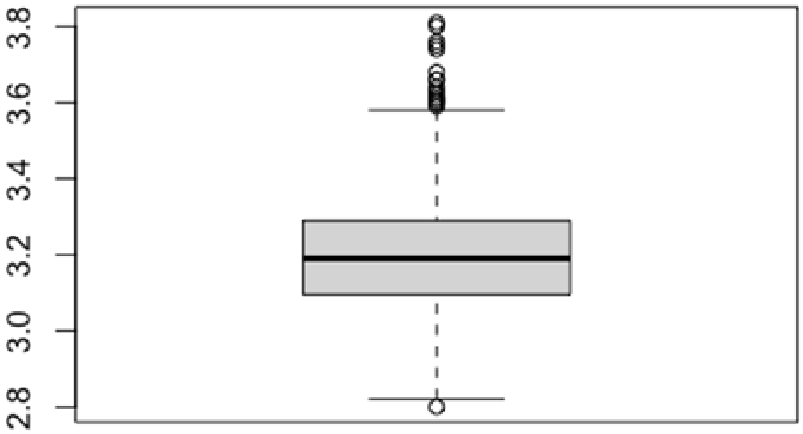


There are many outliers to remove, and the reduced data set is called wineQualitySubsample8, which yields the following dimensions and boxplot.

```
## [1] 1687    12
```

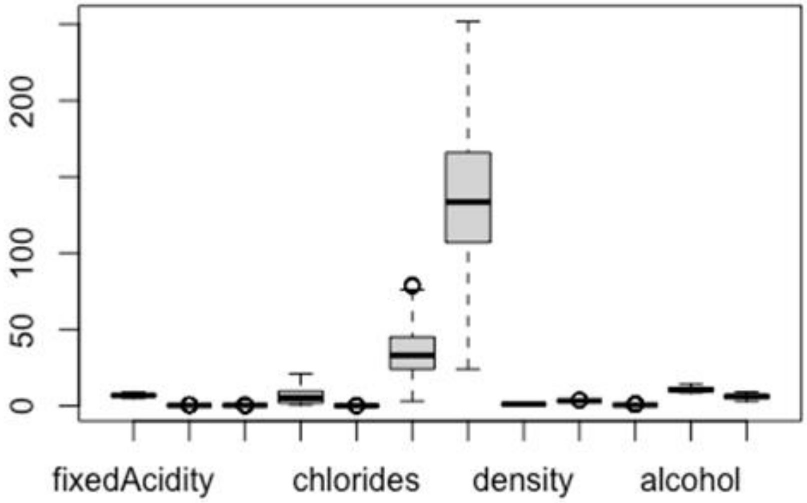


The next variable to preprocess is “pH” (column 8), as the “density” variable now has no outliers. Should new ones be created in the cleaning process, they will be removed. The “pH” boxplot is below.

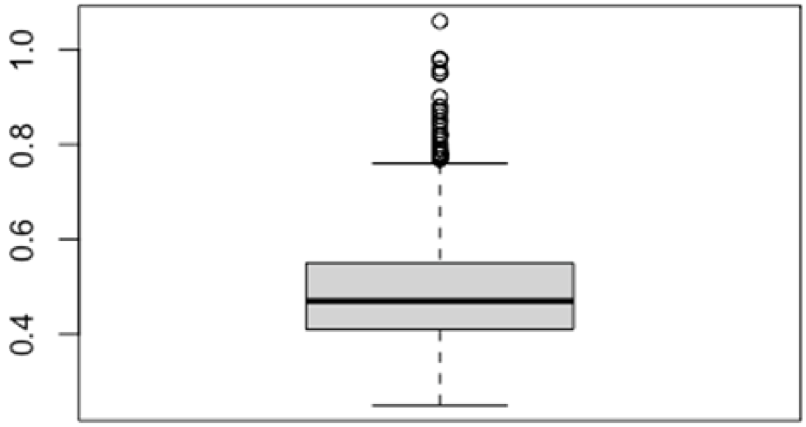


A new data set titled wineQualitySubsample9 is created with the observed dimensions and boxplot below.

```
## [1] 1670   12
```

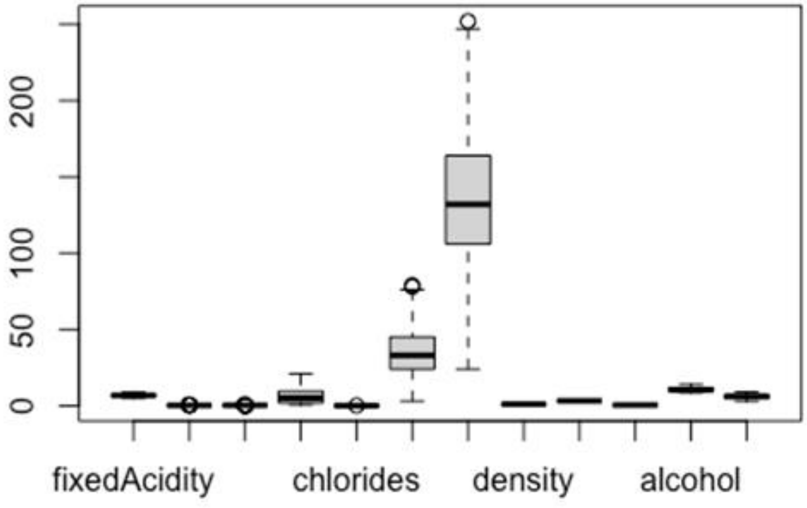


The variable titled “sulphates” is the next to be processed, and this occurs at column 10. Its boxplot shows there are many outliers to be removed, which should hopefully take care of some of the new outliers that pop up in other variables.



Cleaning this variable effectively reduces the data set by nearly 300 observations and removes the outliers that emerged in many of the other variables throughout preprocessing. The final data set is titled wineQualitySubsample10, and the following dimensions and boxplot are observed.

```
## [1] 1383    12
```



Outlier Summary:
After outlier removal, the data set now has 1383 observations. Therefore, 617 observations were removed, empowering us to work off leaner and more insightful data.

Create the training and testing datasets:

We will use the 80/20 rule to create two separate data sets. Eighty percent of the data (1106 observations) will be used to train the model, and the other 20% (277 observations) will test the efficacy of the developed model. A random sample is generated to create these new data sets, and they will be saved to the file paths "wineTrain.txt" and "wineTest.txt" for the classification phase. However, we must conduct principal component analysis on the predictor variables for dimension reduction.

Apply PCA to the training dataset to reduce the dimensions:

There are 10 quantitative predictor variables and one categorical response variable, which is "quality." Therefore, we can use the simple *prcomp()* command in R. Our goal is to select the number of principal components that account for 80% of the cumulative proportion of variance in wine quality. The following R output summarized the PCA.

