Stout DDA Intern Case Study

Sharleen Kong

2022-04-09

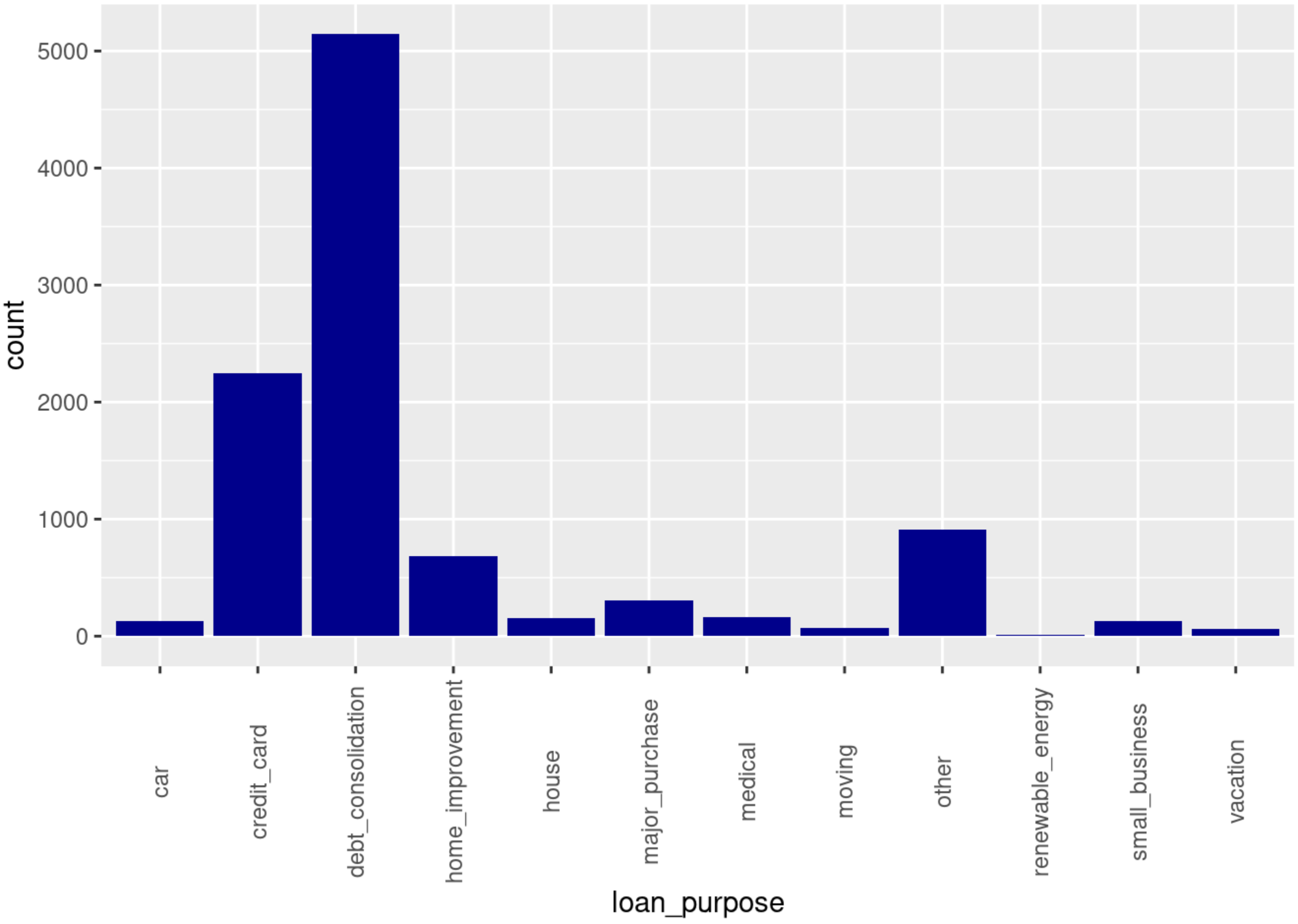
Case Study 1

Description This data set represents thousands of loans made through the Lending Club platform. The dataset records information of the person who is getting the loan (e.g. job title, job years, annual income, etc.) as well as their credit history such as total number of credit lines, Delinquencies, etc. In addition, the dataset also includes the information of the loan received by each person, including loan amount, term, interest rate, etc. This dataset helps to learn the relationship between a person's credit history and the loan they can receive, thus helps to better predict the lending decision.

