Project 2 SDS 322E

Joshua Bryer

5/7/2022

Project Dataset

```r
# load in data
credit<- readr::read_csv("clean_dataset.csv")
```

Credit Card Application Analysis

We are working with the "clean_dataset.csv" dataset, which contains 690 observations records. I found the dataset on the website Kaggle and the data itself comes from credit card applications.In this dataset, missing values have been inferred, numeric variables have been scaled, and the data has been cleaned already.There are sixteen columns providing information about the applicant's Social Status, Financial History, and whether or not they were approved. Information about their social status includes their Gender, marrital status, Age, Ethnicity, Citizenship, years employed, employment status and Business Industry.Their financial history includes the applicants income,debt, years employed.Finally whether or not they were approved is represented by the column Approved.

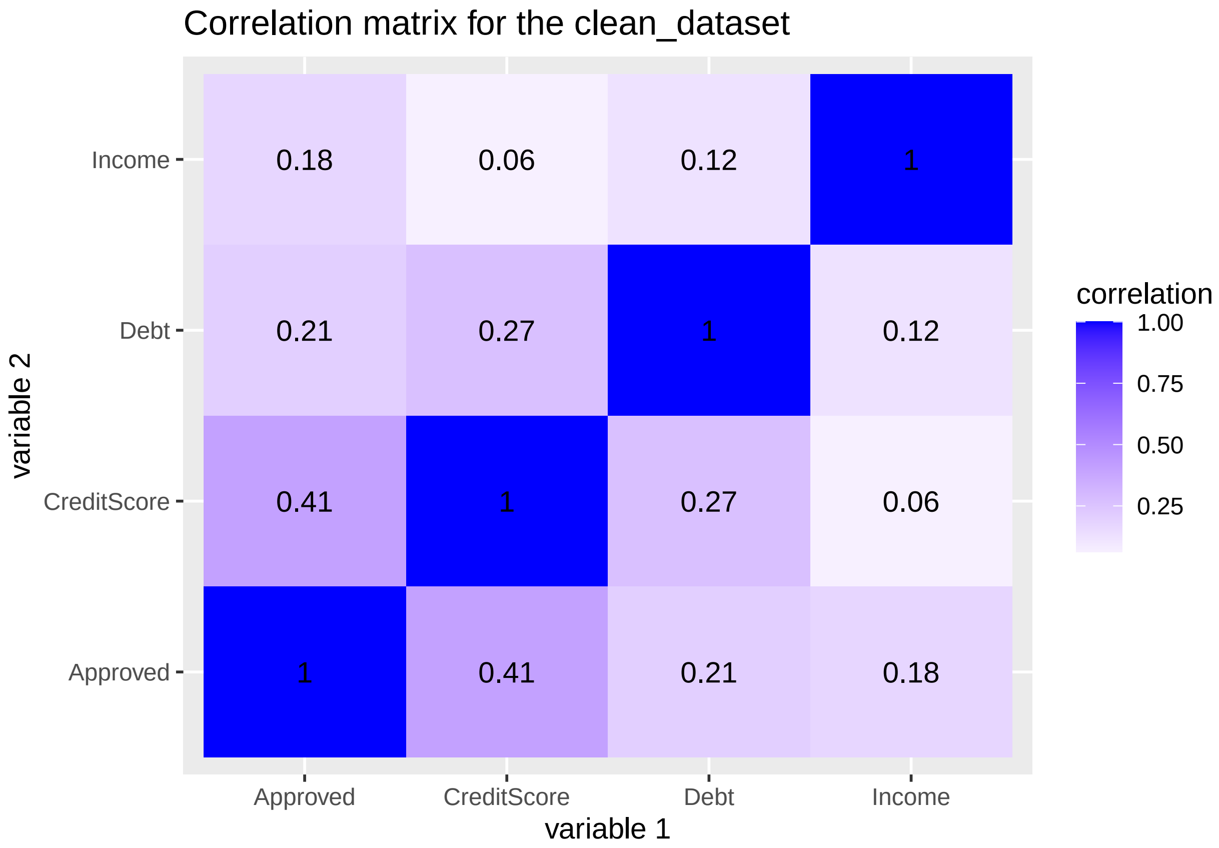
I want to work with this dataset because I would like to work with data of this variety in the future and I thought it would be a good preview into that.

