Final Project

```
# Suppress dplyr summarise grouping warning messages
options(dplyr.summarise.inform = FALSE)

## Add R libraries here
library(tidyverse)
library(tidymodels)
library(discrim)


# Load data
loans_df <- read_rds("C:/Users/hecto/Downloads/loan_data.rds")
```

Introduction

Improving loan processes is important for lenders and borrowers. Keeping default rates low is important for the bank because when customers are defaulting on their loans, it costs the bank money. Another reason for the bank to keep default rates low is because it helps the bank keep a good reputation in the market which will help the bank attract more customers. We will be analyzing the loans data set that we have received from a local bank to develop solutions.

The goal of the analysis of the loans data set is to identify which factors could be contributing to customers defaulting on their loans. To uncover the factors that are contributing to customers defaulting on their loans we must ask questions between variables in the data set. The following questions that will be asked are:

1. Are there differences in loan default rates by annual income and loan purpose?
2. Is there a difference in loan default rates by homeowner ship and average loan amount?
3. Is there a difference in default rates by the term of the loan?
4. Is there a difference in loan default rate by application type?
5. Is there a relationship between the customers that default and those that do not based on their total credit lines and the loan purpose?
6. Is there a relationship between how many customers default on their loans based on interest rates and loan amounts?

The following section is the exploratory data analysis section where these questions will be discussed. Followed by where the best of three classification models will be discussed. In this section we will dicuss the ROC, are under ROC curve, and the confusion matrix in detail. Followed by the recommendations section based off the analysis of the six questions. In the conclusion section we will revisit the recommendations.

Exploratory data analysis

One key finding from the first table is that the credit card and medical loan purpose groups had more than half of the customers default on their loans. For credit card loans the default rate was 53% and the average income was the lowest out of all with 69458.32. For medical loans, the default loan rate was 60% and it had the highest average income of 74581.36. The rest of the loan purposes had default loan rates under 30% and average income between 73,000 and 75,000. This is important for the business because we can see that all these loan purposes have similar average incomes, but in for credit card loans and medical loans customers are defaulting on their loans at a high rate.

From the second table a key finding was that the customers that rent have the highest default rate at 42.79% but have the lowest average loan amount at 14,996.46. Another key finding is that the bank's customers mostly come from the mortgage group with 1937 customers, but only 628 customers have defaulted on their loans resulting in a low default rate of 32.42% compared to the group of customers that rent. The group of customers that own is the smallest for the bank with 507, but only 189 customers have defaulted on their loan. The default rate for this group is 37.27%. This is important for the bank because even though the group that rents has the lowest average loan amount, it has the highest loan default rate.

A key finding for the third table is that most of the bank's customers are given a loan term of three years. 2588 in total have been given a three-year loan term, but only 693 customers have defaulted on their loan. The three-year term has a loan default rate of 26.77%. The bank has given 1,522 customers a loan term of 5 years and 837 customers have defaulted. The 5-year term has a default rate of 54.99%. This is important for the bank as there has been more success with customers completing 3-year terms than 5 year terms.

The bar graph shows us that joint application type has a higher loan default rate than the individual loan rate. The joint application type has a loan default rate of 45% and the individual application type has a default rate of about 36%. This is important to the bank as this is about a 9% difference between the two application types.

Based off the first table we know that loans for medical and credit card purposes are significantly higher than the rest. From the histogram we can see that throughout all the loans most customers that default has between 10 to 15 years of credit history, but this is understandable since we can see that most customers that don't default also have 10 to 15 years of credit history. This is important for the bank because although most of their customers that don't default have between 10 to 15 years of credit history, they also are the ones that are raising the default rate for the bank.

From the scatter plot we can see that at about 9% interest rate is where we start to see customers that default on their loans, but there is also still a good mix of customers that do not default on their loans. Beginning at about 14% interest rate is where we start seeing customers that default on their loans no matter the loan amount. Between 4% and 8% interest rate is where there are zero customers that default on their loan no matter the loan amount. Most customers that do not default on their loans are below the 20,000-loan amount on the scatter plot. For customers that default on their loans, the dots are spread more across the scatter plot. The key findings from the scatter plot are important to the bank because it gives insight into loan default rates in correlation with interest rates and loan amounts.

Best classification model

The best classification model out of the three was the logistic regression model. When looking at the logistic regression model ROC curve the closer the curve is to 1, the better the model is predicting accurately between classes. Calculating the area under the ROC curve we get a result of .99. This result is extremely close to 1, so we can conclude that the model does a good job at predicting accurately between the classes in the data set. Furthermore, the confusion matrix will give us insight into how accurately the model predicted. A confusion matrix is a summary table that shows us the number of correct predictions and errors made by the model. The correct number of correct predictions are called true positives and true negatives. The number of wrong predictions or errors are called false positives and false negatives. When we look at the confusion matrix of the logistic regression model, we can see that the model had 360 true positives. The model accurately predicted the number of customers that did default on their loan. The model had 629 true negatives meaning that the model accurately predicted the number of customers that did not default on their loans. The model had 16 false negatives meaning that it predicted 16 customers would not default on their loans, but in fact did default. The model had 23 false positives meaning that it predicted 23 times that a customer would default on their loans, but they did not default on their loans. Adding up the true positives and true negatives we get a total of 989 times that the logistic regression model predicted accurately.

