Raymond Sutanto, Cheng Lin Tsai, Yi Jen Chen
August 31, 2022
IDS 572 Assignment 2
Prof. Siddhartha Bhattacharyya

Questions:
Part A

1.(a) Your team's ultimate goal is to help clients determine whether they should invest in p2p loans. What is the final decision that you will help the client make? What is the objective, and how will you evaluate 'better' vs 'worse' decisions? What is the goal of predictive models for this ? What will be the potential target variables?

The clients that are mentioned in this scenario are lenders or investors, so basically they are interested in investing in the right loans with hopes of achieving profits. We hope that our data analysis will help the investors in finding the right loans that can maximize their overall returns. Even if our initial observation has proven that investing in high-risk loans can yield the highest profits, they also have the highest risk of defaults (borrowers unable to repay the loan principal and interests). So by all means, we hope that the loans that we recommend can maximize their overall yields but also minimize their overall risk in this loan investment.

To find the right loans, we need to set objectives, which are mainly based on our analysis of the factors that are very prevalent in the default of loans. Moreover, we want to predict the influence of these factors in the probability of the loans, that range from grades A-G (A being the safest and G being the highest risk), from being default. This objective can be derived from the loan characteristics as well as the borrower's profile. Finding the probability of the loan from being default (loan default rate) can be done by building a model that can associate the factors as well as the strength of each factor to the default rate, which is then compared with the loan expected return. This is basically the goal of the predictive models that we aim to create, which is to find out the significance of each factor in the default rate and furthermore compare it with the expected return.

Having the above approach would be the better decision-making since investors would aim to maximize their yields while also considering how to minimize default risk from their investment, which is derived from understanding the model of default risk from each of their investments. Not having the above approach and simply just investing "blindly" (we assume that they would pick the highest risk loans since they yield the highest returns) would be the worst decision-making since there is a chance of default and if that is the case these investors would obtain quantifiable losses from their loan investments.

Based on the above explanations, we have classified our potential target variables into two categories.

- Borrower's profile/characteristics: this includes the borrower's credit grade, FICO score, borrower's annual income, numbers of delinquencies, last loan delinquency, revolving balance, debt-to-income ratio, etc
- Loan profile: this includes the size of the loan, the loan interest rate, and the purpose of the loan

1.(b) Take a look at the data attributes. How would you categorize these attributes, in broad terms, considering what they pertain to? Before doing any analyses, what do you think may be the important attributes to consider for your decision task?

We can define the attributes based on three definitions. The first definition is basically whether the attribute is a quantitative or qualitative attribute. We can further classify a quantitative or qualitative attribute into several categories.

NB: We use this code to find the categories: head(lcdf) %>% str()

- Quantitative attribute
    - Numeric data attributes => these attributes contain values in numerical format with no specific interpretation. Examples of variables with these attributes include loan amount, total payment, annual income, DTI ratio, etc
    - Scaled data attributes => these attributes contain values in numerical format with a specific interpretation. For instance, the variable issue_d, which is the month the loan is funded, is labeled as POSIXct, which is a date.
- Qualitative attribute
    - Nominal attributes: These attributes are just variables with different names without any specific order. Furthermore, nominal attributes provide enough information to distinguish one object from another. Examples of nominal attributes in our dataset may include member_id, zip_code, state, etc.
    - Ordinal attributes: These attributes are variables with a specific ordering/sequential classification. Examples of ordinal attributes are grades and sub-grades.
    - Boolean/Binary Attributes: These attributes are variables which have two values (0 and 1, Yes or No, etc). For example, the variable ‘application_type’ has only two outputs, either ‘individual’ and ‘joint app’.