Issues with this dataset There are many missing data in some of the columns (e.g. job titles), which will cause problem in the data analysis process.

Marginal distribution of purpose of loan

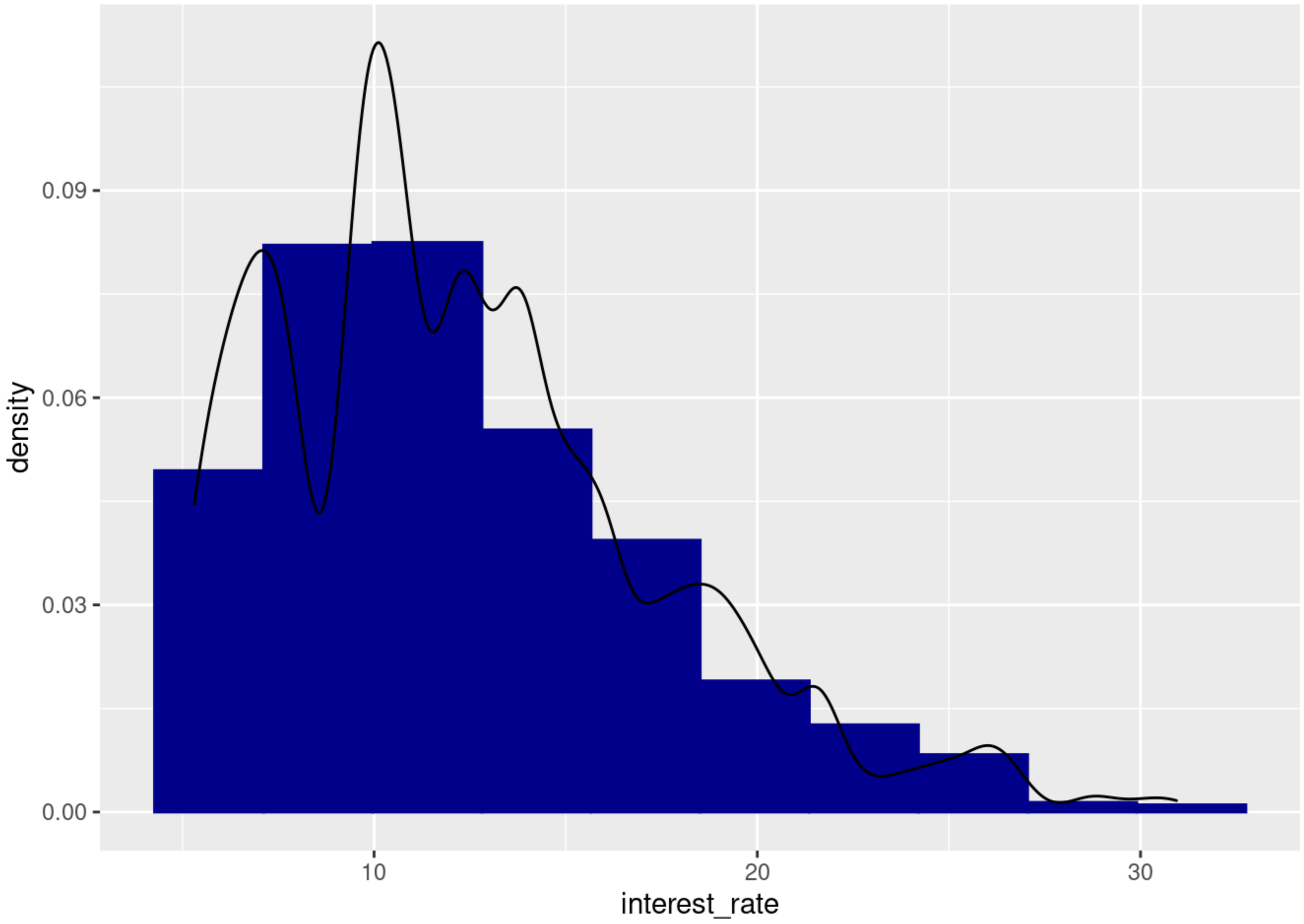
```
ggplot(data = loans_full_schema, aes(x = loan_purpose)) +
geom_bar(fill="darkblue")+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=0.5))
```



From the bar chart above, we observed that most loans are applied for the purpose of debt consolidation, followed by credit card, other, and home improvement.

