Loan Default Data

Aaron Matos

```
library(rmarkdown)
library(MASS)
library(tidyverse)
```

```
## -- Attaching packages -----------------------------------------------------------------------

## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.3.0

## -- Conflicts ----------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(ISLR)
library(kknn)


loan_data <- readRDS(file = "/cloud/project/Final Project/loan_data.rds")

#Create training and test data
set.seed(314)
train_index <- sample(1:nrow(loan_data), floor(0.7*nrow(loan_data)))

# training
loan_training <- loan_data[train_index, ]

# test
loan_test <- loan_data[-train_index, ]

# Function for analyzing confusion matrices
cf_matrix <- function(actual_vec, pred_prob_vec, positive_val,
                      cut_prob = 0.5, search_cut = FALSE) {

  if (search_cut == FALSE) {
  actual <- actual_vec == positive_val; pred <- pred_prob_vec >= cut_prob
  P <- sum(actual); N <- length(actual) - P; TP <- sum(actual & pred)
  FN <- P - TP; TN <- sum(!(actual) & !(pred)); FP <- N - TN

  if (TP != 0) { Precision <- TP/(TP + FP); Recall <- TP/(TP + FN)
                 F1 <- 2*((Precision*Recall)/(Precision + Recall))}

  if(TP == 0) { Precision = 0; Recall = 0; F1 = 0 }

  model_results <- list(confusion_matrix =
    data.frame(metric = c("Correct", "Misclassified", "True Positive",
                          "True Negative","False Negative", "False Positive"),
               observations = c(TN + TP, FN + FP, TP, TN, FN, FP),
```

```r
                rate = c((TN + TP)/(N + P), (FN + FP)/(N + P), TP/P, TN/N, FN/P, FP/N),
                pct_total_obs = c((TN + TP), (FN + FP), TP, TN, FN, FP)*(1/(N + P)),
                stringsAsFactors = FALSE),
    F1_summary =
    data.frame(metric = c("Precision", "Recall", "F1 Score"),
                value = c(Precision, Recall, F1),
                stringsAsFactors = FALSE))
return(model_results) }

  if (search_cut == TRUE) {
    optimal_cut = data.frame(cut_prob = seq(0,1, by = 0.05),
                             correct_rate = NA, F1_score = NA,
                             false_pos_rate = NA, false_neg_rate = NA)

    for (row in (1:nrow(optimal_cut))) {
      actual <- actual_vec == positive_val
      pred <- pred_prob_vec >= optimal_cut$cut_prob[row]
      P <- sum(actual); N <- length(actual) - P
      TP <- sum(actual & pred); FN <- P - TP
      TN <- sum(!(actual) & !(pred)); FP <- N - TN

      if (TP != 0) { Precision <- TP/(TP + FP); Recall <- TP/(TP + FN)
          F1 <- 2*((Precision*Recall)/(Precision + Recall))}

      if(TP == 0) { Precision = 0; Recall = 0; F1 = 0 }

      optimal_cut[row, 2:5] <- c((TN + TP)/(N + P), F1, FP/N, FN/P)
    }
return(optimal_cut)
  }
}
```

Exploratory Data Analysis Section

Do loan default rates differ by customer age?

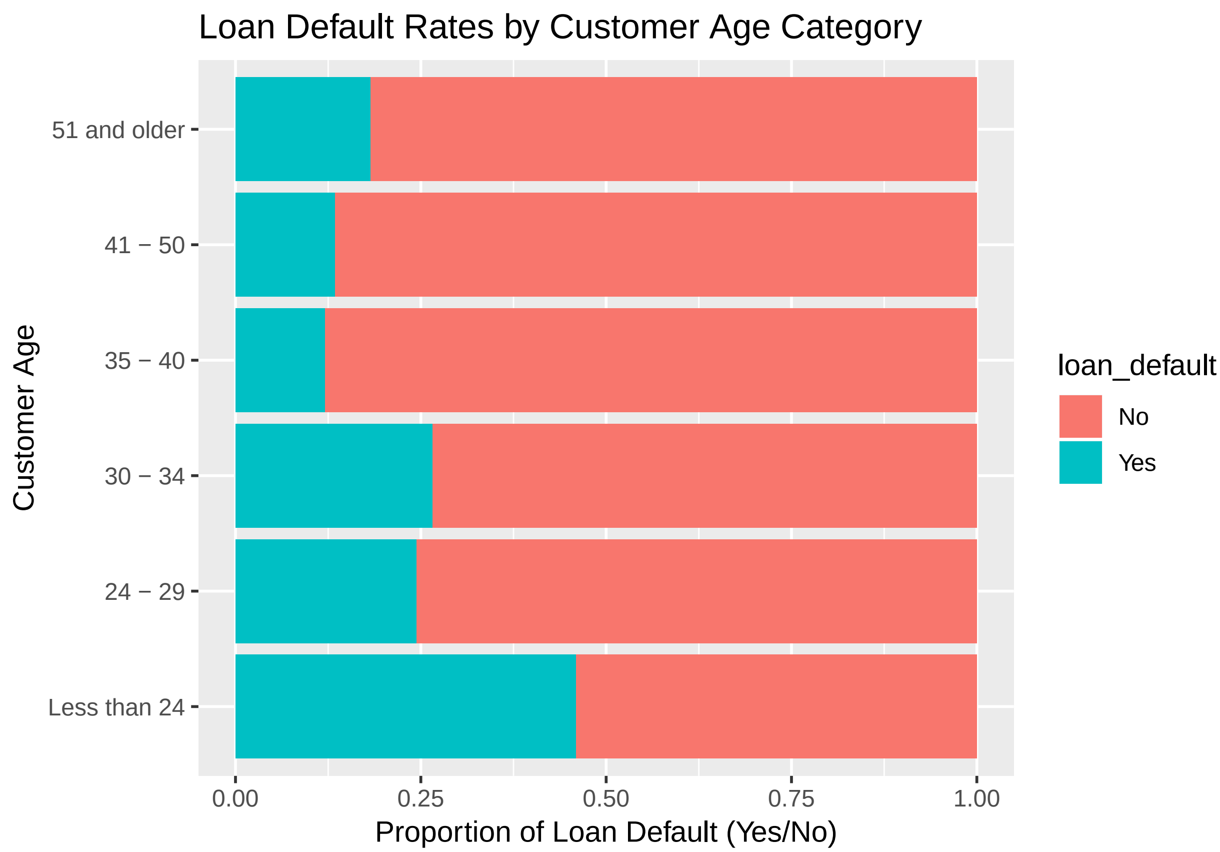
Findings: Yes, customers between 35 and 50 years old have significantly lower default rates than other customers. Customer age appears to be a strong predictor of loan default.

```r
default_by_age <- loan_data %>% group_by(age_category) %>%
                  summarise(total_customers = n(),
                            customers_who_defaulted = sum(loan_default == "Yes")) %>%
                  mutate(default_rate = customers_who_defaulted / total_customers)


default_by_age
```

```
## # A tibble: 6 x 4
##   age_category total_customers customers_who_defaulted default_rate
##   <fct>                  <int>                   <int>        <dbl>
## 1 Less than 24             557                     256        0.460
## 2 24 - 29                  742                     181        0.244
## 3 30 - 34                  519                     138        0.266
## 4 35 - 40                  754                      91        0.121
## 5 41 - 50                  685                      92        0.134
```