The second definition is based on how the attribute pertains to a specific stakeholder. For our dataset, we classify the attributes by this format:

- Borrower’s Attributes: These attributes are to identify the financial standing of the borrower. Examples of these attributes may include annual income, FICO score, DTI ratios, delinquencies, balance, number of credit lines, etc.
- Loan’s Attributes: These attributes are basically how the loan is defined/described. Examples of these attributes may include the loan amount, purpose of the loan, issue date, etc.
- Lending Platform’s Attributes: These attributes are provided by the Lending Club platform as guidance for investors when choosing a loan. Examples of these attributes may include the grade, subgrade, loan interest rate, monthly installment, investor’s total funded loan amount, etc.
- Loan Performance’s Attributes: These attributes are also provided by the Lending Club platform as indicators of the loan’s performance in the marketplace, which further

broaden the guidance for the investors in choosing a loan. Examples of these include the status of the loan (default to fully paid), total payment received for total amount funded, last monthly payment received, etc.

The third definition is based on how the attribute is considered a discrete or continuous variable.

- Discrete variables have finite/limited values that can be counted, and they can be either quantitative or qualitative. Examples of these attributes are zip_code, state, application type, etc. Continuous data have an infinite number of states.
- Continuous variables have infinite values that can be measured, and usually they are quantitative in nature. Examples of these attributes are interest rates, loan amount, annual income, etc.

Furthermore, we need to keep in mind several important attributes in our decision-making process.

- First off, we need to realize that some variables can be changed in any period of time. For instance, a person's FICO score changes periodically. This will impact our model creation due to constant changes in some attributes' values that can result in an unstable model, which is not something we want as this can mislead an investor's judgment based on our model.
- Also, there are some attributes that can have overlapping information, which can result in redundancy information. For instance, a change in loan status will change the past due amount, total payment, initial funding, etc and if we put in these variables as the dependent variables, obviously this redundancy will create a less accurate model with a lower predictive power.

Considering the above explanation, we will classify our attributes based on how each variable pertains to a specific stakeholder.

- Borrower's Attributes: the financial standing of the borrower will help in the decision-making process of the lender as to whether the borrower has a chance of defaulting the loan.
- Loan's Attributes: these can help lenders in deciding whether the loan can have positive yields.
- Lending Platform Attributes: these can help the lender in determining the loan default rate.
- Loan Performance Attributes: these can help the lender in determining whether the loan is a good investment.

2. Data exploration

(a) some questions to consider:
(i) What is the proportion of defaults ('charged off' vs 'fully paid' loans) in the data?
How does the default rate vary with loan grade? Does it vary with sub-grade? And is this what you would expect, and why?

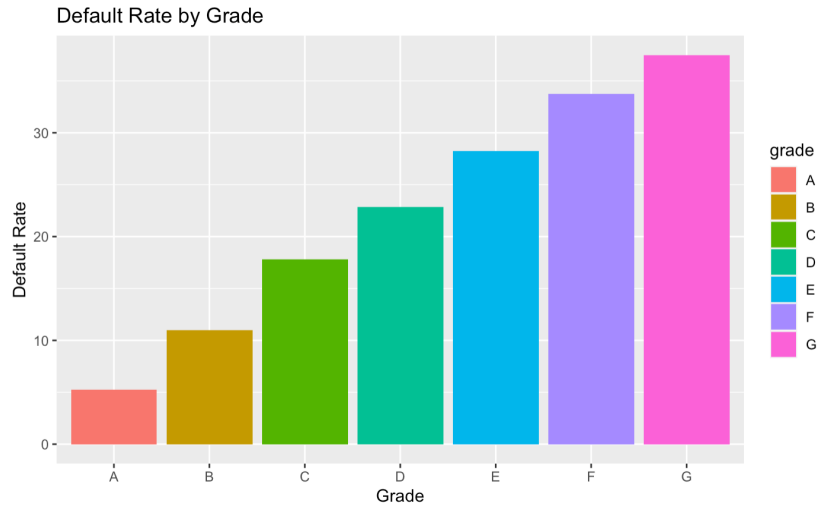
From our dataset, we find out that the number of loans that are charged off are 13785, whereas the number of loans that are fully paid are 86215.

