MXN600: Credit Risk 2023

Joanna Salerno, Pavan Asopa, Sevin Nejadi, Fernanda Martins Giuriati

2023-10-26

Introduction

The credit risk models our lending start up company uses are of the utmost importance to the functioning and ultimate success of the company. As we were recently acquired by a regional Australian bank, the success of our company also impacts the success of the bank. Recently, some of the bank's senior financial analysts have raised concerns about the credit risk models we have been using. They reviewed our models' performance benchmarks and feel as though our models are not suitable for use in a setting in which they are subject to strict regulatory requirements.

Thus, we have been tasked by the bank's management to rebuild our credit risk model from the ground up. As per management, our main objective is to use information known at the time of a loan application to build a model that predicts loan default. We will follow a standard statistical analysis process, which will be guided by the following questions:

1. How does this new model perform compared to the one used previously? How can it be expected to perform on new loan applications?
2. What are the important variables in this model and how do they compare to variables that are traditionally important for predicting credit risk in the banking sector?

Furthermore, management has consulted with an expert statistician, who has suggested we also account for variation in trends that may exist either between different jurisdictions or over time. The following questions will guide this second part of our analysis:

3. Can accounting for this variation (e.g., state/zip-code and time) improve performance benchmarks?
4. Are there any surprising differences in variables that are important for predicting credit risk?
5. Does credit risk change over time or between states? This is not something the bank has previously investigated and results may inform modified loan policies in the future.

This report will document our entire analysis process, beginning with data exploration and cleaning, to model building and interpretations of our results.

Setup

We will first load in the required libraries for our data exploration and analysis process.

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
```

```
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'MASS'
##
##
## The following object is masked from 'package:dplyr':
##
##     select
##
##
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
##
## Attaching package: 'lmerTest'
##
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
##
## The following object is masked from 'package:stats':
##
##     step
##
##
## This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
##
## Type 'citation("pROC")' for a citation.
##
##
## Attaching package: 'pROC'
##
##
## The following objects are masked from 'package:stats':
##
```

```
##     cov, smooth, var
```

Next, we will load in our datasets. We have a total of 4:

(1) A training dataset that we will use to build and train our model
(2) A test dataset that we will use to test the fit of our model(s)
(3) A validation dataset that we will use to assess the performance of our model(s)
(4) An extended dataset that includes the necessary variables for us to account for variation such as location and time

```
train_data <- read.csv("benchmark_training_loan_data.csv")
test_data <- read.csv("benchmark_testing_loan_data.csv")
val_data <- read.csv("benchmark_validation_loan_data.csv")
extended <- read.csv("extendend_version_loan_data.csv")
```

Exploratory Analysis

We will begin by exploring the available data to understand how each variable is distributed and to identify any potential data quality issues. We will also investigate the relationships between the different variables to see whether any variables are highly correlated with one another.

Note: For this exploration portion of our analysis, we will be using the training dataset.

We will first explore the training dataset to understand its structure and the variables it is comprised of.

```
head(train_data)
```

```
##   X loan_amnt      term int_rate emp_length home_ownership annual_inc
## 1 1      2500 36 months    13.98    4 years           RENT      20004
## 2 2      5000 36 months    15.95    4 years           RENT      59000
## 3 3      7000 36 months     9.91  10+ years       MORTGAGE      53796
## 4 4      2000 36 months     5.42  10+ years           RENT      30000
## 5 5      8000 36 months     6.03        n/a       MORTGAGE      77736
## 6 6      6250 36 months    17.27    4 years       MORTGAGE      28000
##   verification_status            purpose   dti delinq_2yrs inq_last_6mths
## 1        Not Verified              other 19.86           0              5
## 2        Not Verified debt_consolidation 19.57           0              1
## 3        Not Verified              other 10.80           3              3
## 4        Not Verified debt_consolidation  3.60           0              0
## 5            Verified              other  6.07           0              0
## 6            Verified              other 13.76           0              0
##   open_acc pub_rec revol_bal revol_util total_acc repay_fail credit_age_yrs
## 1        7       0       981     0.2130        10          0       4.931319
## 2        7       0     18773     0.9990        15          1      16.222527
## 3        7       0      3269     0.4720        20          0      13.549451
## 4        7       0         0     0.0000        15          0      36.791209
## 5       12       0      4182     0.1360        49          0      15.302198
## 6        2       1         0     0.0846        15          1      12.126374
```

At first glance of the first 6 rows of this dataset, we notice there is a value of n/a in the employment length column. There are also a few zero values in a few columns. This will prompt us to further explore the data for any true missing values.

```r
dim(train_data)
```

```
## [1] 23052    19
```

The training dataset contains a total of 23,052 observations of 19 variables.

```r
str(train_data)
```

```
## 'data.frame':    23052 obs. of  19 variables:
##  $ X                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ loan_amnt          : int  2500 5000 7000 2000 8000 6250 8000 16000 7000 13000 ...
##  $ term               : chr  "36 months" "36 months" "36 months" "36 months" ...
##  $ int_rate           : num  13.98 15.95 9.91 5.42 6.03 ...
##  $ emp_length         : chr  "4 years" "4 years" "10+ years" "10+ years" ...
##  $ home_ownership     : chr  "RENT" "RENT" "MORTGAGE" "RENT" ...
##  $ annual_inc         : num  20004 59000 53796 30000 77736 ...
##  $ verification_status: chr  "Not Verified" "Not Verified" "Not Verified" "Not Verified" ...
##  $ purpose            : chr  "other" "debt_consolidation" "other" "debt_consolidation" ...
##  $ dti                : num  19.86 19.57 10.8 3.6 6.07 ...
##  $ delinq_2yrs        : int  0 0 3 0 0 0 0 0 2 0 ...
##  $ inq_last_6mths     : int  5 1 3 0 0 0 1 0 1 1 ...
##  $ open_acc           : int  7 7 7 7 12 2 8 5 3 14 ...
##  $ pub_rec            : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ revol_bal          : int  981 18773 3269 0 4182 0 9287 11006 6082 38433 ...
##  $ revol_util         : num  0.213 0.999 0.472 0 0.136 0.0846 0.619 0.651 0.965 0.565 ...
##  $ total_acc          : int  10 15 20 15 49 15 37 36 11 31 ...
##  $ repay_fail         : int  0 1 0 0 0 1 0 0 0 0 ...
##  $ credit_age_yrs     : num  4.93 16.22 13.55 36.79 15.3 ...
```

Upon further investigation of the structure of the training data, we can see that 14 of the 19 variables are currently of numeric type, and the remaining 5 variables are characters.

We will now remove variable X, which is the number of each record, from all datasets, as it is not necessary for this analysis.

```r
train_data <- train_data[, -1]
val_data <- val_data[, -1]
test_data <- test_data[, -1]
extended <- extended[, -1]
```

We will now further investigate each variable to determine whether there is any missing data or outliers.

```r
summary(train_data)
```

```
##    loan_amnt         term              int_rate      emp_length
##  Min.   :  500   Length:23052       Min.   : 5.42   Length:23052
##  1st Qu.: 5400   Class :character   1st Qu.: 9.63   Class :character
##  Median : 9862   Mode  :character   Median :11.99   Mode  :character
##  Mean   :11128                      Mean   :12.18
##  3rd Qu.:15000                      3rd Qu.:14.72
##  Max.   :35000                      Max.   :24.11
```

```
##  home_ownership       annual_inc      verification_status   purpose
##  Length:23052       Min.   :   2000   Length:23052        Length:23052
##  Class :character   1st Qu.:  40032   Class :character    Class :character
##  Mode  :character   Median :  58000   Mode  :character    Mode  :character
##                     Mean   :  68435
##                     3rd Qu.:  82000
##                     Max.   :2039784
##       dti          delinq_2yrs     inq_last_6mths      open_acc
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   : 1.000
##  1st Qu.: 8.248   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 6.000
##  Median :13.550   Median :0.0000   Median : 1.000   Median : 9.000
##  Mean   :13.426   Mean   :0.1502   Mean   : 1.078   Mean   : 9.354
##  3rd Qu.:18.740   3rd Qu.:0.0000   3rd Qu.: 2.000   3rd Qu.:12.000
##  Max.   :29.950   Max.   :9.0000   Max.   :33.000   Max.   :47.000
##     pub_rec        revol_bal        revol_util       total_acc
##  Min.   :0.000   Min.   :     0   Min.   :0.0000   Min.   : 1.00
##  1st Qu.:0.000   1st Qu.:  3686   1st Qu.:0.2600   1st Qu.:13.00
##  Median :0.000   Median :  8918   Median :0.4990   Median :20.00
##  Mean   :0.057   Mean   : 14383   Mean   :0.4932   Mean   :22.12
##  3rd Qu.:0.000   3rd Qu.: 17367   3rd Qu.:0.7302   3rd Qu.:29.00
##  Max.   :4.000   Max.   :952013   Max.   :1.1900   Max.   :90.00
##    repay_fail    credit_age_yrs
##  Min.   :0.000   Min.   : 0.5055
##  1st Qu.:0.000   1st Qu.: 9.0302
##  Median :0.000   Median :12.5440
##  Mean   :0.152   Mean   :13.7856
##  3rd Qu.:0.000   3rd Qu.:17.1429
##  Max.   :1.000   Max.   :60.6209
```

Included above is the numeric distributions of each of the included variables. Based on the above summary, it appears as though there may exist one or multiple outliers in a few variables: annual income, inquiries in the last 6 months, open accounts, revolving balances, and total accounts. However, we will need to further investigate the distribution of all variables to better visualize this to determine whether these actually appear to be outliers.

Before producing some exploratory plots, we will briefly explore the data to see whether there are missing values for us to handle.

```
colSums(is.na(train_data))
```

```
##           loan_amnt                term            int_rate          emp_length
##                   0                   0                   0                   0
##      home_ownership          annual_inc   verification_status             purpose
##                   0                   0                   0                   0
##                 dti          delinq_2yrs       inq_last_6mths            open_acc
##                   0                   0                   0                   0
##             pub_rec           revol_bal          revol_util           total_acc
##                   0                   0                   0                   0
##          repay_fail       credit_age_yrs
##                   0                   0
```

Based on the above output, it appears as though this dataset does not contain any missing data in the form of NA values. Now, we will explore the n/a values present in the columns which include string data.

```
sum(train_data == 'n/a')
```

```
## [1] 591
```

```
na_rows <- train_data %>% filter_all(any_vars(. %in% c('n/a')))
head(na_rows)
```

```
##   loan_amnt      term int_rate emp_length home_ownership annual_inc
## 1      8000 36 months     6.03        n/a       MORTGAGE      77736
## 2      1450 36 months     7.51        n/a           RENT      10000
## 3      1800 36 months     5.42        n/a            OWN      29184
## 4      4000 36 months    17.19        n/a            OWN      37200
## 5      2250 36 months     5.42        n/a           RENT      52500
## 6      5500 36 months     8.49        n/a       MORTGAGE      36780
##   verification_status            purpose   dti delinq_2yrs inq_last_6mths
## 1            Verified              other  6.07           0              0
## 2     Source Verified              other 22.20           0              0
## 3        Not Verified     small_business 23.68           0              2
## 4        Not Verified   home_improvement  9.35           0              0
## 5        Not Verified debt_consolidation 16.43           0              0
## 6        Not Verified                car 16.54           0              2
##   open_acc pub_rec revol_bal revol_util total_acc repay_fail credit_age_yrs
## 1       12       0      4182      0.136        49          0      15.302198
## 2        9       0       709      0.032        10          0       6.271978
## 3        7       0     30037      0.235        24          0      36.711538
## 4        6       0      4598      0.575        14          1      12.881868
## 5        9       0      4425      0.360        25          0      18.978022
## 6        8       0      8926      0.498        27          1      31.857143
```

Based on the above output, there are a total of 591 n/a (string) in this dataset. These all appear to be from the employment length column. We will move forward and assume that this is not truly missing data, but rather, means that the applicant is not currently employed. We will further examine the distribution of each variable by creating exploratory plots, and this will further clarify the meaning of some of the values contained within each column of the dataset.

Exploratory Plots

Data Preparation

Before we can create exploratory plots, there are a few variables we will need to convert to factors. This will help us in creating our visualizations as well as conducting our analyses, so we can consider the selected variables as factors with different levels rather than simply strings.

Training dataset:

```
train_data$term <- as.factor(train_data$term)
train_data$emp_length <- as.factor(train_data$emp_length)
train_data$home_ownership <- as.factor(train_data$home_ownership)
train_data$verification_status <- as.factor(train_data$verification_status)
train_data$purpose <- as.factor(train_data$purpose)
train_data$repay_fail <- as.factor(train_data$repay_fail)
```

Test dataset:

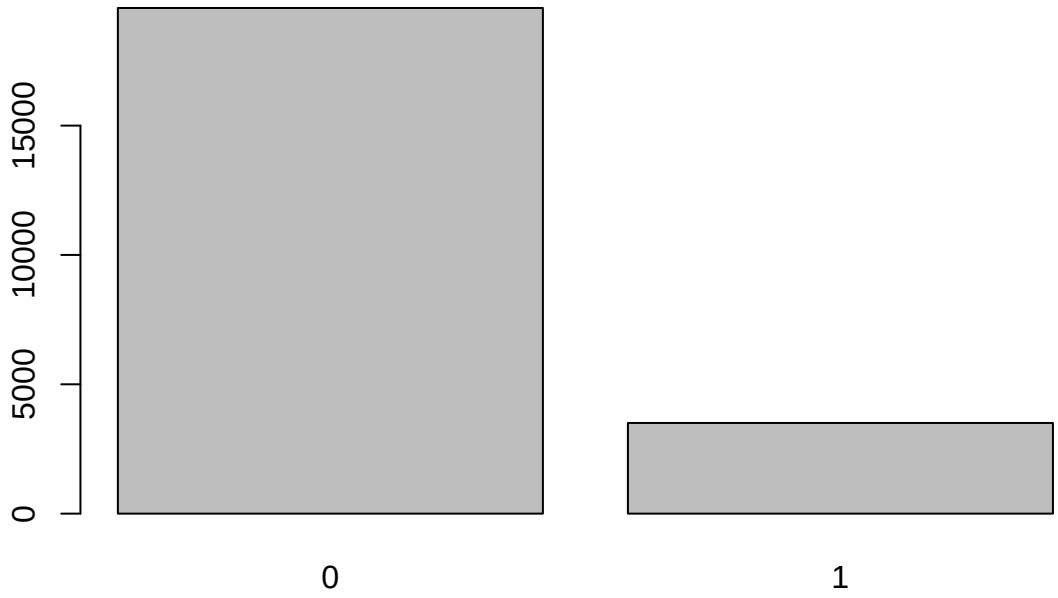
```
test_data$term <- as.factor(test_data$term)
test_data$emp_length <- as.factor(test_data$emp_length)
test_data$home_ownership <- as.factor(test_data$home_ownership)
test_data$verification_status <- as.factor(test_data$verification_status)
test_data$purpose <- as.factor(test_data$purpose)
test_data$repay_fail <- as.factor(test_data$repay_fail)
```

Validation dataset:

```
val_data$term <- as.factor(val_data$term)
val_data$emp_length <- as.factor(val_data$emp_length)
val_data$home_ownership <- as.factor(val_data$home_ownership)
val_data$verification_status <- as.factor(val_data$verification_status)
val_data$purpose <- as.factor(val_data$purpose)
val_data$repay_fail <- as.factor(val_data$repay_fail)
```

Now, we'd like to briefly check whether there is an imbalance in the cases of repay_fail.

```
plot(train_data$repay_fail)
```

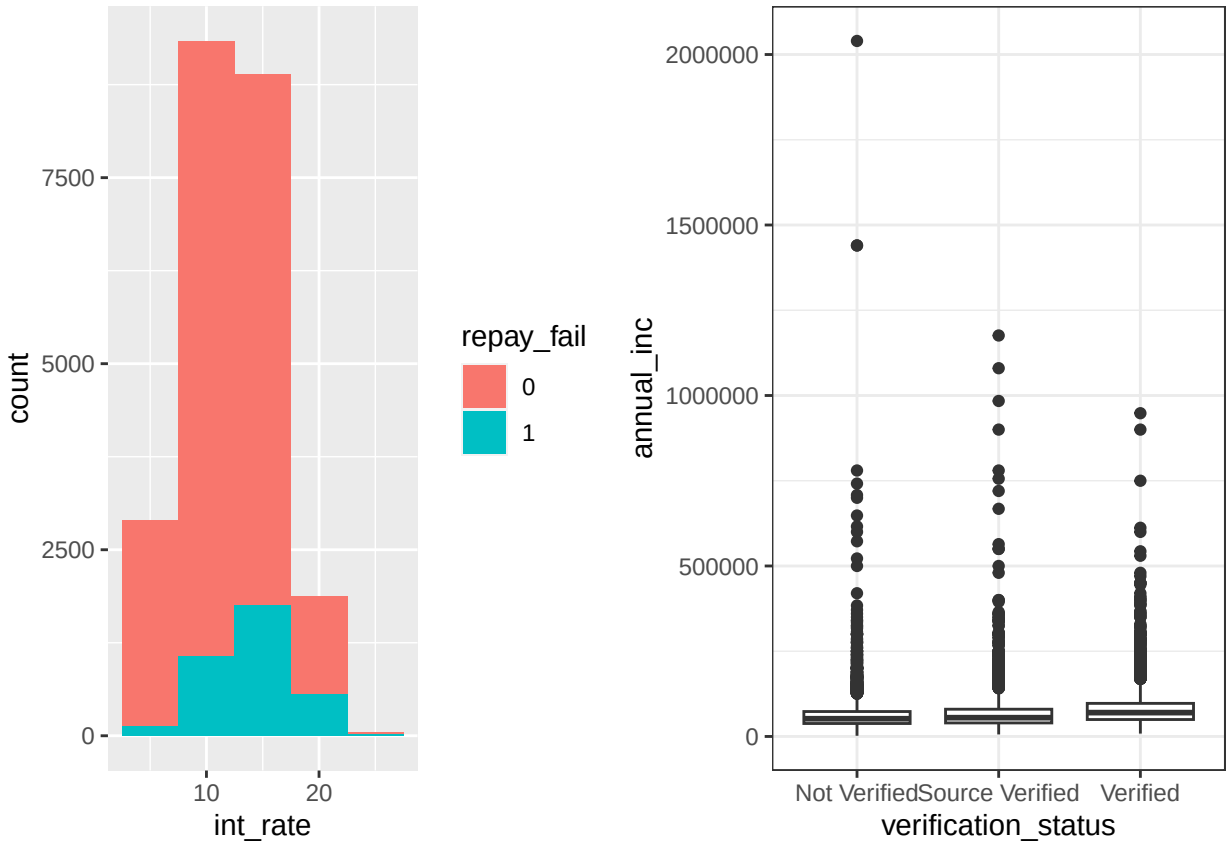


Despite the significant class imbalance (repay_fail==0 v. repay_fail==1) in the dataset, logistic regression still maintains its probabilistic nature. However, it's important to note that imbalanced data can affect the model's predictive performance, potentially leading to a bias towards the majority class.

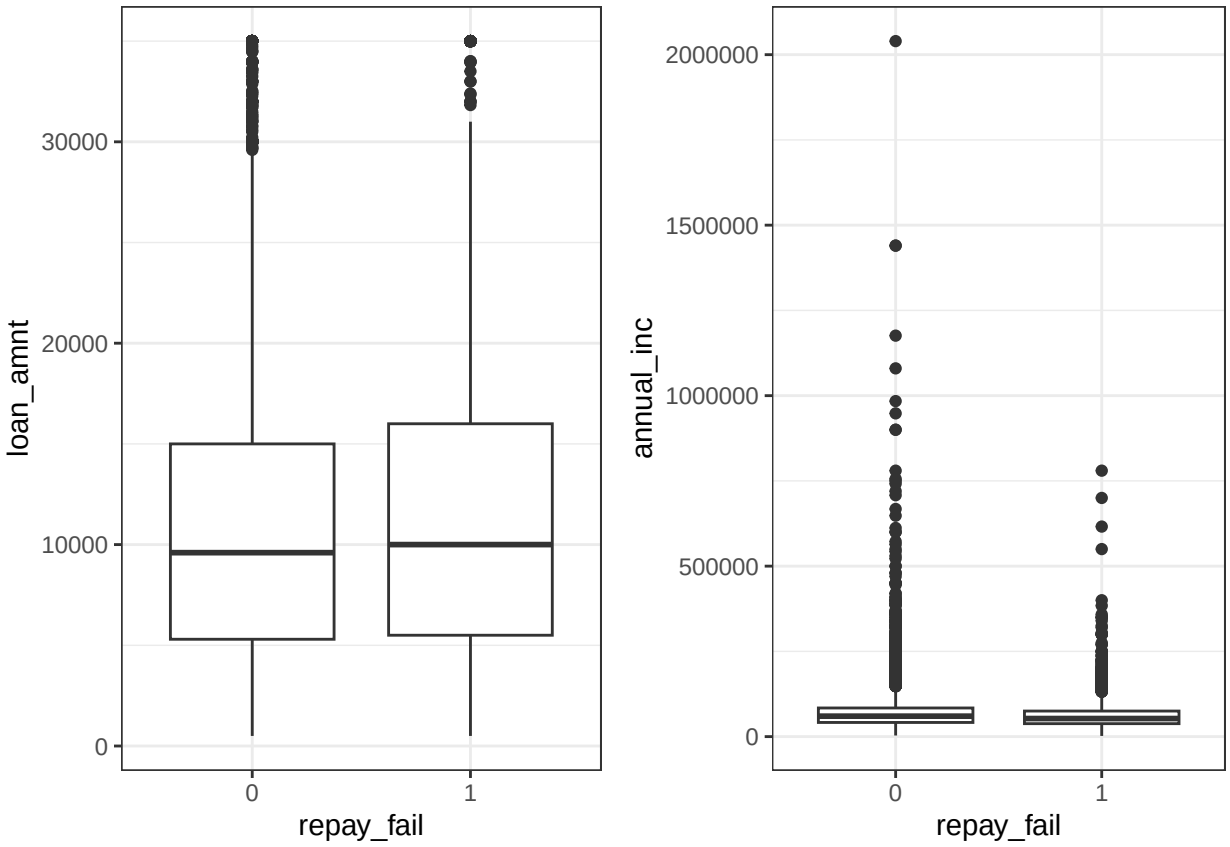
Variable Distributions

Now, we will create some plots to visualize the distributions of some of the variables in this dataset.

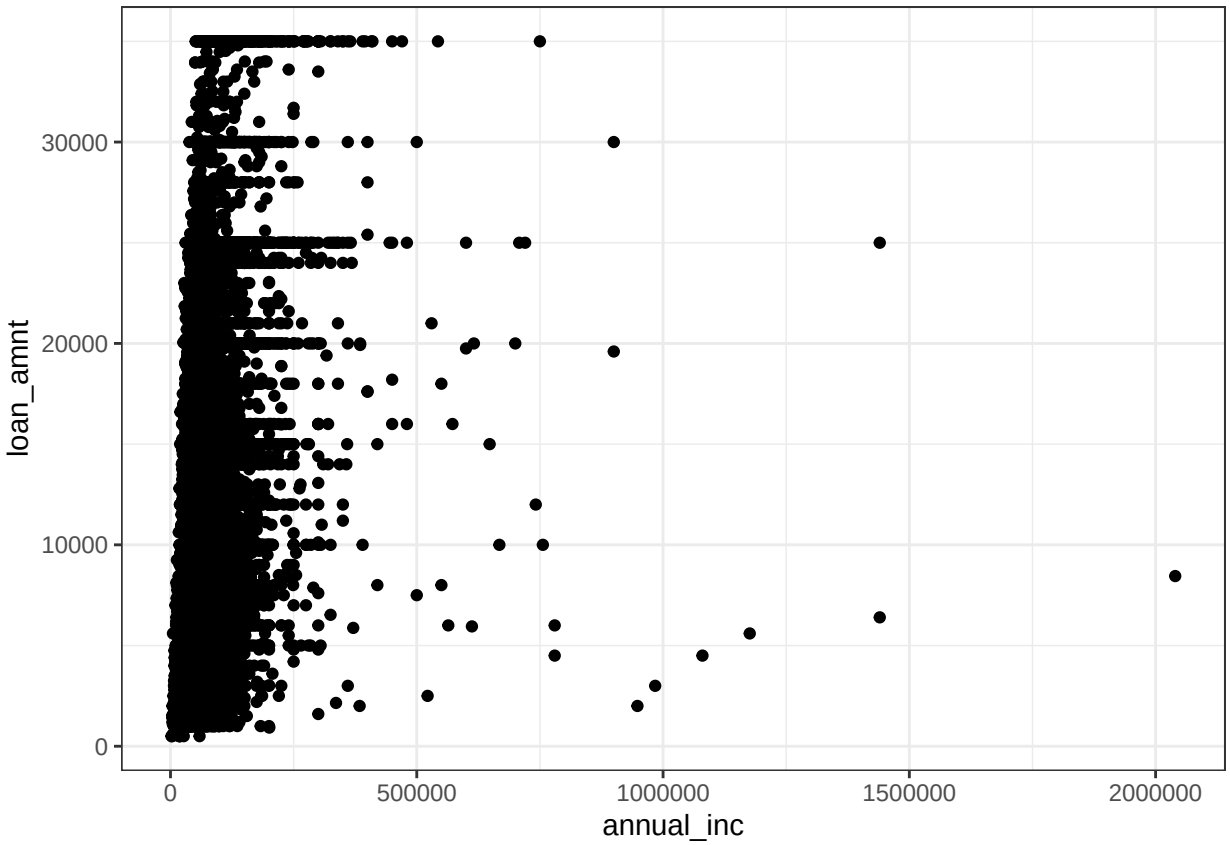
```
p1 <- ggplot(data = train_data, aes(int_rate, fill = repay_fail)) +
  geom_histogram(binwidth=5)