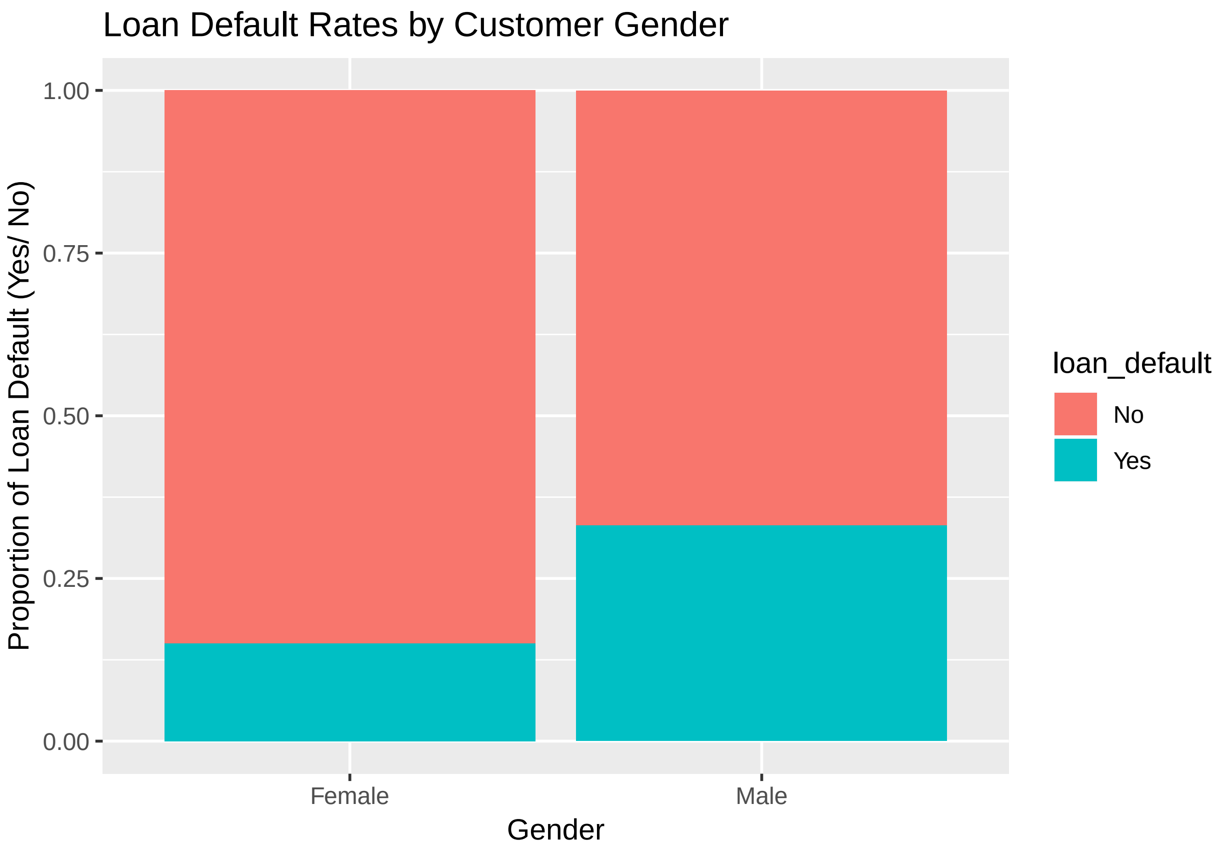
```
## 6 51 and older              488                        89        0.182
```

```
ggplot(data = loan_data, mapping = aes(x = age_category, fill = loan_default)) +
  geom_bar(position = "fill") +
  labs(title = "Loan Default Rates by Customer Age Category",
        x = "Customer Age",
        y = "Proportion of Loan Default (Yes/No)") +
  coord_flip()
```



Question 1:

```
#Question1: Which applicant gender (female/ male) was more likely to default on their loan?
default_by_gender <- loan_data %>%
                  group_by(gender) %>%
                  summarise(total_customers = n(),
                            customers_defaulted = sum(loan_default == "Yes"))%>%
                   mutate(default_rate = customers_defaulted / total_customers) %>%
                    arrange(desc(default_rate))

#Bar Chart
ggplot(data = loan_data, mapping = aes(x = gender, fill = loan_default)) + geom_bar(position = "fill")
```