I expect to find are that years employed(YearsEmployed) will be correlated with Income(Income). I expect that there will be a negative correlation between Debt(Debt) and credit score(CreditScore). Additionally, I expect that there will be a positive correlation between age and income.

Exploratory Data Analysis

My main variables are income, debt, credit score, and whether the credit card application was approved

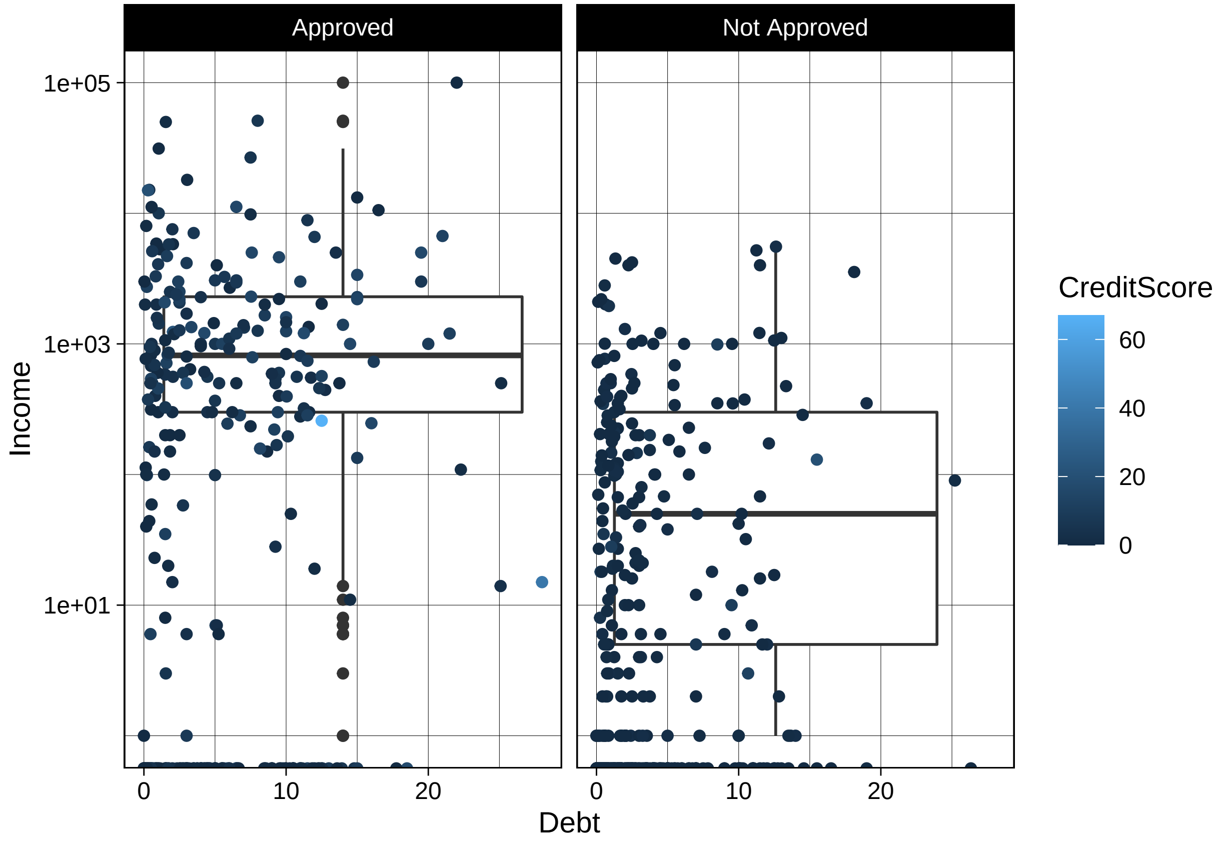
```r
#Selecting variables
credit_num <- credit %>%
  select(Income,CreditScore,Debt,Approved)

cor(credit_num, use = "pairwise.complete.obs") %>%
  # Save as a data frame
  as.data.frame %>%
  # Convert row names to an explicit variable
  rownames_to_column %>%
  # Pivot so that all correlations appear in the same column
  pivot_longer(-1, names_to = "other_var", values_to = "correlation") %>%
  ggplot(aes(rowname, other_var, fill = correlation)) +
  # Heatmap with geom_tile
  geom_tile() +
  # Change the scale to make the middle appear neutral
  scale_fill_gradient2(low="red",mid="white",high="blue") +
  # Overlay values
  geom_text(aes(label = round(correlation,2)), color = "black", size = 4) +
  # Give title and labels
  labs(title = "Correlation matrix for the clean_dataset", x = "variable 1", y = "variable 2")
```