Distribution of loan interest rate

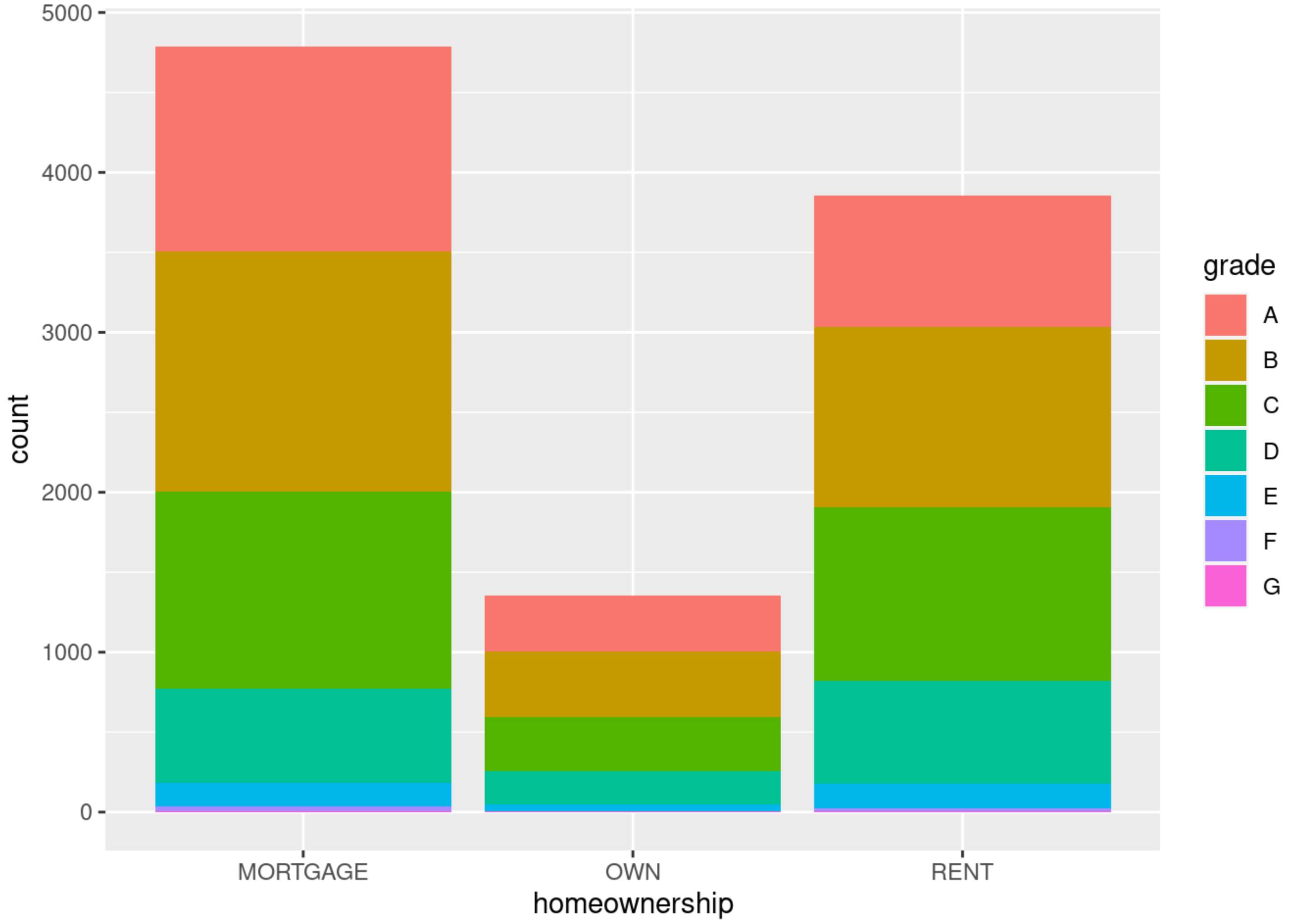
```
ggplot(loans_full_schema, aes(x = interest_rate)) +
geom_histogram(color = "darkblue",fill="darkblue",aes(y = after_stat(density)),bins = 10)+geom_density()
```



From the histogram above, we can observe that most loans in this data set has interest rate between 5% and 15%.