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8653 1.2175 1.1108 1.03685 1.00531 0.90188 0.83395
## Proportion of Variance 0.3163 0.1348 0.1122 0.09773 0.09188 0.07394 0.06322
## Cumulative Proportion  0.3163 0.4511 0.5632 0.66095 0.75283 0.82677 0.89000
##                            PC8     PC9    PC10    PC11
## Standard deviation     0.74268 0.61329 0.51900 0.11390
## Proportion of Variance 0.05014 0.03419 0.02449 0.00118
## Cumulative Proportion  0.94014 0.97433 0.99882 1.00000
```

Based on our requirements, we will use the first six principal components. The various linear combinations for the PCA are below and highlighted in green are variables with |r| > 0.5.

```
## Rotation (n x k) = (11 x 11):
 ##                             PC1          PC2         PC3         PC4
 ## fixedAcidity       -0.129334724  0.632756675 -0.08079289  0.15304481
 ## volatileAcidity     0.004577761 -0.105877055  0.54416598  0.17917780
 ## citricAcid         -0.028690075  0.286537317 -0.49001284  0.44006701
 ## residualSugar      -0.416892399  0.005143333  0.24476442 -0.02746874
 ## chlorides          -0.340831686 -0.005872142 -0.17888228 -0.35768298
 ## freeSulfurDioxide  -0.286618423 -0.247042553  0.07107378  0.55567759
 ## totalSulfurDioxide -0.390236342 -0.226788290  0.06840626  0.38802675
 ## density            -0.503318760  0.027167999 -0.02014628 -0.15904831
 ## pH                  0.103710990 -0.593307968 -0.30600952 -0.09138116
 ## sulphates          -0.048248284 -0.202425845 -0.49338905  0.13898976
 ## alcohol             0.437738253 -0.007269841  0.13462342  0.33138286
 ##                             PC5          PC6          PC7           PC8
 ## fixedAcidity        0.188025659  0.152524923 -0.209876943 -6.338472e-01
 ## volatileAcidity     0.683855029 -0.325847274 -0.038368002  2.534922e-02
 ## citricAcid         -0.001326129 -0.624457462  0.155832686  2.519524e-01
 ## residualSugar      -0.097407160 -0.093831518  0.572226585 -1.011243e-05
 ## chlorides           0.142131612 -0.165093246 -0.574825406  3.080674e-01
 ## freeSulfurDioxide  -0.360031927  0.226187184 -0.170218392  2.766512e-02
 ## totalSulfurDioxide  0.065428613  0.015321367 -0.312593764 -5.416160e-02
 ## density             0.039917516 -0.078375217  0.275623326 -2.008638e-01
 ## pH                  0.005853865 -0.357338327 -0.017988761 -6.109057e-01
 ## sulphates           0.575982237  0.509234149  0.263268470  1.447761e-01
 ## alcohol            -0.003199524  0.002070346  0.006079928 -3.160155e-02
 ##                           PC9        PC10         PC11
 ## fixedAcidity        0.13291438 -0.06703214  0.162338203
 ## volatileAcidity    -0.04994582 -0.28650664  0.004280330
 ## citricAcid         -0.06098111 -0.04262149  0.007326674
 ## residualSugar       0.41689787  0.17100813  0.465539191
 ## chlorides           0.49959308 -0.02209309  0.031544001
 ## freeSulfurDioxide   0.16632068 -0.55150729 -0.025389765
 ## totalSulfurDioxide -0.27926302  0.67530518  0.043006709
 ## density            -0.01327230 -0.03528791 -0.771205515
 ## pH                  0.11397633 -0.06665766  0.128643549
 ## sulphates           0.10205463 -0.02011201  0.042189727
 ## alcohol             0.65076194  0.33997325 -0.374528029
```

PCA Conclusions:

Based on the observed |r|values of the variables across six principal components, we will move forward with only using “density,” “fixedAcidity,” “pH,” “volatileAcidity,” “citricAcid,” “freeSulfurDioxide,” and “sulphates.” The removal of four variables effectively reduces the dimensions of our data set by 36%. We can now write two new files named “wineTestReduced.txt” and “wineTrainReduced.txt” that only have the seven variables that came out of PCA and the “quality” response variable. Below is a summary of the variables included in these outputs.

Variable	Type
density	integer
fixedAcidity	integer
pH	integer
volatileAcidity	integer
citricAcid	integer
freeSulfurDioxide	integer
sulphates	integer

quality	factor (response)

V. Applying Classification Techniques and Performing Predictions

A. CreditCardDefaultData

The CreditCardDefaultData usecase is a binary classification that attempts to predict if a borrower will default on a loan. In this case, as described above, there is a combination of quantitative and qualitative variables. For our first classification of this data, we will leverage the logistic regression technique.

1. Logistic Regression

The logic regression is intended to be used to predict the likelihood of an outcome based on a set of input variables. In this case, we are working to predict whether a borrower will default on a given loan based on several quantitative and qualitative variables developed in the previous sections of this report.

The first step in our logistic regression was to create the logistic model. The PAY_0-PAY_6 variables were of interest to us during this step. These are numeric variables that represent whether the borrower is delayed in payment over the last 6 months, and how far. After experimenting with the data and creating multiple models, we decided to use the PAY_x variables as factor variables in the model, as it produced better predictive results.

```
Call:
glm(formula = default.payment.next.month ~ as.factor(EDUCATION) + 
    as.factor(MARRIAGE) + AGE + as.factor(PAY_2) + as.factor(PAY_3) + 
    as.factor(PAY_4) + as.factor(PAY_5) + as.factor(PAY_6) + 
    BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 + BILL_AMT5 + 
    BILL_AMT6, family = "binomial", data = trainDataSet)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.0094  -0.7158  -0.6178   0.7326   2.2722  