The logistic regression model was compared to an QDA model and a random forest model. The QDA model had a good area under the ROC curve of .97 but the logistic regression model had a slightly better area under the ROC curve of .99. The random forest model also had an area under ROC curve of .97. The QDA model had 347 true positives, 47 false positives, 598 true negatives, and 36 false negatives. The random forest model had 314 true positives, 27 false positives, 618 true negatives, and 69 false negatives. Comparing the ROC curve and confusion matrix of these two models to the logistic regression, we can see that the logistic regression model performed the best.

Recommendations

My first recommendation to reduce loan default rates would be for the bank to look into raising the average annual income requirement for customers that want a credit card or medical loan. From the table we can see that customers in all the loan purposes groups have similar average income except for the credit card loan purpose group. In all the groups the average annual loan amount is also similar across groups with a loan amount between 16,000 and 17,000. Raising the average annual income requirement for credit cards and medical purposes will help the bank as customers reduce the loan default rate as customers with a high income will be more likely to make regular payments.

My second recommendation for the bank would be to reduce the amount of loans they give to customers that rent. Customers that rent make up the second biggest customer demographic for the bank at 1,666 but have had 713 of them default on their loans. This makes them the group with the highest loan default rate at 42.79%. Customers that have a mortgage make up the biggest demographic for the bank at 1,937 customers and have a lower number of customers that rent at 628. This is important to note because the bank has 271 more customers that have a mortgage than customers that rent. Overall, customers that rent are raising the bank's loan default rate. Lowering the number of loans, they give out to customers that rent can help lower the loan default rate.

My third recommendation is to reduce the number of five-year loan terms that the bank gives out. The loan default rate for five-year terms is at 54.99% which is significantly higher than the loan default rate for three-year terms that is at 26.77. The bank has a total of 2,588 customers with a three-year loan term and 1,522 customers with a five-year loan term. This is a 1,066 difference between the two groups, but the low default loan rate of 26.77% compared to 54.99% of five-year terms shows how successful the three year terms have been for the bank.

My fourth recommendation would be to reduce the number of loans that are given to joint applicants. From the bar graph we can see that joint applications have a 45% default rate loan, and individual applications have about a 36% default rate loan. This could be because for an individual application the borrower is solely responsible. For a joint loan there is more than one borrower involved which could lead to problems which can then lead to a default loan. The bank could look into reducing the number of joint applications they approve and see if the loan default rate gets better. My fifth recommendation would be for the bank to investigate changing the requirement for number of years of credit history when it comes to credit card and medical loan purposes. For all loan purposes most of the customers have between 10 to 15 years of credit. For this reason it makes sense that for debt consolidation, small business, and home improvement the customers that default are mostly customers that have between 10 to 15 years of credit history, but the difference is that the number of customers that default for these loan purposes are not that high compared to the number for medical and credit card purposes. Overall, looking into increasing or decreasing the number of years of credit history for medical and credit card purposes could bring change to the loan default rate for these loan purposes.

My last recommendation for the bank is to investigate trying to reduce interest rates. Looking at the scatter plot, starting at about 9% is where loan defaults start no matter the loan amount. At about 8% interest rate and below there are zero customers that default on their loans. Between 9% and 13% there is a good mix of customers that do not default and those that do. My recommendation is for the bank to investigate trying to reduce interest rates below about 14% because that's where default loans start happening no matter the loan amount. Lowering the interest rate for loans could negatively impact the bank as it can lead to a reduction in profit. Lowering interest rates below about 14% could help lower default rates and it can also positively impact the bank by an increase in demand for loans from customers.