From the heat map of the correlations between the variables, it can be observed that the strongest correlations are between Credit Score and Approved, debt and credit score, and debt and approved.

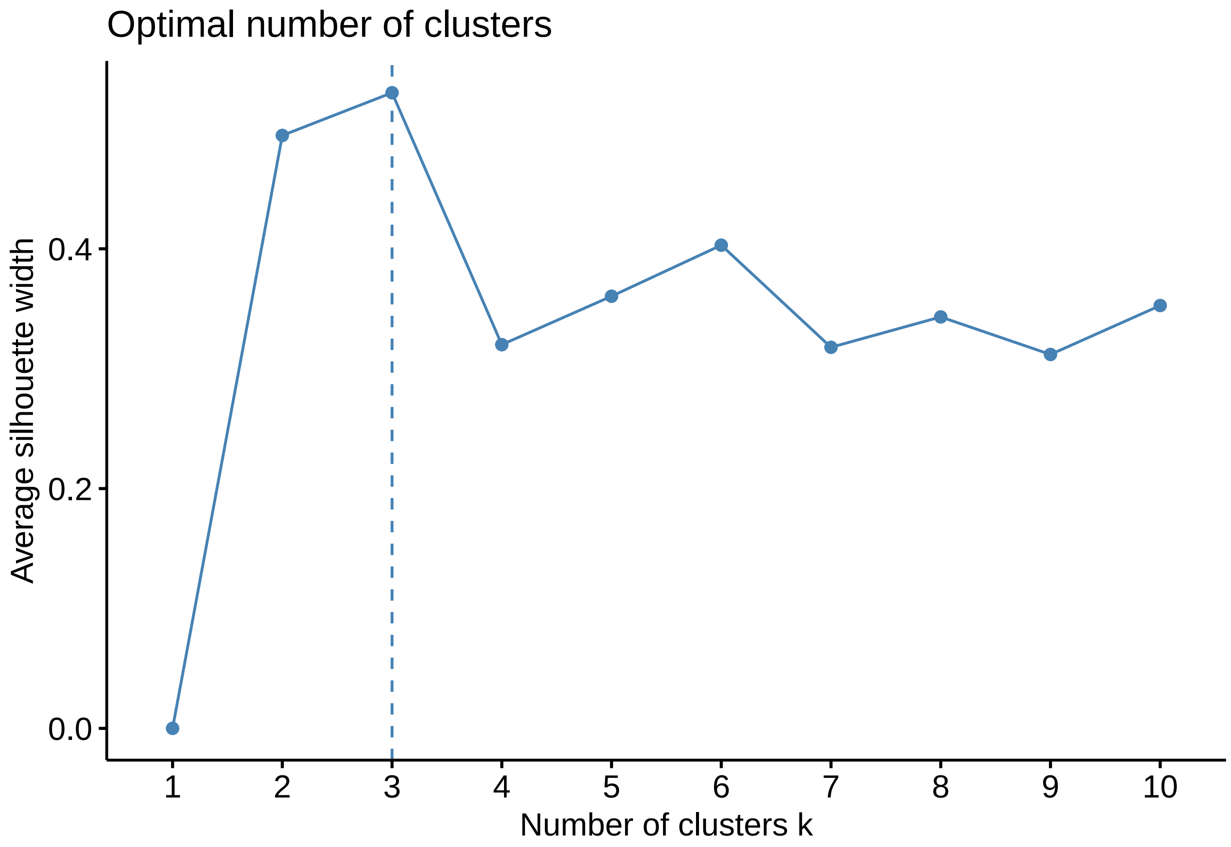
```
#Plot of my main variables income, debt, and approved status
credit%>%
  mutate(Approved=if_else(Approved==0, "Not Approved", "Approved"))%>%
  ggplot(aes(Debt,Income))+
  geom_boxplot()+
  geom_point(aes(color=CreditScore))+
  facet_wrap(vars(Approved),labeller = labeller(c("hi","hello")))+
  theme_linedraw()+
  scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

## Warning: Removed 295 rows containing non-finite values (stat_boxplot).
```