Coefficients: (3 not defined because of singularities)
                        Estimate Std. Error z value Pr(>|z|)   
(Intercept)           -2.834e+00  1.215e+00  -2.333  0.01966 * 
as.factor(EDUCATION)2  1.436e-01  1.341e-01   1.071  0.28428   
as.factor(EDUCATION)3  2.024e-01  1.726e-01   1.173  0.24077   
as.factor(EDUCATION)4 -1.407e+01  8.098e+02  -0.017  0.98614   
as.factor(EDUCATION)5 -5.388e-01  6.552e-01  -0.822  0.41089   
as.factor(EDUCATION)6 -1.398e+01  8.400e+02  -0.017  0.98672   
as.factor(MARRIAGE)1   1.891e+00  1.172e+00   1.613  0.10666   
as.factor(MARRIAGE)2   1.852e+00  1.176e+00   1.576  0.11508   
as.factor(MARRIAGE)3   6.913e-01  1.356e+00   0.510  0.61026   
AGE                   -8.300e-03  7.028e-03  -1.181  0.23757   
as.factor(PAY_2)-1    -5.309e-02  3.117e-01  -0.170  0.86477   
as.factor(PAY_2)0     -1.167e-01  3.627e-01  -0.322  0.74770   
as.factor(PAY_2)1      1.665e+00  1.446e+00   1.151  0.24954   
as.factor(PAY_2)2      1.159e+00  3.553e-01   3.261  0.00111 **
as.factor(PAY_2)3      1.472e+00  5.442e-01   2.706  0.00681 **
as.factor(PAY_2)4     -4.785e-01  9.082e-01  -0.527  0.59827   
as.factor(PAY_2)5     -1.131e+00  2.058e+03  -0.001  0.99956   
as.factor(PAY_2)7      2.898e+01  2.058e+03   0.014  0.98877   
as.factor(PAY_3)-1    -2.152e-01  3.741e-01  -0.575  0.56517   
as.factor(PAY_3)0      2.481e-01  4.295e-01   0.577  0.56360   
as.factor(PAY_3)1     -1.608e+01  1.455e+03  -0.011  0.99118   
as.factor(PAY_3)2      2.566e-01  4.242e-01   0.605  0.54520   
as.factor(PAY_3)3      1.249e+00  7.126e-01   1.752  0.07970 . 
as.factor(PAY_3)4     -1.166e+00  1.561e+00  -0.747  0.45508   
as.factor(PAY_3)5      1.510e+01  1.455e+03   0.010  0.99172   
as.factor(PAY_3)6             NA         NA      NA       NA   
as.factor(PAY_3)7      1.444e+01  1.455e+03   0.010  0.99208   
as.factor(PAY_4)-1     3.039e-01  3.873e-01   0.785  0.43274   
as.factor(PAY_4)0      2.457e-01  4.391e-01   0.560  0.57578   
as.factor(PAY_4)2      5.862e-01  4.531e-01   1.294  0.19577   
as.factor(PAY_4)3      1.129e+00  8.599e-01   1.313  0.18924   
as.factor(PAY_4)4     -8.486e-01  1.407e+00  -0.603  0.54639   
as.factor(PAY_4)5     -2.799e+01  1.597e+03  -0.018  0.98601   
as.factor(PAY_4)7      1.372e+00  1.150e+00   1.193  0.23278  
as.factor(PAY_5)-1    -2.257e-01  3.564e-01  -0.633  0.52659  
as.factor(PAY_5)0     -1.543e-01  4.089e-01  -0.377  0.70581  
as.factor(PAY_5)2      4.388e-01  4.361e-01   1.006  0.31436  
as.factor(PAY_5)3      1.305e+00  9.308e-01   1.402  0.16095  
as.factor(PAY_5)4      1.500e+01  6.571e+02   0.023  0.98179  
as.factor(PAY_5)5      3.718e-02  1.604e+03   0.000  0.99998  
as.factor(PAY_5)7             NA         NA      NA       NA  
as.factor(PAY_6)-1    -1.671e-01  2.719e-01  -0.615  0.53885  
as.factor(PAY_6)0     -2.396e-01  3.015e-01  -0.795  0.42690  
as.factor(PAY_6)2      1.373e-01  3.485e-01   0.394  0.69363  
as.factor(PAY_6)3      1.139e+00  7.387e-01   1.542  0.12316  
as.factor(PAY_6)4      1.453e+01  6.732e+02   0.022  0.98277  
as.factor(PAY_6)5     -1.364e+01  2.521e+03  -0.005  0.99568  
as.factor(PAY_6)6      1.391e+01  1.455e+03   0.010  0.99237  
as.factor(PAY_6)7             NA         NA      NA       NA  
BILL_AMT1             -1.754e-05  1.019e-05  -1.721  0.08528 .
BILL_AMT2              1.403e-05  1.356e-05   1.035  0.30069  
BILL_AMT3              7.819e-06  1.223e-05   0.640  0.52248  
BILL_AMT4             -4.684e-06  1.287e-05  -0.364  0.71591  
BILL_AMT5             -9.621e-06  1.535e-05  -0.627  0.53085  
BILL_AMT6             -4.322e-06  1.254e-05  -0.345  0.73038  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2376.9  on 1987  degrees of freedom
Residual deviance: 2043.4  on 1936  degrees of freedom
AIC: 2147.4

Number of Fisher Scoring iterations: 14
```

Logistic Regression Model Analysis

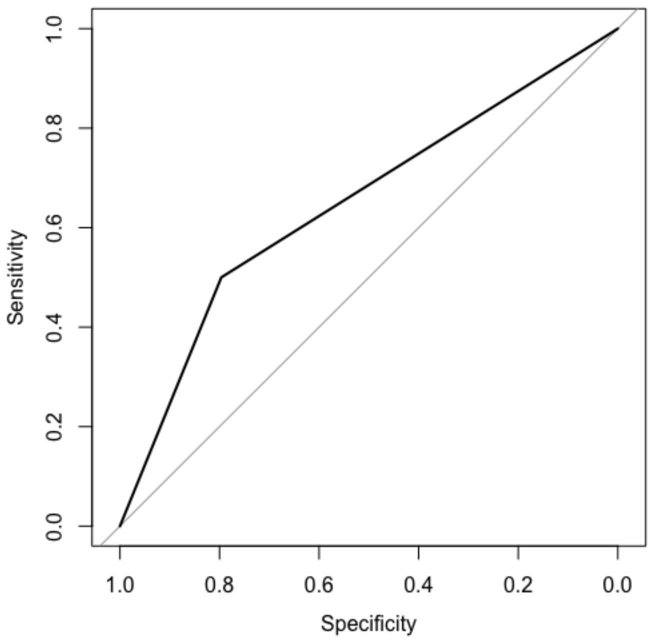
Once the logistic model is planned and run, it is important to analyze the model to understand the quality of the predictions that are being made. To accomplish the model, that was created with a training dataset, is executed against a separate testing dataset.

Some of the key performance indexes of this model against the testing data are:

Accuracy	= 0.7641129	= 76.41%
Error Rate	= 0. 2358871	= 23.59%
Sensitivity (TPR)	= 0.2307692	= 23.08%
Total Negative Rate (TNR)	= 0.9287599	= 92.88%
Specificity (FPR)	= 0.07124011	= 7.12%

The Receiver Operating Characteristic Curve (ROC Curve) will plot the sensitivity against the specificity. The ideal model characteristics will have a low specificity (FPR) and a high sensitivity (TPR). Visually the graph helps understand if this is true. Additionally, it is helpful to calculate the Area Under the Curve (AUC) as this provides a numeric representation of the quality of the model. The graph below represents the ROC curve for our CreditCardDefaultData model, and as can be seen in the output the AUC is 0.6482.