Conclusion

For the conclusion, the results and recommendations will be revisited. The first recommendation was for the bank to increase the average annual income of customers when approving a loan for medical and credit card loan purposes. The default loan rate for these purposes was significantly higher than the rest. The average income in all of the different loan purposes is similar, but the default loan rates for the other loan purposes are low compared to credit card and medical. My recommendation for the bank is that when giving out loans for credit card and medical loan purposes to require customers to make a higher annual income than what the average is right now. Giving out loans to customers that make a higher annual income than the average could result in customers being able to pay their loans which will lead to a decrease in default loan rates for medical and credit card loan purposes. The second recommendation was for the bank to reduce the number of loans that they give out to customers that rent. Customers that rent make up the second biggest customer demographic for the bank and have the highest loan default rate. Lowering the number of loans given out to customers can help lower the loan default rate for the bank. The third recommendation was for the bank to reduce the number of five-year loan terms that are given out. The default loan rate for five-year terms at 54.99% is significantly higher than the default loan rate for three-year terms at 26.77%. The bank has had much more success with customers completing three-year terms than five-year terms. Lowering the number of five-year terms given out could help lower the default loan rate. The fourth recommendation was for the bank to lower the amount of loans that are approved for joint applications. Joint applications have a default loan rate of 45% and individual application default loan rates are at about 36%. Lowering the number of joint applications approved for loans could help lower the default loan rate. The fifth recommendation for the bank was to look into increasing the number of years of credit history for credit card and medical loan purposes. For all loan purposes most customers have between 10 to 15 years of credit history, but only for credit card and medical loan purposes do customers default on their loans at a high rate. Trying out different number of years of credit history needed for a credit card and medical loan purposes can help change the high default loan rates for medical and credit card loan purposes. The last recommendation was to investigate trying to reduce interest rates. Between 4% and 8% interest rates customers do not default on their loans no matter the loan amount. Starting at 9% interest rate is where customers start to default on their loans, but there is still a good mix of those that do not between 9% and 13% interest rate. Staring at 14% interest rate is where customers start to default on their loans no matter the loan amount. The bank could look into trying to reduce interest rate below 14%. Reducing interest rates could have an impact on the banks profit, but it could also attract more customers. Overall, lowering interest rates below 14% should help lower loan default rates.

In conclusion, lowering loan default rates is a difficult challenge as there are many factors that play into why customers default on their loans. Moving forward, when making changes the bank should continue to analyze. With the recommendations mentioned above the bank should start to be able to lower loan default rates. As the bank implements changes, analyzation of data should continue to see if the changes have brought positive change or not.

Question 1

Question: Are there differences in loan default rates by annual income and loan purpose?

Answer: For this table I grouped the table by the purpose of the loan. I then counted the number of customers and then summed up the number of customers that defaulted on their loan. I then calculated the average of each group's annual income. I then calculated the percentage of the default percent of customers that did default on their loans. The table shows us that credit card and medical loans have significantly higher default rates than any of the other loan purposes. The credit card loan purpose group had the lowest average annual income with 69,5458.32 and had the second highest default loan rate at 53.47%. The medical loan purpose group had the second highest average annual income with 74,581.36 but had the highest default loan rate at 60.47%.

```
loans_df %>%
group_by(loan_purpose) %>%
  summarise(number_customers = n(),
            customers_defualt = sum(loan_default == "yes"),
            avg_annual_income = mean(annual_income),
            avg_loan_amount = mean(loan_amount),
            default_percent = 100 * mean(loan_default == 'yes'))
```

```
## # A tibble: 5 × 6
##   loan_purpose       number_customers customers_defualt avg_annual_income
##   <fct>                         <int>             <int>             <dbl>
## 1 debt_consolidation             1218               308            73628.
## 2 credit_card                     879               470            69458.
## 3 medical                         635               384            74581.
## 4 small_business                  853               221            73582.
## 5 home_improvement                525               147            74731.
## # ℹ 2 more variables: avg_loan_amount <dbl>, default_percent <dbl>
```

Question 2

Question: Is there a difference in loan default rates by homeowner ship and average loan amount?

Answer: For this table I grouped the table by home ownership. I then counted the number of customers in each group. I then summed up the number of customers that did default on their loans. I then calculated the average loan amount given to each group. I then calculated the rate at which customers did default on their loans. The table shows us that the group that rents has the highest loan default rate with 42.79% but has the lowest average loan amount. The group that owns has the second highest loan default rate with 37.27% and has the second highest average loan amount with 16,513.95. The group that has a mortgage has the highest number of customers, lowest default rate, and has the highest average loan amount.

```
loans_df %>%
  group_by(homeownership) %>%
  summarise(number_customers = n(),
            customers_defualt = sum(loan_default == "yes"),
            avg_loan_amount = mean(loan_amount),
            default_percent = 100 * mean(loan_default == 'yes'))
```

```
## # A tibble: 3 × 5
##   homeownership number_customers customers_defualt avg_loan_amount
##   <fct>                    <int>             <int>           <dbl>
## 1 mortgage                  1937               628          18199.
## 2 rent                      1666               713          14996.
## 3 own                        507               189          16514.
## # ℹ 1 more variable: default_percent <dbl>
```

Question 3

Question: Is there a difference in default rates by the term of the loan?

Answer: For this table I grouped by the term of the loan. I then calculated the total number of customers and then summed up the number of customers that defaulted on their payments. Then calculated the default percentage of customers that did default on their payments. Based on the table the loans with a five-year term had a significantly higher loan default percent at 54.99% compared to the three-year term at 26.77%. The loan term of three years has a total of 2,558 customers with only 693 customers defaulting on their loan. The five-year loan term had a total of 1,552 with more than half of customers defaulting on their payments with 837.

```
loans_df %>%
  group_by(term) %>%
  summarise(number_customers = n(),
            customers_defualt = sum(loan_default == "yes"),
            default_percent = 100 * mean(loan_default == 'yes'))
```