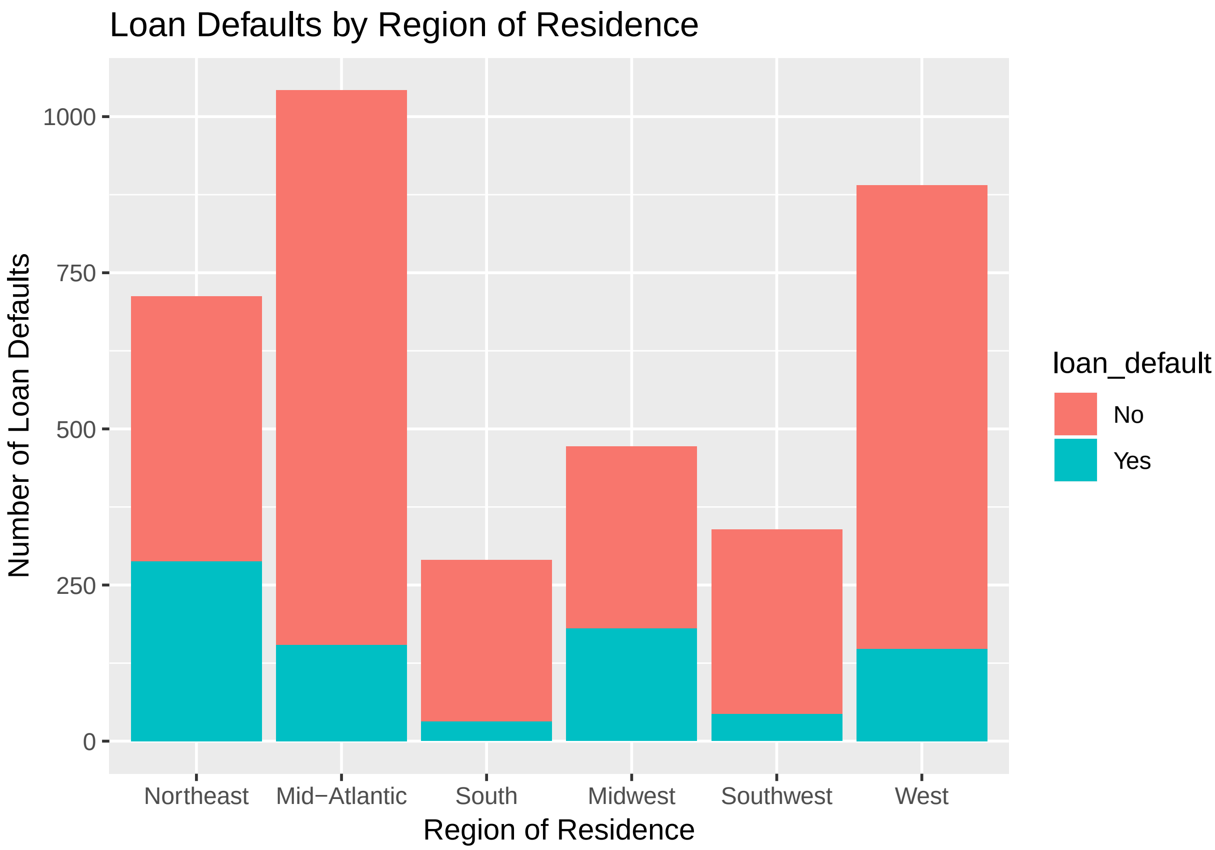


Males have more than twice the default rate on loans than females with rates of 33.2% and 15.0% respectively.

```
#Question2: Which education level is most likely to default in a loan?
default_by_ed <- loan_data %>%
                group_by(highest_ed_level) %>%
                summarise(customers_defaulted = sum(loan_default == "Yes"), number_of_customers = n()) %
  mutate(percent_defaulted = customers_defaulted / number_of_customers) %>%
  arrange(desc(customers_defaulted))
```

It appears that individuals with less formal education default on their loans more frequently. Those with a high school level education showed a 61.6% default rate while < high school exhibited a 43.6% default rate.

```
#Question3: What is the total amount of loan defaults by applicant region of residence?
default_by_region <- loan_data %>% group_by(us_region_residence) %>%
  summarise(number_of_customer = n(),
    customers_defaulted = sum(loan_default == "Yes"),
    default_rate = customers_defaulted / number_of_customer)

#Bar Chart
ggplot(data = loan_data, mapping = aes(x= us_region_residence, fill = loan_default)) +
  geom_bar(stat = "count")+
  labs(title = "Loan Defaults by Region of Residence", x = "Region of Residence", y = "Number of Loan De
```

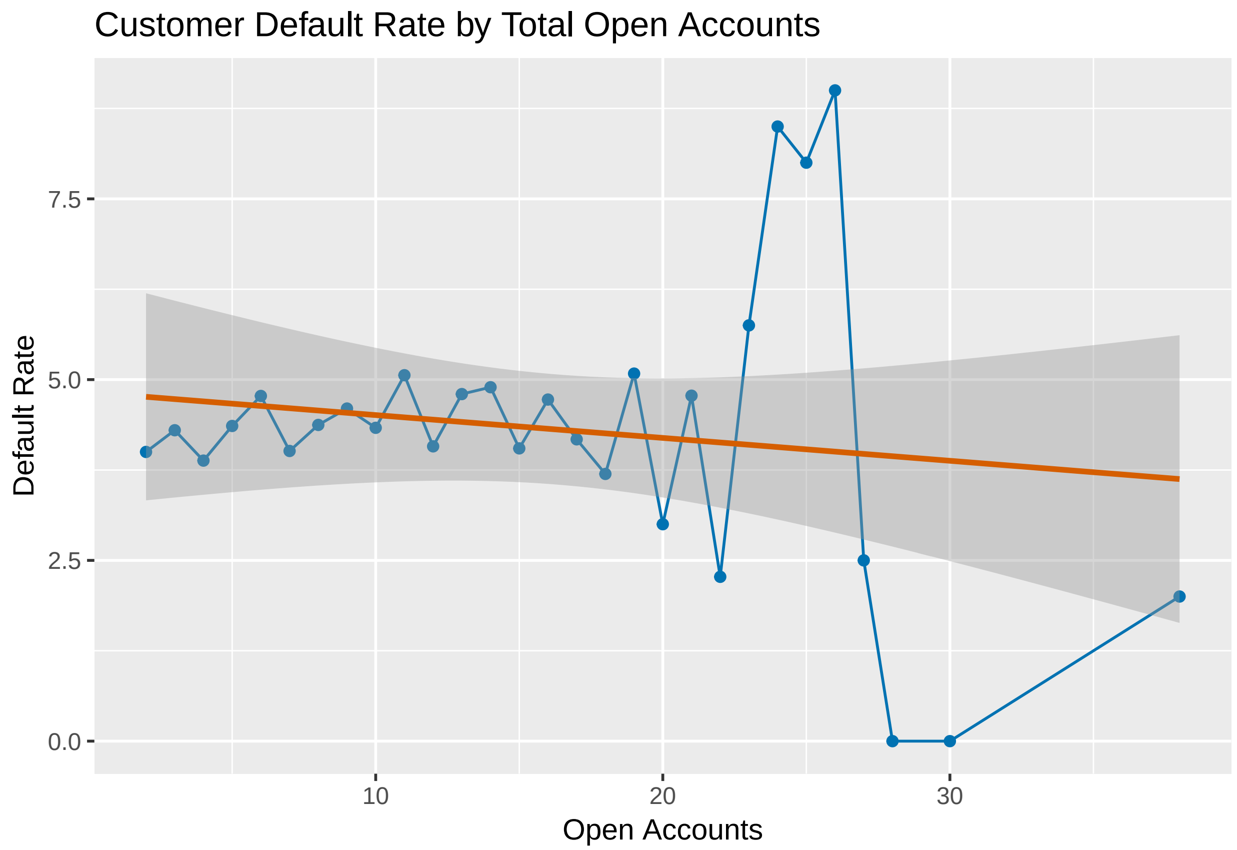


It appears that those individuals living in the Northeast and Midwest have a significantly higher rate of default at 40.4% and 38.3% respectively.

```
#Question4: What is the number of customers that defaulted on their loan based on their adjusted annual
income <-loan_data %>% mutate(income_category = case_when(adjusted_annual_inc < 10000 ~ "Less than $10,
```