```
Call:
roc.default(response = predRnd, predictor = as.numeric(testDataSet$default.payment.next.month))

Data: as.numeric(testDataSet$default.payment.next.month) in 442 controls (predRnd 0) < 54 cases
(predRnd 1).
Area under the curve: 0.6482
```



2. Logistic Regression Analysis Summary

After analysis, this model is not particularly strong. It has an accuracy of 76.41%. I believe that we would be wise to revisit earlier stages of the project and reexamine the data for missing factors.

3. Decision Tree Classification

The decision tree classification is, like the logistic regression, intended to be used to predict the likelihood of an outcome based on a set of input variables. In this case, we are still working to predict whether a borrower will default on a given loan based on several quantitative and qualitative variables developed in the previous sections of this report. The decision tree algorithm is often considered easier to understand and explain, so applying this algorithm to this dataset makes sense.

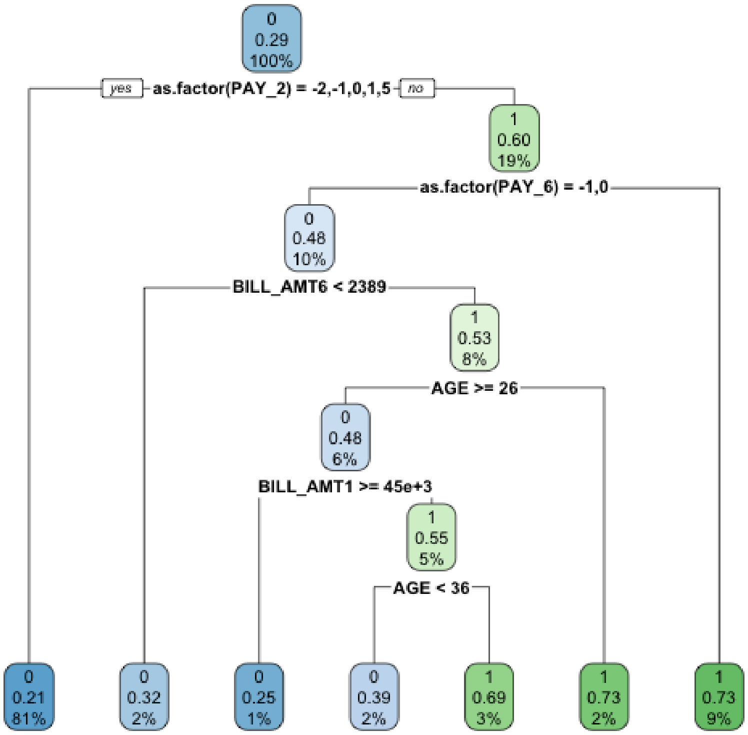
The first step in our decision tree classification is to create the decision tree model. As before, the PAY_0-PAY_6 variables are being used as factor variables in this model.

Decision Tree Model Analysis

The decision tree model created by our tools is shown below in both table output form and graphical form. It is clear from both formats which predictors were used in the tree, as well as the thresholds used in the decisions. The list of predictors included in the tree are: PAY_2, PAY_6, BILL_AMT6, AGE(Used in multiple branches), and BILL_AMT1.

```
n= 1988 

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 1988 567 0 (0.7147887 0.2852113)  
    2) as.factor(PAY_2)=-2,-1,0,1,5 1605 336 0 (0.7906542 0.2093458) *
    3) as.factor(PAY_2)=2,3,4,7 383 152 1 (0.3968668 0.6031332)  
```

```
6) as.factor(PAY_6)=-1,0 200  97 0 (0.5150000 0.4850000)
 12) BILL_AMT6< 2388.5 44  14 0 (0.6818182 0.3181818) *
 13) BILL_AMT6>=2388.5 156  73 1 (0.4679487 0.5320513)
   26) AGE>=25.5 123  59 0 (0.5203252 0.4796748)
     52) BILL_AMT1>=44660.5 28   7 0 (0.7500000 0.2500000) *
     53) BILL_AMT1< 44660.5 95  43 1 (0.4526316 0.5473684)
      106) AGE< 35.5 44  17 0 (0.6136364 0.3863636) *
      107) AGE>=35.5 51  16 1 (0.3137255 0.6862745) *
   27) AGE< 25.5 33   9 1 (0.2727273 0.7272727) *
7) as.factor(PAY_6)=-2,2,3,4,6,7 183  49 1 (0.2677596 0.7322404) *
```

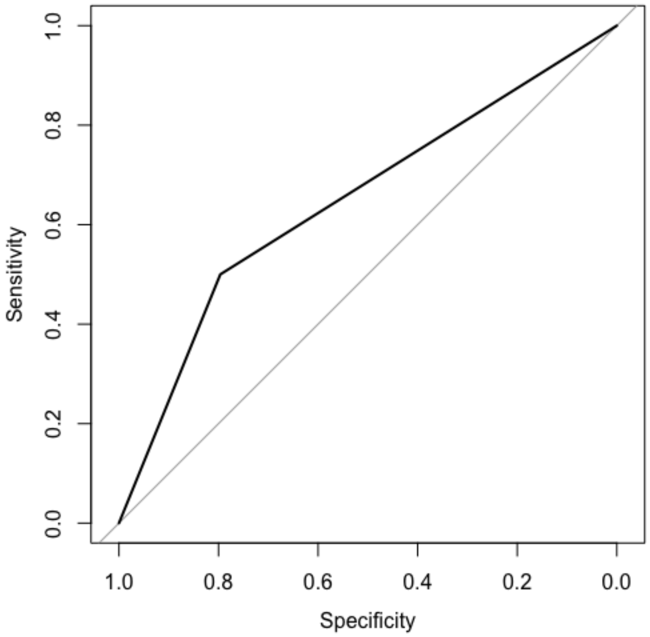


Key performance indexes of this model against the testing data are:

Accuracy	= 0.7641129	= 75.40%
Error Rate	= 0. 2358871	= 24.60%
Sensitivity (TPR)	= 0.2307692	= 23.93%
Total Negative Rate (TNR)	= 0.9287599	= 91.29%
Specificity (FPR)	= 0.08707124	= 8.71%

ROC Curve and AUC Value:

```
Call:
roc.default(response = predDT, predictor = as.numeric(testDataSet$default.payment.next.month))

