CSCI-4047-901
Tyler Burleson
Final Project

A) Data Cleaning:

For my data cleaning I started by opening the CSV file and making a copy with the identifier columns, low variance, and too many missing value columns deleted. This left me with 12 remaining columns. I then went ahead and did a find and replace on the keyword Unknown, and replaced it with NA. This would allow me to omit unknown/ NA data for upcoming tasks. Next I utilized kNN to fill in any missing data from the remaining columns. Once I had the completed data set, I cleaned the data of outliers by following the steps in the workshop for applicant income then co-applicant income. Once both of these were clean I combined the parameters to have a cleaned set of data that followed the rules of both columns.

Write a section that answers the following questions:
a. How many empty values did each column contain?

- ApplicantIncome - *211*
- CoapplicantIncome - *184*
- Owns_Car - *7484*

b. How many outliers did ApplicantIncome contain?

- *167*

c. How many outliers did CoapplicantIncome contain?

- 191

d. Which columns did you delete, and why?

- Loan_ID - We don’t need an identifier column
- Applicant_ID - We don’t need an identifier column
- Owns_Car - Too many unknown records

B) Visualization

CSCI-4047-901
Tyler Burleson
Final Project

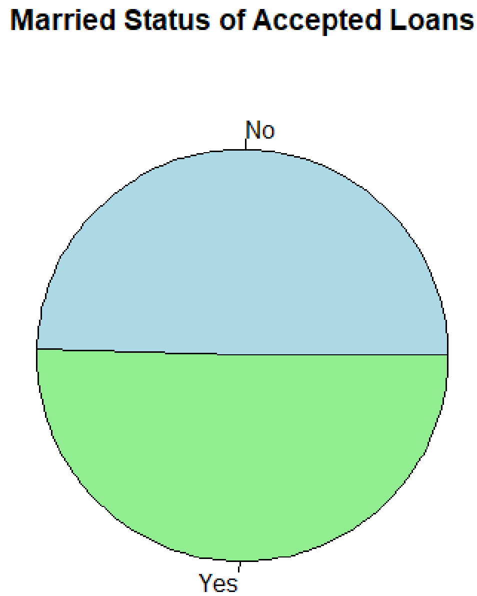


Figure 1

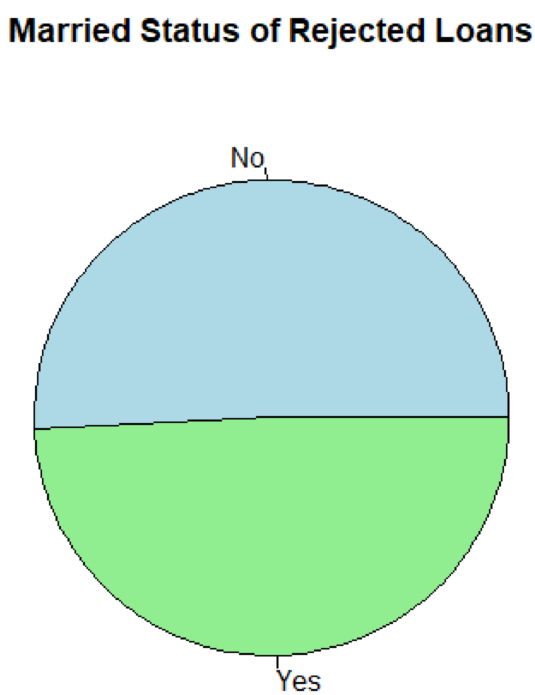


Figure 2

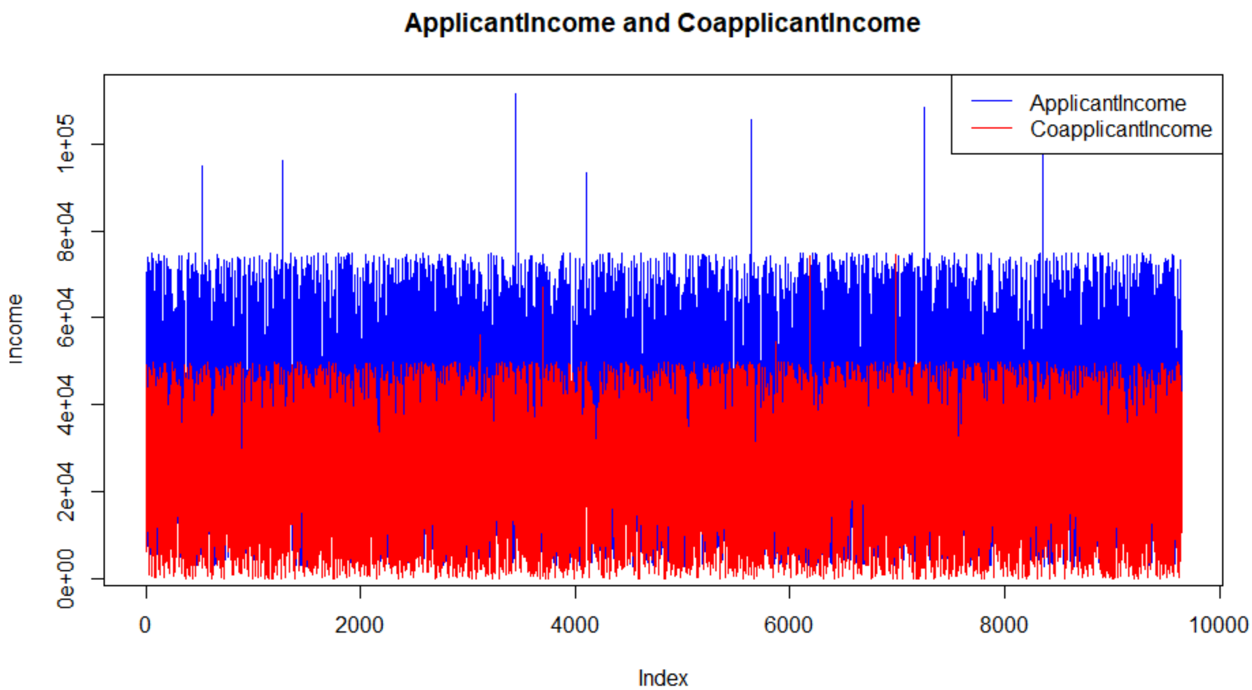


Figure 3

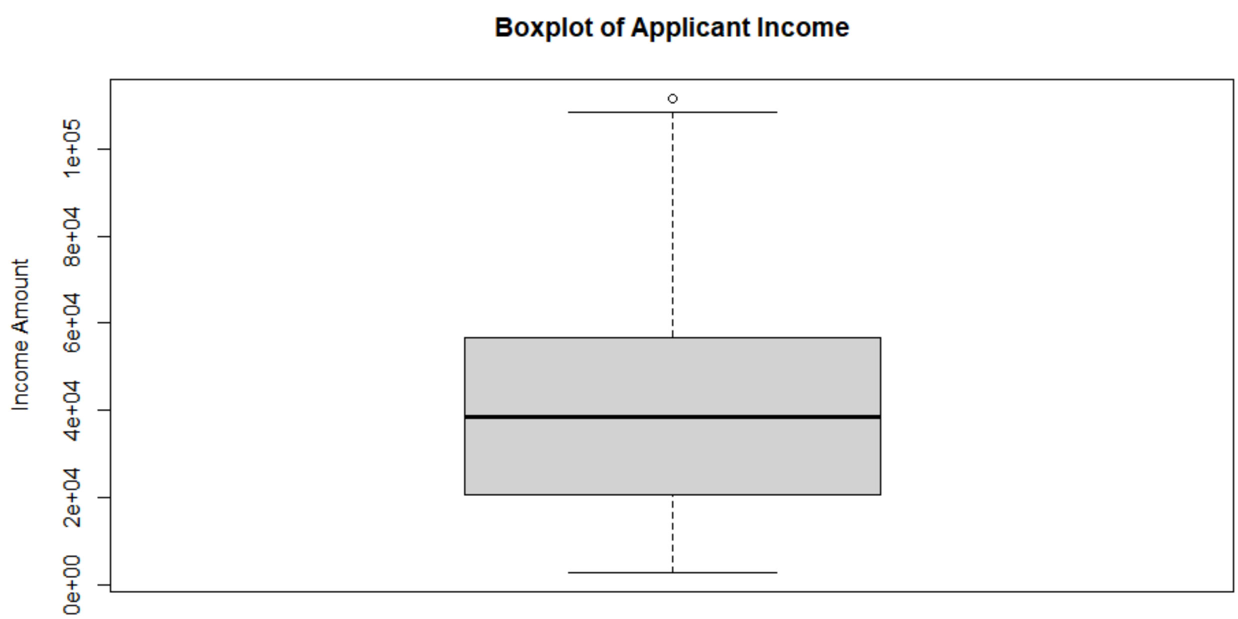


Figure 4-A

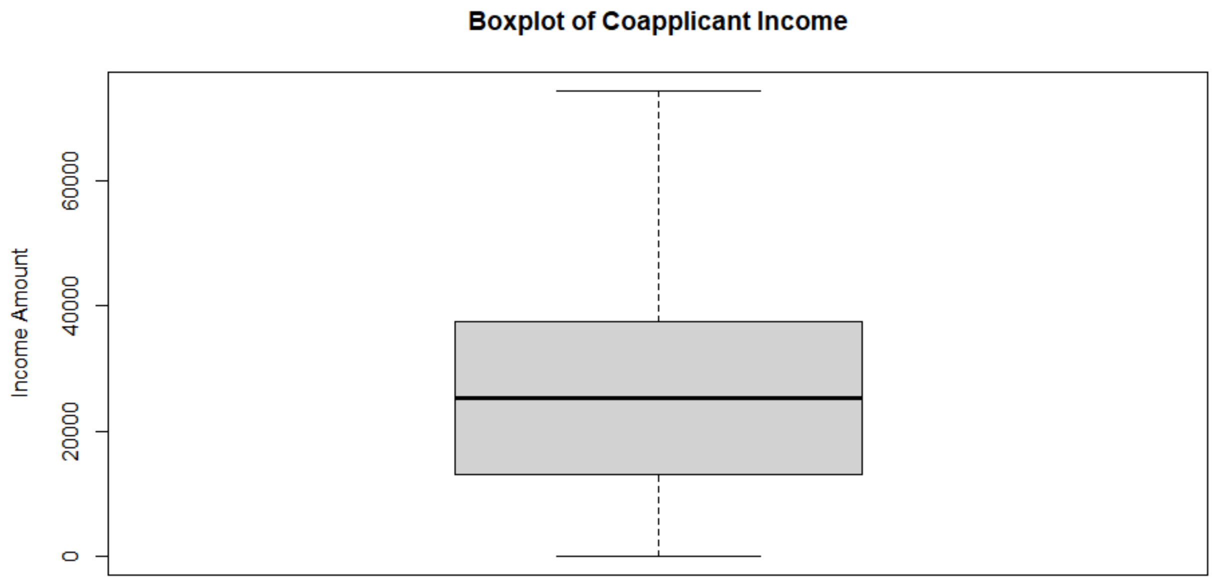


Figure 4-B

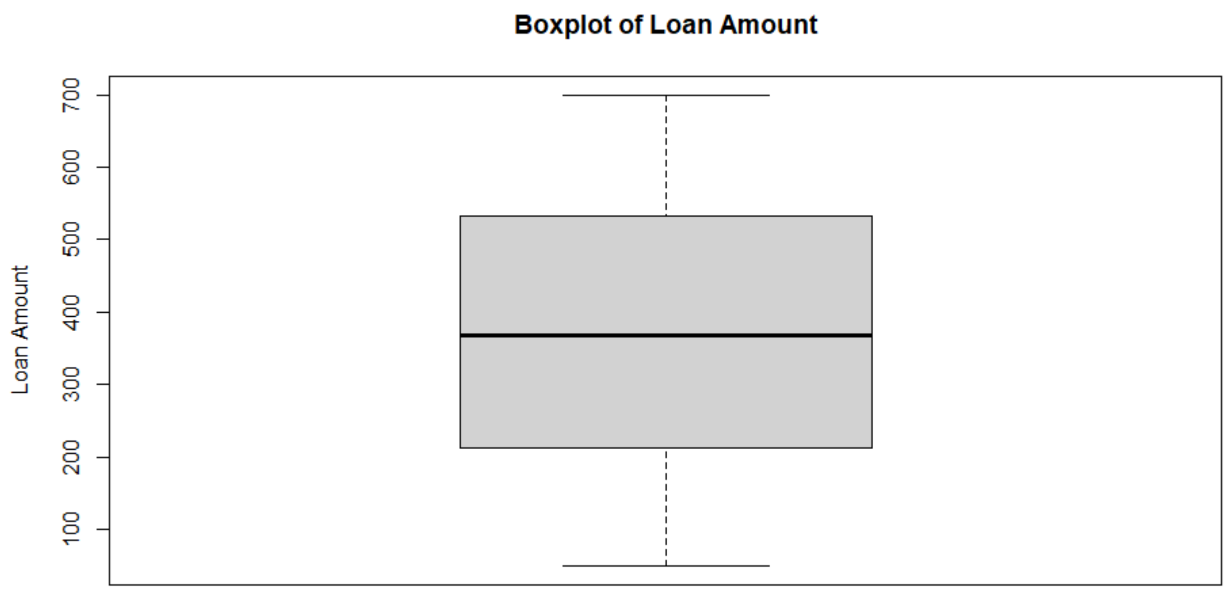