p2 <- ggplot(data = train_data, mapping = aes(x = verification_status, y = annual_inc)) +
  geom_boxplot() +
  theme_bw()

p3 <- ggplot(data = train_data, mapping = aes(x = repay_fail, y = loan_amnt)) +
  geom_boxplot() +
  theme_bw()

p4 <- ggplot(data = train_data, mapping = aes(x = repay_fail, y = annual_inc)) +
  geom_boxplot() +
  theme_bw()

p5 <- ggplot(data = train_data, mapping = aes(x = annual_inc, y = loan_amnt)) +
  geom_point() +
  theme_bw()

ggarrange(p1, p2)
```



```
ggarrange(p3, p4)
```




```
ggarrange(p5)
```



Explanations:

Plot1 is a histogram that displays the distribution of interest rates in the dataset. The bars in the histogram are filled based on whether the loan was repaid successfully or not, where fill colours represent the two categories 0 for no failure and 1 for failure of loan repayment. Based on the plot, a trend can be seen that as the interest rate increases so does the chances of loan repayment failure.

Plot2 is a boxplot that shows the distribution of annual incomes based on different verification statuses. Each box represent a category of verification status. Based on the plot, we can see that we have similar sized boxes for all three categories which suggests that the median income for applicants in these categories is relatively consistent. However, we can see the presence of outliers in all three verification_status categories, indicating that there are individuals with very high income within each group. The largest outliers appear in the "not verified" group, which may be concerning for the bank.

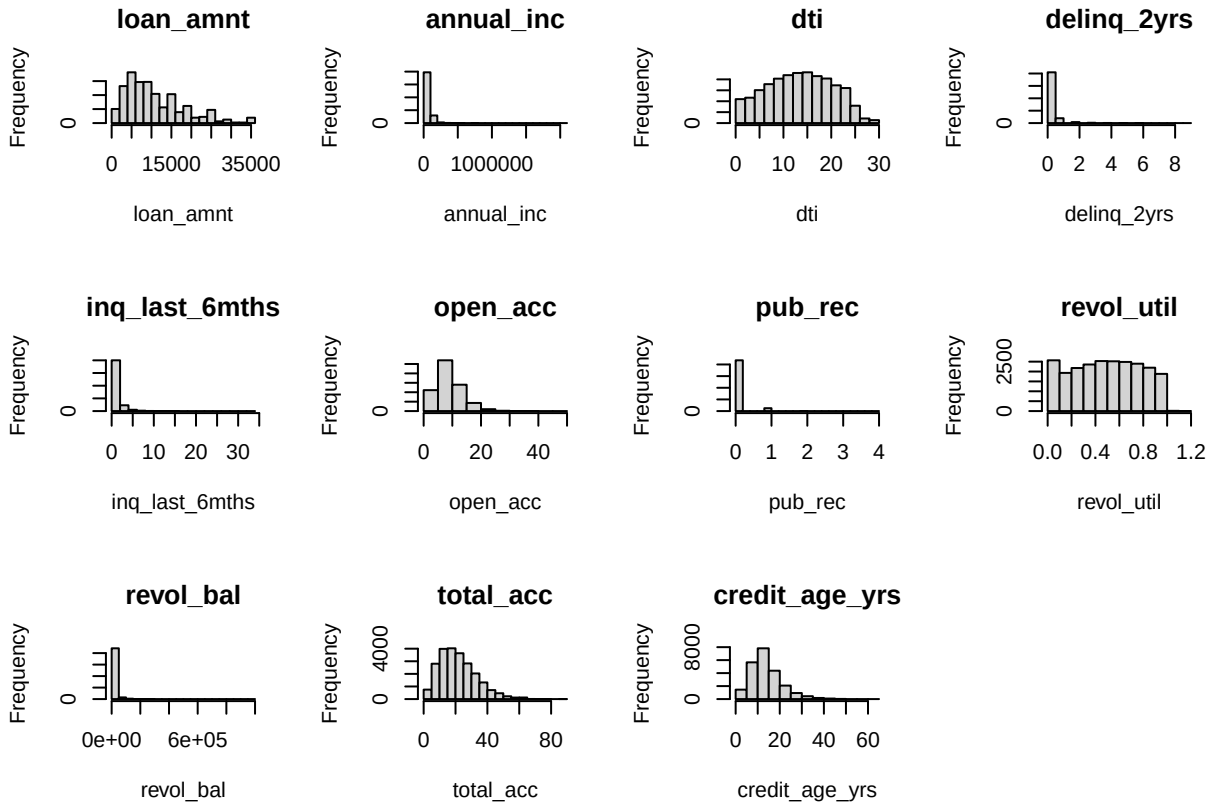
Plot3 is a boxplot that compares the loan amounts for both successful and unsuccessful loan repayments. Based on the plot, it can be seen that the median loan amount for both categories of loan repayment status are more or less consistent. The presence of outliers in both categories indicates that there are individuals with very high loan amounts within each group.

Plot4 is a boxplot that compares annual income between successful and unsuccessful loan repayments. Based on the plot, it can be seen that most individuals that successfully paid off their loans are more or less consistent with the ones who failed with an exception of outliers in both categories. The outliers in terms of larger annual incomes of $750,000 or more consistently paid their loans back in full.

Plot5 is scatter plot that displays the relationship between annual income and loan amount. No apparent trends appear in this plot, indicating that individuals with a range of annual incomes also request to borrow loans for a range of loan amounts.

Now, we will visualize the distribution of each of the numeric variables in the dataset, using histograms:

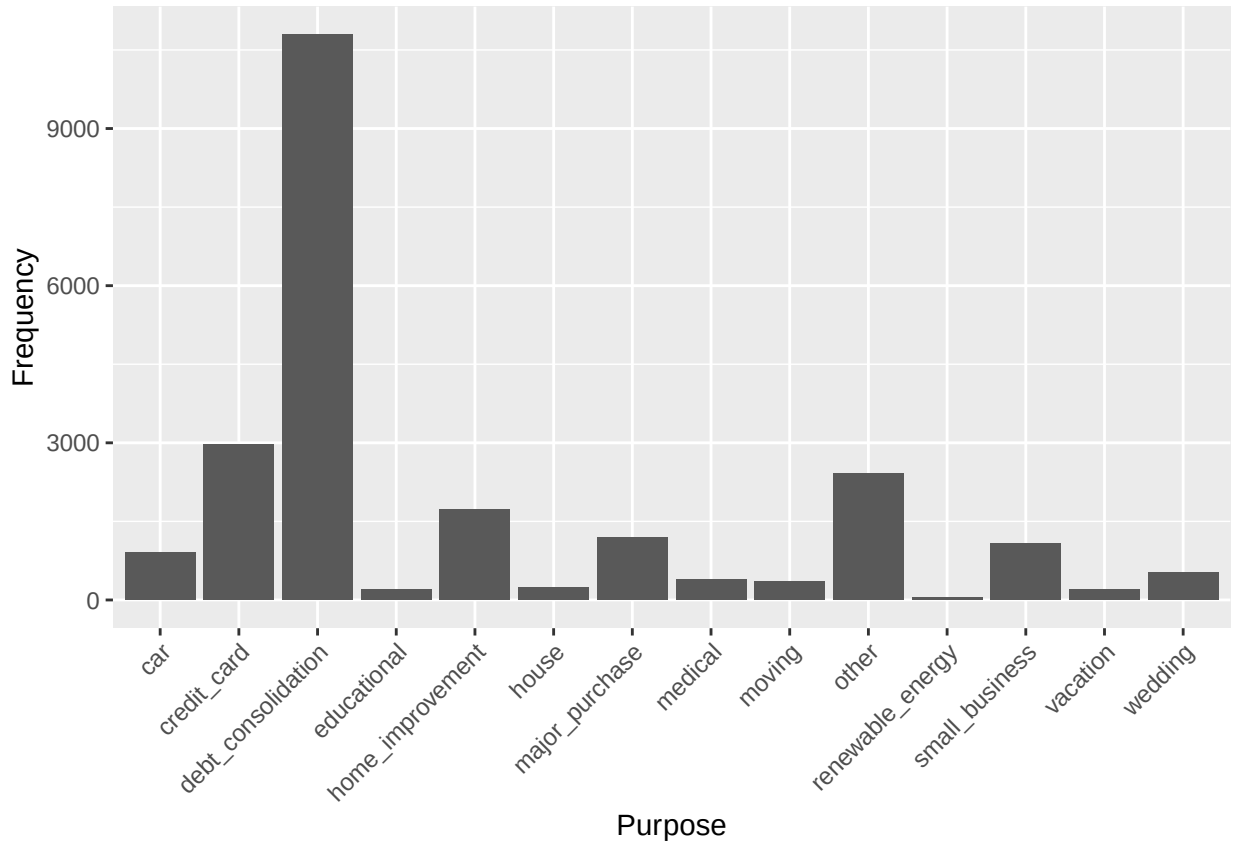
```
numeric_vars <- c("loan_amnt", "annual_inc", "dti", "delinq_2yrs", "inq_last_6mths",
                  "open_acc", "pub_rec", "revol_util", "revol_bal", "total_acc",
                  "credit_age_yrs")
par(mfrow=c(3,4))
for (var in numeric_vars) {
  hist(train_data[[var]], main=var, xlab=var)
}
```



The above plots shows the distribution of the numeric variables within the dataset. These reveal that several variables, namely 'annual_inc,' 'delinq_2yrs,' 'inq_last_6mths,' 'pub_rec,' and 'revol_bal,' exhibit significant outliers as well as largely varying ranges of values, which can potentially impact the integrity of our analysis.

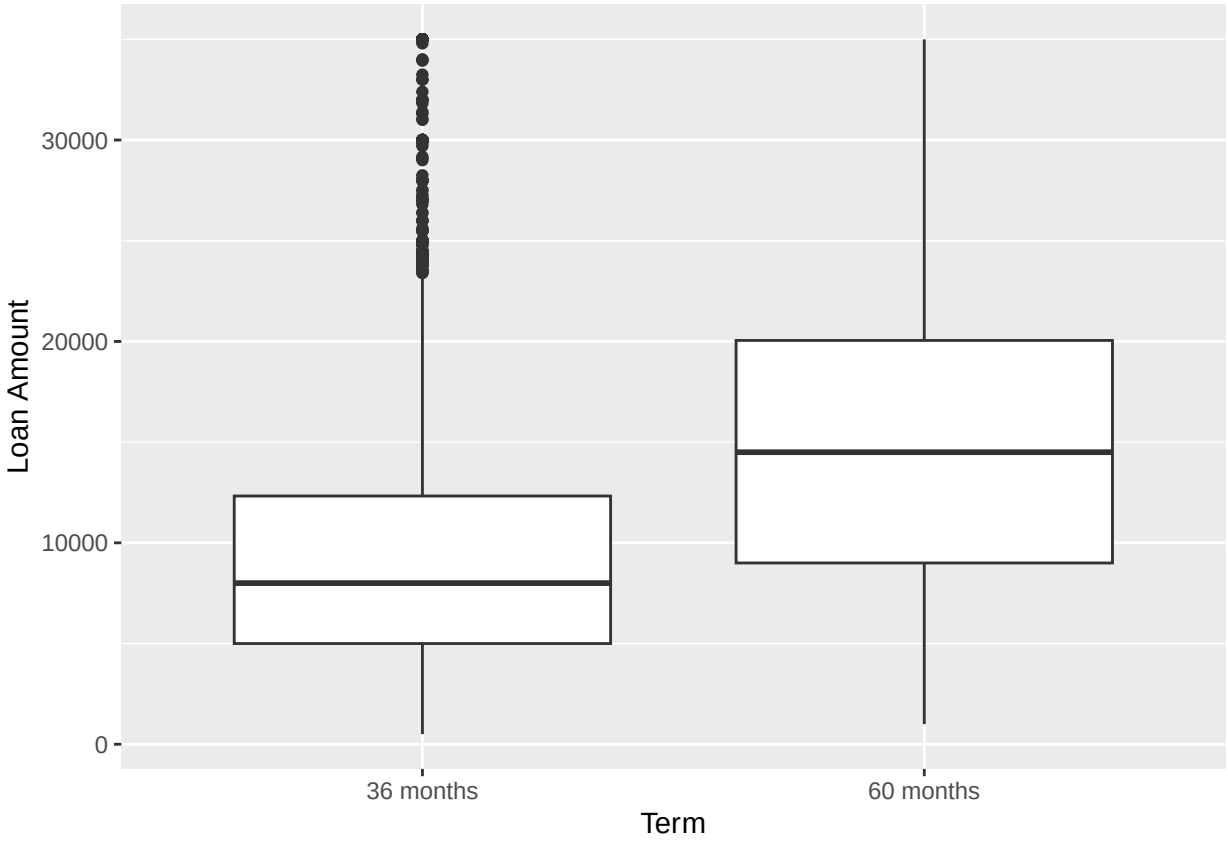
To address the presence of these outliers and large ranges of values and to ensure the robustness of our subsequent statistical modeling, we have chosen to apply a scaling transformation to the numeric covariates in the dataset. The resulting scaled variables will enable us to conduct our analysis with a dataset that has been preprocessed to reduce the undue impact of outliers, so that we can more accurately and appropriately compare our chosen numeric variables.

```
ggplot(train_data, aes(x = purpose)) +
  geom_bar() +
  labs(x = "Purpose", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



The above bar plot represents the frequency of each unique value in the purpose variable. From the plot, it can be seen that the most common reasons to take out loans are for debt consolidation, credit cards, other, and home improvement.

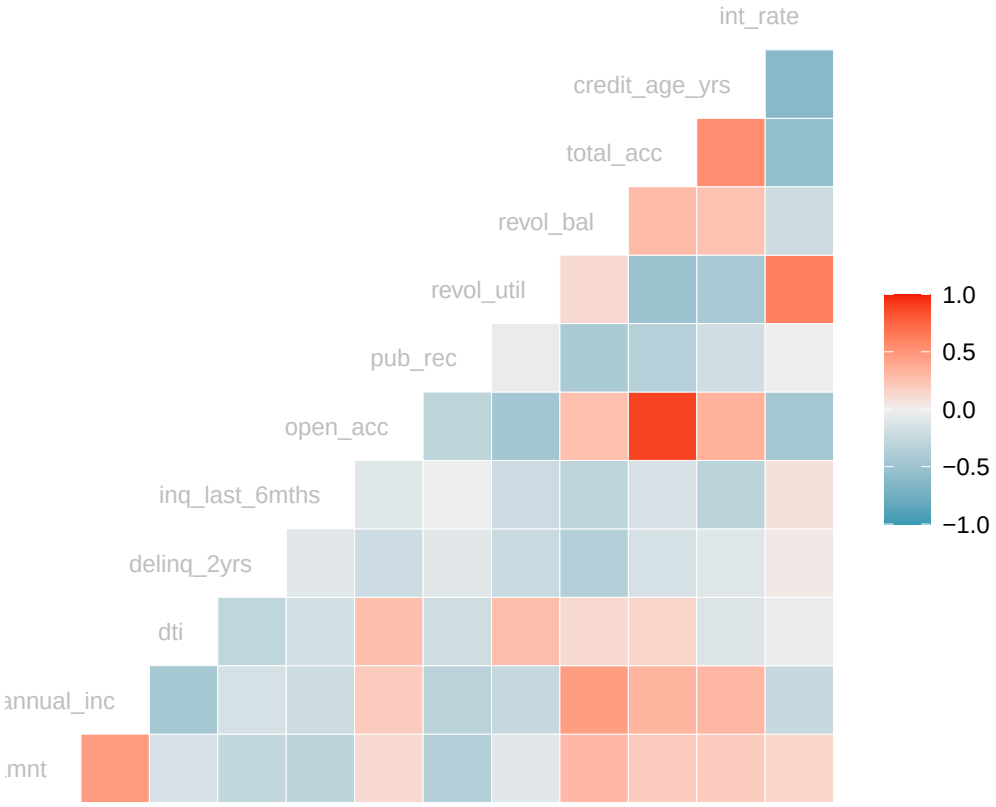
```
ggplot(train_data, aes(x = term, y = loan_amnt)) +
  geom_boxplot() +
  labs(x = "Term", y = "Loan Amount")
```



The above plot consists of two side-by-side boxplots, one for each loan term category. Each boxplot provides information about the distribution of loan amounts for its respective term category. Based on the plot, we can see that, on average, loans with 60-month terms have higher loan amounts than those with 36-month terms, which is an expected behaviour. Also, we can observe some outliers in the 36-month terms, which indicates that there are a few loan amounts which are significantly higher than the typical range of loan amounts for this term category.

Below, we create a correlation matrix of the numeric variables to help us understand the relationship between the different numeric variables.

```
columns <- c("loan_amnt", "annual_inc", "dti", "delinq_2yrs", "inq_last_6mths", "open_acc", "pub_rec",
corr_matrix <- cor(train_data[columns], method = "pearson")
ggcorr(corr_matrix, hjust = 1, size = 3, color = "grey")
```



The correlation matrix (heatmap) reveals several noteworthy patterns:

Upon reviewing the correlation matrix, it becomes evident that certain features exhibit a high degree of positive correlation, such as "total_acc" and "open_acc." Additionally, "credit_age_yrs" and "total_acc," as well as "revol_util" and "int_rate," while not displaying correlations as strong as those between "total_acc" and "open_acc," still exhibit a notable level of correlation.

Following our examination of this matrix, each variable within pairs of highly correlated predictors will be analyzed while fitting models with the goal of identifying and refining the best model.

New Credit Risk Model

Keeping our above exploratory analysis in mind, we also reviewed articles on the factors that traditionally impact an individual's credit risk, especially when applying for loans. Both Understanding the Five Cs of Credit and Top 5 Factors Affecting Credit Risk When Taking A Personal Loan discussed the same 5 major factors impacting credit risk: 1) capacity, 2) capital, 3) conditions, 4) collateral, and 5) character. Factors related to capacity include an individual's employment history and job stability, as well as their annual income. Debt to equity ratios are also assessed as part of capacity. Capital demonstrates an individual's net worth and allows them to list assets that could be used as a reserve in unforeseen circumstances. Lenders will look at current market conditions as well to assess the relative risk presented to the applicant in repaying the loan. Collateral includes assets an applicant may pledge as security to be used in the case that they fail to repay the loan. Finally, character encompasses many factors, including an individual's credit history (how long they have had credit for?), repayment history (are they typically paying on time or not?), any previous loan defaults, and derogatory records for the individual. These 5 factors have been cited in multiple financial resources as those most important in evaluating and determining an applicant's credit risk.

Taking the above research into consideration, as well as our exploratory analysis of the available data, we have been able to match up some of the variables within the available data with those cited as being traditionally important for determining credit risk. For example, we have annual income (*annual_inc*), employment length (*emp_length*), revolving balances (*revol_bal*), revolving utilization, (*revol_util*), and debt-to-income ratio (*dti*) variables which all relate to the first factor listed above, capacity. We do not seem to have any relevant data to account for capital, but perhaps interest rate (*int_rate*) may be an indication of some of the market conditions, as interest rates tend to be higher during financial crises or recessions. We also have home ownership (*home_ownership*), which includes whether an applicant owns a home or is renting, and this could be considered under the factor of collateral. Finally, the following variables may relate to the factor of character: their credit age (*credit_age_yrs*), which demonstrates part of their credit history; total accounts (*total_acc*), which may demonstrate credit usage; inquiries in the last 6 months (*inq_last_6mths*), which demonstrates the frequency the individual is applying for credit or conducting activities which require credit check inquiries; the number of 30+ days past-due incidences of delinquency for the past 2 years (*delinq_2yrs*); number of derogatory public records (*pub_rec*). Other variables that seem relevant and potentially important to provide more information on the requested loan within the context of the applicant include the loan's purpose (*purpose*), amount (*loan_amnt*), and term length (*term*). Finally, we feel as though verification status (*verification_status*) may be important, as this indicates whether an applicant's income was officially verified. If an applicant's income is verified, this may prove beneficial to their application. Thus, we have listed below the initial 16 variables we wish to further investigate in relation to credit risk:

annual_inc, emp_length, verification_status, dti, credit_age_yrs, home_ownership, total_acc, revol_bal, revol_util, int_rate, term, purpose, loan_amnt, inq_last_6mths, delinq_2yrs, pub_rec

Variable Standardization

Now, we'll move on to building our first model. We'll first need to standardize our numeric variables, because, as mentioned above, many of them have differing ranges and some include outliers. Scaling these numeric variables will allow for more accurate comparison so that all variables are on a similar scale.