Data: as.numeric(testDataSet$default.payment.next.month) in 435 controls (predDT 0) < 61 cases
(predDT 1).
Area under the curve: 0.6272
```



4. Decision Tree Classification Summary

The results produced by the decision tree model are very similar to the results that were produced by the logistic regression model. The decision tree did not use all of the predictor variables that were provided to it in the model. In this case it only used 5 of 14 total predictors in the decision tree.

5. Ensemble Methods

Ensemble methods represent an approach that combines several "simple building block" models in an attempt to create a single potentially powerful model. In this case, we will utilize three common ensemble methods to analyze our creditCardDefaultData. The three methods we will leverage are: Bagging Model, Random Forest Model, and Boosting Model. Each of these models will build on the decision tree model.

Bagging Method Enhancement:

The bagging model attempts to decrease the variability in the basic decision tree classification model by incorporating several random samples of data from the original dataset to create multiple independent decisions trees. These decision trees are all done in parallel without dependency upon one another, and are eventually combined into a single output. The following analysis utilizes a bagging model classification with 20 bootstrap replications.

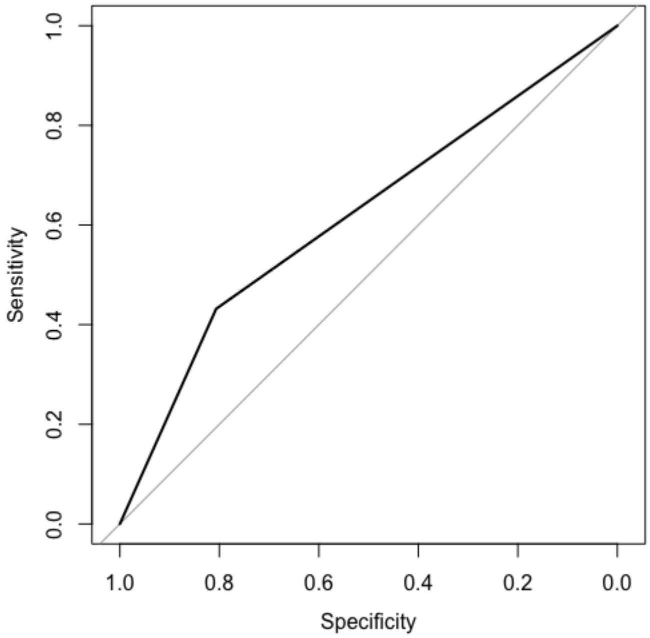
```
Call: bagging.data.frame(formula = default.payment.next.month ~ (EDUCATION) +
    (MARRIAGE) + AGE + (PAY_2) + (PAY_3) + (PAY_4) + (PAY_5) +
    (PAY_6) + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 +
    BILL_AMT5 + BILL_AMT6, data = trainDataSet, nbagg = 20)
```

Accuracy = 0.7399194 = 73.99%

Error Rate	= 0. 2600806	= 26.01%
Sensitivity (TPR)	= 0. 3247863	= 32.48%
Total Negative Rate (TNR)	= 0. 8680739	= 86.81%
Specificity (FPR)	= 0. 1319261	= 13.19%

ROC Curve and AUC Value:

```
Call:
roc.default(response = predBag, predictor = as.numeric(testDataSet$default.payment.next.month))

Data: as.numeric(testDataSet$default.payment.next.month) in 408 controls (predBag 0) < 88 cases
(predBag 1).
Area under the curve: 0.6191
```



Random Forest Method Enhancement

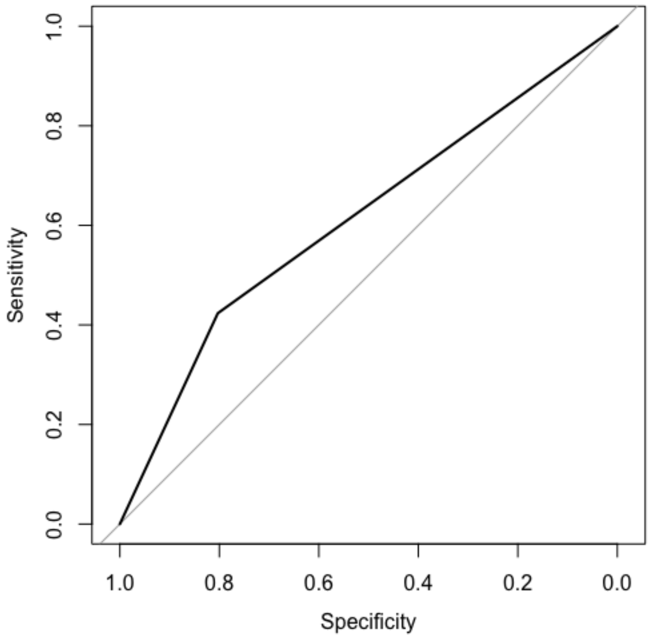
The random forest model is similar to the bagging algorithm discussed above, but with a slight variation. Each time that the random forest method prepares to make a split decision, it takes a random sample of some number of the predictor variables to consider for the next split. The result of this sampling will decorrelate the individual trees. At the end, similar to bagging, the models are combined into a single output. The following model is a random forest model representing the CreditCardDefault data.

```
Call:
 randomForest(formula = default.payment.next.month ~ (EDUCATION) +      (MARRIAGE) + AGE + (PAY_2) +
(PAY_3) + (PAY_4) + (PAY_5) +      (PAY_6) + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 +
BILL_AMT5 + BILL_AMT6, data = trainDataSet)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 26.26%
Confusion matrix:
     0   1 class.error
0 1288 133  0.09359606
1  389 178  0.68606702
```

Accuracy	= 0.7379032	= 73.79%
Error Rate	= 0.2620968	= 26.21%
Sensitivity (TPR)	= 0.3076923	= 30.08%
Total Negative Rate (TNR)	= 0.8707124	= 87.07%
Specificity (FPR)	= 0.1292876	= 12.93%

ROC Curve and AUC Value:

```
Call:
roc.default(response = predRF, predictor = as.numeric(testDataSet$default.payment.next.month))

Data: as.numeric(testDataSet$default.payment.next.month) in 411 controls (predRF 0) < 85 cases
(predRF 1).
Area under the curve: 0.6132
```



Boosting Model Application:

The Boosting method is similar to the bagging algorithm discussed above, but with a slight variation. The boosting method arranges the underlying models in a serial fashion to create a dependency. It uses outputs from the previous model to enhance the input to the next model. At the end, similar to bagging, the models are combined into a single output. The following model is a random forest model representing the CreditCardDefault data.

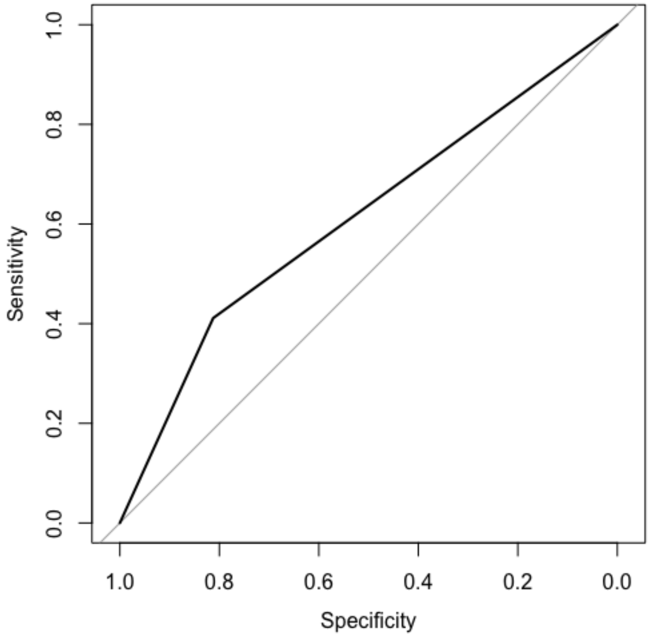
.