Marginal distribution of loan grade given homeownership

```
ggplot(data = loans_full_schema, aes(x = homeownership)) +
geom_bar(aes(fill=grade))
```



From the bar chart above, we can see that there is not much differentiation in loan grade among person with different types of homeownership.

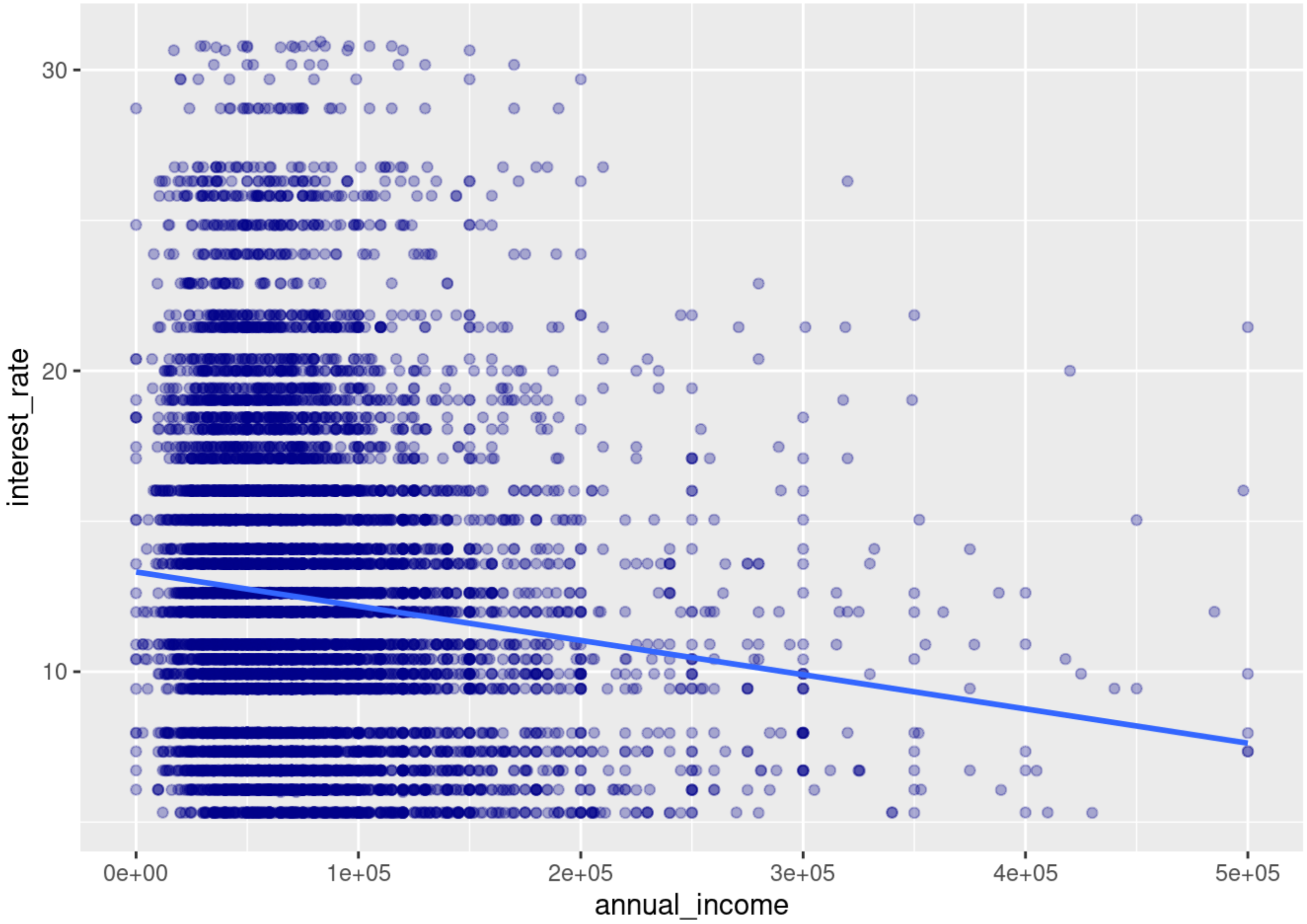
Relationship between annual income and interest rate

```
ggplot(loans_full_schema, aes(x= annual_income,y=interest_rate))+ geom_point(alpha=0.3,color="darkblue")+
  xlim(c(0,500000))+  # remove some outliers to make pattern more obvious
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 19 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 19 rows containing missing values (geom_point).
```