Training dataset standardization:

```
numeric_columns <- sapply(train_data, is.numeric)  # Identify numeric columns

# Scale the numeric columns while keeping column names
train_data[, numeric_columns] <- scale(train_data[, numeric_columns])

train_scaled_data <- train_data %>%
  mutate_if(is.numeric, scale)
```

Test dataset standardization:

```
numeric_columns <- sapply(test_data, is.numeric)  # Identify numeric columns

# Scale the numeric columns while keeping column names
test_data[, numeric_columns] <- scale(test_data[, numeric_columns])

test_scaled_data <- test_data %>%
  mutate_if(is.numeric, scale)
```

Validation dataset standardization:

```
numeric_columns <- sapply(val_data, is.numeric)  # Identify numeric columns
```

```r
# Scale the numeric columns while keeping column names
val_data[, numeric_columns] <- scale(val_data[, numeric_columns])

val_scaled_data <- val_data %>%
  mutate_if(is.numeric, scale)
```

Variable Selection

First, we have decided to perform stepwiseAIC to identify the most important predictors of loan default, from the list of 16 variables we identified above: annual_inc, emp_length, verification_status, dti, credit_age_yrs, home_ownership, total_acc, revol_bal, revol_util, int_rate, term, purpose, loan_amnt, inq_last_6mths, delinq_2yrs, and pub_rec. However, due to computing limitations, none of the group members succeeded in running it with a full model that includes interactions between variables. Instead, we decided to use the sum of variables as the full model.

```r
# Full model for backwards selection:
full_model <- glm(repay_fail ~ loan_amnt + term + int_rate + emp_length + home_ownership + annual_inc +
                    verification_status + purpose + dti + delinq_2yrs + inq_last_6mths + open_acc +
                    pub_rec + revol_bal + revol_util + total_acc + credit_age_yrs,
                  data = train_scaled_data, family = "binomial")

# Model with no variables present for forwards selection:
null_model <- glm(data = train_scaled_data,
    formula = repay_fail ~ 1,
    family = "binomial")

# Perform backward and forward selection:
backward_sel_model <- stepAIC(full_model, direction = "backward", trace = 0)

forward_sel_model <- stepAIC(
  null_model,
  scope = formula(full_model),
  direction = "forward",
  trace = 0) # prevents automatic output of stepAIC function
```

Now, we can view the model formulas and AIC values to ultimately choose the best one:

```r
formula(backward_sel_model)
```

```
## repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util
```

```r
formula(forward_sel_model)
```

```
## repay_fail ~ int_rate + purpose + annual_inc + inq_last_6mths +
##     term + revol_util + emp_length + revol_bal + pub_rec
```

```r
AIC(backward_sel_model)
```

```
## [1] 18160.57
```

```
AIC(forward_sel_model)
```

```
## [1] 18160.57
```

Based on the above output, both models produced the same formula (same list of covariates), as well as the same AIC values ($AIC = 18160.57$). Thus, we can select either to move forward with. We will choose the backward selection model, and will print the formula below again for review.

```
formula(backward_sel_model)
```

```
## repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util
```

According to stepwiseAIC, the best predictors of repay_fail are term, int_rate, emp_length, annual_inc, purpose, inq_last_6mths, pub_rec, revol_bal, and revol_util. We will move forward with these covariates to fine tune a model that not only fits the data well but improves predictive performance compared to the credit risk model the bank has been using.

First, we'll create a model using the same exact formula suggested by stepAIC:

```
model1 <- glm(repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
                inq_last_6mths + pub_rec + revol_bal + revol_util,
              data = train_scaled_data, family = "binomial")
```

Print the model's AIC:

```
AIC(model1)
```

```
## [1] 18160.57
```

As we already saw above and just reviewed, this model has an $AIC = 18160.57$. Next, we will add a few interactions between some of the covariates that make the most sense based on our credit risk research. Specifically, we want to investigate the interactions between:

- term and int_rate, as interest rate makes sense in the context of the length of time the interest is spread out over
- term and emp_length, as perhaps shorter terms may be less risky with more stable employment histories
- term and annual_inc, as perhaps shorter terms may be less risky with higher annual incomes
- term and purpose, as perhaps credit risk changes depending on the term and loan's purpose (perhaps longer term loans are more appropriate for typically longer term purposes)
- int_rate and emp_length, as perhaps higher interest rates may not be as risky for individuals with stable employment histories
- int_rate and annual_inc, as perhaps higher interest rates may not be as risky for individuals with higher annual incomes

```
model2 <- glm(repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
                inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate +
                term * emp_length + term * annual_inc + term * purpose +
                int_rate * emp_length + int_rate * annual_inc,
              data = train_scaled_data, family = "binomial")
```

Print the model's AIC:

```
AIC(model2)
```

```
## [1] 18171.08
```

The AIC of the new model is higher, with a value of 18171.08. This is to be expected, though, as this new model is penalized for having more covariates than the first model we fit. We will now perform a χ^2-test to confirm whether the addition of these interactions are adding value to the model compared to the model without any interaction terms:

```
anova(model1, model2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util
## Model 2: repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util + term *
##     int_rate + term * emp_length + term * annual_inc + term *
##     purpose + int_rate * emp_length + int_rate * annual_inc
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     23020      18097
## 2     22982      18031 38   65.486 0.003667 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As the χ^2-test has a $p < 0.05$ ($p = 0.0037$), this provides us with enough evidence to reject the null hypothesis that model 1 without interactions and model 2 with interactions explain the same amount of variation in the data. In other words, model 2 is significantly better at explaining the variation in the data compared to model 1. Thus, we will consider interactions between covariates. However, we will now decrease the number of interactions in our model, as model 2 included quite a few interactions. We will only consider the interaction between term and interest rate, as it seems to make sense that these two covariates together would have a large impact on loan default risk. Furthermore, including fewer interactions will assist in our model's interpretability, so that it may ultimately be more useful to the bank. Our goal is twofold in that we want to build an improved model, but this model needs to also be easy to interpret and apply in context. Thus, we want to minimize the complexity of this new model as much as possible so that we can better communicate and interpret our findings, and so that the bank can more easily use and understand our model.

```
model3 <- glm(repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
                inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate,
              data = train_scaled_data, family = "binomial")
```

Print the model's AIC:

```
AIC(model3)
```

```
## [1] 18156.43
```

The AIC of model 3 is lower than that of both model 1 and model 2. We will now conduct χ^2-tests to confirm whether model 3 is significantly better at explaining the variation in the data than both model 1 and model 2.

Compare model 3 and model 1:

```r
anova(model1, model3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util
## Model 2: repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util + term *
##     int_rate
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     23020      18097
## 2     23019      18090  1   6.1363  0.01324 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compare model 3 and model 2:

```r
anova(model2, model3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util + term *
##     int_rate + term * emp_length + term * annual_inc + term *
##     purpose + int_rate * emp_length + int_rate * annual_inc
## Model 2: repay_fail ~ term + int_rate + emp_length + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util + term *
##     int_rate
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1     22982      18031
## 2     23019      18090 -37   -59.35   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As both χ^2-tests have a $p < 0.05$, this indicates that model 3 is significantly better at explaining the variation in the data, as compared to both model 1 and model 2. Thus, for this reason, as well as the fact that model 3 should be relatively easy to interpret, especially in comparison with model 2, we will move forward with model 3.

```r
summary(model3)
```

```
##
## Call:
## glm(formula = repay_fail ~ term + int_rate + emp_length + annual_inc +
##     purpose + inq_last_6mths + pub_rec + revol_bal + revol_util +
##     term * int_rate, family = "binomial", data = train_scaled_data)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -2.29468    0.12282 -18.683  < 2e-16 ***
## term60 months              0.51118    0.05204   9.822  < 2e-16 ***
```

```
## int_rate                    0.45314    0.03098  14.628  < 2e-16 ***
## emp_length1 year            0.06302    0.08575   0.735 0.462370
## emp_length10+ years         0.11638    0.07058   1.649 0.099181 .
## emp_length2 years          -0.11476    0.08219  -1.396 0.162605
## emp_length3 years           0.01722    0.08269   0.208 0.835023
## emp_length4 years          -0.02186    0.08692  -0.251 0.801428
## emp_length5 years           0.07258    0.08697   0.834 0.404005
## emp_length6 years           0.05234    0.09811   0.534 0.593679
## emp_length7 years          -0.08676    0.10959  -0.792 0.428582
## emp_length8 years           0.15873    0.11192   1.418 0.156120
## emp_length9 years          -0.02471    0.12257  -0.202 0.840243
## emp_lengthn/a               0.55108    0.11965   4.606 4.11e-06 ***
## annual_inc                 -0.34587    0.03102 -11.150  < 2e-16 ***
## purposecredit_card         -0.05265    0.12504  -0.421 0.673737
## purposedebt_consolidation   0.21394    0.11291   1.895 0.058121 .
## purposeeducational          0.25429    0.22997   1.106 0.268842
## purposehome_improvement     0.16802    0.13201   1.273 0.203076
## purposehouse                0.34081    0.21135   1.613 0.106844
## purposemajor_purchase       0.05782    0.14520   0.398 0.690459
## purposemedical              0.60441    0.17369   3.480 0.000502 ***
## purposemoving               0.49819    0.18466   2.698 0.006978 **
## purposeother                0.45453    0.12268   3.705 0.000211 ***
## purposerenewable_energy     0.38635    0.41018   0.942 0.346246
## purposesmall_business       1.02101    0.13076   7.808 5.80e-15 ***
## purposevacation             0.44727    0.22561   1.983 0.047421 *
## purposewedding             -0.03855    0.17998  -0.214 0.830382
## inq_last_6mths              0.21509    0.01757  12.243  < 2e-16 ***
## pub_rec                     0.06002    0.01672   3.589 0.000332 ***
## revol_bal                   0.09438    0.02080   4.537 5.70e-06 ***
## revol_util                  0.11653    0.02300   5.067 4.05e-07 ***
## term60 months:int_rate     -0.11027    0.04444  -2.481 0.013086 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19652  on 23051  degrees of freedom
## Residual deviance: 18090  on 23019  degrees of freedom
## AIC: 18156
##
## Number of Fisher Scoring iterations: 5
```

Logit Link Function

Next, we will fit a model using the logit link function, with our final list of covariates from model 3.

```
fit_logit <- glm(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
                   inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate,
                 data = train_scaled_data, family = "binomial")

AIC(fit_logit)
```

```
## [1] 18156.43
```

This model has the lowest AIC value we have received thus far, and this, as well as the results of our χ^2-tests, indicate that this is the best model at the moment. Thus, we will proceed with this model.

```
summary(fit_logit)
```

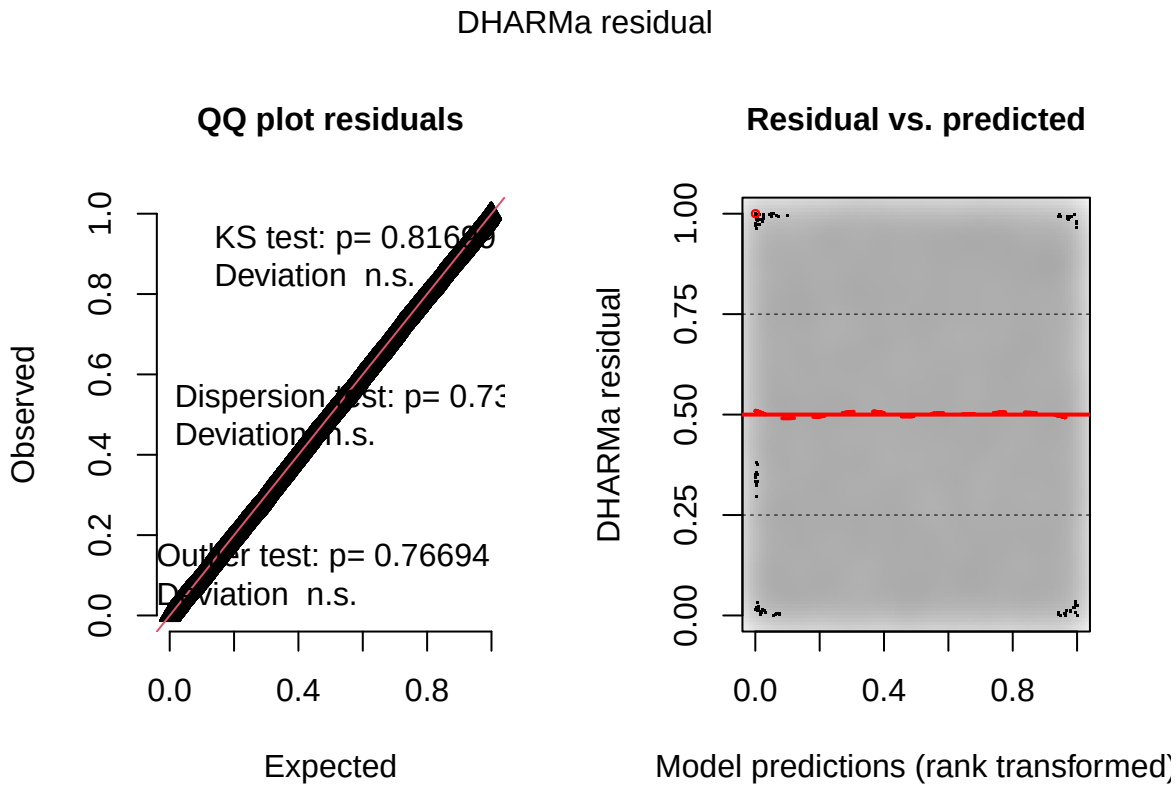
```
##
## Call:
## glm(formula = repay_fail ~ int_rate + emp_length + annual_inc +
##     purpose + inq_last_6mths + pub_rec + revol_bal + revol_util +
##     term * int_rate, family = "binomial", data = train_scaled_data)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.29468    0.12282 -18.683  < 2e-16 ***
## int_rate                    0.45314    0.03098  14.628  < 2e-16 ***
## emp_length1 year            0.06302    0.08575   0.735 0.462370
## emp_length10+ years         0.11638    0.07058   1.649 0.099181 .
## emp_length2 years          -0.11476    0.08219  -1.396 0.162605
## emp_length3 years           0.01722    0.08269   0.208 0.835023
## emp_length4 years          -0.02186    0.08692  -0.251 0.801428
## emp_length5 years           0.07258    0.08697   0.834 0.404005
## emp_length6 years           0.05234    0.09811   0.534 0.593679
## emp_length7 years          -0.08676    0.10959  -0.792 0.428582
## emp_length8 years           0.15873    0.11192   1.418 0.156120
## emp_length9 years          -0.02471    0.12257  -0.202 0.840243
## emp_lengthn/a               0.55108    0.11965   4.606 4.11e-06 ***
## annual_inc                 -0.34587    0.03102 -11.150  < 2e-16 ***
## purposecredit_card         -0.05265    0.12504  -0.421 0.673737
## purposedebt_consolidation   0.21394    0.11291   1.895 0.058121 .
## purposeeducational          0.25429    0.22997   1.106 0.268842
## purposehome_improvement     0.16802    0.13201   1.273 0.203076
## purposehouse                0.34081    0.21135   1.613 0.106844
## purposemajor_purchase       0.05782    0.14520   0.398 0.690459
## purposemedical              0.60441    0.17369   3.480 0.000502 ***
## purposemoving               0.49819    0.18466   2.698 0.006978 **
## purposeother                0.45453    0.12268   3.705 0.000211 ***
## purposerenewable_energy     0.38635    0.41018   0.942 0.346246
## purposesmall_business       1.02101    0.13076   7.808 5.80e-15 ***
## purposevacation             0.44727    0.22561   1.983 0.047421 *
## purposewedding             -0.03855    0.17998  -0.214 0.830382
## inq_last_6mths              0.21509    0.01757  12.243  < 2e-16 ***
## pub_rec                     0.06002    0.01672   3.589 0.000332 ***
## revol_bal                   0.09438    0.02080   4.537 5.70e-06 ***
## revol_util                  0.11653    0.02300   5.067 4.05e-07 ***
## term60 months               0.51118    0.05204   9.822  < 2e-16 ***
## int_rate:term60 months     -0.11027    0.04444  -2.481 0.013086 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19652  on 23051  degrees of freedom
## Residual deviance: 18090  on 23019  degrees of freedom
## AIC: 18156
```

```
##
## Number of Fisher Scoring iterations: 5
```

Logit Model Goodness of Fit

Before we try to interpret this model's summary, we will first check the goodness-of-fit to determine whether this model is actually a good fit for the data. We will first look at the goodness-of-fit for "seen" data (using the training dataset).

```
# simulate residuals from the model:
res_logit_seen = simulateResiduals(fit_logit)

# plot observed quantile versus expected quantile to assess distribution fit, and
# predicted value versus standardised residuals for unmodelled pattern in the residuals
plot(res_logit_seen)
```



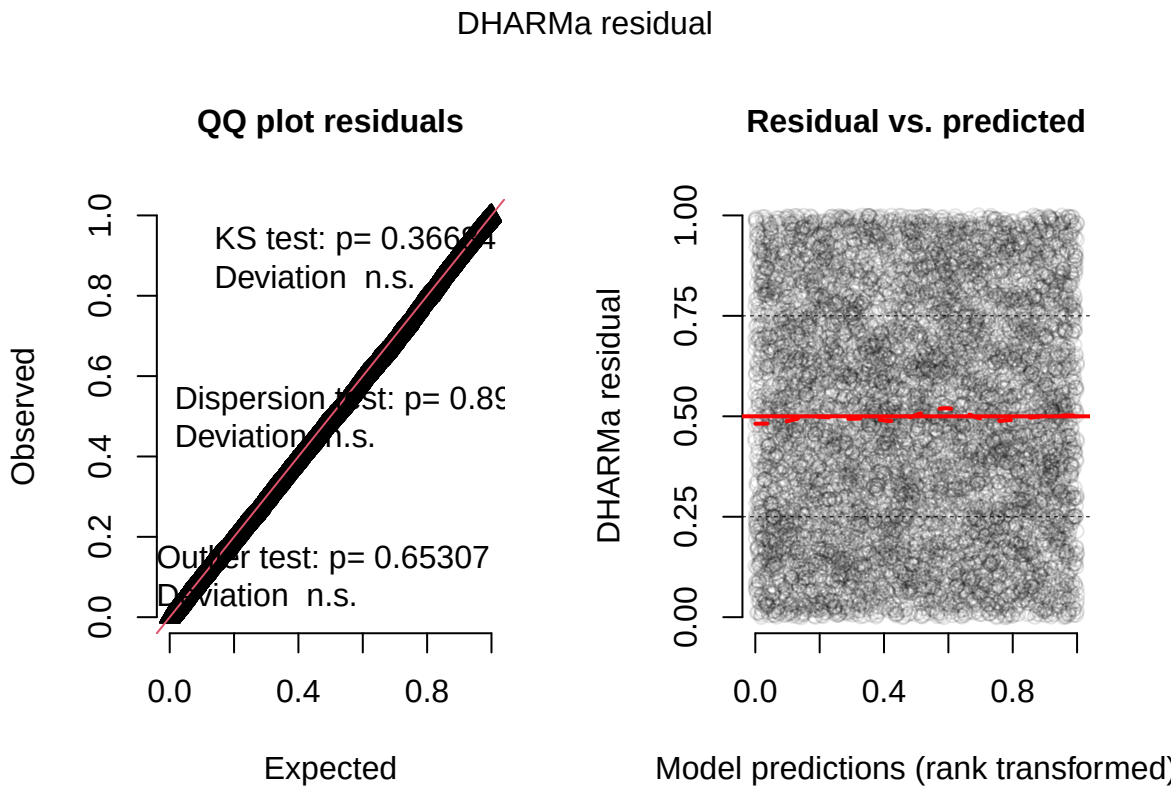
The above DHARMA residuals plots shows no sign of overdispersion. The distribution test (KS), dispersion test, and outlier test all are non-significant, which means there is evidence to suggest that the distribution of the simulated quantiles follows a uniform distribution. Additionally, in the second plot, the quantiles of the residuals (red lines) show a uniform pattern for predicted values.

Now, we will use the test dataset, or "unseen" data, to assess whether this model is still a good fit for unseen data.

```
# first fit the model on unseen data (test dataset)
fit_logit_unseen <- glm(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
                          inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate,
                        data = test_scaled_data, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# simulate residuals from the model:
res_logit_unseen = simulateResiduals(fit_logit_unseen)

# plot observed quantile versus expected quantile to assess distribution fit, and
# predicted value versus standardised residuals for unmodelled pattern in the residuals
plot(res_logit_unseen)
```



As can be seen in the above residuals plots, once again, this model appears to be a good fit for the data. The QQ plot of residuals does not indicate any deviations from normality, overdispersion, or outliers in the residuals. In other words, it appears that this model is also a reasonable fit for unseen data, which helps us to validate that our model is appropriate.

Logit Model: Performance

Next, we will analyse the performance of our logit model, including for prediction. After all, this is one of the main reasons we've been tasked with conducting this analysis.

We will first analyse the model's performance on "seen" data:

Training dataset ROC:

```
# ROC curve on scaled train data
prob_logit_train=predict(fit_logit,type=c("response"))
g <- roc(train_scaled_data$repay_fail ~ prob_logit_train)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

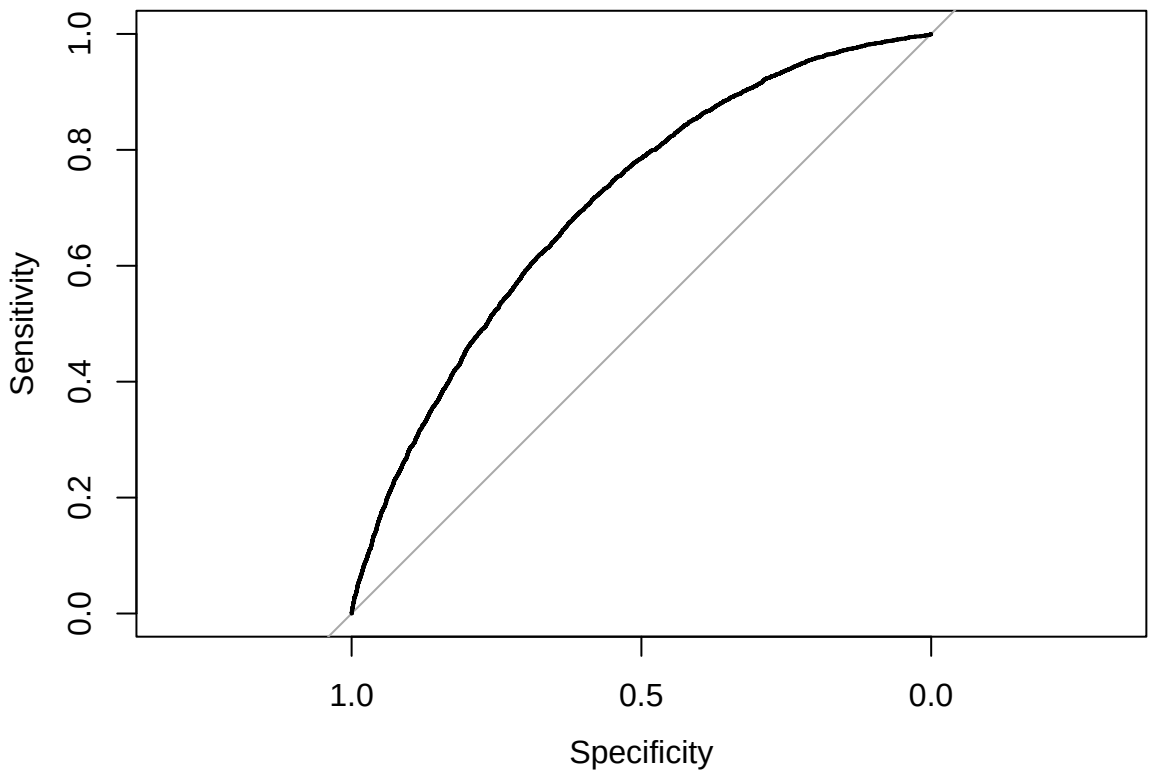
```
AUC <- g$auc
AUC
```

```
## Area under the curve: 0.7051
```

```
Gini <- 2*(AUC - 1/2)
Gini
```

```
## [1] 0.4101659
```

```
plot(g)
```



Now, we will assess this model's predictive performance on "unseen" data. In other words, we'll assess how well this model performs on predicting credit risk for new applicants.