```
adaboost(formula = default.payment.next.month ~ (EDUCATION) +
    (MARRIAGE) + AGE + (PAY_2) + (PAY_3) + (PAY_4) + (PAY_5) +
    (PAY_6) + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 +
    BILL_AMT5 + BILL_AMT6, data = trainDataSet, nIter = 20)
default.payment.next.month ~ (EDUCATION) + (MARRIAGE) + AGE +
    (PAY_2) + (PAY_3) + (PAY_4) + (PAY_5) + (PAY_6) + BILL_AMT1 +
    BILL_AMT2 + BILL_AMT3 + BILL_AMT4 + BILL_AMT5 + BILL_AMT6
Dependent Variable: default.payment.next.month
No of trees:20
The weights of the trees
are:0.7821230.69754470.66475850.64218680.64414670.66501310.63705070.64923750.60986920.61814590.627196
10.61411150.58483230.57903640.55388510.52876480.52701730.50889970.476790.4651539
```

Accuracy	= 0.7258065	= 72.58%
Error Rate	= 0.2741935	= 27.42%
Sensitivity (TPR)	= 0.3760684	= 37.61%
Total Negative Rate (TNR)	= 0.8337731	= 83.38%
Specificity (FPR)	= 0.1662269	= 16.62%

ROC Curve and AUC Value:

```
Call:
roc.default(response = predBST$class, predictor = as.numeric(testDataSet$default.payment.next.month))

Data: as.numeric(testDataSet$default.payment.next.month) in 389 controls (predBST$class 0) < 107
cases (predBST$class 1).
Area under the curve: 0.6118
```



6. Ensemble Method Summary

The ensemble methods provide yet another few views of the creditCardDefault data. Through the use of the Bagging, Random Forest, and Boosting methods, we were able to see that the accuracy of our predictions ranged from 72.58% to 73.99% accuracy. The ROC curves for these methods are very similar, as are the AUC values, ranging from 0.6118 to 0.6191.

B. Wine Quality

1. k-Nearest Neighbor Classification

Because the Wine Quality data set has a categorical response variable with more than two levels, we cannot use logistic regression. Therefore, we began the model building process by starting with k-Nearest Neighbor Classification. We will also try out various other statistical learning methods and then compare the accuracy outputs to land on the most helpful model.

kNN Model Analysis

Upon a first run through the data, we observed an accuracy rate of 44% and, thus, were concerned that not normalizing the variables or revisiting the PCA was harming the predictive validity of the model. We started over by normalizing the variables. After doing that, we took the square root of the sample size of the training data set, which is 33. Therefore, we will create a for loop to iterate through k-values between 25 and 50 to find the ideal number for classification.

```
## [1] "k value =25 -->0.48014440433213"
 ## [1] "k value =26 -->0.476534296028881"
 ## [1] "k value =27 -->0.48014440433213"
 ## [1] "k value =28 -->0.494584837545126"
 ## [1] "k value =29 -->0.494584837545126"
 ## [1] "k value =30 -->0.509025270758123"
 ## [1] "k value =31 -->0.498194945848375"
 ## [1] "k value =32 -->0.490974729241877"
 ## [1] "k value =33 -->0.498194945848375"
 ## [1] "k value =34 -->0.476534296028881"
 ## [1] "k value =35 -->0.494584837545126"
 ## [1] "k value =36 -->0.476534296028881"
 ## [1] "k value =37 -->0.469314079422383"
 ## [1] "k value =38 -->0.472924187725632"
 ## [1] "k value =39 -->0.469314079422383"
 ## [1] "k value =40 -->0.451263537906137"
 ## [1] "k value =41 -->0.458483754512635"
 ## [1] "k value =42 -->0.462093862815884"
 ## [1] "k value =43 -->0.451263537906137"
 ## [1] "k value =44 -->0.451263537906137"
 ## [1] "k value =45 -->0.447653429602888"
 ## [1] "k value =46 -->0.458483754512635"
 ## [1] "k value =47 -->0.447653429602888"
 ## [1] "k value =48 -->0.451263537906137"
 ## [1] "k value =49 -->0.447653429602888"
 ## [1] "k value =50 -->0.447653429602888"
```

The highest output is detected at k=30. We will then use the knn() function to produce a prediction for the test data set. After constructing a confusion matrix, the below results were observed:

- Accuracy: 50.18%
- Error: 49.82%

Our hunch that non-normalized data was impairing model validity was correct. However, this number still struck us as low. Therefore, we decided to revisit the PCA and increase from 6 to 7 principal components. By increasing the cumulative variance, we can now add the “residualSugar” and “chlorides” variables.

```
## Rotation (n x k) = (11 x 11):
 ##                             PC1          PC2         PC3         PC4
 ## fixedAcidity       -0.129334724  0.632756675 -0.08079289  0.15304481
 ## volatileAcidity     0.004577761 -0.105877055  0.54416598  0.17917780
## citricAcid         -0.028690075  0.286537317 -0.49001284  0.44006701
## residualSugar      -0.416892399  0.005143333  0.24476442 -0.02746874
## chlorides          -0.340831686 -0.005872142 -0.17888228 -0.35768298
## freeSulfurDioxide  -0.286618423 -0.247042553  0.07107378  0.55567759
## totalSulfurDioxide -0.390236342 -0.226788290  0.06840626  0.38802675
## density            -0.503318760  0.027167999 -0.02014628 -0.15904831
## pH                  0.103710990 -0.593307968 -0.30600952 -0.09138116
## sulphates          -0.048248284 -0.202425845 -0.49338905  0.13898976
## alcohol             0.437738253 -0.007269841  0.13462342  0.33138286
##                             PC5          PC6          PC7           PC8
## fixedAcidity        0.188025659  0.152524923 -0.209876943 -6.338472e-01
## volatileAcidity     0.683855029 -0.325847274 -0.038368002  2.534922e-02
## citricAcid         -0.001326129 -0.624457462  0.155832686  2.519524e-01
## residualSugar      -0.097407160 -0.093831518  0.572226585 -1.011243e-05
## chlorides           0.142131612 -0.165093246 -0.574825406  3.080674e-01
## freeSulfurDioxide  -0.360031927  0.226187184 -0.170218392  2.766512e-02
## totalSulfurDioxide  0.065428613  0.015321367 -0.312593764 -5.416160e-02
## density             0.039917516 -0.078375217  0.275623326 -2.008638e-01
## pH                  0.005853865 -0.357338327 -0.017988761 -6.109057e-01
## sulphates           0.575982237  0.509234149  0.263268470  1.447761e-01
## alcohol            -0.003199524  0.002070346  0.006079928 -3.160155e-02
##                           PC9        PC10         PC11
## fixedAcidity        0.13291438 -0.06703214  0.162338203
## volatileAcidity    -0.04994582 -0.28650664  0.004280330
## citricAcid         -0.06098111 -0.04262149  0.007326674
## residualSugar       0.41689787  0.17100813  0.465539191
## chlorides           0.49959308 -0.02209309  0.031544001
## freeSulfurDioxide   0.16632068 -0.55150729 -0.025389765
## totalSulfurDioxide -0.27926302  0.67530518  0.043006709
## density            -0.01327230 -0.03528791 -0.771205515
## pH                  0.11397633 -0.06665766  0.128643549
## sulphates           0.10205463 -0.02011201  0.042189727
## alcohol             0.65076194  0.33997325 -0.374528029
```

We will then build another for loop that iterates through 25 and 50, but this time for an expanded, normalized data set. The result of the for loop is below.