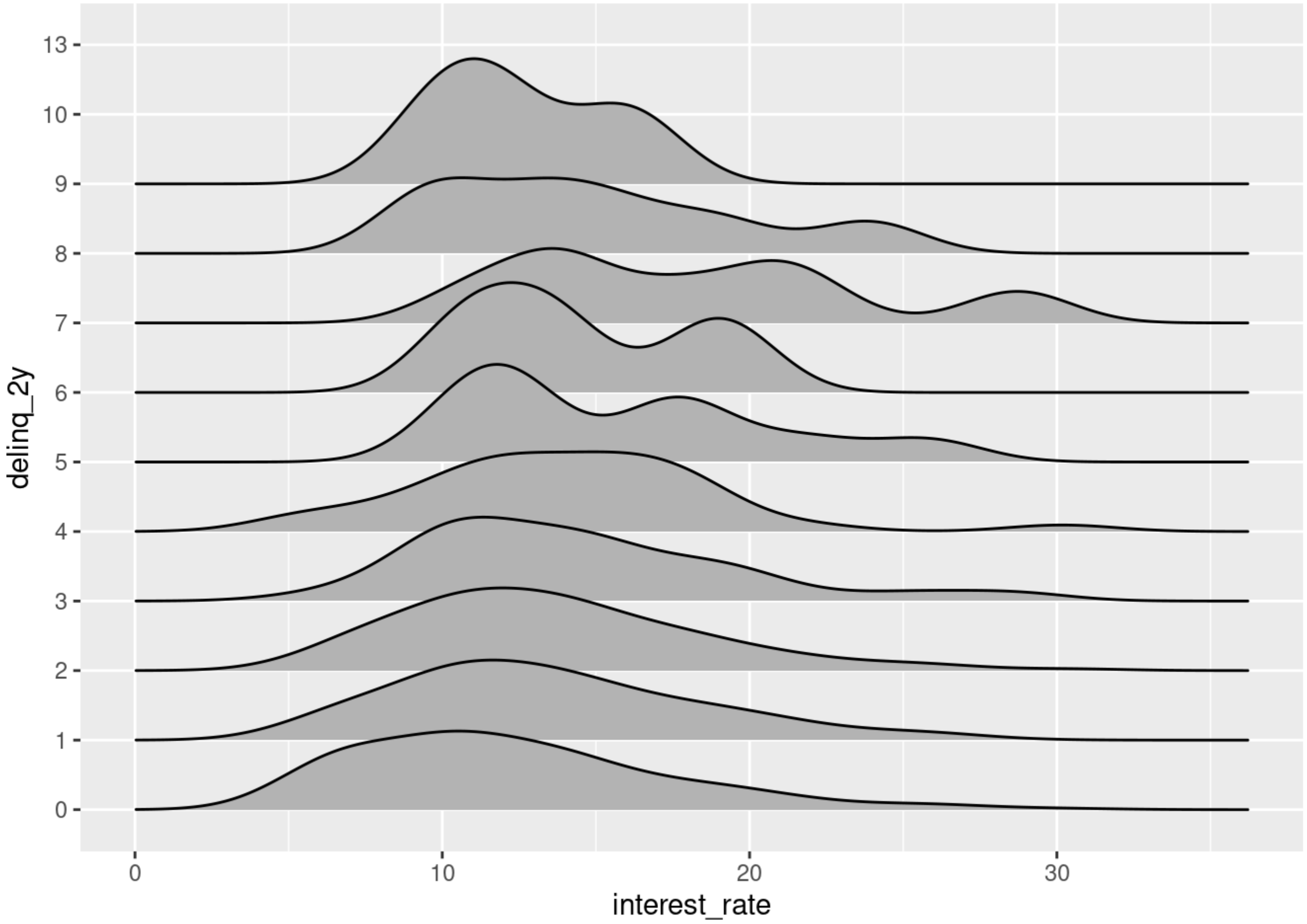


From the scatter plot and the fitted line using linear regression method, we can see that person with higher annual income generally receives a lower interest rate while people with lower income receives higher interest rate.