	loan_status	n
	<chr>	<int>
1	Charged Off	13785
2	Fully Paid	86215

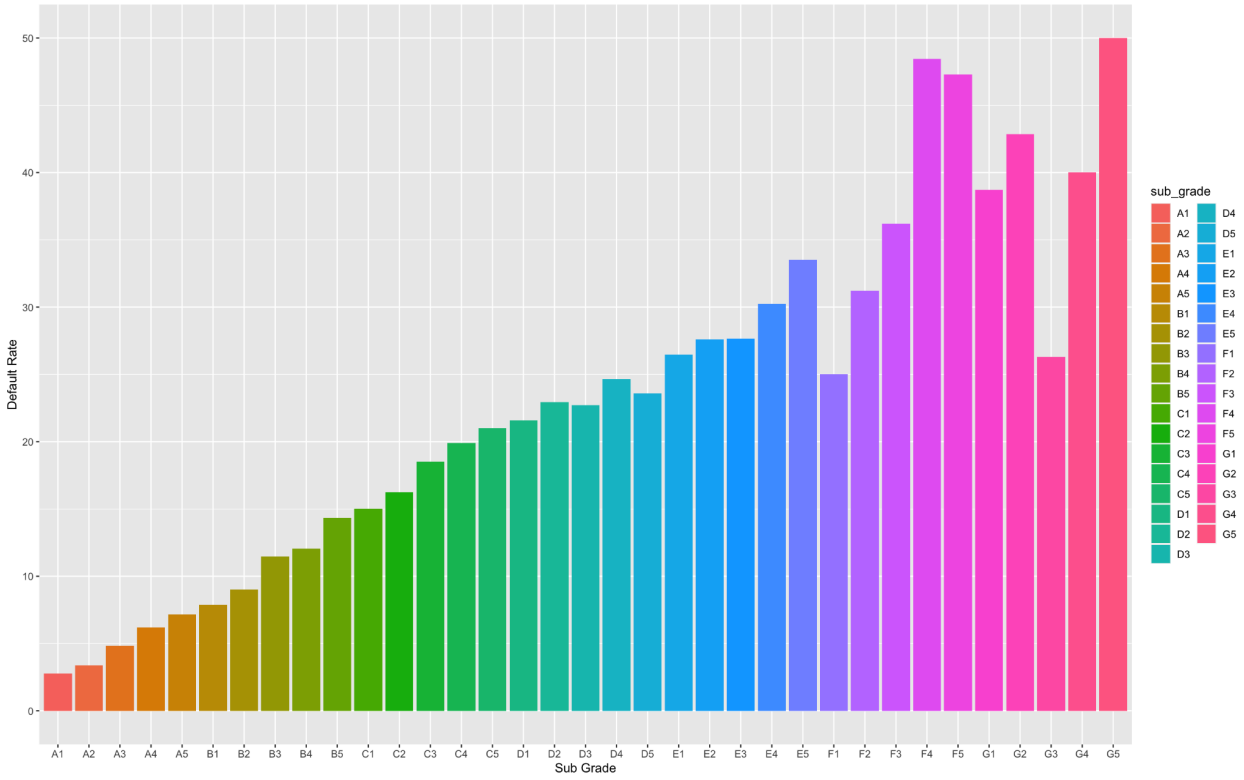
The proportion of defaults is calculated by dividing the number of loans that are charged off by the total loans in the marketplace. In this case, it is 13785/(13785+86215)= 13.785%.

From our analysis, we noticed how the default rate increases as the grade gets worse from A to G. We defined higher grade as the safer, which resulted in a lower default rate; and the lower grade as the risker, which resulted in a higher default rate.

	A	B	C	D	E	F	G
Charged Off	1187	3723	4738	2858	1010	239	30
Fully Paid	21401	30184	21907	9635	2569	469	50



Sub-grade also has a similar trend with grade in relation to the default rate, in which as the subgrade worsens, the higher the default rate, and vice versa. If we look in the bar chart, there are a few outliers such as F1 which has a lower default rate than all subgrades under grade E, and G3 which has a lower default rate than most subgrades under grade F. Overall, it is consistent with what we expected because we think higher grade borrowers have a more financial ability to pay back the loans.