Clustering

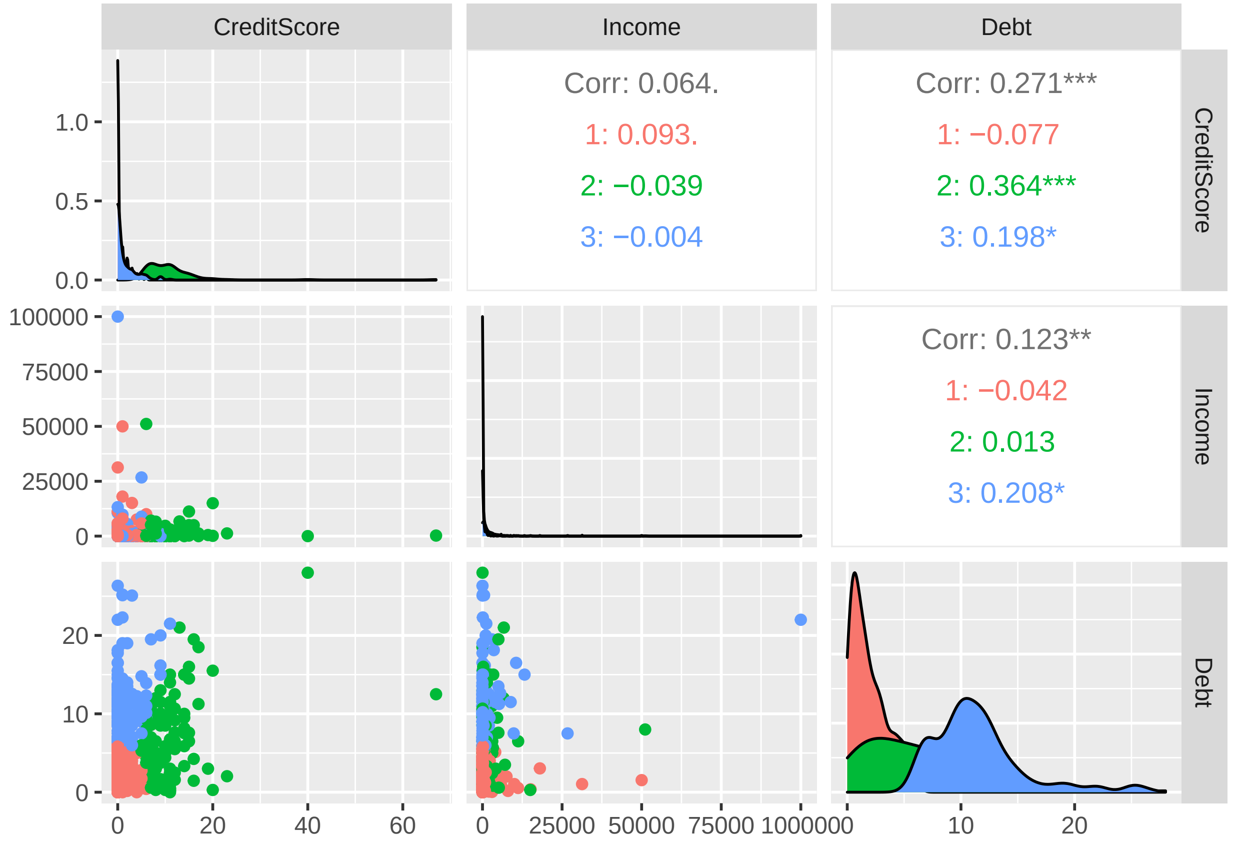
```
#Selecting numeric variables
credit2<-credit%>%
  select(CreditScore,Income,Debt)%>%
  scale()

fviz_nbclust(credit2, pam, method = "silhouette")
```



```
# Use the function pam to find 3 clusters
pam_results <- credit2 %>%
  pam(k = 3)

# Save cluster assignment as a column in your dataset
credit_pam <- credit %>%
  mutate(cluster = as.factor(pam_results$clustering))

# Convert Approved to Logical variable
credit_cluster<-credit%>%
  mutate(Approved=if_else(Approved==0,F,T))

# visualize the clusters after dimension reduction
```