Relationship between delinquencies on lines of credit in the last 2 years and interest rate

```
library(ggridges)
loans_full_schema$delinq_2y <- as.factor(loans_full_schema$delinq_2y)
ggplot(loans_full_schema, aes(x = interest_rate, y =delinq_2y)) + geom_density_ridges()
```

```
## Picking joint bandwidth of 1.76
```



From the ridgeline plot, we can observe that fewer delinquencies generally yields a lower interest rate, especially for less than 4 delinquencies.

Create feature set We have to remove rows with empty values here. Also, the last few columns should not be included since they are part of the "loan received", whose interest rate is our prediction task. Also, some of the previous credit history items may be colinear, so we just have to pick one from each category (e.g. delinquencies, credit line num, etc.)

```
loans <- loans_full_schema %>%
dplyr::select(`emp_length`,`annual_income`, `debt_to_income`, `earliest_credit_line`,'total_credit_utilize
d',"public_record_bankrupt","delinq_2y","inquiries_last_12m","account_never_delinq_percent","num_accounts_1
20d_past_due","interest_rate") %>%
  na.omit()
loans$delinq_2y <- as.numeric(loans$delinq_2y)
```

Training set and testing set partitioning

```
set.seed(1)

train.index <- sample(row.names(loans), dim(loans)[1]*0.6)
train.df <- loans[train.index,]

test.index <- setdiff(row.names(loans), train.index)
test.df <- loans[test.index,]
```

Linear regression

```
lm_loan <-lm(interest_rate~. ,data = train.df)
summary(lm_loan)
```

```
##
## Call:
## lm(formula = interest_rate ~ ., data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.6314  -3.5805  -0.7766   2.5964  19.9604
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -1.691e+02  1.880e+01  -8.991  < 2e-16 ***
## emp_length                   -2.319e-03  1.831e-02  -0.127  0.89921
## annual_income                -5.765e-06  1.106e-06  -5.212 1.93e-07 ***
## debt_to_income                5.984e-02  5.200e-03  11.508  < 2e-16 ***
## earliest_credit_line          9.337e-02  9.401e-03   9.932  < 2e-16 ***
## total_credit_utilized         3.993e-06  1.393e-06   2.866  0.00417 **
## public_record_bankrupt        8.984e-01  2.023e-01   4.442 9.11e-06 ***
## delinq_2y                     5.557e-01  1.112e-01   4.996 6.05e-07 ***
## inquiries_last_12m            2.362e-01  2.708e-02   8.722  < 2e-16 ***
## account_never_delinq_percent -7.898e-02  8.248e-03  -9.575  < 2e-16 ***
## num_accounts_120d_past_due           NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.787 on 5321 degrees of freedom
## Multiple R-squared:  0.09873,    Adjusted R-squared:  0.09721
## F-statistic: 64.77 on 9 and 5321 DF,  p-value: < 2.2e-16
```

```
pred_test <- predict(lm_loan,test.df)
```

```
## Warning in predict.lm(lm_loan, test.df): prediction from a rank-deficient fit
## may be misleading
```

```
pred_train <- predict(lm_loan,train.df)
```

```
## Warning in predict.lm(lm_loan, train.df): prediction from a rank-deficient fit
## may be misleading
```

```
# RMSE in test
rmse_ols_test <- sqrt(mean((test.df$interest_rate-pred_test)^2))
rmse_ols_test
```

```
## [1] 4.766929
```

```
# RMSE in train
rmse_ols_train <- sqrt(mean((train.df$interest_rate-pred_train)^2))
rmse_ols_train
```

```
## [1] 4.782728
```