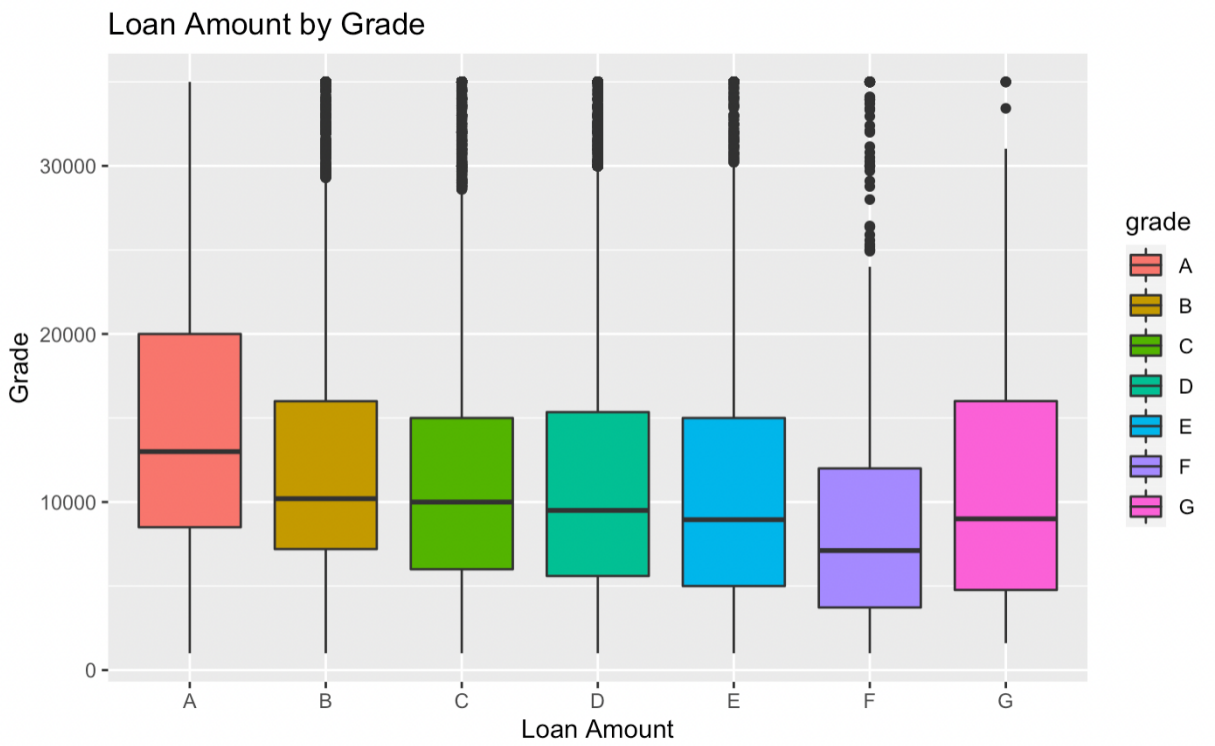


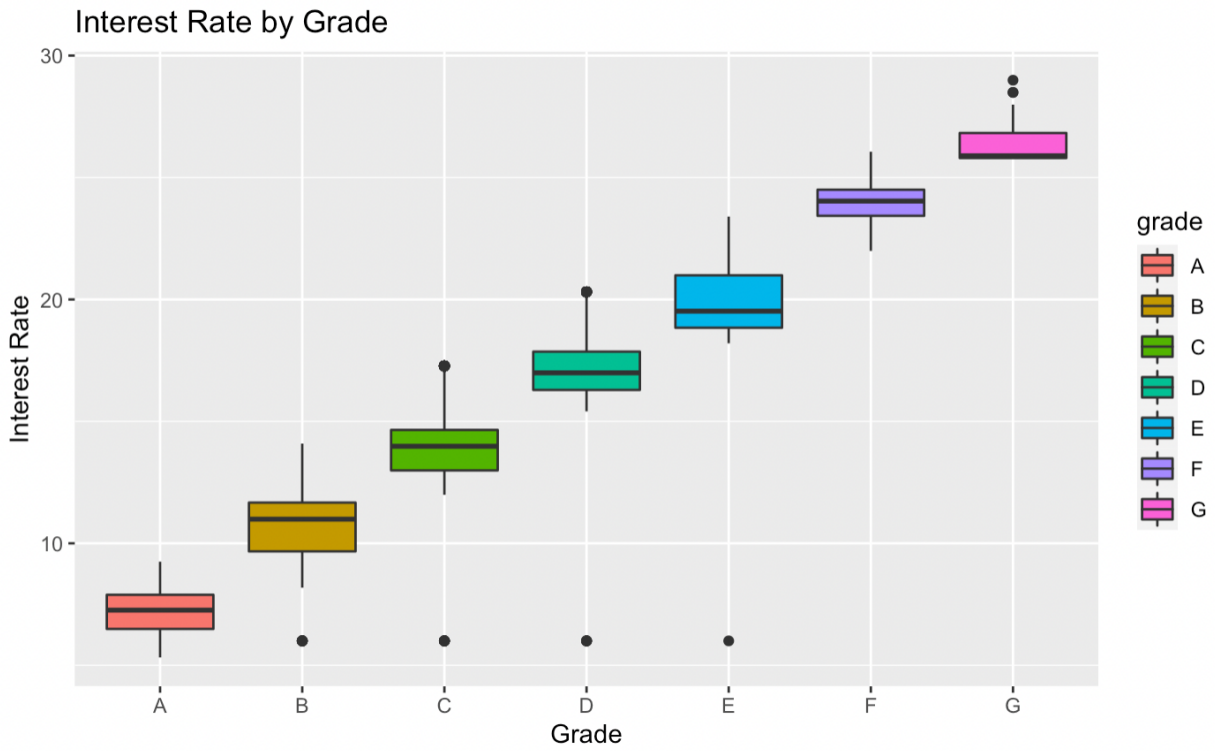
(ii) How many loans are there in each grade? And do loan amounts vary by grade? Does interest rate for loans vary with grade, subgrade? Look at the average, standard-deviation, min and max of interest rate by grade and subgrade. Is this what you expect, and why?

The amount of loans do vary by grade, and it is shown in pictures below, but we don't see any pattern based on the level of grade. The interest rate gets higher with lower grades or subgrades.