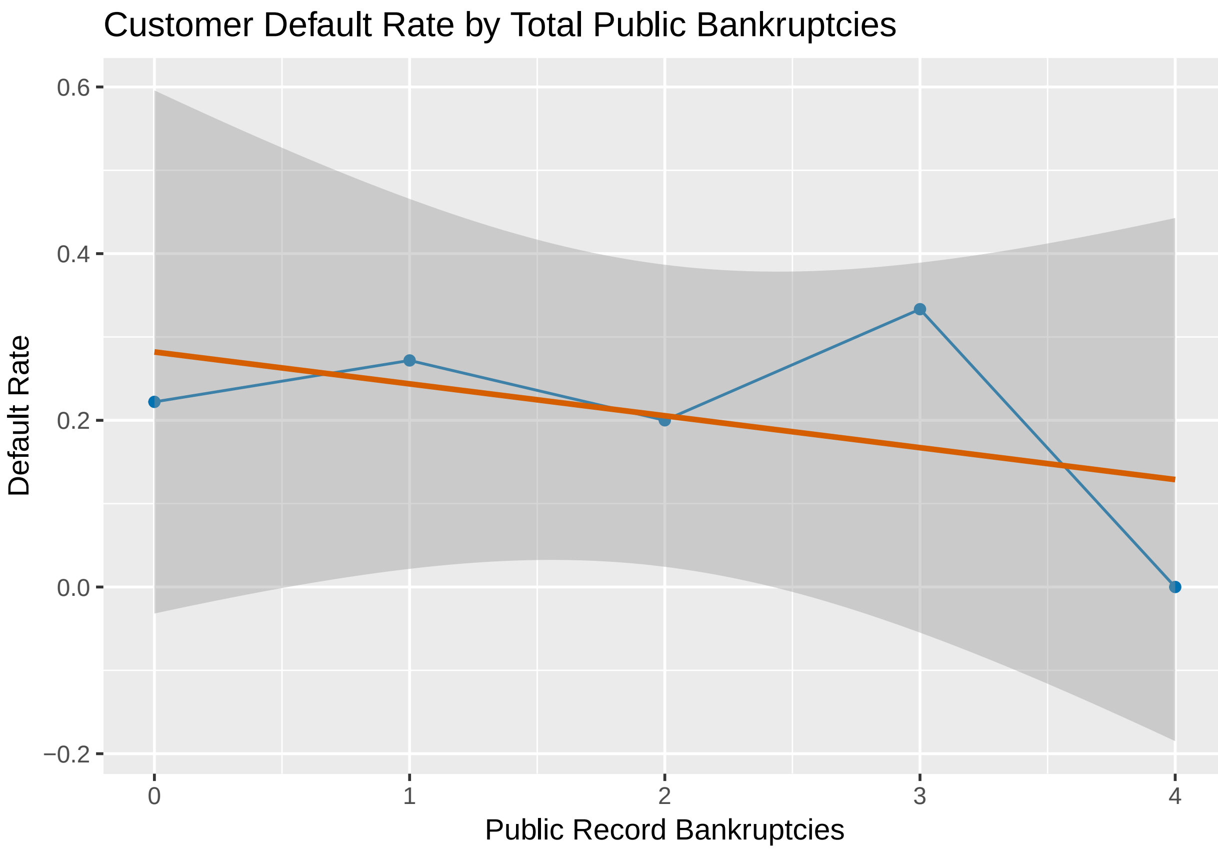
Apparent in the "income" data frame is the relationship between defaulting on loans and levels of income. Those making less than 10k/year showed a default rate of 38.5%. Individuals making 10-50k have a default rate of 25.9%.

```
#Question5: What is the relationship between average total open accounts for an applicant and number of
open_accounts <- loan_data %>%
              group_by(open_acc) %>%
              summarise(customer_defaulted = sum(loan_default == "Yes"),
                        number_of_customers = n(), default_rate = number_of_customers / customer_defaul
                        default_rate1 = ifelse(default_rate == "Inf", 0, default_rate)) %>% arrange(des

#Line Chart
ggplot(data = open_accounts, mapping = aes(y = default_rate1, x = open_acc)) +
  geom_line(color = "#0072B2") +
  geom_point(color = "#0072B2") +
  geom_smooth(method = "lm", color = "#D55E00")+
  labs(title = "Customer Default Rate by Total Open Accounts", x = "Open Accounts", y = "Default Rate")
```



The relationship between open accounts and default rates seems to be a negative one. As open accounts increase, default rates seem to decline. However this could be indicative of outliers in our data set or a lack of individuals with more than 28 open accounts.

```
#Question6: What is the relationship between public record bankruptcies and number of defaults?
bankruptcies <- loan_data %>%
              group_by(pub_rec_bankruptcies) %>%
              summarise(customer_defaulted = sum(loan_default == "Yes"),
                        number_of_customers = n(),
                        default_rate = customer_defaulted / number_of_customers)

#Line Chart
ggplot(data = bankruptcies, mapping = aes(y = default_rate, x = pub_rec_bankruptcies)) +
  geom_line(color = "#0072B2") +
  geom_point(color = "#0072B2") +
  geom_smooth(method = "lm", color = "#D55E00") +
  labs(title = "Customer Default Rate by Total Public Bankruptcies", y = "Default Rate", x = "Public Rec
```



Surprisingly the relationship between public bankruptcies and loan defaults is negative as well. Again, this could be due to the limited number of individuals in the data set with >= 2 bankruptcies.

```
#Question7: What is the average fico score of applicants who have/ have not defaulted on their loan?
fico_score_defaults <- loan_data %>%
              mutate(fico_score_ranges = case_when(fico_score  <= 579 ~ "Poor", between(fico_score, 580
  group_by(fico_score_ranges, loan_default) %>%
  summarise(average_fico = mean(fico_score)) %>%
  spread(key = fico_score_ranges, value = average_fico)
```

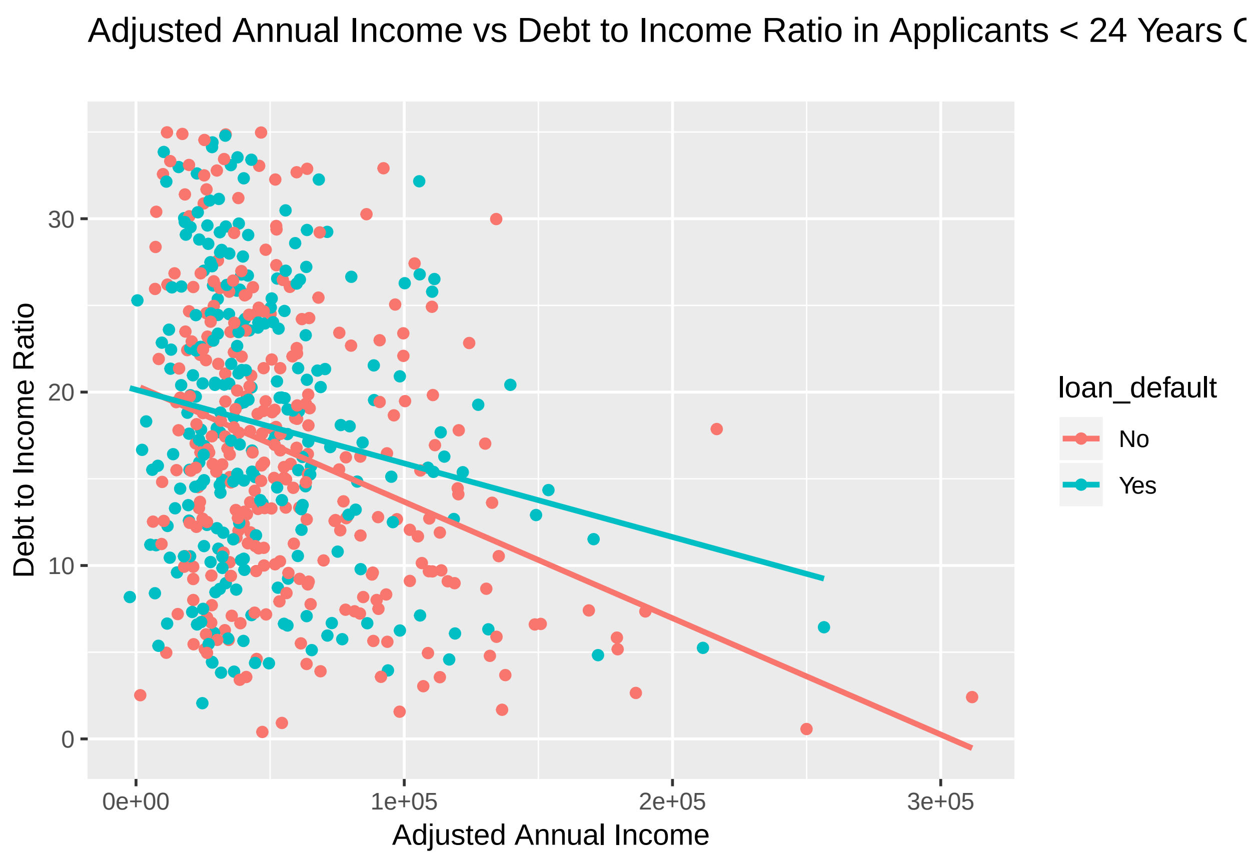
In our fico score defaults dataframe above, we see the average fico score for individuals in all four ranges, poor, fair, good, and exceptional, who did or did not default on their loans. There is no obvious difference in score between the two groups aside from the "Poor" fico score. Those who defaulted in the poor group had on average a 48 point lower fico score than their counterparts who did not default.

```
#Question8: What number of credit inquires per customer has the highest default rates?

inq_defaults <- loan_data %>%
              group_by(inq_last_6mths) %>%
              summarise(number_of_customers = n(),
                customer_defaulted = sum(loan_default == "Yes"),
                        customer_not_defaulted = sum(loan_default == "No"),
                default_rate = customer_defaulted / number_of_customers)
```