Regression tree

```
library(rpart)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

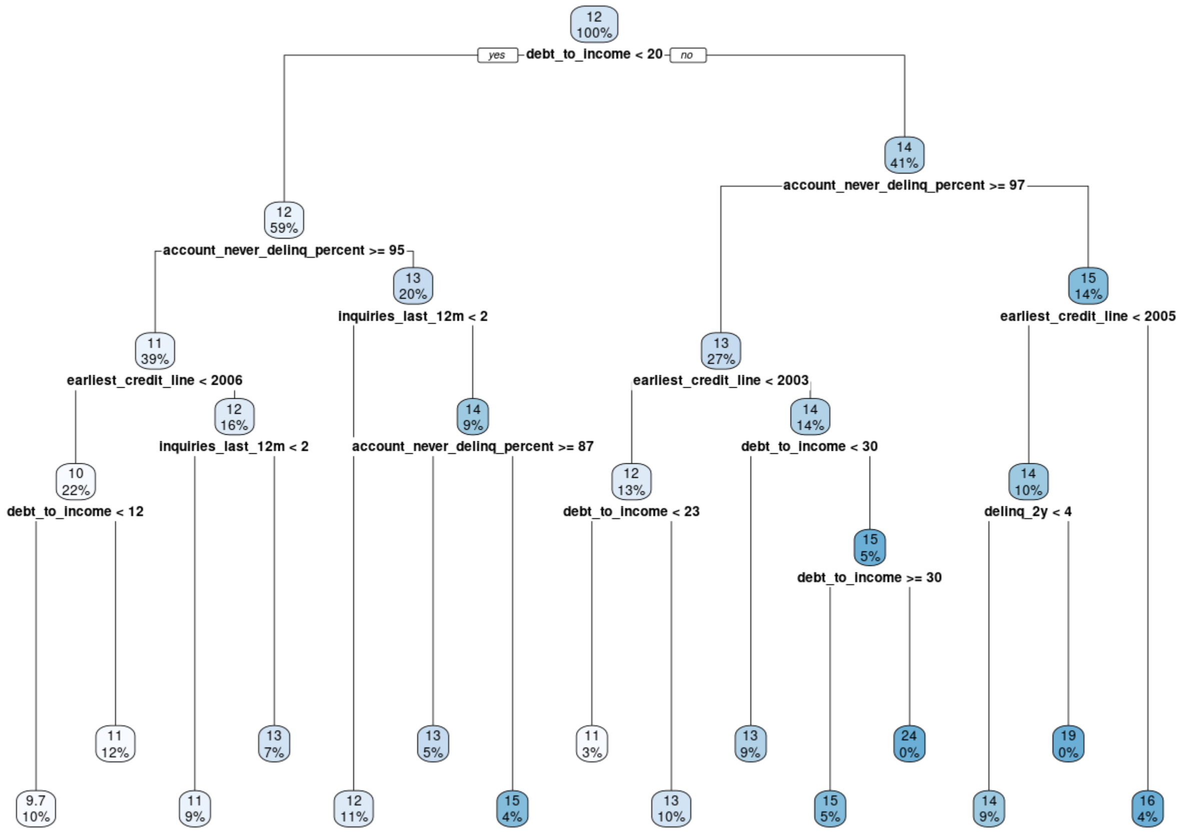
```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart.plot)
set.seed(555)

####  Train the model
tree.fit <- train(interest_rate ~ ., data=train.df,
                  method = 'rpart',
                  trControl=trainControl(method = 'cv', number=5),tuneLength = 30)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
rpart.plot(tree.fit$finalModel)
```



```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
tree_pred_test <- predict(tree.fit,test.df)
tree_pred_train <- predict(tree.fit,train.df)

accuracy(tree_pred_test,test.df$interest_rate)
```

```
##                   ME     RMSE      MAE      MPE    MAPE
## Test set -0.1876158 4.808252 3.767317 -17.0927 36.0996
```

```
accuracy(tree_pred_train,train.df$interest_rate)
```

```
##                     ME     RMSE      MAE       MPE     MAPE
## Test set 2.424051e-15 4.740589 3.722076 -15.15548 34.72969
```

Test result The RMSE of test set using regression tree is 4.81 while that of linear regression is 4.77. Linear regression has a slightly better performance here.

Future Improvements In order to increase model accuracy, ensemble models can be used. Also, I assume that there are some degree of colinearity between some variables such as Number of current accounts that are 120 days past due and Number of current accounts that are 30 days past due. To make better models, I can select features more carefully with techniques such as PCA, colinearity matrix and so on to pick the best features.