##Clustering vizualization and Summary

```
# Visualize cluster data using ggpairs

ggpairs(credit_pam, columns = c("CreditScore","Income","Debt"), aes(color = cluster))
```



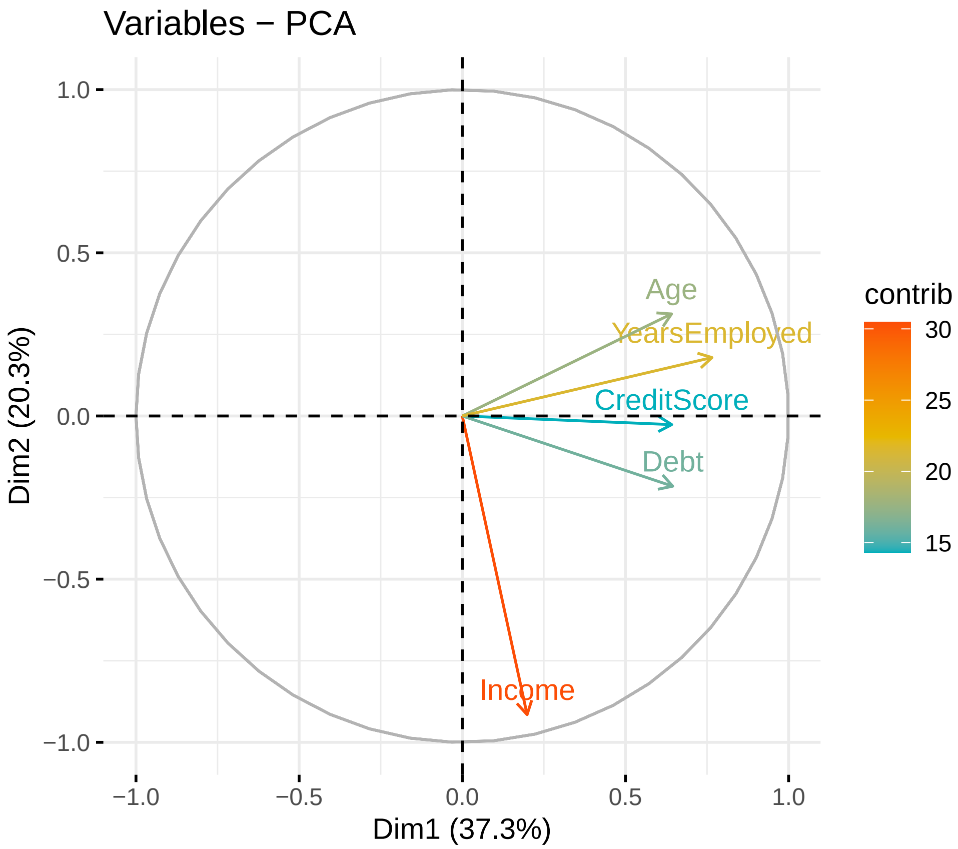
```
# Finding means of each variable for each cluster
credit_pam %>%
  select(CreditScore,Income,Debt,cluster)%>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 3 x 4
##   cluster CreditScore Income  Debt
##   <fct>         <dbl>  <dbl> <dbl>
## 1 1             0.658   615.  1.93
## 2 2            11.1    1991.  6.73
## 3 3             1.37   1490. 11.4
```