Validation dataset ROC:

```
# ROC curve on scaled validation data
prob_logit_val=predict(fit_logit,newdata=val_scaled_data,type=c("response"))
g <- roc(val_scaled_data$repay_fail ~ prob_logit_val)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

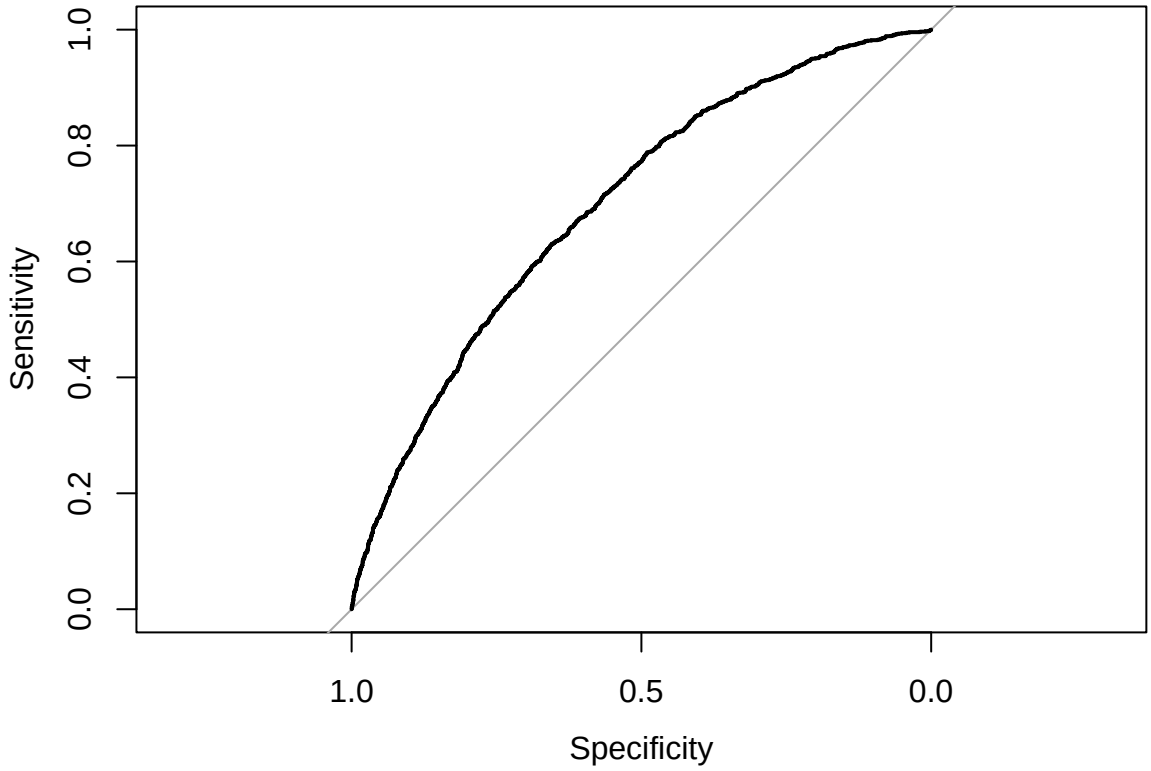
```
AUC <- g$auc
AUC
```

```
## Area under the curve: 0.6976
```

```
Gini <- 2*(AUC - 1/2)
Gini
```

```
## [1] 0.3951212
```

```
plot(g)
```



For "seen" data, our model has a $Gini = 0.410$, which is much higher than that of the old model ($Gini = 0.114$). For "unseen" data (using the validation dataset), our model has a $Gini = 0.395$, which is also much higher than that of the old model ($Gini = 0.110$). Additionally, the ROC curves for the new model on both seen and unseen data are positioned closer to the top-left corners compared to the old model. These findings collectively indicate that the new model is not only performing better than the old model on seen data, but also has better predictive performance to predict credit risk on unseen data for new applicants.

Probit Link Function

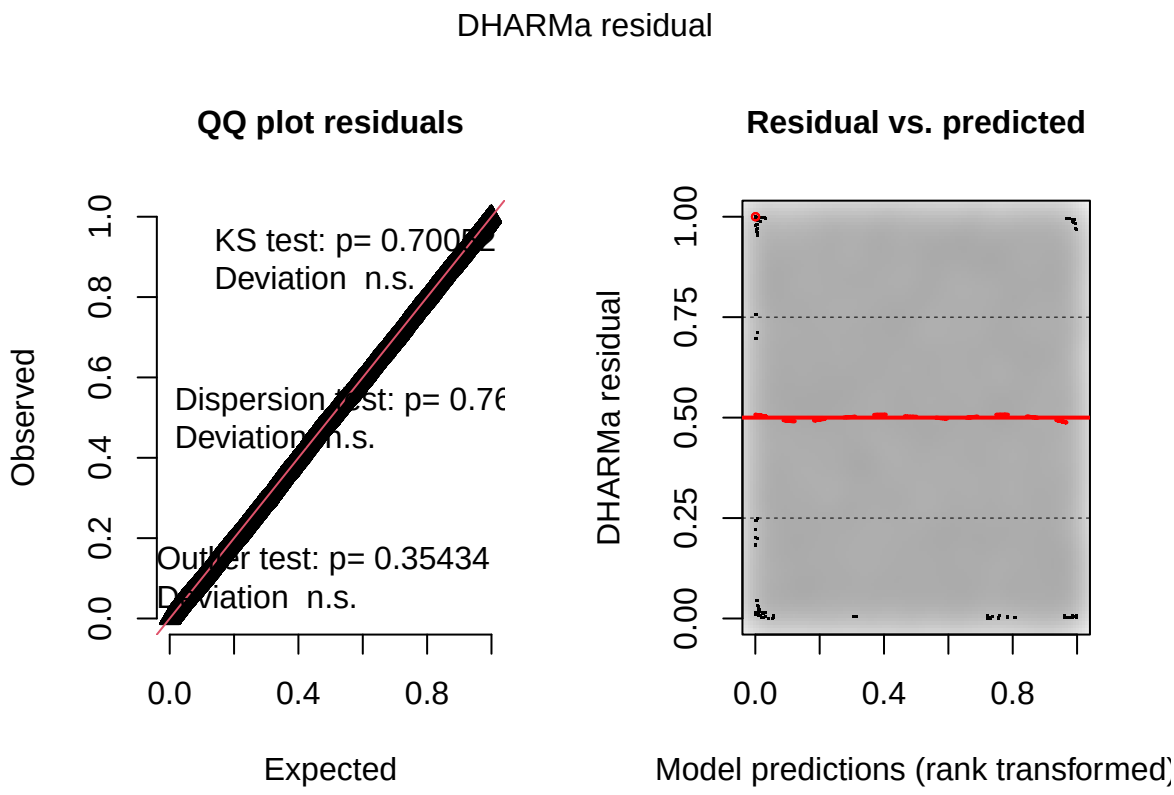
Though we are quite happy with the above selected model (with a logit link function), we will now test whether using a probit link function changes the AIC or performance of our model.

```
fit_probit <- glm(repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
                    pub_rec + revol_bal + revol_util + term * int_rate,
                  data = train_scaled_data, family = binomial("probit"))

AIC(fit_probit)
```

```
## [1] 18150.14
```

Review probit model goodness-of-fit:

```
# simulate residuals from the model:
res_probit = simulateResiduals(fit_probit)

# plot observed quantile versus expected quantile to assess distribution fit, and
# predicted value versus standardised residuals for unmodelled pattern in the residuals
plot(res_probit)
```



As can be seen above, the model fit using the probit link function also appears to be a good fit for the data in that it does not indicate any deviations from normality, overdispersion, or outliers in the residuals.

Probit Model: Performance

Now, we will assess the performance of the model fit using the probit link function for comparison against the logit link function model.

Training dataset:

```
# Probit model: ROC curve on scaled train data
prob_probit_train=predict(fit_probit,type=c("response"))
g <- roc(train_scaled_data$repay_fail ~ prob_probit_train)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

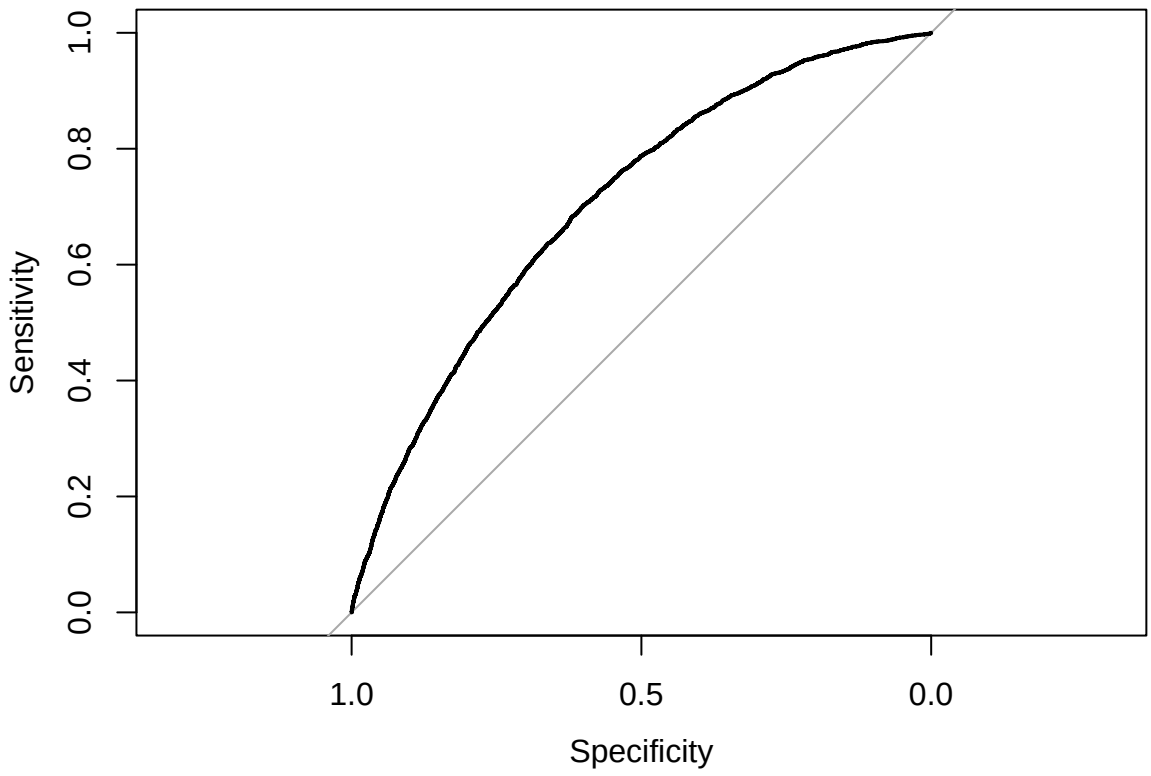
```
AUC <- g$auc
AUC
```

```
## Area under the curve: 0.7052
```

```
Gini <- 2*(AUC - 1/2)
Gini
```

```
## [1] 0.4103512
```

```
plot(g)
```



Validation dataset:

```
# Probit model: ROC curve on scaled validation data
prob_probit_val=predict(fit_probit,newdata = val_scaled_data,type=c("response"))
g <- roc(val_scaled_data$repay_fail ~ prob_probit_val)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

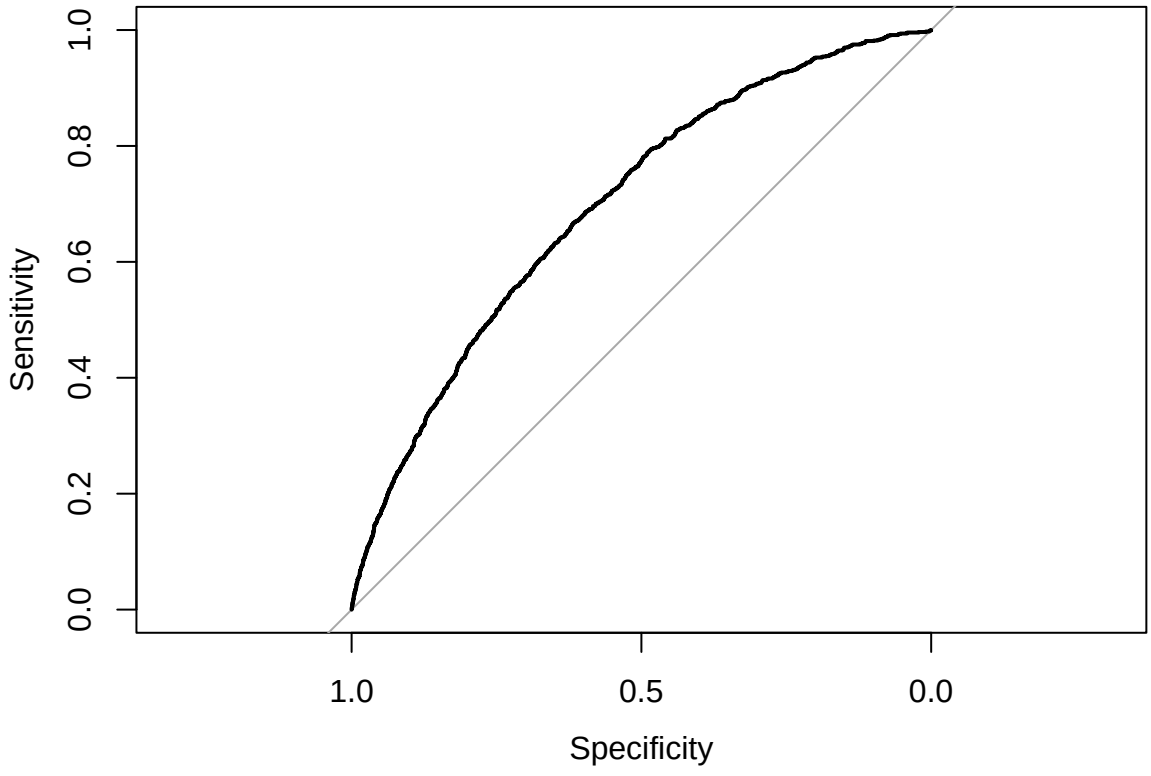
```
AUC <- g$auc
AUC
```

```
## Area under the curve: 0.6968
```

```
Gini <- 2*(AUC - 1/2)
Gini
```

```
## [1] 0.3936362
```

```
plot(g)
```



While the model fit with a probit link function has lower AIC than the model with a logit link function and similarly demonstrates goodness-of-fit for the data, the probit model has a slightly lower Gini score on the validation dataset (Logit model: $Gini = 0.410$ on train, $Gini = 0.395$ on validation vs. Probit model: $Gini = 0.410$ on train, $Gini = 0.394$). Thus, we'll consider the logit model as better than the probit model.

Cloglog Link Function

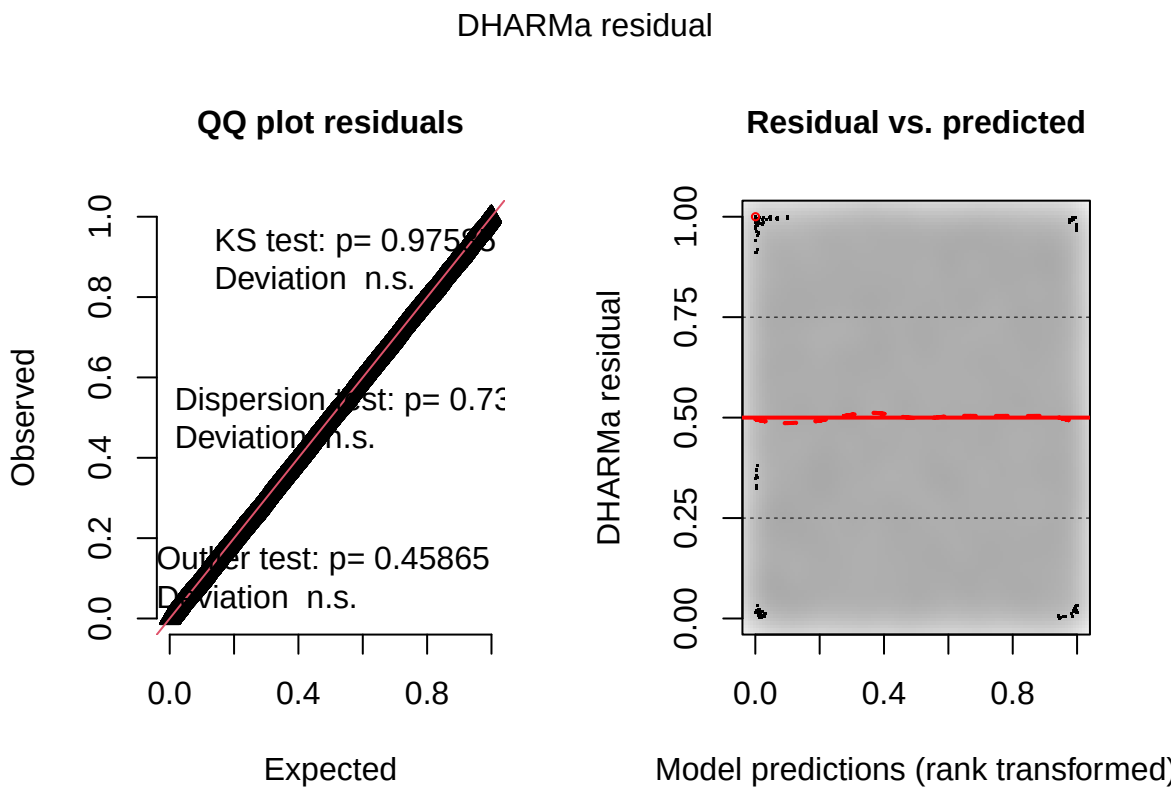
Finally, we will test whether using a complementary log-log link function changes the AIC or performance of our model.

```
fit_cloglog <- glm(repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
                     pub_rec + revol_bal + revol_util + term * int_rate,
                   data = train_scaled_data, family = binomial("cloglog"))

AIC(fit_cloglog)
```

```
## [1] 18183.15
```

Review cloglog model goodness-of-fit:

```
# simulate residuals from the model:
res_cloglog = simulateResiduals(fit_cloglog)

# plot observed quantile versus expected quantile to assess distribution fit, and
# predicted value versus standardised residuals for unmodelled pattern in the residuals
plot(res_cloglog)
```



As can be seen above, the model fit with the cloglog link function appears to be a good fit for the data, as there is no indication of non-normality, overdispersion, or outliers in the residuals. However, the cloglog model has a higher AIC than both the logit and probit models.

Cloglog Model: Performance

Now, we will assess the performance of the model fit using the cloglog link function for comparison against the logit link function model.

Training dataset:

```
# Cloglog model: ROC curve on scaled train data
prob_cloglog_train=predict(fit_cloglog,type=c("response"))
g <- roc(train_scaled_data$repay_fail ~ prob_cloglog_train)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

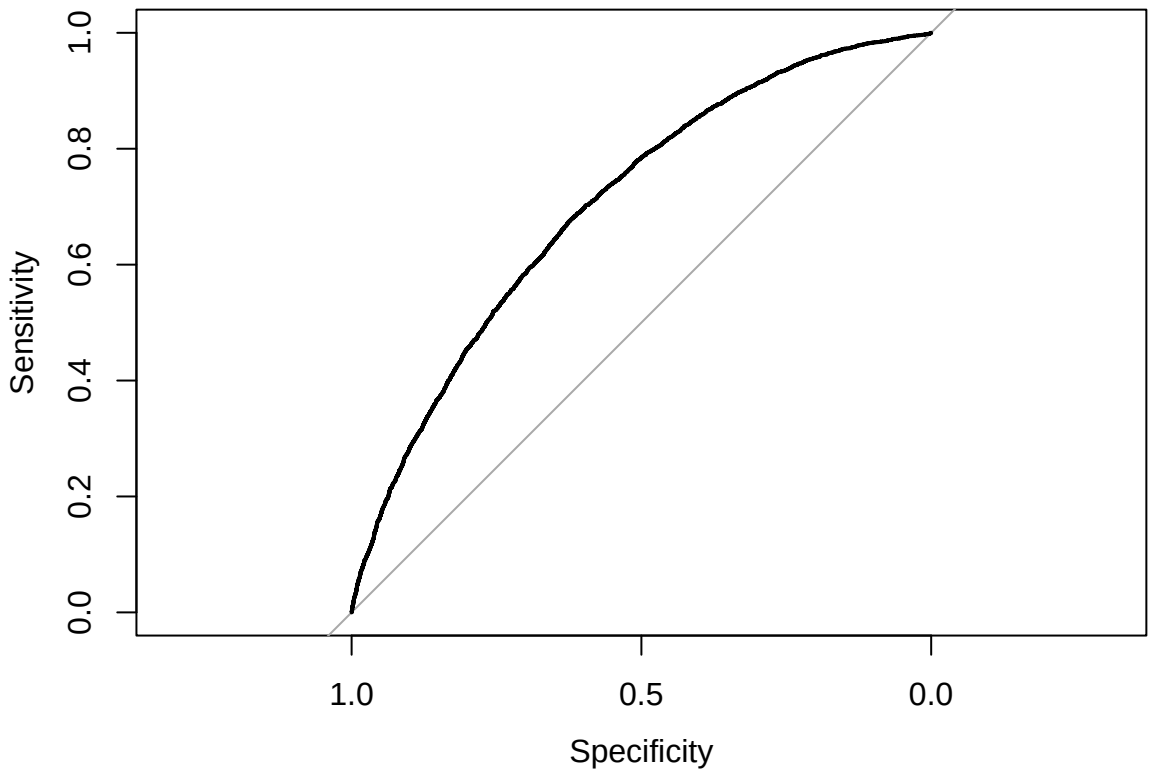
```
AUC <- g$auc
AUC
```

```
## Area under the curve: 0.7037
```

```
Gini <- 2*(AUC - 1/2)
Gini
```

```
## [1] 0.4074574
```

```
plot(g)
```



Validation dataset:

```
# Cloglog model: ROC curve on scaled validation data
prob_cloglog_val=predict(fit_cloglog,newdata = val_scaled_data,type=c("response"))
g <- roc(val_scaled_data$repay_fail ~ prob_cloglog_val)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

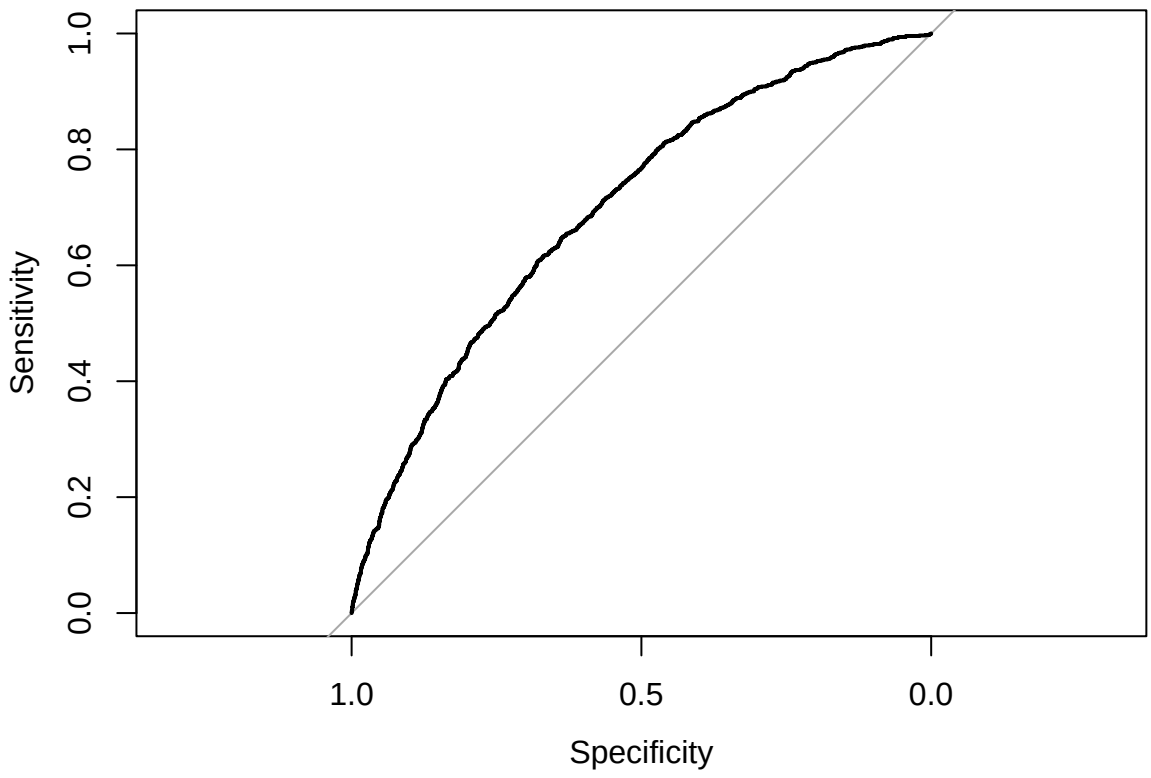
```
AUC <- g$auc
AUC
```

```
## Area under the curve: 0.6973
```

```
Gini <- 2*(AUC - 1/2)
Gini
```

```
## [1] 0.3946075
```

```
plot(g)
```



The cloglog model has a slightly lower Gini score on the the training dataset (Logit model: $Gini = 0.410$ on train, $Gini = 0.395$ on validation vs. Cloglog model: $Gini = 0.407$ on train, $Gini = 0.395$ on validation).

Based on the comparisons conducted between the three different link functions for our GLM model, we will choose to use the model with the logit link function as our final model. This model demonstrated the best combination of AIC value and performance, including predictive performance on unseen data, compared to the other models.

Calculate Odds Ratio and CI

Next, we will calculate the odds ratios and confidence intervals for our final model. As this model was fit using the logit link function, we can exponentiate the estimates to get the odds ratio for each covariate.