```
## [1] "k value =25 -->0.458483754512635"
 ## [1] "k value =26 -->0.472924187725632"
 ## [1] "k value =27 -->0.447653429602888"
 ## [1] "k value =28 -->0.476534296028881"
 ## [1] "k value =29 -->0.462093862815884"
 ## [1] "k value =30 -->0.462093862815884"
 ## [1] "k value =31 -->0.458483754512635"
 ## [1] "k value =32 -->0.469314079422383"
 ## [1] "k value =33 -->0.451263537906137"
 ## [1] "k value =34 -->0.458483754512635"
 ## [1] "k value =35 -->0.469314079422383"
 ## [1] "k value =36 -->0.462093862815884"
 ## [1] "k value =37 -->0.454873646209386"
 ## [1] "k value =38 -->0.458483754512635"
 ## [1] "k value =39 -->0.469314079422383"
 ## [1] "k value =40 -->0.458483754512635"
 ## [1] "k value =41 -->0.454873646209386"
 ## [1] "k value =42 -->0.462093862815884"
 ## [1] "k value =43 -->0.462093862815884"
 ## [1] "k value =44 -->0.462093862815884"
 ## [1] "k value =45 -->0.462093862815884"
 ## [1] "k value =46 -->0.458483754512635"
 ## [1] "k value =47 -->0.462093862815884"
 ## [1] "k value =48 -->0.454873646209386"
 ## [1] "k value =49 -->0.472924187725632"
 ## [1] "k value =50 -->0.462093862815884"
```

The highest outcome is observed at k = 28. The number already looks lower than the output from the leaner data set, but we will still complete the process by running the knn() function. After running the function and completing a confusion matrix, we observed the following validity measures:

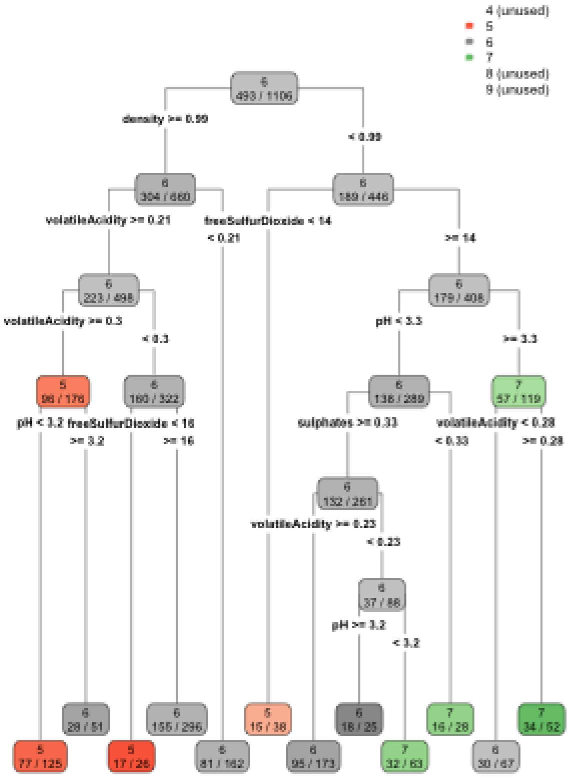
- Accuracy: 45.85%
- Error: 54.15%

2. k-Nearest Neighbor Model Analysis

As we can see, increasing the number of principal components harmed the accuracy of the model. Our initial hunch to settle for an 80% cumulative proportion of variance to remove enough variables was well-founded and resulted in a 50.18% accuracy. This certainly is not as high as we would like, so it’s important to conduct a decision tree analysis and use the ensemble methods.

3. Decision Tree

The next step is to use a decision tree to see if the machine can build a network based on the independent variables that successfully predicts the wine quality. The tree visual is below, but the multiclass nature of the response variable does not necessarily make interpreting the intricacies of it useful. Rather, we must focus on the discrepancies between predicted and actual values when we test the model.



Based on the confusion matrix below that plots actual versus predicted values, we observed the following accuracy and error rates:

```
##        Pred
 ## Actual  3  4  5  6  7  8  9
 ##      3  0  0  0  1  0  0  0
 ##      4  0  0  4  1  0  0  0
 ##      5  0  0 27 46  4  0  0
 ##      6  0  0 13 89 15  0  0
 ##      7  0  0  4 48 13  0  0
 ##      8  0  0  0  8  4  0  0
```

- Accuracy: 46.57%

- Error: 53.43%

4. Decision Tree Summary

The decision tree algorithm was, unfortunately, less accurate than the kNN algorithm. This could be due to the less robust nature of decision trees compared to its accompanying ensemble methods that use summations of series of models and vectors for fitting. Thus, we must proceed with digging deeper into the decision tree-based learning methods.

5. Ensemble Methods

The ensemble methods are natural extensions of the decision tree model that aim to increase accuracy, whether that is through creating a series of decision trees and averaging the observations, decorrelating the variables at each split in the tree, or using vectors to repeatedly fit new models until the closest approximation is found. Because our response variable is greater than two levels and the boosting package in R we are using only takes binary variables, we will only be conducting bagging and random forest.

Bagging Method Enhancement:

We must consider taking the summation of multiple random samples of decision trees and averaging their responses to create a model with greater predictive accuracy. Through this bootstrapping process, we can decrease the variance in responses that is often observed with simple decision trees. After building a bagging model and predicting the test values, we were able to build a confusion matrix.

```
##        Pred
 ## Actual  3  4  5  6  7  8  9
 ##      3  0  0  0  1  0  0  0
 ##      4  0  0  4  1  0  0  0
 ##      5  0  1 43 32  1  0  0
 ##      6  0  1 21 75 18  2  0
 ##      7  0  0  5 28 30  2  0
 ##      8  0  0  0  2  5  5  0
```

Based on the confusion matrix, we observed the following accuracy and error rates:

- Accuracy: 55.23%
- Error: 44.77%

The bootstrapping process clearly improved our model's accuracy compared to both the kNN model and the decision tree. We cannot measure sensitivity and specificity because our product is not binary, so this model appears to be the strongest thus far.

Random Forest Method Enhancement

Through randomly sampling predictor variables to be considered at each break in a series of bootstrapped decision trees, we can de-correlate variables and potentially increase our model's accuracy. The random forest algorithm effectively does this. After running our test data set through the algorithm, we were able to plot its predictions against the actual values in the test sample. The resulting confusion matrix and accuracy and error rates are below.