From the ggpairs plot we can see that for variables debt and credit score the clusters are fairly spaced and that there is a positive correlation between debt and credit score, corr=0.271. In the plot between Debt and Income we can see that the clusters are fairly bunched and that there is a significant correlation between debt and income, corr=0.123. In credit score versus income we can see there is a non-significant correlation of 0.064. From the data table we can see that in cluster one the mean values of credit score, income, and debt were respectively 0.66, 615.21, and 1.93. In cluster two, the mean values of credit score, income, and debt were respectively 11.08, 1991.13, and 6.73.Lastly in cluster three,values of credit score, income, and debt were respectively 1.37, 1490.39, and 11.44

Dimensionality Reduction

```
#Select variables and perform PCA
pca.credit<-credit %>%
  select(CreditScore,Income,Debt,Age,YearsEmployed) %>%
  scale()%>%
  prcomp()

#Visualize PCA data
fviz_pca_var(pca.credit, col.var = "contrib",gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"))
```



From the vizualization we can see that all variables contribute positively to dimension one and that variables creditscore, debt, and income contribute negatively to dimension two. To score high on one of the variables Income, Debt, or credit score indicates a high value on the other two and indicates a low value in age or years employed. To score highly on variables age or years employed indicates a high score on the other and indicates a low score on income, debt or credit score.