It appears that customers with a greater amount of credit inquiries default at a higher rate.

```
#Question9: What is the relationship between adjusted_ann_inc and dti as it relates to loan_default amo

less_than_24 <- loan_data %>% filter(age_category == "Less than 24")
```

```
ggplot(data = less_than_24,
       mapping = aes(x = adjusted_annual_inc, y = dti,
                     color = loan_default)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
      labs(title = "Adjusted Annual Income vs Debt to Income Ratio in Applicants < 24 Years Old \n",
          x = "Adjusted Annual Income", y = "Debt to Income Ratio")
```



It does not appear that there is any relationship between annual adjusted income and dti as they relate to loan default rates in individuals under 24 years old. It is intuitive however, that those with a greater annual adjusted income with a lower dti would be at the very least, be slightly less likely to default.

Variable Selection

Mixed Variable Selection with Logistic Regression

```
#full model
upper_loan_model <- glm(loan_default ~ .,
                        data = loan_training,
                        family = "binomial")

#Null Model
lower_loan_model <- glm(loan_default ~ 1,
                        data = loan_training,
                        family = "binomial")
#mixed selection
results_loan_mixed <- step(lower_loan_model,
                           scope = list(lower = lower_loan_model, upper = upper_loan_model),
```

```r
                          direction = "both", trace = 0)

summary(results_loan_mixed)
```

```
##
## Call:
## glm(formula = loan_default ~ fico_score + highest_ed_level +
##     us_region_residence + age_category + gender + dti + bc_util +
##     inq_last_6mths + adjusted_annual_inc + residence_property,
##     family = "binomial", data = loan_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4480  -0.4407  -0.2182  -0.0628   3.7058
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      1.053e+01  6.610e-01  15.931  < 2e-16
## fico_score                      -1.754e-02  9.604e-04 -18.263  < 2e-16
## highest_ed_levelHigh School      1.191e+00  2.678e-01   4.448 8.68e-06
## highest_ed_levelBachelors       -1.409e+00  2.155e-01  -6.537 6.29e-11
## highest_ed_levelMasters         -1.414e+00  2.412e-01  -5.861 4.59e-09
## highest_ed_levelPhD or Doctorate -8.834e-01  2.663e-01  -3.317 0.000910
## us_region_residenceMid-Atlantic -1.759e+00  1.891e-01  -9.301  < 2e-16
## us_region_residenceSouth        -1.627e+00  2.968e-01  -5.483 4.19e-08
## us_region_residenceMidwest      -1.728e-01  2.050e-01  -0.843 0.399363
## us_region_residenceSouthwest    -1.794e+00  2.926e-01  -6.131 8.74e-10
## us_region_residenceWest         -1.560e+00  1.944e-01  -8.024 1.03e-15
## age_category24 - 29             -8.908e-01  2.005e-01  -4.444 8.85e-06
## age_category30 - 34             -8.827e-01  2.203e-01  -4.007 6.15e-05
## age_category35 - 40             -1.835e+00  2.275e-01  -8.066 7.27e-16
## age_category41 - 50             -1.836e+00  2.319e-01  -7.917 2.44e-15
## age_category51 and older        -1.086e+00  2.260e-01  -4.804 1.56e-06
## genderMale                       1.032e+00  1.336e-01   7.728 1.09e-14
## dti                              2.115e-02  8.820e-03   2.398 0.016483
## bc_util                          9.028e-03  2.690e-03   3.356 0.000789
## inq_last_6mths                   1.691e-01  6.040e-02   2.799 0.005120
## adjusted_annual_inc             -3.637e-06  1.775e-06  -2.049 0.040505
## residence_propertyOwn           -2.735e-01  1.364e-01  -2.005 0.045009
##
## (Intercept)                      ***
## fico_score                       ***
## highest_ed_levelHigh School      ***
## highest_ed_levelBachelors        ***
## highest_ed_levelMasters          ***
## highest_ed_levelPhD or Doctorate ***
## us_region_residenceMid-Atlantic  ***
## us_region_residenceSouth         ***
## us_region_residenceMidwest
## us_region_residenceSouthwest     ***
## us_region_residenceWest          ***
## age_category24 - 29              ***
## age_category30 - 34              ***
## age_category35 - 40              ***
```

```
## age_category41 - 50              ***
## age_category51 and older         ***
## genderMale                       ***
## dti                              *
## bc_util                          ***
## inq_last_6mths                   **
## adjusted_annual_inc              *
## residence_propertyOwn            *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2805.4  on 2620  degrees of freedom
## Residual deviance: 1520.7  on 2599  degrees of freedom
## AIC: 1564.7
##
## Number of Fisher Scoring iterations: 6
```

```
optimal_loan_model <- glm(loan_default ~ fico_score + highest_ed_level +
    us_region_residence + age_category + gender + dti + bc_util +
    inq_last_6mths + adjusted_annual_inc + residence_property,
    family = "binomial", data = loan_training)

summary(optimal_loan_model)
```

```
##
## Call:
## glm(formula = loan_default ~ fico_score + highest_ed_level +
##     us_region_residence + age_category + gender + dti + bc_util +
##     inq_last_6mths + adjusted_annual_inc + residence_property,
##     family = "binomial", data = loan_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4480  -0.4407  -0.2182  -0.0628   3.7058
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     1.053e+01  6.610e-01  15.931  < 2e-16
## fico_score                     -1.754e-02  9.604e-04 -18.263  < 2e-16
## highest_ed_levelHigh School     1.191e+00  2.678e-01   4.448 8.68e-06
## highest_ed_levelBachelors      -1.409e+00  2.155e-01  -6.537 6.29e-11
## highest_ed_levelMasters        -1.414e+00  2.412e-01  -5.861 4.59e-09
## highest_ed_levelPhD or Doctorate -8.834e-01  2.663e-01  -3.317 0.000910
## us_region_residenceMid-Atlantic -1.759e+00  1.891e-01  -9.301  < 2e-16
## us_region_residenceSouth       -1.627e+00  2.968e-01  -5.483 4.19e-08
## us_region_residenceMidwest     -1.728e-01  2.050e-01  -0.843 0.399363
## us_region_residenceSouthwest   -1.794e+00  2.926e-01  -6.131 8.74e-10
## us_region_residenceWest        -1.560e+00  1.944e-01  -8.024 1.03e-15
## age_category24 - 29            -8.908e-01  2.005e-01  -4.444 8.85e-06
## age_category30 - 34            -8.827e-01  2.203e-01  -4.007 6.15e-05
## age_category35 - 40            -1.835e+00  2.275e-01  -8.066 7.27e-16
## age_category41 - 50            -1.836e+00  2.319e-01  -7.917 2.44e-15
## age_category51 and older       -1.086e+00  2.260e-01  -4.804 1.56e-06
```

```
## genderMale                        1.032e+00  1.336e-01   7.728 1.09e-14
## dti                               2.115e-02  8.820e-03   2.398 0.016483
## bc_util                           9.028e-03  2.690e-03   3.356 0.000789
## inq_last_6mths                    1.691e-01  6.040e-02   2.799 0.005120
## adjusted_annual_inc              -3.637e-06  1.775e-06  -2.049 0.040505
## residence_propertyOwn            -2.735e-01  1.364e-01  -2.005 0.045009
##
## (Intercept)                      ***
## fico_score                       ***
## highest_ed_levelHigh School      ***
## highest_ed_levelBachelors        ***
## highest_ed_levelMasters          ***
## highest_ed_levelPhD or Doctorate ***
## us_region_residenceMid-Atlantic  ***
## us_region_residenceSouth         ***
## us_region_residenceMidwest
## us_region_residenceSouthwest     ***
## us_region_residenceWest          ***
## age_category24 - 29              ***
## age_category30 - 34              ***
## age_category35 - 40              ***
## age_category41 - 50              ***
## age_category51 and older         ***
## genderMale                       ***
## dti                              *
## bc_util                          ***
## inq_last_6mths                   **
## adjusted_annual_inc              *
## residence_propertyOwn            *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2805.4  on 2620  degrees of freedom
## Residual deviance: 1520.7  on 2599  degrees of freedom
## AIC: 1564.7
##
## Number of Fisher Scoring iterations: 6
```