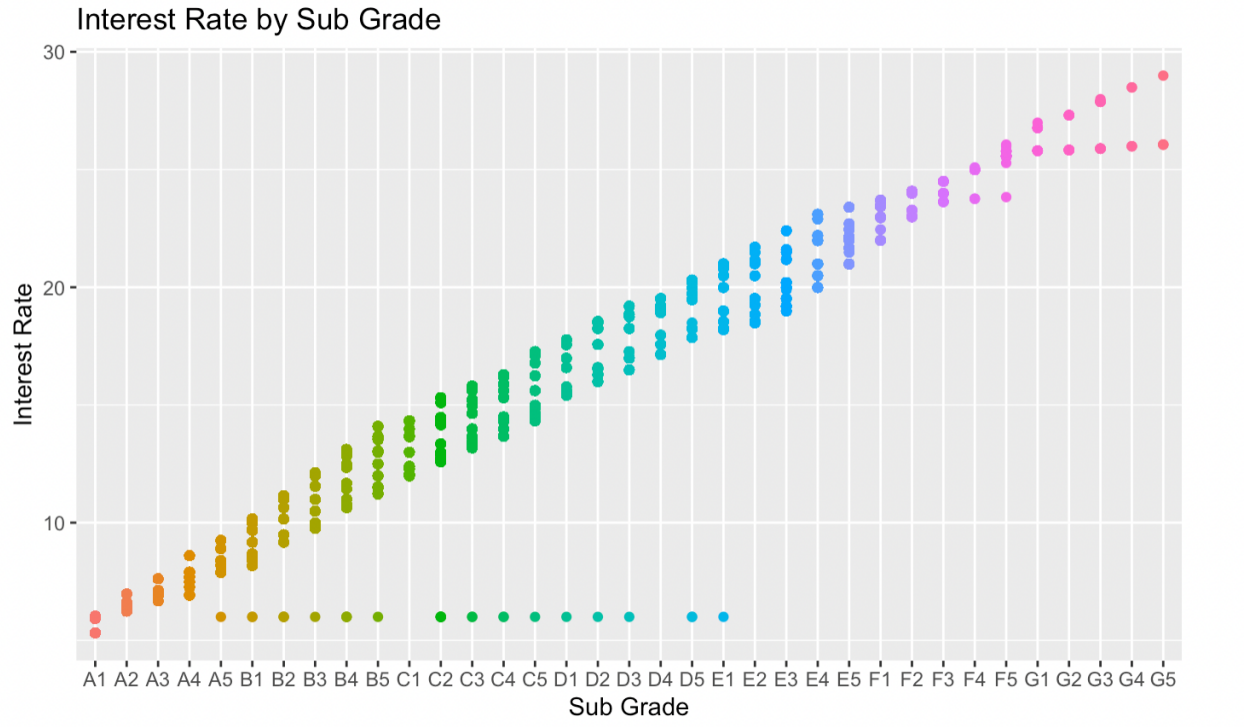
This is what we expect since loans with higher grades tend to be paid, which is why they get lower interest rates, and people with lower grades tend to default, hence the higher interest rates. By looking at the average, standard-deviation, min and max interest rate by grade and subgrade, the results are totally the same as what we expected. The interest rate follows the rule as we mentioned before, and the standard deviations part are almost in reasonable scope (around 1).