Figure 4-C

C) Descriptive Analytics

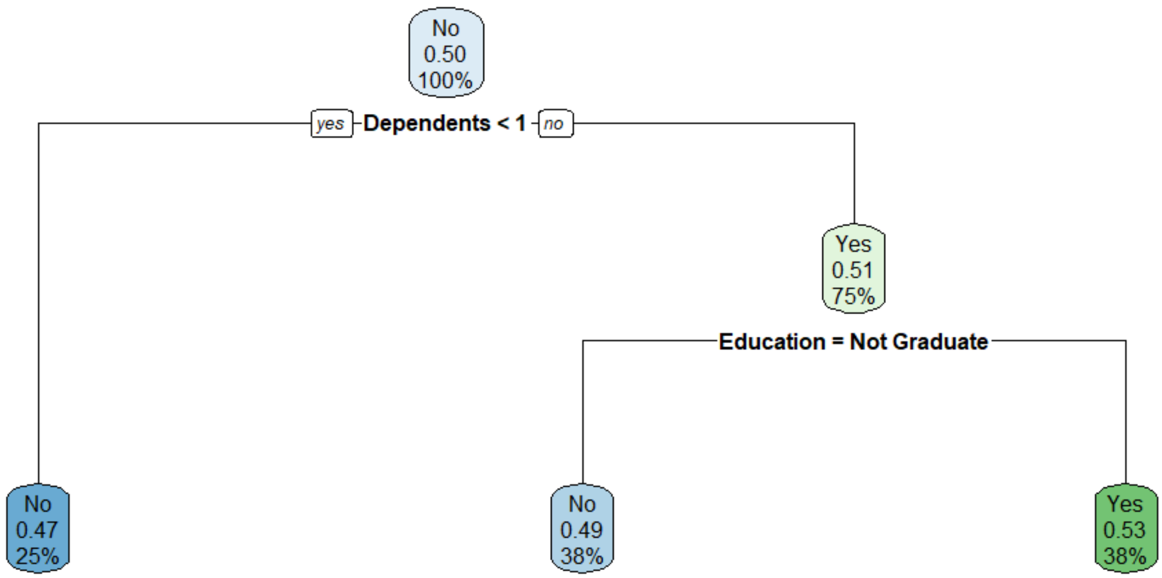
CSCI-4047-901
Tyler Burleson
Final Project

```
> summary(cleaned_Data$ApplicantIncome)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3002   20811   38519   38768   56634  111410 
> summary(cleaned_Data$CoapplicantIncome)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0   12952   25249   25182   37581   74440 
> summary(cleaned_Data$LoanAmount)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   50.0   212.0   367.0   370.9   532.0   699.0 
```