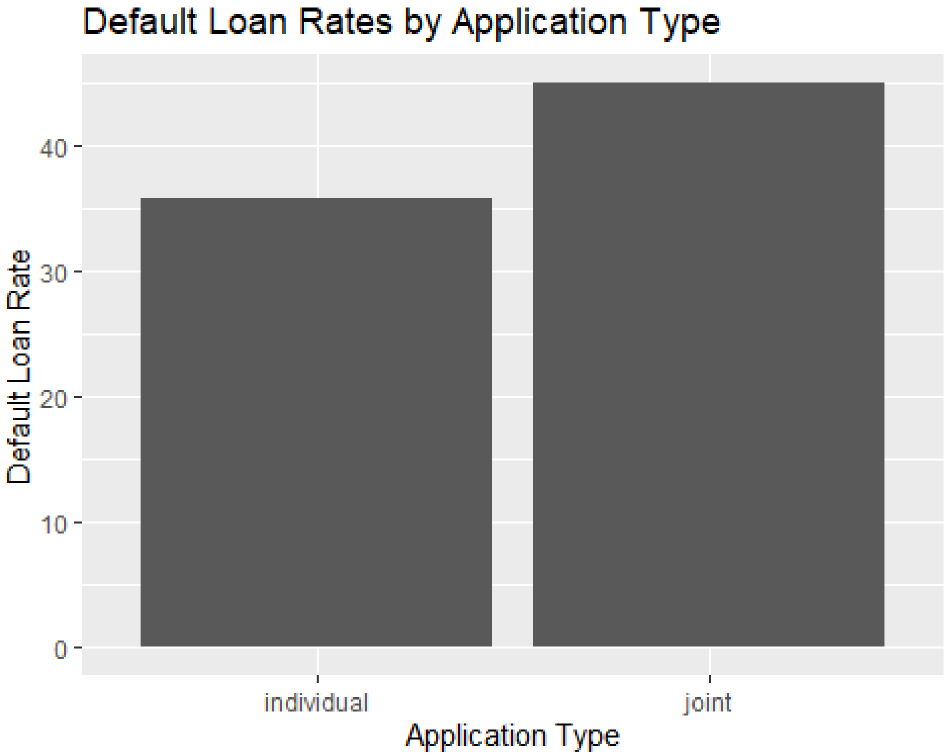
```
## # A tibble: 2 × 4
##   term       number_customers customers_defualt default_percent
##   <fct>                 <int>             <int>           <dbl>
## 1 three_year             2588               693            26.8
## 2 five_year              1522               837            55.0
```

Question 4

Question: Is there a difference in loan default rate by application type?

Answer: This graph was created by first creating a table by grouping the table by application type. Then counted the number of customers in each group. I then summed up the customers that did default on their payments. After I calculated the default percentage of customers that did default on their payments. After creating the table I passed it on to ggplot to create a bar graph. I placed the application type on the x axis and the default loan rates on the y axis. From the bar graph we can see that joint application type has a default loan rate of 45% and the individual application type has a loan default rate of about 36%.

```
DF1 <- loans_df %>%
  group_by(application_type) %>%
  summarise(number_customers = n(),
            customers_defualt = sum(loan_default == "yes"),
            default_percent = 100 * mean(loan_default == 'yes'))

ggplot(DF1, aes(x = application_type, y = default_percent)) +
  geom_bar(stat = 'identity') +
  labs(title = 'Default Loan Rates by Application Type', x = 'Application
Type', y = 'Default Loan Rate')
```