Classification and Cross-validation

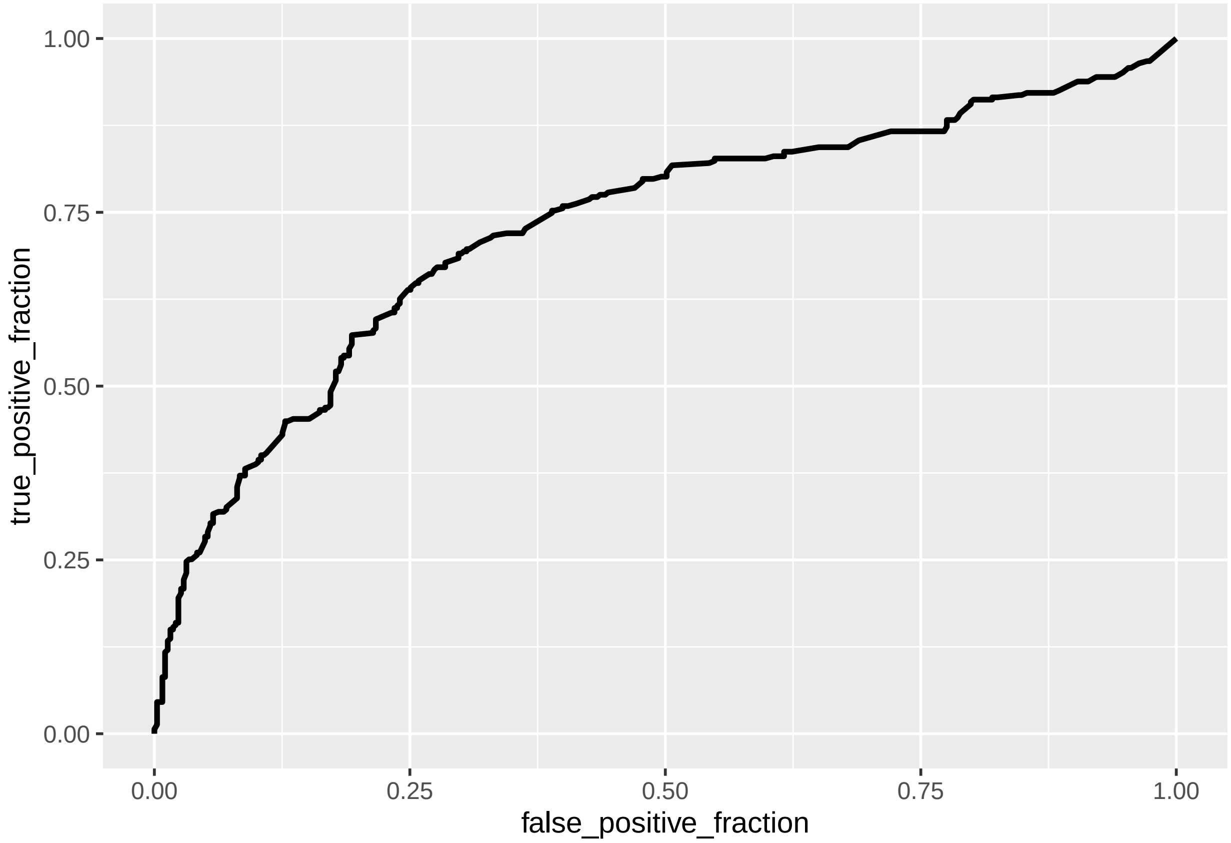
Classification

```
#Fitting data to glm model
fit <- glm(Approved ~ CreditScore+Debt, data = credit, family = "binomial")

log_credit <- credit %>%
  mutate(probability = predict(fit, type = "response"),
         predicted = ifelse(probability > 0.5, 1, 0)) %>%
  # Give a name to the rows
  rownames_to_column("Applicant") %>%
  select(Approved, CreditScore, Debt, probability, predicted)
```

```r
table(log_credit$Approved, log_credit$CreditScore) %>% addmargins
```

```
##
##         0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##   0   297  36  26   8   3   2   3   1   0   0   4   1   1   0   0   0   0   0
##   1    98  35  19  20  12  16  20  15  10  10   4  18   7   1   8   4   3   2
##   Sum 395  71  45  28  15  18  23  16  10  10   8  19   8   1   8   4   3   2
##
##        19  20  23  40  67 Sum
##   0     0   1   0   0   0 383
##   1     1   1   1   1   1 307
##   Sum   1   2   1   1   1 690
```

```r
# Calculating ROC curve
ROC <- ggplot(log_credit) +
  geom_roc(aes(d = Approved, m = CreditScore+Debt), n.cuts = 0)
ROC
```



```r
#Calculating AUC for the ROC
calc_auc(ROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.7277281
```

Cross Validation

```r
# Choose number of folds
k = 10
```

```r
set.seed(32)

# Randomly order rows in the dataset
data <- credit[sample(nrow(credit)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Use a for loop to get diagnostics for each test set
diags_k <- NULL

for(i in 1:k){
  # Create training and test sets
  train <- data[folds != i, ] # all observations except in fold i
  test <- data[folds == i, ]  # observations in fold i

    # Train model on training set (all but fold i)
  fit <- glm(Approved ~ CreditScore+Debt, data = train, family = "binomial")

  # Test model on test set (fold i)
  df <- data.frame(
    probability = predict(fit, newdata = test, type = "response"),
    outcome = test$Approved)

  # Consider the ROC curve for the test dataset
  ROC <- ggplot(df) +
    geom_roc(aes(d = outcome, m = probability))

  # Get diagnostics for fold i (AUC)
  diags_k[i] <- calc_auc(ROC)$AUC
}
# Average performance
mean(diags_k)
```

```
## [1] 0.7716617
```

Based on the ROC plot and the AUC output we see that the model does a fair job in predicting new observations as the AUC value is 0.73. Based on the mean of our cross validation we can see that the model does not show signs of overfitting as the AUC value in fact increases.