Above you can see our optimal loan model which is the result of a step wise search direction of "both". The optimal model produces an AIC score of 1564.7 while the upper model produces a score of 1570.8. The optimal model removes 5 of the 15 original variables to produce it's outcome. Due to the high P values associated with adjusted_annual_inc and residence_propertyOwn, we explored removing them from our optimal model. Removing them ultimately resulted in an increased AIC score which is indicative of a lesser quality model.

Predictive Modeling

Classification Method 1: Predicting loan_default

```
lda_loan_default <- lda(loan_default ~ .,
                        data = loan_training,
                        CV =  FALSE)
                  names(lda_loan_default)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```r
lda_pred_training <- predict(lda_loan_default, newdata = loan_training)

lda_results_training <- data.frame(loan_training, lda_pred_0.5 = lda_pred_training$class, lda_pred_trai


cf_matrix(actual_vec = lda_results_training$loan_default, pred_prob_vec = lda_results_training$Yes, pos
```

```
##    cut_prob correct_rate  F1_score false_pos_rate false_neg_rate
## 1      0.00    0.2266311 0.3695179    1.000000000     0.00000000
## 2      0.05    0.6852346 0.5758355    0.390231870     0.05723906
## 3      0.10    0.7706982 0.6411940    0.268376912     0.09595960
## 4      0.15    0.8157192 0.6824458    0.201282684     0.12626263
## 5      0.20    0.8431896 0.7074733    0.154908732     0.16329966
## 6      0.25    0.8580694 0.7190332    0.125308337     0.19865320
## 7      0.30    0.8653186 0.7187251    0.103601381     0.24074074
## 8      0.35    0.8695155 0.7159468    0.088307844     0.27441077
## 9      0.40    0.8752385 0.7154047    0.071040947     0.30808081
## 10     0.45    0.8760015 0.7058824    0.059694129     0.34343434
## 11     0.50    0.8733308 0.6897196    0.052787370     0.37878788
## 12     0.55    0.8782907 0.6911907    0.040453873     0.39898990
## 13     0.60    0.8744754 0.6680121    0.032560434     0.44276094
## 14     0.65    0.8702785 0.6443515    0.026640355     0.48148148
## 15     0.70    0.8676078 0.6272825    0.022200296     0.50841751
## 16     0.75    0.8630294 0.5988827    0.016280217     0.54882155
## 17     0.80    0.8580694 0.5694444    0.011840158     0.58585859
## 18     0.85    0.8500572 0.5259349    0.008386778     0.63299663
## 19     0.90    0.8374666 0.4593909    0.006413419     0.69528620
## 20     0.95    0.8157192 0.3263598    0.002960039     0.80303030
## 21     1.00    0.7733689 0.0000000    0.000000000     1.00000000
```

```r
loandefaultlda <- cf_matrix(actual_vec = lda_results_training$loan_default, pred_prob_vec = lda_results_

loandefaultlda
```

```
## $confusion_matrix
##           metric observations      rate pct_total_obs
## 1        Correct         2249 0.8580694    0.85806944
## 2  Misclassified          372 0.1419306    0.14193056
## 3  True Positive          476 0.8013468    0.18161007
## 4  True Negative         1773 0.8746917    0.67645937
## 5 False Negative          118 0.1986532    0.04502098
## 6 False Positive          254 0.1253083    0.09690958
##
## $F1_summary
##      metric     value
## 1 Precision 0.6520548
## 2    Recall 0.8013468
## 3  F1 Score 0.7190332
```

```r
#Analysis: In the training data, the model that had the default probability cut-off value of .5 had a F

lda_pred_test <- predict(lda_loan_default, newdata = loan_test)
```

```
lda_results_test <- data.frame(loan_test,
                               lda_pred_0.5 = lda_pred_test$class, lda_pred_test$posterior)

lda_results_test <- lda_results_test %>%
                mutate(lda_pred_0.25 = ifelse(Yes >= 0.25, "Yes", "No"))

cf_matrix(actual_vec = lda_results_test$loan_default, pred_prob_vec = lda_results_test$Yes, positive_va
```

```
## $confusion_matrix
##            metric observations      rate pct_total_obs
## 1         Correct          944 0.8398577    0.83985765
## 2  Misclassified          180 0.1601423    0.16014235
## 3  True Positive          170 0.6719368    0.15124555
## 4  True Negative          774 0.8886338    0.68861210
## 5 False Negative           83 0.3280632    0.07384342
## 6 False Positive           97 0.1113662    0.08629893
##
## $F1_summary
##      metric     value
## 1 Precision 0.6367041
## 2    Recall 0.6719368
## 3  F1 Score 0.6538462
```

```
#Make Predictions : The test data in comparison to the training data had a weaker F1 score when using t
```