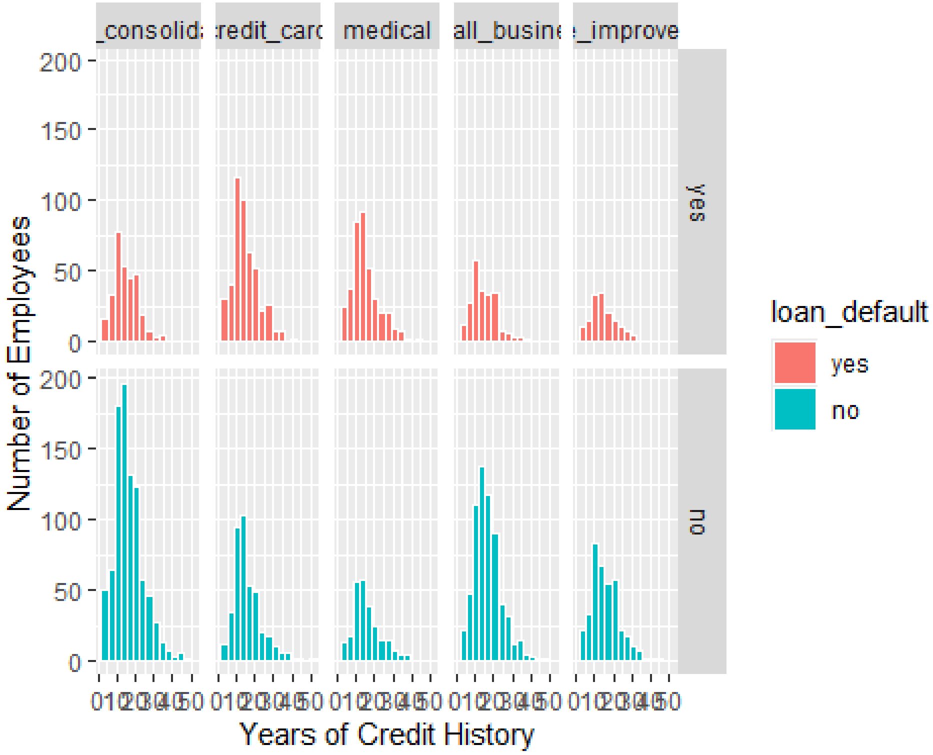


Question 5

Question: Is there a relationship between the customers that default and those that don't based on their total credit lines and the loan purpose?

Answer: For this histogram I used the variables total credit lines, loan purpose, and loan default. I placed the years of credit history on the x axis and the number of customers on the y axis. Each rectangle separates the loan purpose. Orange represents the customers that did default on their loan and the teal represents customers that did not default on their loan. From the histogram we can see that most customers that did default on their loan come from the credit card loan purpose group. Most customers that did default on their loan from the credit card group had between 10 to 15 years of credit card history. The medical group has the second highest number of customers that default on their loans followed by debt consolidation, small business, and home imporvement. One important thing to note is that in all of these groups, most of the customers that default on their loans has between 10 to 15 years of credit history. This makes sense as most of the customers that do not default also have 10 to 15 years of credit history.

```
ggplot(loans_df, aes(x = years_credit_history, fill = loan_default)) +
  geom_histogram(color = "white", bins = 15) +
  facet_grid(loan_default ~ loan_purpose) +
  labs(Title = 'Years of Credit History by loan purpose', x = 'Years of
Credit History', y = 'Number of Employees')
```

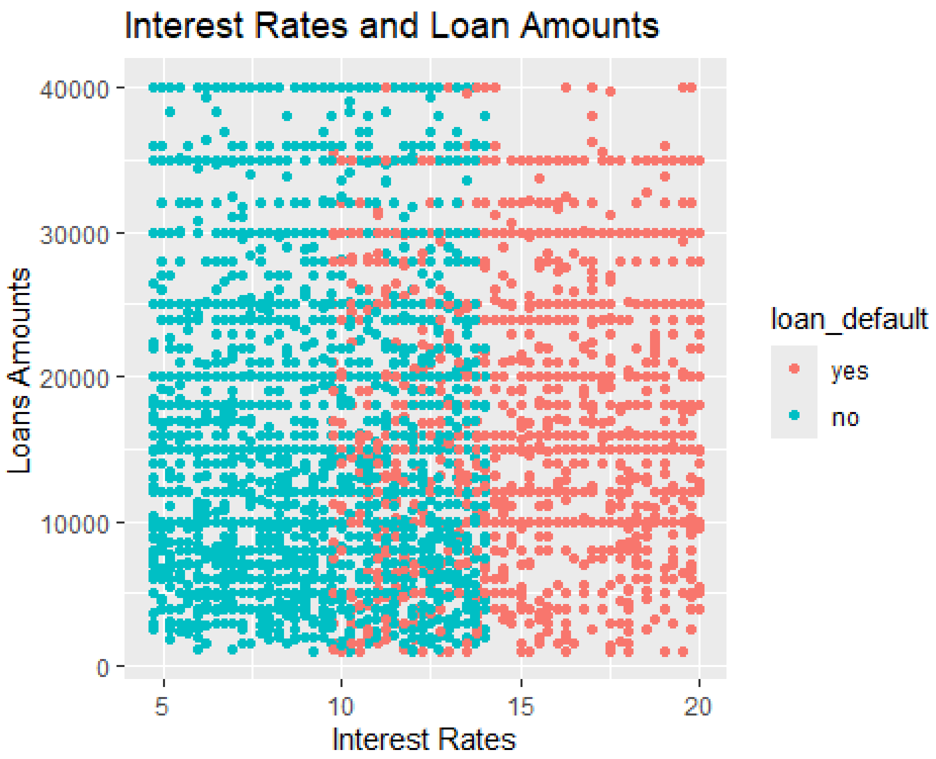


## Question 6

Question: Is there a relationship between how many customers default on their loans based on interest rates and loan amounts?