```
summary_logit <- summary(fit_logit)

parameter_estimates <- summary_logit$coef[, "Estimate"]
standard_errors <- summary_logit$coef[, "Std. Error"]
p_values <- summary_logit$coef[, "Pr(>|z|)"]

# Calculate the odds ratios by exponentiating the parameter estimate values
odds_ratios <- exp(parameter_estimates)

# Calculate the confidence intervals for the odds ratios
lower_ci <- exp(parameter_estimates - 1.96 * standard_errors)  # 1.96 corresponds to a 95% confidence i
upper_ci <- exp(parameter_estimates + 1.96 * standard_errors)

model_results <- data.frame(
  Estimate = parameter_estimates,
  OddsRatio = odds_ratios,
  LowerCI = lower_ci,
  UpperCI = upper_ci,
  p_value = p_values
)

# Display results in a table
knitr::kable(
  model_results,
  col.names = c("Estimate", "Odds Ratio", "Lower Bound (2.5%)", "Upper Bound (97.5%)", "p-value")
)
```

	Estimate	Odds Ratio	Lower Bound (2.5%)	Upper Bound (97.5%)	p-value
(Intercept)	-2.2946810	0.1007935	0.0792294	0.1282268	0.0000000
int_rate	0.4531371	1.5732399	1.4805611	1.6717202	0.0000000
emp_length1 year	0.0630227	1.0650510	0.9002801	1.2599786	0.4623703
emp_length10+ years	0.1163787	1.1234212	0.9782767	1.2901004	0.0991814
emp_length2 years	-0.1147634	0.8915771	0.7589263	1.0474134	0.1626049
emp_length3 years	0.0172216	1.0173707	0.8651494	1.1963751	0.8350230
emp_length4 years	-0.0218610	0.9783763	0.8251176	1.1601015	0.8014284
emp_length5 years	0.0725754	1.0752739	0.9067528	1.2751148	0.4040049
emp_length6 years	0.0523446	1.0537388	0.8693955	1.2771695	0.5936789
emp_length7 years	-0.0867569	0.9169000	0.7396625	1.1366070	0.4285822
emp_length8 years	0.1587255	1.1720161	0.9411695	1.4594839	0.1561202
emp_length9 years	-0.0247079	0.9755948	0.7672478	1.2405186	0.8402435
emp_lengthn/a	0.5510784	1.7351232	1.3724050	2.1937057	0.0000041

	Estimate	Odds Ratio	Lower Bound (2.5%)	Upper Bound (97.5%)	p-value
annual_inc	-0.3458713	0.7076035	0.6658644	0.7519590	0.0000000
purposecredit_card	-0.0526453	0.9487165	0.7425053	1.2121973	0.6737369
purposedebt_consolidation	0.2139387	1.2385467	0.9926639	1.5453347	0.0581207
purposeeducational	0.2542904	1.2895463	0.8216344	2.0239289	0.2688423
purposehome_improvement	0.1680222	1.1829629	0.9132830	1.5322756	0.2030756
purposehouse	0.3408083	1.4060837	0.9291963	2.1277218	0.1068445
purposemajor_purchase	0.0578245	1.0595290	0.7971016	1.4083547	0.6904589
purposemedical	0.6044068	1.8301663	1.3020944	2.5724009	0.0005018
purposemoving	0.4981937	1.6457459	1.1459823	2.3634567	0.0069776
purposeother	0.4545316	1.5754352	1.2387273	2.0036663	0.0002113
purposerenewable_energy	0.3863451	1.4715924	0.6586234	3.2880463	0.3462455
purposesmall_business	1.0210112	2.7760005	2.1483945	3.5869476	0.0000000
purposevacation	0.4472715	1.5640389	1.0050940	2.4338199	0.0474209
purposewedding	-0.0385535	0.9621803	0.6761697	1.3691693	0.8303821
inq_last_6mths	0.2150886	1.2399717	1.1980021	1.2834116	0.0000000
pub_rec	0.0600174	1.0618550	1.0276187	1.0972320	0.0003315
revol_bal	0.0943838	1.0989815	1.0550732	1.1447170	0.0000057
revol_util	0.1165282	1.1235892	1.0740659	1.1753959	0.0000004
term60 months	0.5111847	1.6672652	1.5055788	1.8463155	0.0000000
int_rate:term60 months	-0.1102704	0.8955920	0.8208873	0.9770950	0.0130857

Displaying only those variables whose p-value < 0.05:

```
# Filter rows where p-value is less than 0.05
significant_results <- model_results[model_results$p_value < 0.05, ]

# Print the significant results
knitr::kable(
  significant_results,
  col.names = c("Estimate", "Odds Ratio", "Lower Bound (2.5%)", "Upper Bound (97.5%)", "p-value")
)
```

	Estimate	Odds Ratio	Lower Bound (2.5%)	Upper Bound (97.5%)	p-value
(Intercept)	-2.2946810	0.1007935	0.0792294	0.1282268	0.0000000
int_rate	0.4531371	1.5732399	1.4805611	1.6717202	0.0000000
emp_lengthn/a	0.5510784	1.7351232	1.3724050	2.1937057	0.0000041
annual_inc	-0.3458713	0.7076035	0.6658644	0.7519590	0.0000000
purposemedical	0.6044068	1.8301663	1.3020944	2.5724009	0.0005018
purposemoving	0.4981937	1.6457459	1.1459823	2.3634567	0.0069776
purposeother	0.4545316	1.5754352	1.2387273	2.0036663	0.0002113
purposesmall_business	1.0210112	2.7760005	2.1483945	3.5869476	0.0000000
purposevacation	0.4472715	1.5640389	1.0050940	2.4338199	0.0474209
inq_last_6mths	0.2150886	1.2399717	1.1980021	1.2834116	0.0000000
pub_rec	0.0600174	1.0618550	1.0276187	1.0972320	0.0003315
revol_bal	0.0943838	1.0989815	1.0550732	1.1447170	0.0000057
revol_util	0.1165282	1.1235892	1.0740659	1.1753959	0.0000004
term60 months	0.5111847	1.6672652	1.5055788	1.8463155	0.0000000

	Estimate	Odds Ratio	Lower Bound (2.5%)	Upper Bound (97.5%)	p-value
int_rate:term60 months	-0.1102704	0.8955920	0.8208873	0.9770950	0.0130857

Interpretations

Listed below are interpretations of the covariates of our final model:

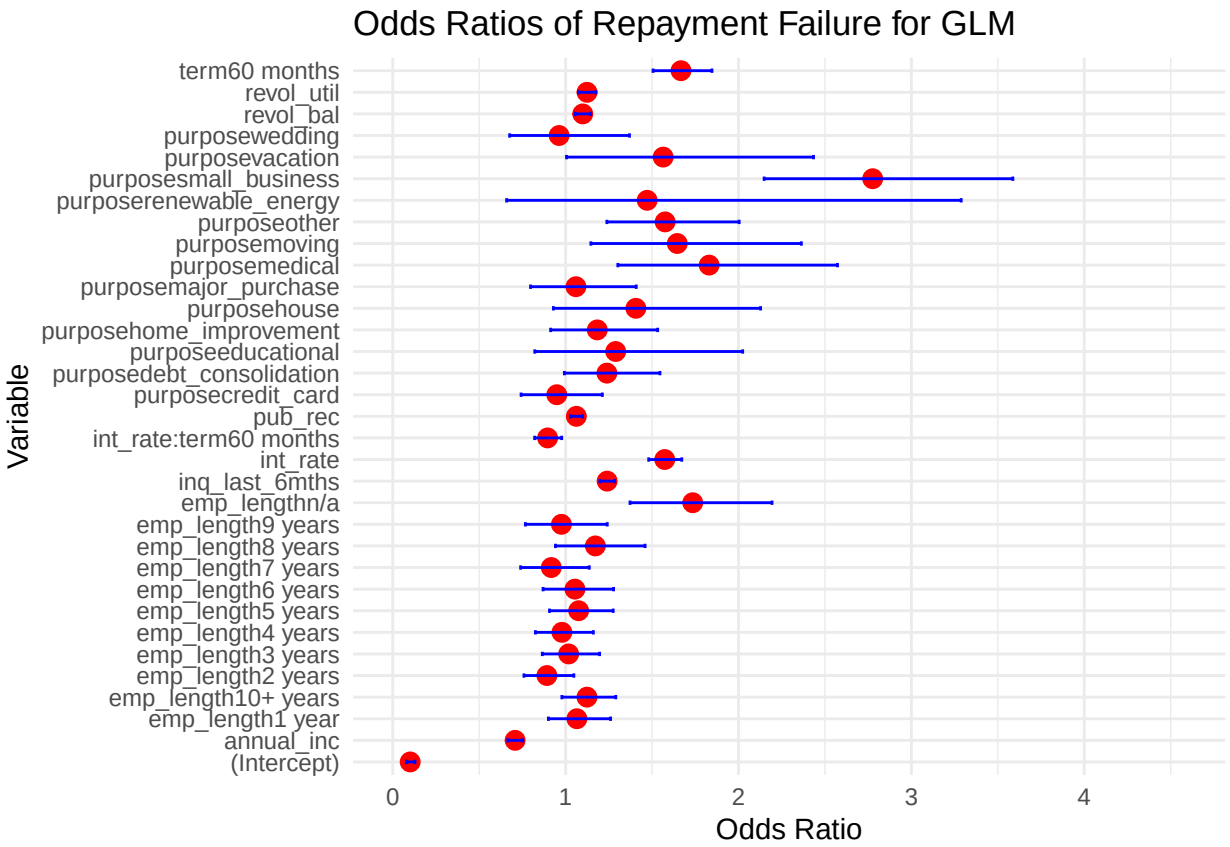
- The intercept or baseline refers to individuals who have been employed for less than one year, are seeking a car loan, and have a 36-month loan term when all other numeric predictors are equal to zero, and is statistically significant (95% CI: [0.08,0.13], $p < 2e-16$).
- For a one-unit increase in the interest rate, the odds of "repay_fail" occurring increase by a factor of approximately 1.57 (95% CI: [1.48,1.67], $p < 2e-16$). As can be seen by the low p-value, this is a very significant effect.
- For individuals with 1 to 10 plus years of employment, none of these employment lengths have a significant effect on the odds of "repay_fail" occurring compared to the reference category (less than one year). Their odds ratios are close to 1, and all $p > 0.05$, indicating that these employment lengths do not significantly impact the likelihood of loan repayment failure.
- For individuals with missing/unspecified employment lengths or unemployed individuals ("n/a"), the odds of "repay_fail" occurring increase by a factor of approximately 1.73 (95% CI: [1.37,2.19], $p = 4.11e-06$), compared to baseline (less than one year). This indicates that the absence of employment length information is associated with a higher likelihood of loan repayment failure.
- For a one-unit increase in annual income, the odds of "repay_fail" occurring decrease by a factor of approximately 0.71 (95% CI: [0.67,0.75], $p < 2e-16$).
- For loans with a stated purpose of "credit card", "debt consolidation", "educational", "home improvement", "house", "major purchase", "renewable energy", and "wedding", there is no significant effect on the odds of "repay_fail" occurring compared to the reference category (car). As all $p > 0.05$, these loan purposes do not significantly impact the likelihood of loan repayment failure.
  - For loans with a stated purpose of "medical", the odds of "repay_fail" occurring increase by a factor of approximately 1.83 (95% CI: [1.30,2.57], $p = 0.000502$), compared to the baseline (car).
  - For loans with a stated purpose of "moving", the odds of "repay_fail" occurring increase by a factor of approximately 1.65 (95% CI: [1.15,2.36], $p = 0.006978$), compared to baseline (car).
  - For loans with a stated purpose of "other", the odds of "repay_fail" occurring increase by a factor of approximately 1.58 (95% CI: [1.24,2.00], $p = 0.000211$), compared to baseline (car).
  - For loans with a stated purpose of "small business", the odds of "repay_fail" occurring increase by a factor of approximately 2.78 (95% CI: [2.15,3.59], $p = 5.80e-15$), compared to baseline (car).
  - For loans with a stated purpose of "vacation", the odds of "repay_fail" occurring increase by a factor of approximately 1.56 (95% CI: [1.01,2.43], $p = 0.047421$), compared to baseline (car).
  - As mentioned above, the specific loan purposes of "medical," "moving," "other," "small_business", and "vacation" are all associated with a higher likelihood of loan repayment failure, as each had a statistically significant effect with a $p < 0.05$. The effect of small business loan purposes is the most significant with the smallest p-value, while vacation loan purposes have a lesser degree of effect.

- For a one-unit increase in the number of inquiries in past 6 months, the odds of "repay_fail" occurring increase by a factor of approximately 1.24 (95% CI: [1.20,1.28], $p < 2e-16$). As can be seen by the low p-value, this is a very significant effect.
- For a one-unit increase in the number of derogatory public records, the odds of "repay_fail" occurring increase by a factor of approximately 1.06 (95% CI: [1.03,1.10], $p = 0.000332$).
- For a one-unit increase in revolving balance, the odds of "repay_fail" occurring increase by a factor of approximately 1.10 (95% CI: [1.06,1.14], $p = 5.70e-06$).
- For a one-unit increase in revolving credit utilization, the odds of "repay_fail" occurring increase by a factor of approximately 1.12 (95% CI: [1.07,1.18], $p = 4.05e-07$).
- Having a 60-month loan term increases the odds of "repay_fail" occurring by a factor of approximately 1.67 (95% CI: [1.51,1.85], $p < 2e-16$), compared to the reference category (36-month loan term).
- The negative estimate for the interaction between interest rate and the 60-month term suggests that the effect of interest rate on "repay_fail" is less severe for loans with a 60-month term compared to the reference category of the 36-month term. In other words, while interest rate still has a significant impact on loan repayment on its own, its impact differs for loans with longer terms.
  - The interaction between interest rate and the 60-month loan term decreases the odds of "repay_fail" by a factor of approximately 0.90 (95% CI: [0.82,0.98], $p = 0.013086$), compared to baseline. The relationship between interest rate and loan repayment may differ depending on the term length, with the effect changing for longer-term loans.

Odds Ratio Plot with CIs

Below, we will create a plot that depicts the odds ratio of repayment failure for our final GLM credit risk model, including confidence interval ranges. These estimates are all compared to the baseline, which includes the 36-month loan term, less than one year of employment, and the loan purpose of "car".

```
plot_data <- data.frame(
  Variable = rownames(model_results),
  OddsRatio = model_results$OddsRatio,
  LowerCI = model_results$LowerCI,
  UpperCI = model_results$UpperCI
)

ggplot(plot_data, aes(x = OddsRatio, xmin = LowerCI, xmax = UpperCI, y = Variable)) +
  geom_point(size = 3, color="red") +
  geom_errorbarh(height = 0.2, color = "blue") +
  xlim(c(0, max(plot_data$UpperCI + 1))) +
  theme_minimal() +
  labs(x = "Odds Ratio", y = "Variable") +
  ggtitle("Odds Ratios of Repayment Failure for GLM")
```



Accuracy and Confusion Matrix

Finally, we will create a confusion matrix to demonstrate our final model's performance of classifying whether applicants will default on their payments or not.

```
confusion_glm <- ifelse(prob_logit_val > 0.5, "1", "0")
confusion_glm <- as.factor(confusion_glm)

require(caret)
```

```
## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
cm <- confusionMatrix(data=confusion_glm,
                      reference=val_scaled_data$repay_fail,
                      positive="1")
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6497 1131
##          1   27   28
##
##                Accuracy : 0.8493
##                  95% CI : (0.8411, 0.8572)
##     No Information Rate : 0.8491
##     P-Value [Acc > NIR] : 0.4951
##
##                   Kappa : 0.0329
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.024159
##             Specificity : 0.995861
##          Pos Pred Value : 0.509091
##          Neg Pred Value : 0.851730
##              Prevalence : 0.150853
##          Detection Rate : 0.003644
##    Detection Prevalence : 0.007159
##       Balanced Accuracy : 0.510010
##
##        'Positive' Class : 1
##
```

Included above is a confusion matrix, which depicts our model's true and false positive/negative rates when predicting on unseen (validation) data. This can be used to better understand how well the model predicts new credit applications in terms of how accurately it classifies new applicants. The confusion matrix shows that out of 7,683 applicants, 6,497 of them will not default and have been accurately classified as so, and 28 of them will default and have been accurately classified as so. However, using our model, 1,131 applicants are falsely classified as non-defaulters when in fact, they will default, and 27 applicants are falsely classified as defaulters when they will not default.

Part 1 Discussion

In summary, the goal of the first part of our analysis was to rebuild a credit risk model from the ground up. We began with an exploration of the available data, which made us aware of the rather large ranges of some of our numeric variables. Thus, we chose to standardise all of the numeric variables in the data so that they can be more easily and directly compared and that those variables with larger ranges would not be favoured by our model. We also converted the categorical variables within the data to factors to prepare for model building.

We conducted our own loan application and credit risk research, and the results of this, in conjunction with our exploratory analyses, ultimately led us to select an initial list of 16 variables: annual_inc, emp_length, verification_status, dti, credit_age_yrs, home_ownership, total_acc, revol_bal, revol_util, int_rate, term, purpose, loan_amnt, inq_last_6mths, delinq_2yrs, pub_rec. We then decided to run stepwiseAIC to extract a good model formula from these variables. StepwiseAIC provided us with the following model formula:

repay_fail ~ term + int_rate + emp_length + annual_inc + purpose + inq_last_6mths + pub_rec + revol_bal + revol_util

We then wanted to consider whether interactions between any of the variables were important or improved the model. So, we tested a few interactions between some of the variables. However, we ultimately did not want our model to include too many interaction terms to avoid unnecessary complexity, which would introduce more challenges in interpretations and later applications. So, we opted to only keep the interaction between interest rate and loan term, as this made the most logical sense to us. This is because an interest rate on its own does not mean much until it is combined with either a loan amount or a loan term, as an interest rate is applied in relation to an amount over a period of time. In this case, we chose to select loan term, as loan amount was not included in the formula provided by stepwiseAIC. This led us to our final model formula:

repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate

We justified arriving at this model by conducting χ^2-tests between different tested models to determine whether new models with different variables and interactions were significantly better at explaining the variability in the data. We then took a look at the goodness-of-fit of our selected model, and we were pleased to find that our model fit the data reasonably well. We produced residuals plots which did not indicate any deviations from normality, overdispersion, or outliers in the model residuals. In other words, these plots demonstrated that our model was a good fit for the data and did a relatively good job of explaining the variation in the data. We also compared the AIC values of our final chosen model with the other models we fit, and the chosen model has the lowest AIC value.

Just to be sure, we also tested different link functions with the same final model formula. We tested both a probit and complementary log-log link function. The probit model, while demonstrating a lower AIC value than the initial logit model, did not do as well in terms of prediction. Ultimately, we want our model to not only fit the data well, but also to do a good job of predicting whether applicants will default or not. The complementary log-log model had a higher AIC value than the initial logit model, and it did not perform as well on seen data. Thus, we ultimately selected the logit link function as the best link function for our final model.

We produced receiver operating characteristic (ROC) curves and calculated Gini scores to test our model's performance. Our goal was to build a model that improved the ROC curve from those of the bank's previous models, so that the curve reached more towards the upper-left-hand corner of the graph area. This would mean that the model did a better job at correctly distinguishing between the classes of repayment (repay_fail == 1 vs repay_fail == 0). The Gini score is a numeric representation of the area under the ROC curve, which also indicates a model's classification performance.

In terms of our final model's performance, it does much better than the credit risk model the bank was previously using. While the previous models yielded a $Gini = 0.114$ on the training (seen) dataset and a $Gini = 0.110$ on the validation (unseen) dataset, our models produced a $Gini = 0.410$ on the training dataset and a $Gini = 0.395$ on the validation dataset. This demonstrates that our model both performs well on seen data and also demonstrates good predictive performance on new, unseen data.

Thus, in terms of answering the first question we set out to answer with our analysis (see below):

1. How does this new model perform compared to the one used previously? How can it be expected to perform on new loan applications?

We can say that our new model outperforms the one used previously. It does a relatively good job on new loan applications. As our confusion matrix demonstrated, out of 7,683 applicants included in the training dataset, 6,497 of them will not default and our model has accurately classified them as so, and 28 of them will default and have been accurately classified as so. However, our model falsely classified 1,131 applicants as non-defaulters and 27 applicants as defaulters. While there is still certainly room for improvement, this model is still performing much better than the previous model, so we have successfully addressed question 1.

In terms of the second question of our analysis (see below):

2. What are the important variables in this model and how do they compare to variables that are traditionally important for predicting credit risk in the banking sector?

Our model summary demonstrated that the important (and in this case, important means that the variables demonstrated a significant effect on the risk of loan repayment) variables for predicting credit risk are as follows:

- Interest rate
- Unspecified employment length (whether information is missing or the individual is unemployed)
- Annual income
- Loan purposes of medical, moving, small business, vacation or other
- Number of inquiries in the last 6 months
- Number of derogatory public records
- Revolving balance and utilization
- 60-month loan term
- Interest rate * 60-month loan term

Only two of the important covariates in our model were associated with a significant decrease in the risk of credit default: annual income, which was associated with a decreased risk of default by a factor of 0.71 (95% CI: [0.67,0.75], $p < 2e-16$), and the interaction between interest rate and the 60-month loan term, which was associated with a decreased risk of default by a factor of 0.90 (95% CI: [0.82,0.98], $p = 0.013086$). All other listed important covariates are associated with an increase in the risk of credit default. Of these, the covariates associated with the larger increased risks of credit default are: the loan purposes of small business, with an increased risk of default by a factor of 2.78 (95% CI: [2.15,3.59], $p = 5.80e-15$), and medical, with an increased risk of default by a factor of 1.83 (95% CI: [1.30,2.57], $p = 0.000502$). Applicants with a value of "n/a" for employment length, whether that means they are unemployed or simply did not provide this information, also had a relatively large increased risk of default compared to other covariates, by a factor of 1.73 (95% CI: [1.37,2.19], $p = 4.11e-06$).

We found the relationship between interest rate and loan term on loan default to be interesting. While on their own, both interest rate and the 60-month loan term were associated with an increase in the odds that an applicant will fail to repay, the interaction between these two variables was associated with a decrease in the odds of repayment failure. This indicates that the relationship between interest rate and loan repayment may differ depending on the term length, with the effect changing for longer-term loans.

We believe the important variables in our model generally make sense when considering the variables that are traditionally important when predicting credit risk in the banking sector, which we determined through conducting some research on our own. For example, in the banking sector, the "5 Cs" have been said to be important in determining an individual's credit risk, and these include: capacity, capital, conditions, collateral, and capital. Our final model's important variables include annual income, employment length, revolving balances, and revolving utilization, which are all related to an individual's capacity to repay a loan. Term may also play a role in capacity, as the term can be compared against an individual's income and employment stability, for example, to determine their capacity to repay during the term period. Interest rate also interacts with the term, and taken together, these can indicate to the bank how well an individual should be able to repay their loan. Interest rate may also indicate current market conditions to the bank, and the number of inquiries in the last 6 months as well as derogatory public records for an individual both play a role in determining character to the bank.

We were somewhat initially surprised by some of the significant loan purposes. We did not initially find much in the research that indicated the effect of loan purpose on application approval, but it does seem to make sense that expenses such as medical bills, particularly if medical expenses are ongoing and high, and small businesses, especially depending on the economy, might make it harder for an individual to repay a loan on time. Perhaps the purpose of other is associated with an increase in risk of repayment failure, as individuals do not necessarily demonstrate that the purpose is meant for the short term and temporary. Moving surprised us, but perhaps if moving expenses are so astronomical and individuals are moving a long

distance and getting set up with new jobs in new cities, it may take a while to get back on their feet, thus potentially affecting their ability to repay on time during the transition period. Finally, the purpose of vacations was initially surprising, but if individuals are taking out loans to pay for their vacations, it potentially means that they do not have the funds in the first place to pay for vacation. Thus, they are spending money without necessarily showing that they have the means to pay it back.

Something that also surprised us was the factors that did not appear to be significant when building and assessing our new credit risk model. In particular, we were surprised that stepwiseAIC did not indicate debt to income ratio, loan amount, home ownership, or an individual's credit age in years to be part of the best model formula. We also thought that verification status might play an important role to the bank, so that they could have some assurance that an individual's reported annual income was actually correct. Furthermore, the number of 30+ days past-due incidences of delinquency for the past 2 years was not indicated by stepwiseAIC to be an important factor, though we initially thought it would be. For this model, we decided to build upon and tune the initial model provided by stepwiseAIC, and thus, we did not include these variables. However, we were surprised, as some, such as debt to income ratio, home ownership, and credit age, are variables traditionally used in the banking sector to help determine an individual's credit risk.

Now, we will move on to the second part of our analysis and extend the final model we have presented above.

Extended Credit Risk Model

So far, we've been able to address management's concerns regarding the previous credit risk model and have presented a new model that performs much better. We have also identified some of the important variables for predicting credit risk and have compared these to the factors which are traditionally important within the banking sector. We are now going to address the second part of management's concerns, which includes using an extended version of the initial dataset. We will set out to answer whether accounting for variation in trends over jurisdiction (location) or time changes or even improves performance benchmarks. Management has previously never considered whether credit risk changes between different states or over time, so we will investigate this. We will choose to use the same basic model formula as that in our new credit risk model, but we will add in the modification of accounting for variations in location and time.

We'll first take a look at the structure of the extended dataset.