grade <chr>	nLoans <int>	defaults <int>	sumLoans <dbl>	avgInterest <dbl>	stdInterest <dbl>	minInterest <dbl>	maxInterest <dbl>	avgLoanAMt <dbl>
A	22588	1187	327649125	7.173848	0.9669664	5.32	9.25	14505.451
B	33907	3723	428494575	10.753559	1.4431575	6.00	14.09	12637.348
C	26645	4738	319762050	13.847765	1.1859154	6.00	17.27	12000.828
D	12493	2858	148590825	17.190576	1.2220189	6.00	20.31	11893.927
E	3579	1010	41583800	19.927656	1.3755560	6.00	23.40	11618.832
F	708	239	6564925	23.980438	0.9163869	21.99	26.06	9272.493
G	80	30	946075	26.425625	0.8490767	25.80	28.99	11825.938


Loan Amount by Grade
Grade
30000
20000
10000
0
A
B
C
D
E
F
G
Loan Amount
grade
A
B
C
D
E
F
G

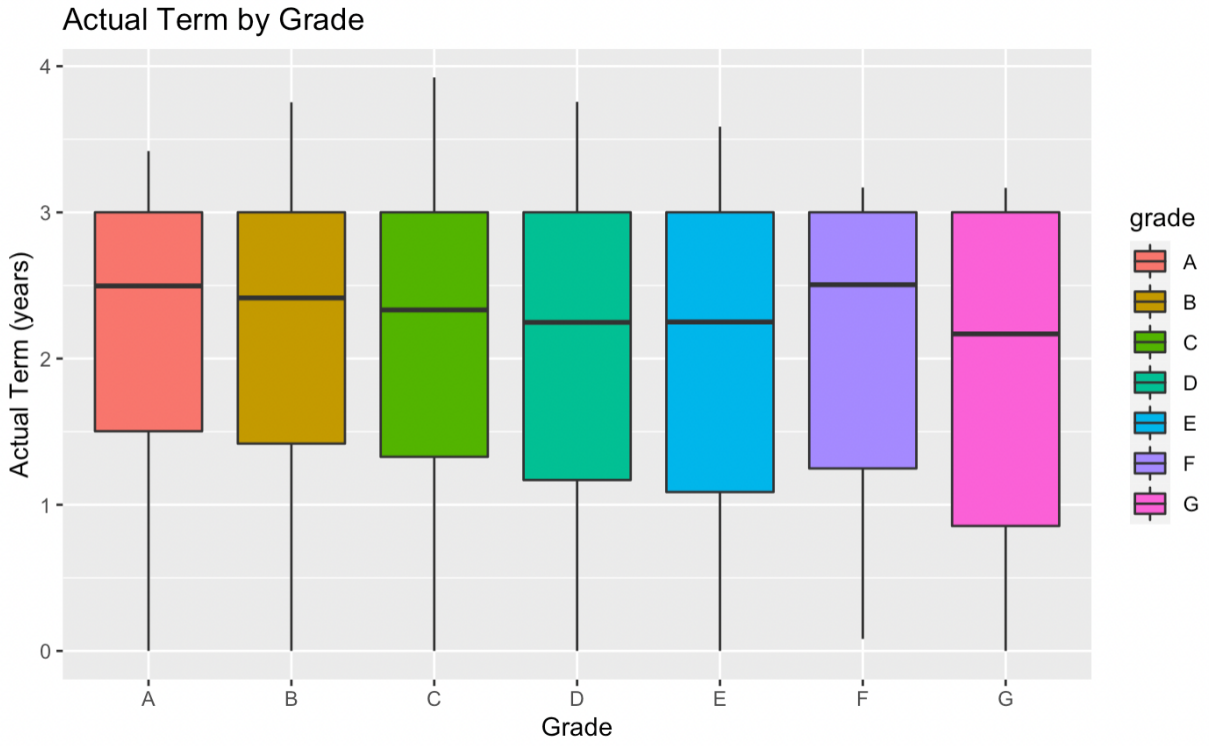

Interest Rate by Grade
Interest Rate
30
20
10
A
B
C
D
E
F
G
Grade
grade
A
B
C
D
E
F
G

sub_grade <chr>	nLoans <int>	defaults <int>	sumLoans <dbl>	avgInterest <dbl>	stdInterest <dbl>	minInterest <dbl>	maxInterest <dbl>	avgLoanAMt <dbl>
A1	3774	105	54621675	5.680069	0.3474851	5.32	6.03	14473.152
A2	3431	116	48499650	6.415494	0.1662589	6.24	6.97	14135.718
A3	3706	179	53865600	7.094107	0.3247008	6.68	7.62	14534.700
A4	5138	319	75401500	7.475851	0.3573953	6.92	8.60	14675.263
A5	6539	468	95260700	8.241788	0.4244667	6.00	9.25	14568.084
B1	6228	491	80444900	8.870010	0.7217524	6.00	10.16	12916.651
B2	6880	619	89162825	9.959382	0.8155856	6.00	11.14	12959.713
B3	7193	825	91849325	10.845931	0.8873289	6.00	12.12	12769.265
B4	7103	855	87767175	11.731457	0.8397941	6.00	13.11	12356.353