Classification Method 2: Predicting loan_default

```
qda_loan_default <- qda(loan_default ~ .,
                        data = loan_training,
                        CV = FALSE)
                  names(qda_loan_default)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "ldet"    "lev"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```
qda_pred_training <- predict(qda_loan_default, newdata = loan_training)

qda_results_training <- data.frame(loan_training, qda_pred_0.5 = qda_pred_training$class, qda_pred_trai

cf_matrix(actual_vec = qda_results_training$loan_default, pred_prob_vec = qda_results_training$Yes, pos
```

```
##    cut_prob correct_rate  F1_score false_pos_rate false_neg_rate
## 1      0.00    0.2266311 0.3695179     1.00000000      0.0000000
## 2      0.05    0.7355971 0.6033200     0.30883078      0.1127946
## 3      0.10    0.7737505 0.6319056     0.25061667      0.1430976
## 4      0.15    0.7905380 0.6460348     0.22496300      0.1565657
## 5      0.20    0.8004578 0.6534129     0.20818944      0.1700337
## 6      0.25    0.8080885 0.6585200     0.19437593      0.1835017
## 7      0.30    0.8168638 0.6643357     0.17809571      0.2003367
## 8      0.35    0.8248760 0.6723769     0.16576221      0.2070707
## 9      0.40    0.8283098 0.6729651     0.15737543      0.2205387
## 10     0.45    0.8328882 0.6745914     0.14701529      0.2356902
## 11     0.50    0.8351774 0.6737160     0.14010853      0.2491582
## 12     0.55    0.8347959 0.6676899     0.13517514      0.2676768
## 13     0.60    0.8347959 0.6630350     0.13073508      0.2828283
## 14     0.65    0.8382297 0.6634921     0.12234830      0.2962963
```

```
## 15     0.70    0.8389928 0.6585761     0.11593488      0.3148148
## 16     0.75    0.8424266 0.6561199     0.10508140      0.3367003
## 17     0.80    0.8428081 0.6472603     0.09669462      0.3636364
## 18     0.85    0.8454788 0.6412755     0.08534780      0.3905724
## 19     0.90    0.8523464 0.6439742     0.07054761      0.4107744
## 20     0.95    0.8512018 0.6183953     0.05525407      0.4680135
## 21     1.00    0.7733689 0.0000000     0.00000000      1.0000000
```

```r
loandefaultqda <- cf_matrix(actual_vec = qda_results_training$loan_default, pred_prob_vec = qda_results_

loandefaultqda
```

```
## $confusion_matrix
##           metric observations      rate pct_total_obs
## 1        Correct         2183 0.8328882    0.83288821
## 2  Misclassified          438 0.1671118    0.16711179
## 3  True Positive          454 0.7643098    0.17321633
## 4  True Negative         1729 0.8529847    0.65967188
## 5 False Negative          140 0.2356902    0.05341473
## 6 False Positive          298 0.1470153    0.11369706
##
## $F1_summary
##      metric     value
## 1 Precision 0.6037234
## 2    Recall 0.7643098
## 3  F1 Score 0.6745914
```

```r
#Analysis: In the training data, the model that had the default probability cut-off value of .5 had a F

qda_pred_test <- predict(qda_loan_default, newdata = loan_test)

qda_results_test <- data.frame(loan_test,
                               qda_pred_0.5 = qda_pred_test$class, qda_pred_test$posterior)

qda_results_test <- qda_results_test %>%
                  mutate(qda_pred_0.45 = ifelse(Yes >= 0.45, "Yes", "No"))

cf_matrix(actual_vec = qda_results_test$loan_default, pred_prob_vec = qda_results_test$Yes, positive_va
```

```
## $confusion_matrix
##           metric observations      rate pct_total_obs
## 1        Correct          898 0.7989324    0.79893238
## 2  Misclassified          226 0.2010676    0.20106762
## 3  True Positive          154 0.6086957    0.13701068
## 4  True Negative          744 0.8541906    0.66192171
## 5 False Negative           99 0.3913043    0.08807829
## 6 False Positive          127 0.1458094    0.11298932
##
## $F1_summary
##      metric     value
## 1 Precision 0.5480427
## 2    Recall 0.6086957
## 3  F1 Score 0.5767790
```

```r
#Make Predictions: The test data in comparison to the training data had a weaker F1 score when using th
```

Classification Method 3: Predicting loan_default

```
logistic_fit <- glm(loan_default ~ .,
                    data = loan_training,
                    family = "binomial")

logistics_results_training <- data.frame(loan_training,
                                         logistic_prob = predict(logistic_fit, newdata = loan_training,

cf_matrix(actual_vec = logistics_results_training$loan_default, pred_prob_vec = logistics_results_train
```

```
## $confusion_matrix
##            metric observations       rate pct_total_obs
## 1         Correct         2300 0.87752766    0.87752766
## 2  Misclassified          321 0.12247234    0.12247234
## 3  True Positive          375 0.63131313    0.14307516
## 4  True Negative         1925 0.94967933    0.73445250
## 5 False Negative          219 0.36868687    0.08355589
## 6 False Positive          102 0.05032067    0.03891644
##
## $F1_summary
##      metric     value
## 1 Precision 0.7861635
## 2    Recall 0.6313131
## 3  F1 Score 0.7002801
```

```
logisticreg <- cf_matrix(actual_vec = logistics_results_training$loan_default, pred_prob_vec = logistic

logisticreg
```

```
## $confusion_matrix
##            metric observations      rate pct_total_obs
## 1         Correct         2291 0.8740939    0.87409386
## 2  Misclassified          330 0.1259061    0.12590614
## 3  True Positive          447 0.7525253    0.17054559
## 4  True Negative         1844 0.9097188    0.70354826
## 5 False Negative          147 0.2474747    0.05608546
## 6 False Positive          183 0.0902812    0.06982068
##
## $F1_summary
##      metric     value
## 1 Precision 0.7095238
## 2    Recall 0.7525253
## 3  F1 Score 0.7303922
```

```
#Analysis: In the training data, the model that had the default probability cut-off value of .5 had a F

logistic_results_test <- data.frame(loan_test,
                                    logistic_prob = predict(logistic_fit, newdata = loan_test, type = "

logistic_results_test <- logistic_results_test %>% mutate(logistic_pred_0.35 = ifelse(logistic_prob >= 

cf_matrix(actual_vec = logistic_results_test$loan_default, pred_prob_vec = logistic_results_test$logist
```

```
## $confusion_matrix
##            metric observations       rate pct_total_obs
```

```
## 1        Correct          959 0.85320285    0.85320285
## 2  Misclassified          165 0.14679715    0.14679715
## 3  True Positive          160 0.63241107    0.14234875
## 4  True Negative          799 0.91733639    0.71085409
## 5 False Negative           93 0.36758893    0.08274021
## 6 False Positive           72 0.08266361    0.06405694
##
## $F1_summary
##      metric     value
## 1 Precision 0.6896552
## 2    Recall 0.6324111
## 3  F1 Score 0.6597938
```

```
#Make Predictions: The test data in comparison to the training data had a weaker F1 score when using th
```