```
str(extended)
```

```
## 'data.frame':    38419 obs. of  22 variables:
##  $ loan_amnt          : int  2500 5000 7000 2000 3600 8000 6000 25600 19750 6250 ...
##  $ term               : chr  "36 months" "36 months" "36 months" "36 months" ...
##  $ int_rate           : num  13.98 15.95 9.91 5.42 10.25 ...
##  $ emp_length         : chr  "4 years" "4 years" "10+ years" "10+ years" ...
##  $ home_ownership     : chr  "RENT" "RENT" "MORTGAGE" "RENT" ...
##  $ annual_inc         : num  20004 59000 53796 30000 675048 ...
##  $ verification_status: chr  "Not Verified" "Not Verified" "Not Verified" "Not Verified" ...
##  $ issue_d            : chr  "Jul-10" "Jun-10" "Sep-11" "Sep-11" ...
##  $ purpose            : chr  "other" "debt_consolidation" "other" "debt_consolidation" ...
##  $ zip_code           : chr  "487xx" "115xx" "751xx" "112xx" ...
##  $ addr_state         : chr  "MI" "NY" "TX" "NY" ...
##  $ dti                : num  19.86 19.57 10.8 3.6 1.55 ...
##  $ delinq_2yrs        : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ earliest_cr_line   : chr  "Aug-05" "Apr-94" "Mar-98" "Jan-75" ...
##  $ inq_last_6mths     : int  5 1 3 0 4 0 0 1 0 0 ...
##  $ open_acc           : int  7 7 7 7 8 12 5 16 15 2 ...
##  $ pub_rec            : int  0 0 0 0 0 0 0 0 0 1 ...
```

```
##  $ revol_bal          : int  981 18773 3269 0 0 4182 5864 33021 21544 0 ...
##  $ revol_util         : num  0.213 0.999 0.472 0 0 0.136 0.477 0.708 0.987 0.0846 ...
##  $ total_acc          : int  10 15 20 15 25 49 9 32 44 15 ...
##  $ repay_fail         : int  0 1 0 0 0 0 0 0 0 1 ...
##  $ credit_age_yrs     : num  4.93 16.22 13.55 36.79 12.04 ...
```

Data Preparation

Next, we'll need to once again convert the relevant variables to factors. This time, this will include the new variables of zip_code, addr_state, and earliest_cr_line, in addition to the variables converted in the first part of this analysis, as these will be used to account for variations in location (zip_code and addr_state) and time (earliest_cr_line):

```
extended$term <- as.factor(extended$term)
extended$emp_length <- as.factor(extended$emp_length)
extended$home_ownership <- as.factor(extended$home_ownership)
extended$verification_status <- as.factor(extended$verification_status)
extended$purpose <- as.factor(extended$purpose)
extended$repay_fail <- as.factor(extended$repay_fail)
extended$zip_code <- as.factor(extended$zip_code)
extended$addr_state <- as.factor(extended$addr_state)
extended$earliest_cr_line <- as.factor(extended$earliest_cr_line)

# create new location variable, which will nest zip codes within states
extended <- within(extended, location <- factor(as.factor(addr_state):zip_code))
```

```
# location
nlevels(extended$zip_code)
```

```
## [1] 831
```

```
nlevels(extended$addr_state)
```

```
## [1] 50
```

```
nlevels(extended$location) # includes zip nested within each state
```

```
## [1] 905
```

```
# time
nlevels(extended$earliest_cr_line)
```

```
## [1] 528
```

Variable Standardisation

Once again, we need to standardise the numeric variables in the dataset:

```r
numeric_columns <- sapply(extended, is.numeric)  # Identify numeric columns

# Scale the numeric columns while keeping column names
extended[, numeric_columns] <- scale(extended[, numeric_columns])

extended_scaled <- extended %>%
  mutate_if(is.numeric, scale)
```

Fit Models

We will begin to fit a GLMM to address this part of our analysis. We will use the same general model formula in terms of the inclusion of the same covariates. What will differ is the inclusion of random effects in this extended model. We will include random effects to account for variation in time and applicant location. We begin by fitting a model with just the applicant's state as a random effect:

```r
fit_var_1 <- glmer(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
                     inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate +
                     (1|addr_state), data = extended_scaled, family = "binomial")
```

View first extended model summary:

```r
summary(fit_var_1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
##     pub_rec + revol_bal + revol_util + term * int_rate + (1 |      addr_state)
##    Data: extended_scaled
##
##      AIC      BIC   logLik deviance df.resid
##  30131.5  30422.4 -15031.7  30063.5    38385
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.290 -0.450 -0.337 -0.236 96.815
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  addr_state (Intercept) 0.02784  0.1668
## Number of obs: 38419, groups:  addr_state, 50
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -2.334947   0.100813 -23.161  < 2e-16 ***
## int_rate                   0.452134   0.024061  18.791  < 2e-16 ***
## emp_length1 year           0.009103   0.066772   0.136 0.891558
## emp_length10+ years        0.115010   0.054340   2.116 0.034304 *
## emp_length2 years         -0.106259   0.063234  -1.680 0.092878 .
## emp_length3 years          0.008816   0.063782   0.138 0.890070
```

```
## emp_length4 years          -0.054990   0.067339  -0.817 0.414146
## emp_length5 years           0.035343   0.067577   0.523 0.600968
## emp_length6 years          -0.008718   0.076863  -0.113 0.909692
## emp_length7 years           0.009439   0.082783   0.114 0.909223
## emp_length8 years           0.033336   0.088499   0.377 0.706411
## emp_length9 years          -0.084498   0.097777  -0.864 0.387485
## emp_lengthn/a               0.538676   0.092353   5.833 5.45e-09 ***
## annual_inc                 -0.410744   0.029773 -13.796  < 2e-16 ***
## purposecredit_card         -0.002383   0.097934  -0.024 0.980584
## purposedebt_consolidation   0.210013   0.088988   2.360 0.018274 *
## purposeeducational          0.703161   0.156264   4.500 6.80e-06 ***
## purposehome_improvement     0.210364   0.103369   2.035 0.041842 *
## purposehouse                0.231531   0.168311   1.376 0.168942
## purposemajor_purchase       0.014451   0.113421   0.127 0.898613
## purposemedical              0.482738   0.136558   3.535 0.000408 ***
## purposemoving               0.449262   0.146557   3.065 0.002173 **
## purposeother                0.444964   0.096676   4.603 4.17e-06 ***
## purposerenewable_energy     0.580666   0.299382   1.940 0.052435 .
## purposesmall_business       0.974426   0.102726   9.486  < 2e-16 ***
## purposevacation             0.354715   0.176941   2.005 0.044994 *
## purposewedding             -0.007788   0.138788  -0.056 0.955252
## inq_last_6mths              0.220363   0.013655  16.138  < 2e-16 ***
## pub_rec                     0.066326   0.012925   5.131 2.88e-07 ***
## revol_bal                   0.096027   0.016437   5.842 5.15e-09 ***
## revol_util                  0.103142   0.017827   5.786 7.22e-09 ***
## term60 months               0.526546   0.040322  13.059  < 2e-16 ***
## int_rate:term60 months     -0.124492   0.034389  -3.620 0.000294 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 33 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

As can be seen above, this model has an AIC value of 30131.5. Next, we'll add in another random effect of time, which is included as the variable "earliest_cr_line":

```
# fit_var_2 <- glmer(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
#                      inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate +
#                      (1|addr_state) + (1|earliest_cr_line), data = extended_scaled, family = "binomia
```

Adding in time as a random effect caused a failed to converge warning. Thus, we will now increase the tolerance of this model to see if this helps:

```
fit_var_3 <- glmer(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
                     inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate +
                     (1|addr_state) + (1|earliest_cr_line), data = extended_scaled, family = "binomial"
                   control = glmerControl(check.conv.grad = .makeCC("warning", tol = 3e-3, relTol = NULL
```

The model ran successfully, so we'll take a look at the summary:

```r
summary(fit_var_3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
##     pub_rec + revol_bal + revol_util + term * int_rate + (1 |
##     addr_state) + (1 | earliest_cr_line)
##    Data: extended_scaled
## Control: glmerControl(check.conv.grad = .makeCC("warning", tol = 0.003,
##     relTol = NULL))
##
##      AIC      BIC   logLik deviance df.resid
##  30126.5  30425.9 -15028.2  30056.5    38384
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.313 -0.450 -0.336 -0.235 97.929
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  earliest_cr_line (Intercept) 0.01412  0.1188
##  addr_state       (Intercept) 0.02792  0.1671
## Number of obs: 38419, groups:  earliest_cr_line, 528; addr_state, 50
##
## Fixed effects:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -2.336880   0.101223 -23.087  < 2e-16 ***
## int_rate                   0.452207   0.024125  18.744  < 2e-16 ***
## emp_length1 year           0.009184   0.066839   0.137 0.890714
## emp_length10+ years        0.108643   0.054587   1.990 0.046560 *
## emp_length2 years         -0.107684   0.063310  -1.701 0.088964 .
## emp_length3 years          0.010073   0.063853   0.158 0.874650
## emp_length4 years         -0.058248   0.067428  -0.864 0.387666
## emp_length5 years          0.035474   0.067659   0.524 0.600069
## emp_length6 years         -0.011585   0.076969  -0.151 0.880363
## emp_length7 years          0.003067   0.082927   0.037 0.970494
## emp_length8 years          0.027436   0.088638   0.310 0.756920
## emp_length9 years         -0.088871   0.097951  -0.907 0.364246
## emp_lengthn/a              0.535157   0.092577   5.781 7.44e-09 ***
## annual_inc                -0.412429   0.029918 -13.786  < 2e-16 ***
## purposecredit_card        -0.001321   0.098019  -0.013 0.989244
## purposedebt_consolidation  0.211301   0.089085   2.372 0.017697 *
## purposeeducational         0.707953   0.156491   4.524 6.07e-06 ***
## purposehome_improvement    0.212034   0.103471   2.049 0.040442 *
## purposehouse               0.238985   0.168571   1.418 0.156276
## purposemajor_purchase      0.015611   0.113489   0.138 0.890589
## purposemedical             0.483430   0.136666   3.537 0.000404 ***
## purposemoving              0.452355   0.146811   3.081 0.002062 **
## purposeother               0.445638   0.096762   4.606 4.11e-06 ***
## purposerenewable_energy    0.568627   0.299619   1.898 0.057718 .
## purposesmall_business      0.978166   0.102870   9.509  < 2e-16 ***
```

```
## purposevacation            0.353679   0.177158   1.996 0.045890 *
## purposewedding             0.001196   0.138988   0.009 0.993134
## inq_last_6mths             0.220506   0.013676  16.124  < 2e-16 ***
## pub_rec                    0.066125   0.012958   5.103 3.35e-07 ***
## revol_bal                  0.095931   0.016473   5.824 5.76e-09 ***
## revol_util                 0.104121   0.017858   5.831 5.52e-09 ***
## term60 months              0.527379   0.040397  13.055  < 2e-16 ***
## int_rate:term60 months    -0.125166   0.034438  -3.635 0.000278 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 33 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

This model has a AIC value of 30126.5, which is lower than that of the first extended model (fit_var_1). This indicates the new model with both added random effects of state and time is a better fit.

Now, we'll add in one more random effect, which is zip code. Once again, we will fit this model with an increased tolerance, compared to the default tolerance:

```
fit_var_4 <- glmer(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
                     inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate +
                     (1|addr_state) + (1|earliest_cr_line) + (1|zip_code),
                   data = extended_scaled, family = "binomial",
                   control = glmerControl(check.conv.grad = .makeCC("warning", tol = 9e-3, relTol = NULL
```

Now we can view this model summary:

```
summary(fit_var_4)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
##     pub_rec + revol_bal + revol_util + term * int_rate + (1 |
##     addr_state) + (1 | earliest_cr_line) + (1 | zip_code)
##    Data: extended_scaled
## Control: glmerControl(check.conv.grad = .makeCC("warning", tol = 0.009,
##     relTol = NULL))
##
##      AIC      BIC   logLik deviance df.resid
##  30115.5  30423.5 -15021.7  30043.5    38383
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.387 -0.449 -0.335 -0.234 97.879
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  zip_code         (Intercept) 0.02396  0.1548
```

```
##  earliest_cr_line (Intercept) 0.01397  0.1182
##  addr_state       (Intercept) 0.01979  0.1407
## Number of obs: 38419, groups:
## zip_code, 831; earliest_cr_line, 528; addr_state, 50
##
## Fixed effects:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.3330139  0.1004848 -23.218  < 2e-16 ***
## int_rate                    0.4523067  0.0241462  18.732  < 2e-16 ***
## emp_length1 year            0.0090889  0.0669321   0.136 0.891985
## emp_length10+ years         0.0979225  0.0547802   1.788 0.073848 .
## emp_length2 years          -0.1115768  0.0633810  -1.760 0.078338 .
## emp_length3 years           0.0079019  0.0639373   0.124 0.901641
## emp_length4 years          -0.0617825  0.0675330  -0.915 0.360271
## emp_length5 years           0.0281216  0.0677861   0.415 0.678246
## emp_length6 years          -0.0189982  0.0770844  -0.246 0.805326
## emp_length7 years          -0.0056864  0.0830971  -0.068 0.945443
## emp_length8 years           0.0175383  0.0888293   0.197 0.843485
## emp_length9 years          -0.0936973  0.0980543  -0.956 0.339292
## emp_lengthn/a               0.5254195  0.0928050   5.662 1.50e-08 ***
## annual_inc                 -0.4103586  0.0300176 -13.671  < 2e-16 ***
## purposecredit_card          0.0008323  0.0980152   0.008 0.993225
## purposedebt_consolidation   0.2126276  0.0890609   2.387 0.016966 *
## purposeeducational          0.7043029  0.1568087   4.491 7.07e-06 ***
## purposehome_improvement     0.2069884  0.1034732   2.000 0.045456 *
## purposehouse                0.2390422  0.1686342   1.418 0.156331
## purposemajor_purchase       0.0169273  0.1134855   0.149 0.881428
## purposemedical              0.4793397  0.1368244   3.503 0.000459 ***
## purposemoving               0.4593447  0.1470129   3.125 0.001781 **
## purposeother                0.4468827  0.0967649   4.618 3.87e-06 ***
## purposerenewable_energy     0.5777967  0.3001303   1.925 0.054210 .
## purposesmall_business       0.9806033  0.1029394   9.526  < 2e-16 ***
## purposevacation             0.3611436  0.1773067   2.037 0.041667 *
## purposewedding              0.0018225  0.1389716   0.013 0.989537
## inq_last_6mths              0.2195093  0.0137086  16.013  < 2e-16 ***
## pub_rec                     0.0648712  0.0130011   4.990 6.05e-07 ***
## revol_bal                   0.0951630  0.0165181   5.761 8.35e-09 ***
## revol_util                  0.1055772  0.0178935   5.900 3.63e-09 ***
## term60 months               0.5242338  0.0404704  12.954  < 2e-16 ***
## int_rate:term60 months     -0.1245569  0.0344879  -3.612 0.000304 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 33 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Now, we'll try including just two random effects: earliest credit line (time), and then the new location factor we created earlier, which nests zip codes within each state, rather than having to include zip code and state separately:

```
fit_var_5 <- glmer(repay_fail ~ int_rate + emp_length + annual_inc + purpose +
                     inq_last_6mths + pub_rec + revol_bal + revol_util + term * int_rate +
                     (1|location) + (1|earliest_cr_line),
                   data = extended_scaled, family = "binomial",
                   control = glmerControl(optimizer = "Nelder_Mead",
                                          check.conv.grad = .makeCC("warning", tol = 8e-3, relTol = NULL
```

View model summary:

```
summary(fit_var_5)
```

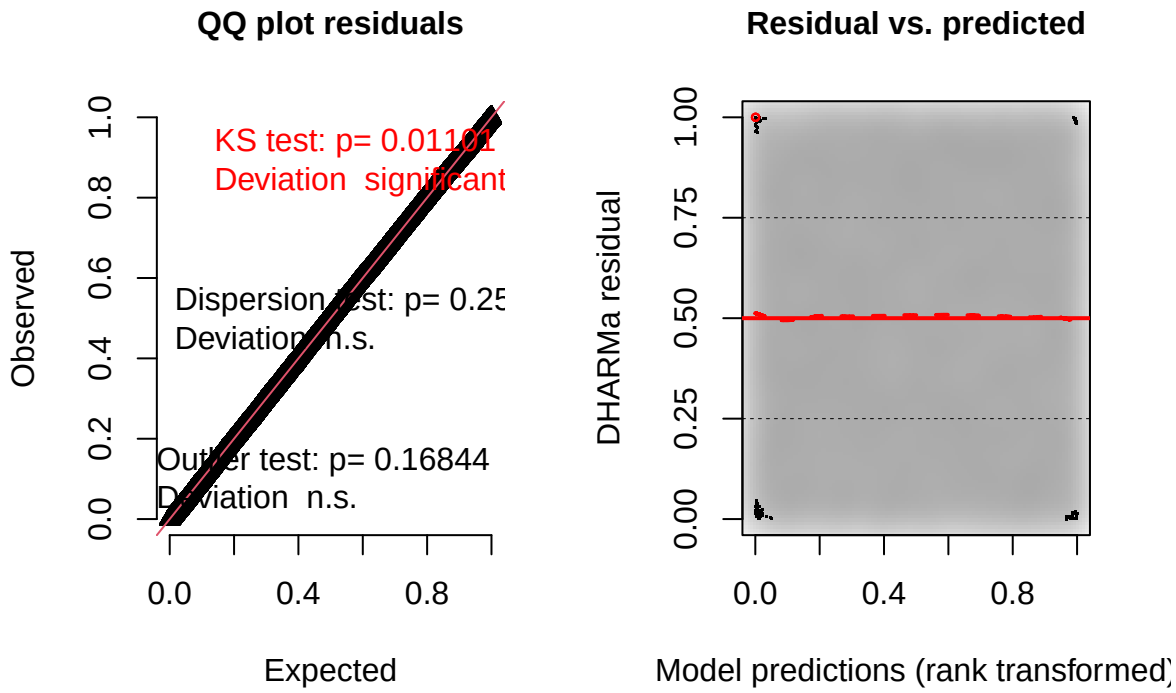
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
##     pub_rec + revol_bal + revol_util + term * int_rate + (1 |
##     location) + (1 | earliest_cr_line)
##    Data: extended_scaled
## Control:
## glmerControl(optimizer = "Nelder_Mead", check.conv.grad = .makeCC("warning",
##     tol = 0.008, relTol = NULL))
##
##      AIC      BIC   logLik deviance df.resid
##  30142.1  30441.5 -15036.0  30072.1    38384
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.468 -0.448 -0.334 -0.234 92.042
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  location         (Intercept) 0.05114  0.2261
##  earliest_cr_line (Intercept) 0.01414  0.1189
## Number of obs: 38419, groups:  location, 905; earliest_cr_line, 528
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -2.304331   0.097729 -23.579  < 2e-16 ***
## int_rate                   0.453831   0.024166  18.780  < 2e-16 ***
## emp_length1 year           0.008895   0.067063   0.133 0.894478
## emp_length10+ years        0.090262   0.054878   1.645 0.100016
## emp_length2 years         -0.112380   0.063486  -1.770 0.076701 .
## emp_length3 years          0.009179   0.064043   0.143 0.886038
## emp_length4 years         -0.060151   0.067660  -0.889 0.373996
## emp_length5 years          0.025726   0.067912   0.379 0.704829
## emp_length6 years         -0.019520   0.077204  -0.253 0.800391
## emp_length7 years         -0.010774   0.083236  -0.129 0.897015
## emp_length8 years          0.011029   0.088992   0.124 0.901372
## emp_length9 years         -0.093617   0.098251  -0.953 0.340674
## emp_lengthn/a              0.524223   0.092945   5.640 1.70e-08 ***
## annual_inc                -0.405295   0.029987 -13.516  < 2e-16 ***
## purposecredit_card         0.004153   0.098212   0.042 0.966269
```