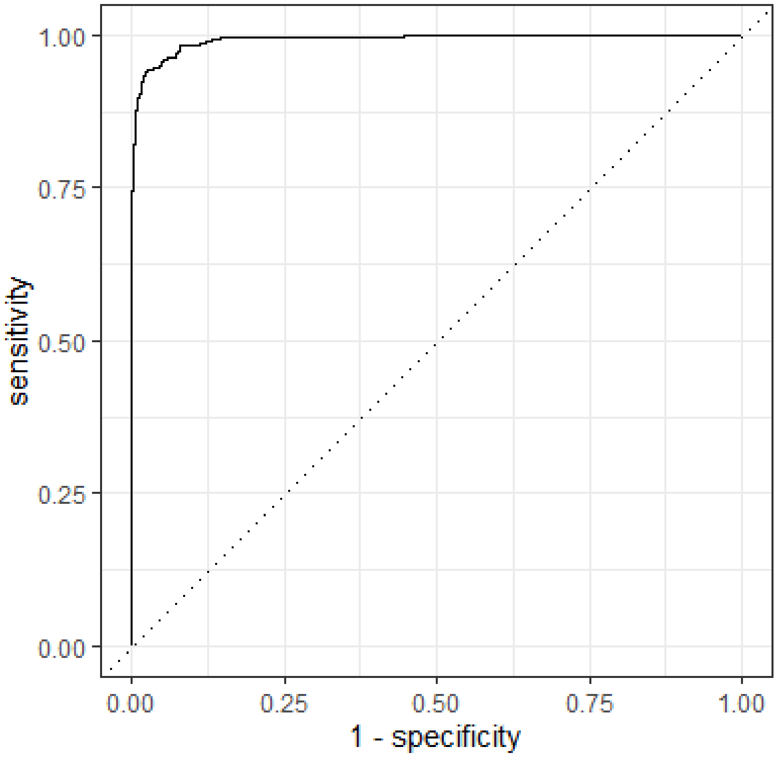
Answer: For this scatter plot I placed interest rates on the x axis and loan amounts on the y axis. The orange points represent the customers that did default and the teal points represent those that did not. From the scatter plot we can see that most customers that do not default on their loan have interest rates between 4 to 10%. Most customers that don't default and have a interest rate between 4 to 10% have loan amounts of 20,000 or less. We can see that starting at 10% interest rates is where customers start to default on their loans, but there is still a good mix of customers that do not default. Starting at about 14% interest rate is where customers start to default on their loans no matter the loan amount.

```r
ggplot(loans_df, aes(x = interest_rate, y = loan_amount, color =
loan_default)) +
  geom_point() +
  labs(title = 'Interest Rates and Loan Amounts', x = 'Interest Rates', y =
'Loans Amounts')
```



Model 1

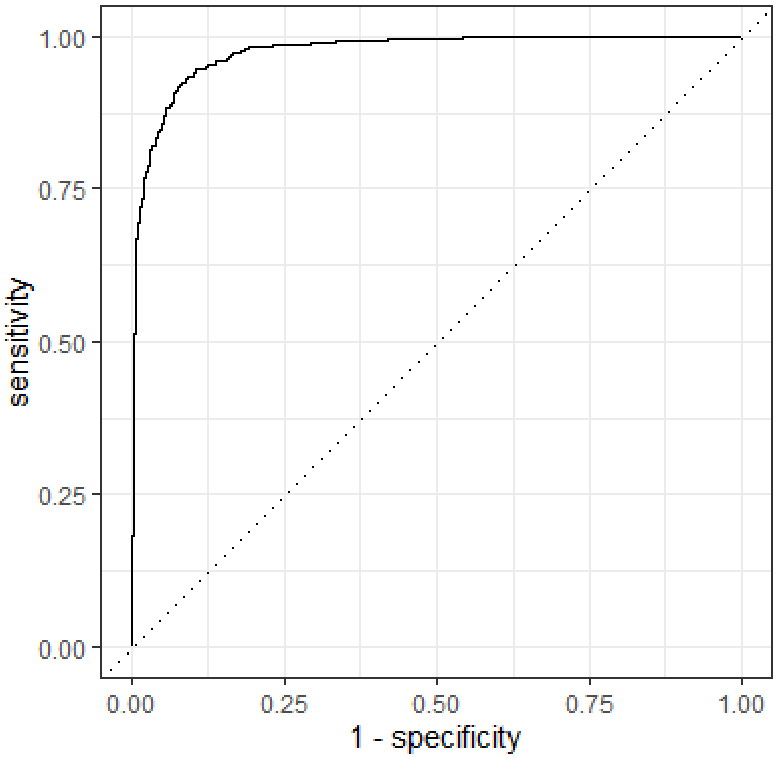
```r
#logistic regression model
#data splitting
set.seed(571)


loans_split <- initial_split(loans_df, prop = .75,
                             strata = loan_default)
loans_training <- loans_split %>% training()

loans_test <- loans_split %>% testing()

set.seed(571)

loans_folds <- vfold_cv(loans_training, v = 5)

#Loans Recipe
loans_recipe <- recipe(loan_default ~ ., data = loans_training) %>%
                       step_YeoJohnson(all_numeric(), -all_outcomes()) %>%
                       step_normalize(all_numeric(), -all_outcomes()) %>%
                       step_dummy(all_nominal(), -all_outcomes())
loans_recipe %>%
  prep() %>%
  bake(new_data = loans_training)

## # A tibble: 3,082 × 20
##    loan_amount installment interest_rate annual_income current_job_years
##          <dbl>       <dbl>         <dbl>         <dbl>             <dbl>
##  1      1.16        1.41         -0.564          1.89              1.10
##  2     -1.62       -1.70         -0.356          0.162            -0.386
##  3      0.956       0.597         0.224          1.32             -0.112
##  4     -1.23       -1.22         -0.854          0.106            -0.386
##  5      1.87        1.44         -0.0270         0.106            -0.678
##  6      0.0114      0.0609        0.0366        -0.847             1.10
##  7      1.63        0.990        -1.08           2.23              1.10
##  8     -1.14       -1.10          0.0366        -1.37              1.10
##  9     -0.556      -0.922         0.523         -1.72             -1.34
## 10     -0.556      -0.543        -1.08          -0.193             1.10
## # ℹ 3,072 more rows
## # ℹ 15 more variables: debt_to_income <dbl>, total_credit_lines <dbl>,
## #   years_credit_history <dbl>, loan_default <fct>,
## #   loan_purpose_credit_card <dbl>, loan_purpose_medical <dbl>,
## #   loan_purpose_small_business <dbl>, loan_purpose_home_improvement
<dbl>,
## #   application_type_joint <dbl>, term_five_year <dbl>,
## #   homeownership_rent <dbl>, homeownership_own <dbl>, …

#specify model
logistic_model <- logistic_reg() %>%
                  set_engine('glm') %>%
                  set_mode('classification')
logistic_model

## Logistic Regression Model Specification (classification)
##
## Computational engine: glm

#creating workflow
loans_wf <- workflow() %>%
            add_model(logistic_model) %>%
            add_recipe(loans_recipe)




#fit the model
loans_logistic_fit <- loans_wf %>%
last_fit(split = loans_split)

#collect_predictions
loans_results <- loans_logistic_fit %>%
                  collect_predictions()

#ROC rucve
roc_curve(loans_results, truth = loan_default, .pred_yes) %>%
  autoplot()
```



```
#Area under ROC curve
roc_auc(loans_results, truth = loan_default, .pred_yes)

## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.993

#Confusion matrix
conf_mat(loans_results, truth = loan_default, estimate = .pred_class)

##           Truth
## Prediction yes  no
##        yes 360  16
##        no   23 629
```