D) Predictive Analysis

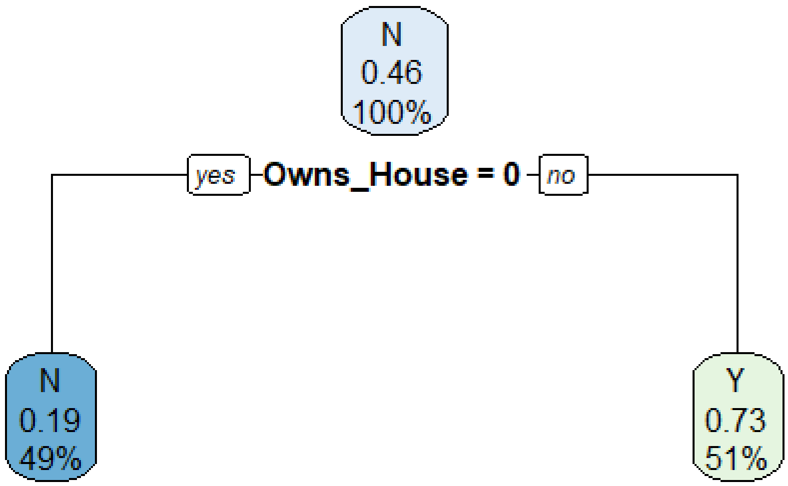
```
Confusion Matrix and Statistics


      Y   N
  Y 720 279
  N 180 750
                                          
               Accuracy : 0.7621          
                 95% CI : (0.7424, 0.7809)
    No Information Rate : 0.5334          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.5252          
                                          
 Mcnemar's Test P-Value : 4.779e-06       
                                          
            Sensitivity : 0.8000          
            Specificity : 0.7289          
         Pos Pred Value : 0.7207          
         Neg Pred Value : 0.8065          
             Prevalence : 0.4666          
         Detection Rate : 0.3733          
   Detection Prevalence : 0.5179          
      Balanced Accuracy : 0.7644          
                                          
       'Positive' Class : Y               
```

CSCI-4047-901
Tyler Burleson
Final Project



Decision Tree Results

```
Confusion Matrix and Statistics



      Y   N
  Y 720 279
  N 180 750

               Accuracy : 0.7621
                 95% CI : (0.7424, 0.7809)
    No Information Rate : 0.5334
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5252

 Mcnemar's Test P-Value : 4.779e-06

            Sensitivity : 0.8000
            Specificity : 0.7289
         Pos Pred Value : 0.7207
         Neg Pred Value : 0.8065
             Prevalence : 0.4666
         Detection Rate : 0.3733
   Detection Prevalence : 0.5179
      Balanced Accuracy : 0.7644

       'Positive' Class : Y
```



Naive Bayes Results

Above are two Confusion Matrices, one utilizes a Decision Tree and the other uses Naive Bayes. Both of these models used the same training data(80% of the clean data) and testing data(20% of the clean data). Surprisingly the results were identical so neither model was better in my testing. Each had an accuracy of 76.21%, a true positive sensitivity of 80%, and a true negative specificity of 72.89%. While the results were identical, each model chose a different route for their respective predictions. The Naive Bayes focused on branching with the Owns_House column while the Decision Tree branched on the Dependent amount and the Education column.