```
## purposedebt_consolidation  0.214997   0.089259   2.409 0.016010 *
## purposeeducational         0.707194   0.157108   4.501 6.75e-06 ***
## purposehome_improvement    0.205037   0.103686   1.977 0.047987 *
## purposehouse               0.253393   0.168896   1.500 0.133540
## purposemajor_purchase      0.017726   0.113695   0.156 0.876106
## purposemedical             0.482956   0.137059   3.524 0.000426 ***
## purposemoving              0.467011   0.147359   3.169 0.001529 **
## purposeother               0.448960   0.096968   4.630 3.66e-06 ***
## purposerenewable_energy    0.596526   0.300442   1.985 0.047089 *
## purposesmall_business      0.986201   0.103149   9.561  < 2e-16 ***
## purposevacation            0.374597   0.177685   2.108 0.035013 *
## purposewedding             0.004378   0.139216   0.031 0.974910
## inq_last_6mths             0.213143   0.013643  15.623  < 2e-16 ***
## pub_rec                    0.062916   0.013013   4.835 1.33e-06 ***
## revol_bal                  0.094470   0.016550   5.708 1.14e-08 ***
## revol_util                 0.108054   0.017920   6.030 1.64e-09 ***
## term60 months              0.517039   0.040505  12.765  < 2e-16 ***
## int_rate:term60 months    -0.124985   0.034533  -3.619 0.000295 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 33 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Goodness-of-Fit

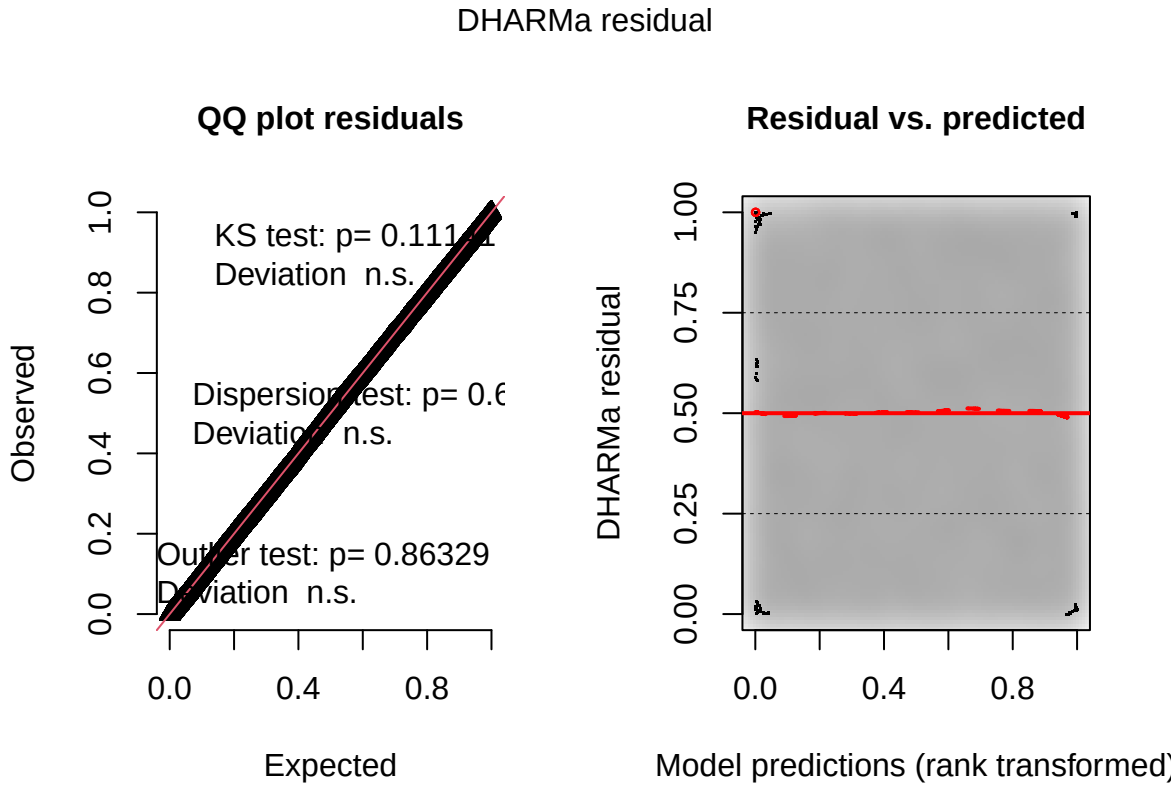
Now, before we begin to interpret these extended models and analyse their performance benchmarks, we will first analyse the goodness-of-fit of the model which has all 3 variables as separate random effects (time, state, and zip code) and of the model which has 2 variables as random effects (time and location, which consists of zip codes nested within states). We'll begin with the former model.

```
# simulate residuals from the model:
res_fit_var_4 = simulateResiduals(fit_var_4)

# plot observed quantile versus expected quantile to assess distribution fit, and predicted value versu
plot(res_fit_var_4)
```



As can be seen above, while this model does not indicate overdispersion or significant outliers in the residuals, there appears to be significant deviation from normality in the residuals, as seen by the significant KS test results on the left. Let's now consider the model with just 2 random effects: time and location, in which location consists of zip codes nested within states.

```
# simulate residuals from the model:
res_fit_var_5 = simulateResiduals(fit_var_5)

# plot observed quantile versus expected quantile to assess distribution fit, and predicted value versu
plot(res_fit_var_5)
```



As can be seen in the above plots, this model that includes location as a random effect of zip codes nested within states appears to be a better fit for the data. Now, there is no indication of significant deviations from normality, nor overdispersion or outliers in the residuals. This indicates to us that this model does a relatively good job of explaining the variation in the data. Thus, we will select this model to further analyse.

Analyse Performance Benchmarks

Now, we can move on to analyse the selected model's performance benchmarks and can compare these to the final, new credit risk model we obtained in the first part of this analysis. We begin by creating a ROC curve and computing a Gini score for the chosen extended model.

```
# ROC curve on scaled extended data for fit_var_5
prob_ext=predict(fit_var_5,type=c("response"))
g <- roc(extended_scaled$repay_fail ~ prob_ext)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

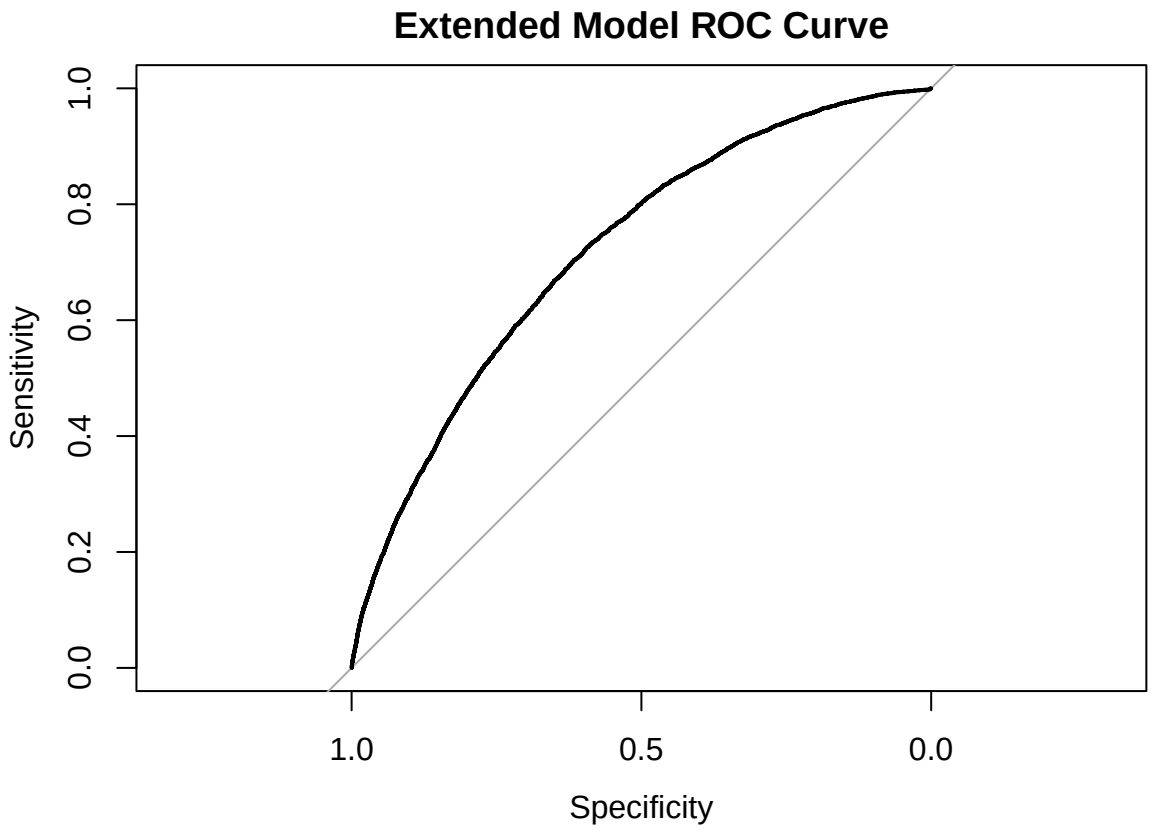
```
AUC <- g$auc
Gini <- 2*(AUC - 1/2)
AUC
```

```
## Area under the curve: 0.7183
```

```
Gini
```

```
## [1] 0.4365468
```

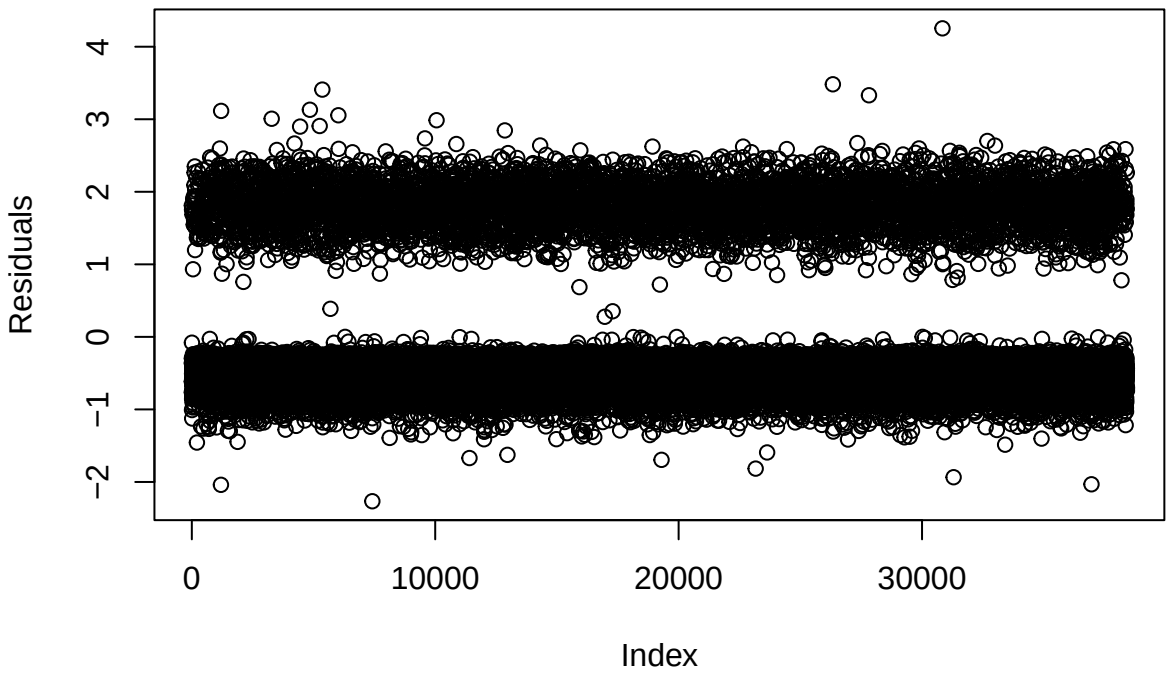
```
plot(g, main = "Extended Model ROC Curve")
```



As can be seen in the above plot, the extended model has a $AUC = 0.7183$ and $Gini = 0.437$. This is an improved score from that of the model from the first part of our analysis, which had a $AUC = 0.7052$ and $Gini = 0.410$. It is important to note that for the above plot, we were only able to assess the performance of our model using "seen" data. Thus, the metrics we are comparing from the part 1 model are also those obtained using "seen" data (called training dataset). Based on our obtained AUC and Gini scores, we can say that our extended model does result in improved performance benchmarks, as this model is better able to predict loan repayment failure. In other words, accounting for variations in time and location improves our model's ability to predict whether applicants will default on their loans or not.

Now, we can further analyse the residuals of this extended model to understand the effect of variations in time and location on credit risk.

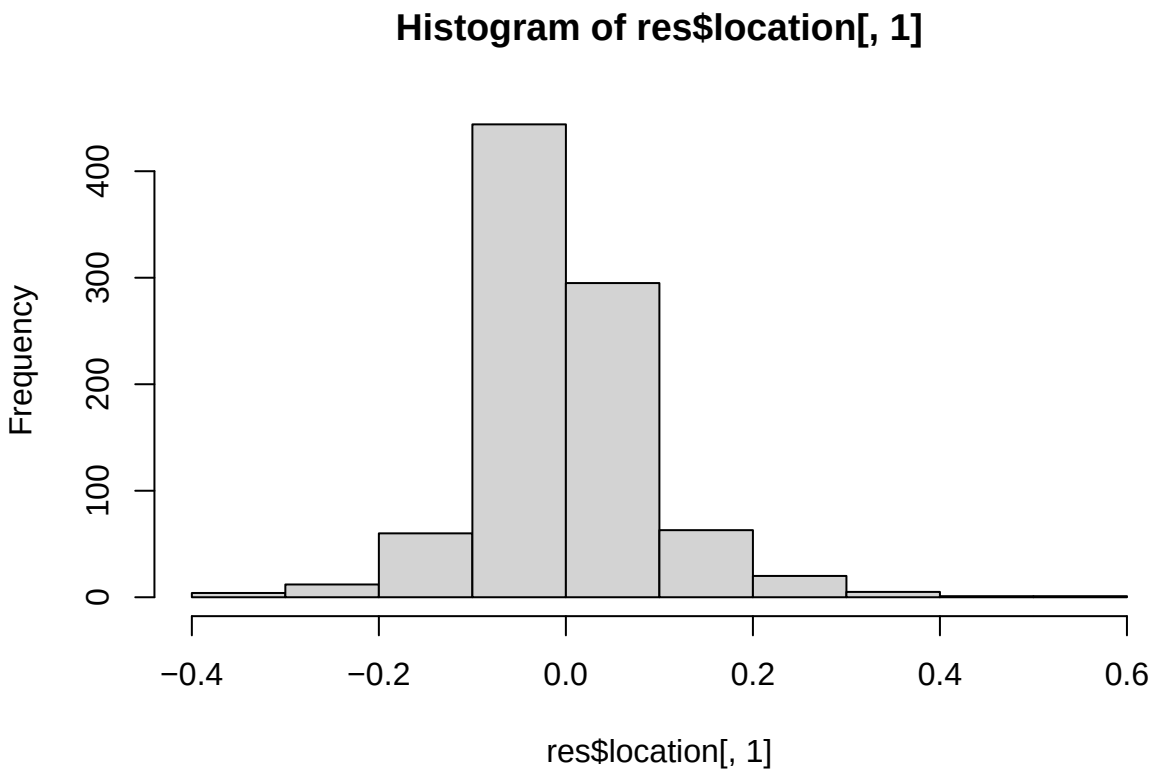
```
plot(residuals(fit_var_5),
     ylab = "Residuals")
```



The above plot is roughly what we would expect to see in terms of the distribution of residuals for a normally distributed binary variable (repay_fail is comprised of just 0s and 1s to indicate whether an applicant failed to repay or not). Now we can further assess the distribution of the random effects:

```
res <- ranef(fit_var_5)

hist(res$earliest_cr_line[,1])
```


```
hist(res$location[,1])
```



As can be seen in the above plots, the residuals for the random effects of both time and location appear to be roughly normally distributed, which indicates to us that the chosen model is a relatively good fit for the data.

Calculate Odds Ratio and CIs

As the selected model was fit with a logit link function, we can exponentiate the estimates for easier interpretations of the significant effects. We will also calculate confidence intervals and present everything in a table for review and further interpretation.

```
summary_fit5 <- summary(fit_var_5)

parameter_estimates2 <- summary_fit5$coef[, "Estimate"]
standard_errors2 <- summary_fit5$coef[, "Std. Error"]
p_values2 <- summary_fit5$coef[, "Pr(>|z|)"]

# Calculate the odds ratios by exponentiating the parameter estimate values
odds_ratios2 <- exp(parameter_estimates2)

# Calculate the confidence intervals for the odds ratios
lower_ci2 <- exp(parameter_estimates2 - 1.96 * standard_errors2)  # 1.96 corresponds to a 95% confidenc
upper_ci2 <- exp(parameter_estimates2 + 1.96 * standard_errors2)

model_results2 <- data.frame(
  Estimate = parameter_estimates2,
```

```
  OddsRatio = odds_ratios2,
  LowerCI = lower_ci2,
  UpperCI = upper_ci2,
  p_value = p_values2
)

# Display results in a table
knitr::kable(
  model_results2,
  col.names = c("Estimate", "Odds Ratio", "Lower Bound (2.5%)", "Upper Bound (97.5%)", "p-value")
)
```

	Estimate	Odds Ratio	Lower Bound (2.5%)	Upper Bound (97.5%)	p-value
(Intercept)	-2.3043311	0.0998255	0.0824239	0.1209011	0.0000000
int_rate	0.4538310	1.5743319	1.5015030	1.6506933	0.0000000
emp_length1 year	0.0088953	1.0089350	0.8846628	1.1506641	0.8944778
emp_length10+ years	0.0902617	1.0944607	0.9828501	1.2187456	0.1000157
emp_length2 years	-0.1123801	0.8937045	0.7891392	1.0121253	0.0767009
emp_length3 years	0.0091787	1.0092209	0.8901672	1.1441972	0.8860376
emp_length4 years	-0.0601509	0.9416224	0.8246757	1.0751533	0.3739959
emp_length5 years	0.0257256	1.0260594	0.8981832	1.1721415	0.7048288
emp_length6 years	-0.0195202	0.9806690	0.8429568	1.1408791	0.8003914
emp_length7 years	-0.0107735	0.9892843	0.8403676	1.1645897	0.8970147
emp_length8 years	0.0110286	1.0110896	0.8492549	1.2037638	0.9013725
emp_length9 years	-0.0936172	0.9106312	0.7511201	1.1040168	0.3406736
emp_lengthn/a	0.5242234	1.6891465	1.4078333	2.0266718	0.0000000
annual_inc	-0.4052951	0.6667800	0.6287201	0.7071439	0.0000000
purposecredit_card	0.0041532	1.0041618	0.8283311	1.2173163	0.9662689
purposedebt_consolidation	0.2149973	1.2398585	1.0408627	1.4768991	0.0160097
purposeeducational	0.7071939	2.0282917	1.4907239	2.7597111	0.0000068
purposehome_improvement	0.2050367	1.2275702	1.0018144	1.5041993	0.0479871
purposehouse	0.2533927	1.2883892	0.9252940	1.7939666	0.1335400
purposemajor_purchase	0.0177259	1.0178839	0.8145519	1.2719726	0.8761063
purposemedical	0.4829562	1.6208590	1.2390193	2.1203736	0.0004256
purposemoving	0.4670106	1.5952182	1.1950486	2.1293873	0.0015286
purposeother	0.4489604	1.5666826	1.2955072	1.8946205	0.0000037
purposerenewable_energy	0.5965259	1.8157996	1.0076896	3.2719682	0.0470893
purposesmall_business	0.9862006	2.6810287	2.1902777	3.2817368	0.0000000
purposevacation	0.3745970	1.4544051	1.0266837	2.0603174	0.0350132
purposewedding	0.0043785	1.0043881	0.7645372	1.3194850	0.9749100
inq_last_6mths	0.2131425	1.2375610	1.2049073	1.2710997	0.0000000
pub_rec	0.0629159	1.0649373	1.0381200	1.0924473	0.0000013
revol_bal	0.0944704	1.0990766	1.0639978	1.1353120	0.0000000
revol_util	0.1080543	1.1141082	1.0756566	1.1539343	0.0000000
term60 months	0.5170389	1.6770544	1.5490627	1.8156214	0.0000000
int_rate:term60 months	-0.1249849	0.8825103	0.8247550	0.9443100	0.0002954

Displaying only those variables whose p-value is less than 0.05:

```r
# Filter rows where p-value is less than 0.05
significant_results2 <- model_results2[model_results2$p_value < 0.05, ]

# Print the significant results
knitr::kable(
  significant_results2,
  col.names = c("Estimate", "Odds Ratio", "Lower Bound (2.5%)", "Upper Bound (97.5%)", "p-value")
)
```

	Estimate	Odds Ratio	Lower Bound (2.5%)	Upper Bound (97.5%)	p-value
(Intercept)	-2.3043311	0.0998255	0.0824239	0.1209011	0.0000000
int_rate	0.4538310	1.5743319	1.5015030	1.6506933	0.0000000
emp_lengthn/a	0.5242234	1.6891465	1.4078333	2.0266718	0.0000000
annual_inc	-0.4052951	0.6667800	0.6287201	0.7071439	0.0000000
purposedebt_consolidation	0.2149973	1.2398585	1.0408627	1.4768991	0.0160097
purposeeducational	0.7071939	2.0282917	1.4907239	2.7597111	0.0000068
purposehome_improvement	0.2050367	1.2275702	1.0018144	1.5041993	0.0479871
purposemedical	0.4829562	1.6208590	1.2390193	2.1203736	0.0004256
purposemoving	0.4670106	1.5952182	1.1950486	2.1293873	0.0015286
purposeother	0.4489604	1.5666826	1.2955072	1.8946205	0.0000037
purposerenewable_energy	0.5965259	1.8157996	1.0076896	3.2719682	0.0470893
purposesmall_business	0.9862006	2.6810287	2.1902777	3.2817368	0.0000000
purposevacation	0.3745970	1.4544051	1.0266837	2.0603174	0.0350132
inq_last_6mths	0.2131425	1.2375610	1.2049073	1.2710997	0.0000000
pub_rec	0.0629159	1.0649373	1.0381200	1.0924473	0.0000013
revol_bal	0.0944704	1.0990766	1.0639978	1.1353120	0.0000000
revol_util	0.1080543	1.1141082	1.0756566	1.1539343	0.0000000
term60 months	0.5170389	1.6770544	1.5490627	1.8156214	0.0000000
int_rate:term60 months	-0.1249849	0.8825103	0.8247550	0.9443100	0.0002954

Interpretations

Accounting for variation in time and location does have an impact on our model. In particular, accounting for this variation results in an addition of four newly significant covariates, which are all loan purpose categories:

- purpose == debt_consolidation

For loans with a stated purpose of debt consolidation, the odds of "repay_fail" occurring increase by a factor of approximately 1.24 (95% CI: [1.04,1.48], $p = 0.016010$), compared to the baseline (car).

- purpose == educational

For loans with a stated purpose of education, the odds of "repay_fail" occurring increase by a factor of approximately 2.03 (95% CI: [1.50,2.76], $p = 0.0000068$), compared to the baseline (car).

- purpose == home_improvement

For loans with a stated purpose of home improvement, the odds of "repay_fail" occurring increase by a factor of approximately 1.23 (95% CI: [1.00,1.50], $p = 0.0479871$), compared to the baseline (car).

- purpose == renewable_energy

For loans with a stated purpose of renewable energy, the odds of "repay_fail" occurring increase by a factor of approximately 1.82 (95% CI: [1.01,3.28], $p = 0.0470893$), compared to the baseline (car).

Beyond these differences, this extended model had the same general set of significant covariates and interactions, in the same directions, though the effect sizes and p-values may vary.

In summary, the full list of important variables in our extended model is as follows:

- Interest rate
- Unspecified employment length (whether information is missing or the individual is unemployed)
- Annual income
- Loan purposes of medical, moving, small business, vacation, other
  - Debt consolidation, educational, home improvement, renewable energy (new with extended model)
- Number of inquiries in the last 6 months
- Number of derogatory public records
- Revolving balance and utilization
- 60-month loan term
- Interest rate * 60-month loan term

Random Effects: Interpretations

Now, we can interpret the role of the random effects of our extended model, which are time and location (consists of zip code nested within each state).