Model 2

```
#QDA MODEL
#Specify model
qda_model <- discrim_regularized(frac_common_cov = 0) %>%
             set_engine('klaR') %>%
             set_mode('classification')

#workflow
qda_wf <- workflow() %>%
          add_model(qda_model) %>%
          add_recipe(loans_recipe)

#Train and evaluate with last_fit()
last_fit_qda <- qda_wf %>%
                last_fit(split = loans_split)

#Collect predictions
qda_results <- last_fit_qda %>%
                    collect_predictions()

# ROC Curve
  roc_curve(qda_results, truth = loan_default, .pred_yes) %>%
  autoplot()
```



```
#Area Under the Curve
roc_auc(qda_results, truth = loan_default, .pred_yes)

## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.975
```

```
#Confusion Matrix
conf_mat(qda_results, truth = loan_default, estimate = .pred_class)

##           Truth
## Prediction yes  no
##        yes 347  47
##        no   36 598
```

Model 3

```
#random forest model
#Specify model
rf_model <- rand_forest(mtry = tune(),
                        trees = tune(),
                        min_n = tune()) %>%
            set_engine('ranger', importance = "impurity") %>%
            set_mode('classification')
rf_model

## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = tune()
##   trees = tune()
##   min_n = tune()
##
## Engine-Specific Arguments:
##   importance = impurity
##
## Computational engine: ranger

#workflow
rf_workflow <- workflow() %>%
               add_model(rf_model) %>%
               add_recipe(loans_recipe)
rf_workflow

## ══ Workflow
═════════════════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: rand_forest()
##
## ── Preprocessor
─────────────────────────────────────────────────────────────────
## 3 Recipe Steps
##
## • step_YeoJohnson()
## • step_normalize()
## • step_dummy()
##
## ── Model
───────────────────────────────────────────────────────────────────────
```

```
## Random Forest Model Specification (classification)
## 
## Main Arguments:
##   mtry = tune()
##   trees = tune()
##   min_n = tune()
## 
## Engine-Specific Arguments:
##   importance = impurity
## 
## Computational engine: ranger

#Create a grid of hyperparameter values to test
set.seed(572)

rf_grid <- grid_random(mtry() %>% range_set(c(4, 12)),
                       trees(),
                       min_n(),
                       size = 10)

# Tune random forest workflow
set.seed(573)

rf_tuning <- rf_workflow %>%
             tune_grid(resamples = loans_folds,
                       grid = rf_grid)
rf_tuning %>%
  show_best()

## Warning in show_best(.): No value of `metric` was given; "roc_auc" will be
## used.

## # A tibble: 5 × 9
##    mtry trees min_n .metric .estimator  mean     n std_err .config
##   <int> <int> <int> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1    12  1704    14 roc_auc binary     0.975     5 0.00162 
Preprocessor1_Model05
## 2    11  1630    22 roc_auc binary     0.973     5 0.00158 
Preprocessor1_Model08
## 3     9  1434    14 roc_auc binary     0.973     5 0.00170 
Preprocessor1_Model02
## 4     7   803    11 roc_auc binary     0.973     5 0.00177 
Preprocessor1_Model01
## 5     7  1472     2 roc_auc binary     0.973     5 0.00180 
Preprocessor1_Model10

# Select best model based on roc_auc
best_rf <- rf_tuning %>%
           select_best(metric = 'roc_auc')
```

```r
best_rf
```

```
## # A tibble: 1 × 4
##    mtry trees min_n .config
##   <int> <int> <int> <chr>
## 1    12  1704    14 Preprocessor1_Model05
```