```
##   3 4   5   6  7  8 9 class.error
 ## 3 0 0   1   1  0  0 0   1.0000000
 ## 4 0 1  14   7  2  0 0   0.9583333
 ## 5 0 1 159 136  7  0 0   0.4752475
 ## 6 0 2  75 362 53  1 0   0.2657201
 ## 7 0 0  14 122 99  0 0   0.5787234
 ## 8 0 0   2  23  8 14 0   0.7021277
 ## 9 0 0   0   1  1  0 0   1.0000000
```

- Accuracy: 59.57%
- Error: 40.43%

De-correlating the variables during the bootstrapping process successfully helped us achieve the highest accuracy rate of any of our models at nearly 60%.

6. Ensemble Methods Summary

The added benefits of the ensemble methods over a simple decision tree were very evident with our accuracy and error rates. For each added layer of reducing variance, the model accuracy went from roughly 47% (decision tree) to 55% (bagging) to 60% (random forest). Therefore, the ensemble methods helped us create the most accurate of all the models we created, with random forest performing best at 60%.

VI. Interpret the Predictions

A. CreditCardDefaultData

To now, we have used several modeling techniques to analyze the creditCardDefault dataset. We have applied Logistic Regression, Decision Tree analysis, Bagging, Random Forest, and Boosting all to the dataset. The application of these models was after a significant amount of time and effort to numerically and visually understand the data, and leverage Principle Component Analysis (PCA) to decimate the number parameters to only those variables that are statistically significant to the dataset and the prediction.

The result of all of this effort has yielded results that are interesting in their own right. When looking at the CreditCardDefault dataset, the goal is to create a model that can accurately predict which borrowers are likely to default on their loan. The summary of the high-level results is listed in the table below.

Model	ACC	ERR Rate	TPR	TNR	FPR	AUC
Logistic Regression	76.41%	23.59%	23.08%	92.88%	7.12%	0.6482
Decision Tree	75.40%	24.60%	23.93%	91.29%	8.71%	0.6272
Bagging	73.99%	26.01%	32.48%	86.81%	13.19%	0.6191
Random Forest	73.79%	26.21%	30.08%	87.07%	12.93%	0.6132
Boosting	72.58%	27.42%	37.61%	83.38%	16.62%	0.6118

Given the above table, it is clear that the accuracy of all of the classification models is relatively close, but that the logistic regression is the best. The AUC (0.6118-0.6482) would indicate that the classifications are all fairly poor in quality, even with greater than 75% accuracy. Another observation is that the negative (not default) prediction accuracy is significantly higher than the positive(default) prediction accuracy. This model is better at predicting that users will not default than the users that will default. The intrinsic value of the model would likely be higher if the ability to predict borrowers that will default were higher.

B. Wine Quality

We have employed a variety of modeling techniques to best test the overall accuracy of our models in determining wine quality. We employed the k-nearest neighbors, decision tree, bagging, and random forest models. We then created confusion matrices to determine how accurate these modeling techniques were. The accuracy and error rates for the various models are shown in the table below.

Model	ACC	ERR Rate

kNN	50.18%	49.82%
Decision Tree	46.57%	53.43%
Bagging	55.23%	44.77%
Random Forest	59.57%	40.43%

The random forest method of testing was shown to be the most accurate, at a rate of 59.57%. There is a clear gap in accuracy between our least accurate model (decision tree) and our most accurate model (random forest). It should be noted that across all models, we were able to best predict what wines would score either a 5 or a 6. Thus, the models were most successful in identifying average quality wines compared to higher or lower levels of quality.

VII. Compare and discuss the results

A. CreditCardDefaultData

The team learned a lot through this project. The mechanics of performing each step was very valuable. We ran into a few anomalies along the way and were able to navigate our way through each one.

The first anomaly we ran into was that the some of the variables (PAY_0-PAY_6) were numeric variables that actually represented a factor indicating the number of months delayed payments were being made. Utilizing these numeric variables as factors proved to improve the predictions. Additionally, when we looked at removing outliers on these variables (PAY_0 – PAY_6), the data was so weighted toward 0 (paying on time) that the tools were identifying any of the values > 2 (2 months delayed on payment) as outliers. Removing information about borrowers that were more than 2 months behind on their payments, would likely have removed very important data, possibly about those most likely to default on a loan. For this reason, we decided to not remove outliers in these variables

Secondly, we learned that PCA application to numeric only variables was not as effective as when we transitioned to the PCAmixdata implementation. This transition allowed us to include quantitative and qualitative variables both in the PCA process, and improved our predictions.

The final anomaly we saw was that our testing dataset contained a level for a categorical variable that was not included in the training dataset. When this happens, the predict function would return an error, as there is a value that was not trained on. There may be more robust ways of dealing with this situation, but, since there was only one sample that fell into this category, we simply removed the sample.

One further observation about the data is that there are significantly more samples in the dataset that predict a borrower will <u>not</u> default on a particular loan. In the training dataset, borrowers are 2.5 times as likely not to default, while in our testing dataset it is over 3 times as likely. As such, the accuracy could be inflated if the model is biased to the negative. In fact, and maybe ironically, if the model predicted every sample in the testing dataset to be "negative", the accuracy would be 76.41% (379/496), which is exactly the accuracy we got for the logistic regression. This can be seen in the results as well. The TNR indicates that the negative predictions are much higher accuracy than the TPR values.

Given more time with this project, we would probably recommend going back to the data preparation phase and try more approaches to combine the variables to improve the accuracy. We could probably create some combinations of the data that would be interesting and possibly increase the accuracy. One example of this could be looking at how the payment delay trended between 6 months ago and 1 month ago. Perhaps if the borrower had fallen further behind on payments, this might have provided additional insight.

Overall, the predictions for the CreditCardDefault data were reasonable, but far from stellar. This was a great learning experience, but the resulting model would not recommended for use in predicting default at any scale.

B. Wine Quality

Conceptually the team had to approach the wine dataset differently than the prior set. This was because the wine data only had a single factor variable amongst its variables. This makes many traditional tests outside of basic correlation somewhat ineffective, so we were forced to change the way we think about testing and modeling the wine data.

An early challenge we came across was the sheer number of outliers that had taken residence within our sample data. The initial attempts at visualization for our data set resulted in plots that had little explanatory power in describing relationships and patterns among our independent variables. So, we felt that the best course of action would be to use post-processed data for our final visualizations.

We found low accuracy rates and high error rates to be a constant struggle throughout the whole analytics process compared to the prior dataset. It is possible that the nature of the response variable involves too much subjective analysis that is not predicated on objective criteria. The individual tastes and preferences of people evaluating a wine's quality is not necessarily empirical in nature. The very nature of the response variable potentially adds confounding factors to the reliability and variance in our data. In addition, the UCI website where this data set was provided stated that the original data set had around 30 predictor variables. For some reason, more than 15 independent variables were removed before the inception of this project, which means no amount of PCA could fully explain the impact of our quantitative predictors on the wine quality. Had we been given the unaltered data set and conducted analysis on every data point instead of a sample, we may have had much more robust models with higher accuracy and lower error rates.