```
summary(fit_var_5)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## repay_fail ~ int_rate + emp_length + annual_inc + purpose + inq_last_6mths +
##     pub_rec + revol_bal + revol_util + term * int_rate + (1 |
##     location) + (1 | earliest_cr_line)
##    Data: extended_scaled
## Control:
## glmerControl(optimizer = "Nelder_Mead", check.conv.grad = .makeCC("warning",
##     tol = 0.008, relTol = NULL))
##
##      AIC      BIC   logLik deviance df.resid
##  30142.1  30441.5 -15036.0  30072.1    38384
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.468 -0.448 -0.334 -0.234 92.042
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  location         (Intercept) 0.05114  0.2261
##  earliest_cr_line (Intercept) 0.01414  0.1189
## Number of obs: 38419, groups:  location, 905; earliest_cr_line, 528
```

```
##
## Fixed effects:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.304331   0.097729 -23.579  < 2e-16 ***
## int_rate                    0.453831   0.024166  18.780  < 2e-16 ***
## emp_length1 year            0.008895   0.067063   0.133 0.894478
## emp_length10+ years         0.090262   0.054878   1.645 0.100016
## emp_length2 years          -0.112380   0.063486  -1.770 0.076701 .
## emp_length3 years           0.009179   0.064043   0.143 0.886038
## emp_length4 years          -0.060151   0.067660  -0.889 0.373996
## emp_length5 years           0.025726   0.067912   0.379 0.704829
## emp_length6 years          -0.019520   0.077204  -0.253 0.800391
## emp_length7 years          -0.010774   0.083236  -0.129 0.897015
## emp_length8 years           0.011029   0.088992   0.124 0.901372
## emp_length9 years          -0.093617   0.098251  -0.953 0.340674
## emp_lengthn/a               0.524223   0.092945   5.640 1.70e-08 ***
## annual_inc                 -0.405295   0.029987 -13.516  < 2e-16 ***
## purposecredit_card          0.004153   0.098212   0.042 0.966269
## purposedebt_consolidation   0.214997   0.089259   2.409 0.016010 *
## purposeeducational          0.707194   0.157108   4.501 6.75e-06 ***
## purposehome_improvement     0.205037   0.103686   1.977 0.047987 *
## purposehouse                0.253393   0.168896   1.500 0.133540
## purposemajor_purchase       0.017726   0.113695   0.156 0.876106
## purposemedical              0.482956   0.137059   3.524 0.000426 ***
## purposemoving               0.467011   0.147359   3.169 0.001529 **
## purposeother                0.448960   0.096968   4.630 3.66e-06 ***
## purposerenewable_energy     0.596526   0.300442   1.985 0.047089 *
## purposesmall_business       0.986201   0.103149   9.561  < 2e-16 ***
## purposevacation             0.374597   0.177685   2.108 0.035013 *
## purposewedding              0.004378   0.139216   0.031 0.974910
## inq_last_6mths              0.213143   0.013643  15.623  < 2e-16 ***
## pub_rec                     0.062916   0.013013   4.835 1.33e-06 ***
## revol_bal                   0.094470   0.016550   5.708 1.14e-08 ***
## revol_util                  0.108054   0.017920   6.030 1.64e-09 ***
## term60 months               0.517039   0.040505  12.765  < 2e-16 ***
## int_rate:term60 months     -0.124985   0.034533  -3.619 0.000295 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 33 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

We can compare the variance of each random effect to determine which has a larger impact on an individual's credit risk, if any.

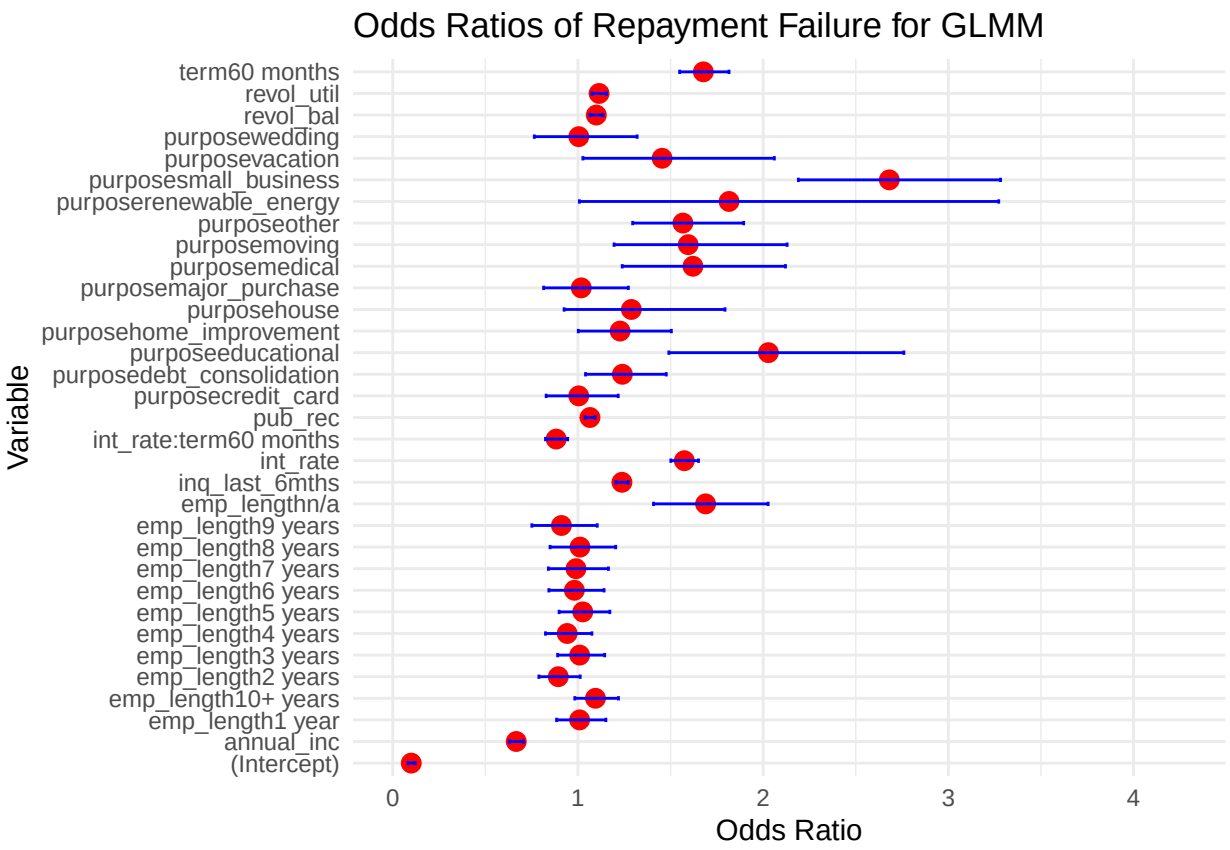
- Location: The location (addr_state:zip_code) random effect has a variance of 0.05114, meaning that the variation in applicants' locations explains about 5% of the variation in credit risk.
- Time: The time (earliest_cr_line) random effect has a variance of 0.01414, meaning that the variation in how long each applicant has had credit explains about 1% of the variation in credit risk.

If we compare the variance of each random effect, we'll see that location has a higher variance (0.05) than time (0.01). Thus, we can say that location has a larger impact on an applicant's credit risk, in comparison to time.

Odds Ratio Plot with CIs

Below, we will create a plot that depicts the odds ratio of repayment failure for our final extended GLMM model, including confidence interval ranges. These estimates are all compared to the baseline, which includes the 36-month loan term, less than one year of employment, and the loan purpose of "car".

```
plot_data <- data.frame(
  Variable = rownames(model_results2),
  OddsRatio = model_results2$OddsRatio,
  LowerCI = model_results2$LowerCI,
  UpperCI = model_results2$UpperCI
)

ggplot(plot_data, aes(x = OddsRatio, xmin = LowerCI, xmax = UpperCI, y = Variable)) +
  geom_point(size = 3, color="red") +
  geom_errorbarh(height = 0.2, color = "blue") +
  xlim(c(0, max(plot_data$UpperCI + 1))) +
  theme_minimal() +
  labs(x = "Odds Ratio", y = "Variable") +
  ggtitle("Odds Ratios of Repayment Failure for GLMM")
```



Accuracy and Confusion Matrix

Finally, we can assess how well our extended model performs in terms of classifying whether applicants will default on their payments or not.

```
confusion_glmm <- ifelse(prob_ext > 0.5, "1", "0")
confusion_glmm <- as.factor(confusion_glmm)

require(caret)
cm_glmm <- confusionMatrix(data=confusion_glmm,
                           reference=extended_scaled$repay_fail,
                           positive="1")
cm_glmm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 32492  5678
##          1   118   131
##
##                Accuracy : 0.8491
##                  95% CI : (0.8455, 0.8527)
##     No Information Rate : 0.8488
##     P-Value [Acc > NIR] : 0.43
##
##                   Kappa : 0.0312
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.022551
##             Specificity : 0.996381
##          Pos Pred Value : 0.526104
##          Neg Pred Value : 0.851244
##              Prevalence : 0.151201
##          Detection Rate : 0.003410
##    Detection Prevalence : 0.006481
##       Balanced Accuracy : 0.509466
##
##        'Positive' Class : 1
##
```

Included above is a confusion matrix for our extended model, which we have built only based on "seen" data. It demonstrates that out of 38,419 applicants, 32,492 of them will not default and have been accurately classified as so, and 131 of them will default and have been accurately classified as so. However, using our extended model, 5,678 applicants are falsely classified as non-defaulters when in fact, they will default, and 118 applicants are falsely classified as defaulters when they will not default.

Part 2 Discussion

In summary, the main goal of this second part of our analysis was to account for variations in location and time to determine whether this improves our model's performance benchmarks.

Specifically, below are the three questions we set out to answer in part 2 of this analysis:

3. Can accounting for this variation (e.g., state/zip-code and time) improve performance benchmarks?
4. Are there any surprising differences in variables that are important for predicting credit risk?
5. Does credit risk change over time or between states? This is not something the bank has previously investigated and results may inform modified loan policies in the future.

In terms of question 3, we can say that yes, accounting for the variation in location and time does improve our model's performance benchmarks. Specifically, our extended model achieved a $Gini = 0.437$, compared to the final model of the first part of our analysis, which had a $Gini = 0.410$. This improved Gini score tells us that our extended model did a slightly better job at predicting whether applicants would fail to repay their loans or not.

In terms of question 4, we were a bit surprised by the inclusion of 4 more significant covariates in our model, after accounting for this variation:

- Debt consolidation
- Educational
- Home improvement
- Renewable energy

Once again, all 4 were associated with an increased risk of repayment failure. Initially, debt consolidation and educational purposes made the most sense to us as being important factors contributing to credit risk, as typically, if an individual is consolidating their debt, they have a decent amount of it. Furthermore, educational expenses, especially in the United States, are rather high and many student loan borrowers are in a great deal of student loan debt. Thus, we are assuming that individuals seeking loans for the purposes of debt consolidation and education will be requesting higher loan amounts, which could be risky in terms of their ability to fully repay them.

However, home improvement and renewable energy loans surprised us as being significant, at least initially. When further considering these, though, we realized that typically, home improvement and the installation or switch to renewable energy are often associated with larger upfront costs, with delayed benefits. For example, when choosing to install solar panels on a home, prices can range between $15,000.00 and $20,000.00, though some homes may cost an upwards of $25,000.00 for full installation How Much Do Solar Panels Cost? (2023 Guide). Furthermore, individuals typically do not begin to see the benefits in terms of decreased energy bills and savings for months to come, thus putting them at a greater initial risk of failing to repay their loan. Prices for education, home improvement, and renewable energy also tend to vary by location; educational expenses will vary based on desired locations and cost of living, as well as whether the school is public or private. In terms of home improvement and renewable energy, associated costs will depend on the market value of homes, as well as current prices of supplies and costs for contractors (if necessary). These all make more sense in the context of this part of our analysis, as we are now accounting for variations in location, thus potentially helping to explain why the above-listed covariates are newly significant in this extended model.

Finally, for the last question (question 5), credit risk does appear to change over time and between states. Credit risk changes more between states than time, as the variance for location was 0.05, compared to that for time, which was 0.01. Thus, location does seem to have a moderate impact on an individual's credit risk, and thus, the bank should consider location when assessing loan applications and policies.

Conclusion

In summary, we have conducted a thorough analysis and built a new credit risk model from the ground up to address the bank's concerns regarding their previous models.

To review, below are the questions that guided our analyses:

1. How does this new model perform compared to the one used previously? How can it be expected to perform on new loan applications?
2. What are the important variables in this model and how do they compare to variables that are traditionally important for predicting credit risk in the banking sector?
3. Can accounting for this variation (e.g., state/zip-code and time) improve performance benchmarks?
4. Are there any surprising differences in variables that are important for predicting credit risk?
5. Does credit risk change over time or between states? This is not something the bank has previously investigated and results may inform modified loan policies in the future.

Our new model has demonstrated significantly improved performance in comparison to the previous models, on both "seen" data and "unseen" data ($Gini = 0.410$ on seen for new model versus $Gini = 0.114$ on seen for old model). It is through the use of this "unseen" data that we were able to assess our new model's performance on new loan applications ($Gini = 0.395$ on unseen for new model versus $Gini = 0.110$ on unseen for old model). The important variables of our new model are:

- Interest rate
- Unspecified employment length (whether information is missing or the individual is unemployed)
- Annual income
- Loan purposes of medical, moving, small business, vacation, other
- Number of inquiries in the last 6 months
- Number of derogatory public records
- Revolving balance and utilization
- 60-month loan term
- Interest rate * 60-month loan term

These mostly align with the variables traditionally used to predict credit risk in the banking sector. A main deviation from tradition is that our new model excluded the variables of debt to income ratio, home ownership, and loan amount, which surprised us.

When we extended our new credit risk model to account for variations in location and time, we did see improved model performance benchmarks in that we received an improved Gini score ($Gini = 0.437$ on extended versus $Gini = 0.410$ on model without accounting for variation). We were only able to assess this model's performance on seen data, and could not assess it's predictive performance on new loan applications. Included below is a plot depicting the ROC curves, including Gini scores, for both newly proposed models. Model 1 refers to the initial new model we built, and model 2 refers to our extended model.

```
# roc curves
roc_logit_train <- roc(train_scaled_data$repay_fail, prob_logit_train)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

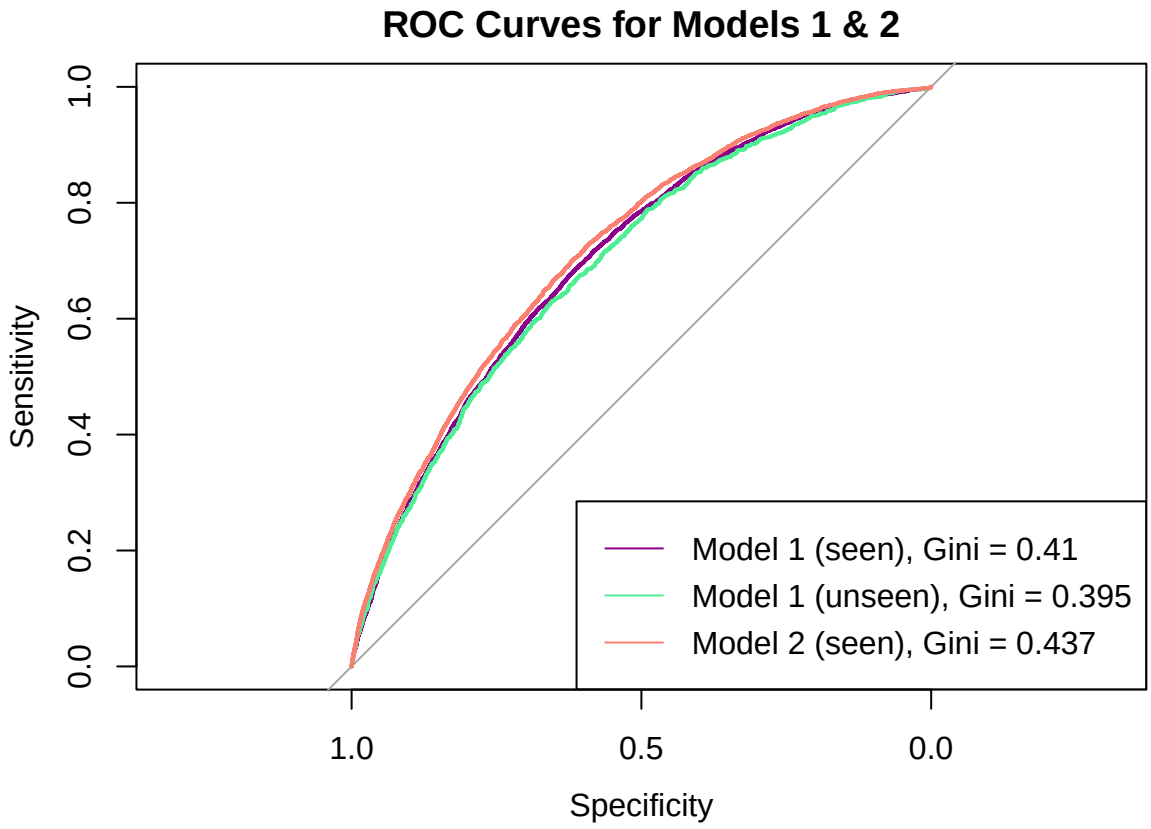
```
roc_logit_val <- roc(val_scaled_data$repay_fail, prob_logit_val)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
roc_ext <- roc(extended_scaled$repay_fail, prob_ext)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# plot all
plot (roc_logit_train, col="darkmagenta", main = "ROC Curves for Models 1 & 2")
plot(roc_logit_val, col = "seagreen2", add = TRUE)
plot(roc_ext, col = "salmon", add = TRUE)

# add legend
legend("bottomright",
       legend = c(paste("Model 1 (seen), Gini =", round(2 * auc(roc_logit_train) - 1, 3)),
                  paste("Model 1 (unseen), Gini =", round(2 * auc(roc_logit_val) - 1, 3)),
                  paste("Model 2 (seen), Gini =", round(2 * auc(roc_ext) - 1, 3))),
       col = c("darkmagenta", "seagreen2", "salmon"), lty = 1)
```



We were initially surprised by the addition of 4 newly significant covariates in our extended model, which were the loan purposes of debt consolidation, educational, home improvement, and renewable energy. However, given that the overall cost of many of these purposes will likely vary based on location, these actually do make sense in the context of our extended model.

Finally, we found that credit risk changes more over location than time, so this is a variable the bank should consider in future loan applications and policies.

While our final model may not be the *best* model, it is a *better* model than the bank has previously used. Thus, we have adequately addressed the bank's concerns by presenting a model that does a better job of predicting loan default, and we hope the bank is delighted by our findings.

Caveats

Though we have attempted to conduct a thorough and meaningful analysis to develop a new credit risk model that performs significantly better than those the bank was previously using, there are still a few caveats to our analyses. We have listed these below:

- We have only included a subset of the possible variables which may help to predict loan default, and there may be other variables that result in a better model with improved performance benchmarks compared to the one we have presented in this report. In particular, we were surprised by the exclusion of certain variables, such as debt to income ratio, home ownership, and loan amount, which we initially assumed would be important for determining an individual's credit risk. Perhaps further investigation of these variables, along with others, may prove to be useful. The bank could even consider investigating interactions between more variables, but as discussed earlier in this report, we opted to avoid the inclusion of all but one possible interaction to prioritize the interpretability of our results.
- Our extended model has a relatively high tolerance value, which could potentially be further reduced by tuning other model parameters. Furthermore, we have not been able to assess the extended model's predictive performance, as we did not have access to any unseen data that included the necessary variables to do so. Assessing predictive performance of the extended model may help to solidify evidence of this model's performance abilities.
- We do not know the time frame of the data that we have used to build and assess our models, and it might be helpful to know this to ensure that we use recent data for model building and prediction tasks. Obtaining the most up-to-date information would help to bolster confidence in the presented new models, as market conditions and the economy may be other factors that play a role in credit risk, and we have not accounted for those in our analyses.
- There is an imbalance within the repay_fail class, in that there exist many more cases of non-failure to repay, which might influence our model to favour that outcome. Thus, our results should be considered with this in mind.

Future Directions

As mentioned above, further research could explore other variables and interactions between variables which were excluded in our analyses. This may further improve performance benchmarks and the model's ability to predict whether applicants will fail to repay their loans or not. Furthermore, if provided with a validation dataset which included the extended variables of location and time, it may be useful to use the extended model to predict loan default on this "unseen" data and compare the extended model's prediction performance with that of the new model from the first part of our analysis. This could further validate whether the extended model improves performance benchmarks.

Bibliography

Listed below are the references we consulted to conduct this analysis, including our research on credit risk as well as resources we consulted when addressing GLMM model convergence errors.

Choksi, N. (n.d.). Top 5 Factors Affecting Credit Risk When Taking a Personal Loan. Forbes. https://www.forbes.com/advisor/in/personal-loan/top-5-factors-affecting-credit-risk-when-taking-a-personal-loan/

CRAN. (n.d.-a). lme4 convergence warnings: Troubleshooting. R-Studio Pubs. https://rstudio-pubs-static.s3.amazonaws.com/33653_57fc7b8e5d484c909b615d8633c01d51.html

CRAN. (n.d.-b). Lme4 performance tips. CRAN - R Project. https://cran.r-project.org/web/packages/lme4/vignettes/lmerperf.html#:~:text=choice%20of%20optimizer,may%20be%20worth%20a% 0try.

Segal, T. (n.d.). Understanding the Five Cs of Credit. Investopedia. https://www.investopedia.com/ask/answers/040115/what-most-important-c-five-cs-credit.asp

The R Foundation. (n.d.). Control of Mixed Model Fitting—R. R Documentation. https://search.r-project.org/CRAN/refmans/lme4/html/lmerControl.html

Wakefield, F. (2023, October 4). How Much Do Solar Panels Cost? (2023 Guide). MarketWatch. https://www.marketwatch.com/guides/solar/solar-panel-cost/#:~:text=Based%20on%20our%20survey%20of,upward%20of%20%24
%20per%20installation.