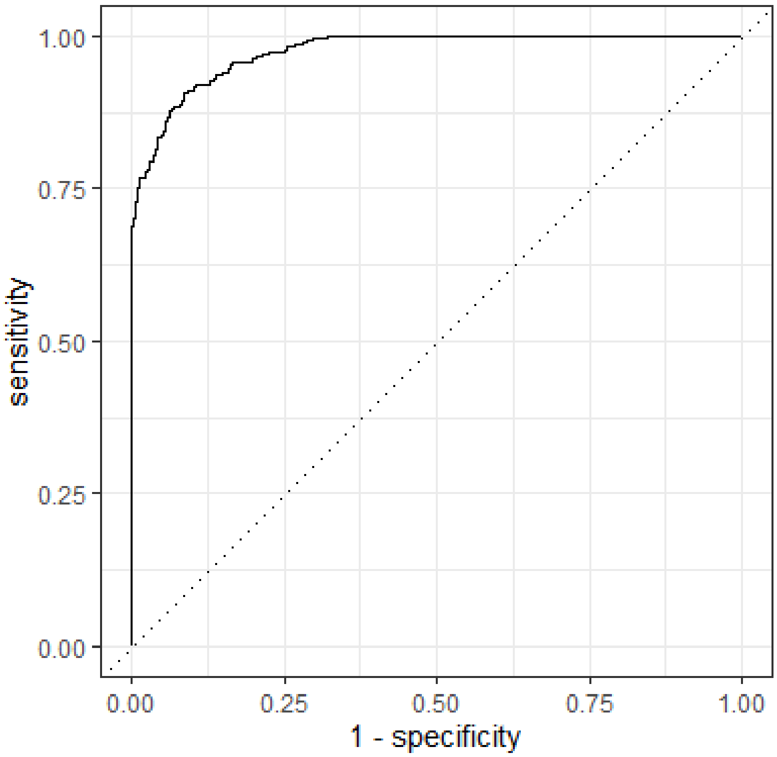
```r
#finalize worflow
final_rf_workflow <- rf_workflow %>%
                     finalize_workflow(best_rf)
final_rf_workflow
```

```
## ══ Workflow
════════════════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: rand_forest()
##
## ── Preprocessor
────────────────────────────────────────────────────────────────
## 3 Recipe Steps
##
## • step_YeoJohnson()
## • step_normalize()
## • step_dummy()
##
## ── Model
───────────────────────────────────────────────────────────────────────
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 12
##   trees = 1704
##   min_n = 14
##
## Engine-Specific Arguments:
##   importance = impurity
##
## Computational engine: ranger
```

```r
#fit the model
rf_last_fit <- final_rf_workflow %>%
               last_fit(loans_split)

#collect predictions
rf_predictions <- rf_last_fit %>%
  collect_predictions()
rf_predictions
```

```
## # A tibble: 1,028 × 7
##    .pred_class .pred_yes .pred_no id                .row loan_default
```

```
.config
##    <fct>           <dbl>    <dbl> <chr>            <int> <fct>
<chr>
##   1 yes         0.999    0.000629 train/test split     1 yes
Preproces…
##   2 yes         0.678    0.322    train/test split     2 yes
Preproces…
##   3 yes         0.917    0.0826   train/test split    12 yes
Preproces…
##   4 yes         1        0        train/test split    13 yes
Preproces…
##   5 no          0.000419 1.00     train/test split    18 no
Preproces…
##   6 yes         1        0        train/test split    24 yes
Preproces…
##   7 no          0.390    0.610    train/test split    28 no
Preproces…
##   8 yes         0.999    0.00119  train/test split    32 yes
Preproces…
##   9 no          0.335    0.665    train/test split    39 no
Preproces…
##  10 no          0.276    0.724    train/test split    44 yes
Preproces…
## # ℹ 1,018 more rows

#ROC curve
rf_predictions %>%
                roc_curve(truth  = loan_default, .pred_yes) %>%
                autoplot()
```



```
#Area under ROC curve
roc_auc(rf_predictions, truth = loan_default, .pred_yes)

## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.975

#Confusion Matrix
conf_mat(rf_predictions, truth = loan_default, estimate = .pred_class)

##           Truth
## Prediction yes  no
##        yes 314  27
##        no   69 618
```

— End of the Project —