BONUS KNN CLASSIFICATION METHOD

```
train.kknn(loan_default ~ .,
          data = loan_training,
          kmax = 40)
```

```
##
## Call:
## train.kknn(formula = loan_default ~ ., data = loan_training,     kmax = 40)
##
## Type of response variable: nominal
## Minimal misclassification: 0.1617703
## Best kernel: optimal
## Best k: 26
```

```
#Best K = 26

knn_loandefault_training <- kknn(loan_default ~ ., train = loan_training,
                                 test= loan_training,
                                 k = 26, distance = 2)

knn_loanresults_training <- data.frame(loan_training,
                                   knn_pred_0.5 = knn_loandefault_training$fitted.values,
                                   knn_loandefault_training$prob)

cf_matrix(actual_vec = knn_loanresults_training$loan_default,
          pred_prob_vec = knn_loanresults_training$Yes,
          positive_val = "Yes",
          search_cut = TRUE)
```

```
##    cut_prob correct_rate    F1_score false_pos_rate false_neg_rate
## 1      0.00    0.2266311 0.369517885   1.0000000000     0.00000000
## 2      0.05    0.5547501 0.504458599   0.5757276764     0.00000000
## 3      0.10    0.7046929 0.605504587   0.3818450913     0.00000000
## 4      0.15    0.8035101 0.692537313   0.2471632955     0.02356902
## 5      0.20    0.8714231 0.768384880   0.1489886532     0.05892256
## 6      0.25    0.8916444 0.782874618   0.0996546621     0.13804714
## 7      0.30    0.9053796 0.789115646   0.0582141095     0.21885522
## 8      0.35    0.9034720 0.766820276   0.0370004933     0.29966330
## 9      0.40    0.9023274 0.747035573   0.0197335964     0.36363636
## 10     0.45    0.8832507 0.670967742   0.0118401579     0.47474747
```

```
## 11      0.50    0.8660816 0.595155709   0.0074000987     0.56565657
## 12      0.55    0.8508203 0.516687268   0.0029600395     0.64814815
## 13      0.60    0.8347959 0.431011827   0.0014800197     0.72390572
## 14      0.65    0.8210607 0.349514563   0.0004933399     0.78787879
## 15      0.70    0.8096147 0.275761974   0.0000000000     0.84006734
## 16      0.75    0.7993132 0.205438066   0.0000000000     0.88552189
## 17      0.80    0.7924456 0.155279503   0.0000000000     0.91582492
## 18      0.85    0.7832888 0.083870968   0.0000000000     0.95622896
## 19      0.90    0.7787104 0.046052632   0.0000000000     0.97643098
## 20      0.95    0.7756581 0.020000000   0.0000000000     0.98989899
## 21      1.00    0.7741320 0.006711409   0.0000000000     0.99663300
```

```r
#test

knn_loandefault_test <- kknn(loan_default ~ ., train = loan_training,
                                 test= loan_test,
                                 k = 26, distance = 2)

knn_loanresult_test <- data.frame(loan_test,
                               knn_pred_0.5 = knn_loandefault_test$fitted.values,
                               knn_loandefault_test$prob)

knn_results_test <- knn_loanresult_test %>%
                    mutate(knn_pred_0.3 = ifelse(Yes >= 0.3, "Yes", "No"))

cf_matrix(actual_vec = knn_loanresult_test$loan_default,
          pred_prob_vec = knn_loanresult_test$Yes,
          positive_val = "Yes",
          cut_prob = .3)
```

```
## $confusion_matrix
##           metric observations       rate pct_total_obs
## 1        Correct          925 0.82295374    0.82295374
## 2   Misclassified          199 0.17704626    0.17704626
## 3   True Positive          135 0.53359684    0.12010676
## 4   True Negative          790 0.90700344    0.70284698
## 5 False Negative          118 0.46640316    0.10498221
## 6 False Positive           81 0.09299656    0.07206406
##
## $F1_summary
##      metric     value
## 1 Precision 0.6250000
## 2    Recall 0.5335968
## 3  F1 Score 0.5756930
```

Summary of Findings and Recommendations

Through our exploratory analysis we discovered a number of relations relating to loan default rates. First we started with gender and discovered that males have more than twice the default rate on loans than females with rates of 33.2% and 15.0% respectively. Next, it appears that individuals with less formal education default on their loans more frequently. Those with a high school level education showed a 61.6% default rate while < high school exhibited a 43.6% default rate. Region seemed to play a role as well. Individuals living in the Northeast and Midwest have a significantly higher rate of default at 40.4% and 38.3% respectively. Apparent in our "income" data frame is the relationship between defaulting on loans and levels of income. Those making less than 10k/year showed a default rate of 38.5%. Individuals making 10-50k have a default rate of 25.9%. It appears that customers with a greater amount of credit inquiries default at a higher rate.

The relationship between open accounts and default rates seems to be a negative one. As open accounts increase, default rates seem to decline. However this could be indicative of outliers in our data set or a lack of individuals with more than 28 open accounts. Surprisingly the relationship between public bankruptcies and loan defaults is negative as well. Again, this could be due to the limited number of individuals in the data set with >= 2 bankruptcies. In our fico score defaults dataframe above, we see the average fico score for individuals in all four ranges, poor, fair, good, and exceptional, who did or did not default on their loans. There is no obvious difference in score between the two groups aside from the "Poor" fico score. Those who defaulted in the poor group had on average a 48 point lower fico score than their counterparts who did not default. It does not appear that there is any relationship between annual adjusted income and dti as they relate to loan default rates in individuals under 24 years old. It is intuitive however, that those with a greater annual adjusted income with a lower dti would at the very least, be slightly less likely to default.

Above, in our variable selection with logistic regression, you can see our optimal loan model which is the result of a step wise search direction of "both". The optimal model produces an AIC score of 1564.7 while the upper model produces a score of 1570.8. The optimal model removes 5 of the 15 original variables to produce it's outcome. Due to the high P values associated with adjusted_annual_inc and residence_propertyOwn, we explored removing them from our optimal model. Removing them ultimately resulted in an increased AIC score which is indicative of a lesser quality model.

Next we attempted to tackle predicitive modeling by fitting linear, quadratic, logistic, and knn models on training data. Ultimately two of the models prevailed when run against our test data. They are the linear discriminant analysis and logistic regression models. The models both posted impressive F1 scores with our lda model at 65.38 and our glm model at 65.98. However, despite the greater F1 score of our glm model, we recommend using our lda model. The lda model posted a 32.81 false negative rate as opposed to the 36.76 false negative rate of our glm model. In this case, a false negative would be predicting that someone would not default on their loan, when actually they did. We as a nation witnessed the possible consequences of sub-prime loans and the damage they can do to our global economy. For that reason it is imperative that lenders place priority on avoiding these false negative outcomes. We recommend that lenders account for all of the variables included in our optimal model when considering lending money to potential borrowers.