B5	6503	933	79270350	12.227378	0.8512147	6.00	14.09	12189.812



(iii) For loans which are fully paid back, how does the time-to-full-payoff vary? For this, calculate the 'actual term' (issue-date to last-payment-date) for all loans. How does this actual-term vary by loan grade (a box-plot can help visualize this).

Based on the chart below, in terms of fully paid back loans, they are paid within 2.5 years on average. There are no significant differences of actual terms between grades. However, based on the boxplot chart below, on average, people pay back all of the amount below 2.5 years. Therefore we can also conclude that if a loan is not paid after 2.5 years, it is less likely to be paid.



(iv) Calculate the annual return. Show how you calculate the percentage annual return.

We calculated the Annual Return as = [(Ending of year price - beginning of year price)/beginning of year price]*100

- For code:

```
lcdf$actualReturn<-ifelse(lcdf$actualTerm>0,((lcdf$total_pymnt-lcdf$funded_amnt)/lcdf$funded_amnt)*(1/lcdf$actualTerm), 0)
summary(lcdf$actualReturn)
```

Is there any return from loans which are ‘charged off’? Explain.

Based on the bar chart below, we can see negative returns from all grades. Furthermore, some charged off loans are paid back partially and also make profit (interest rates can be so high that the investors end up making money out of defaulted loans).

How does return from charged-off loans vary by loan grade?

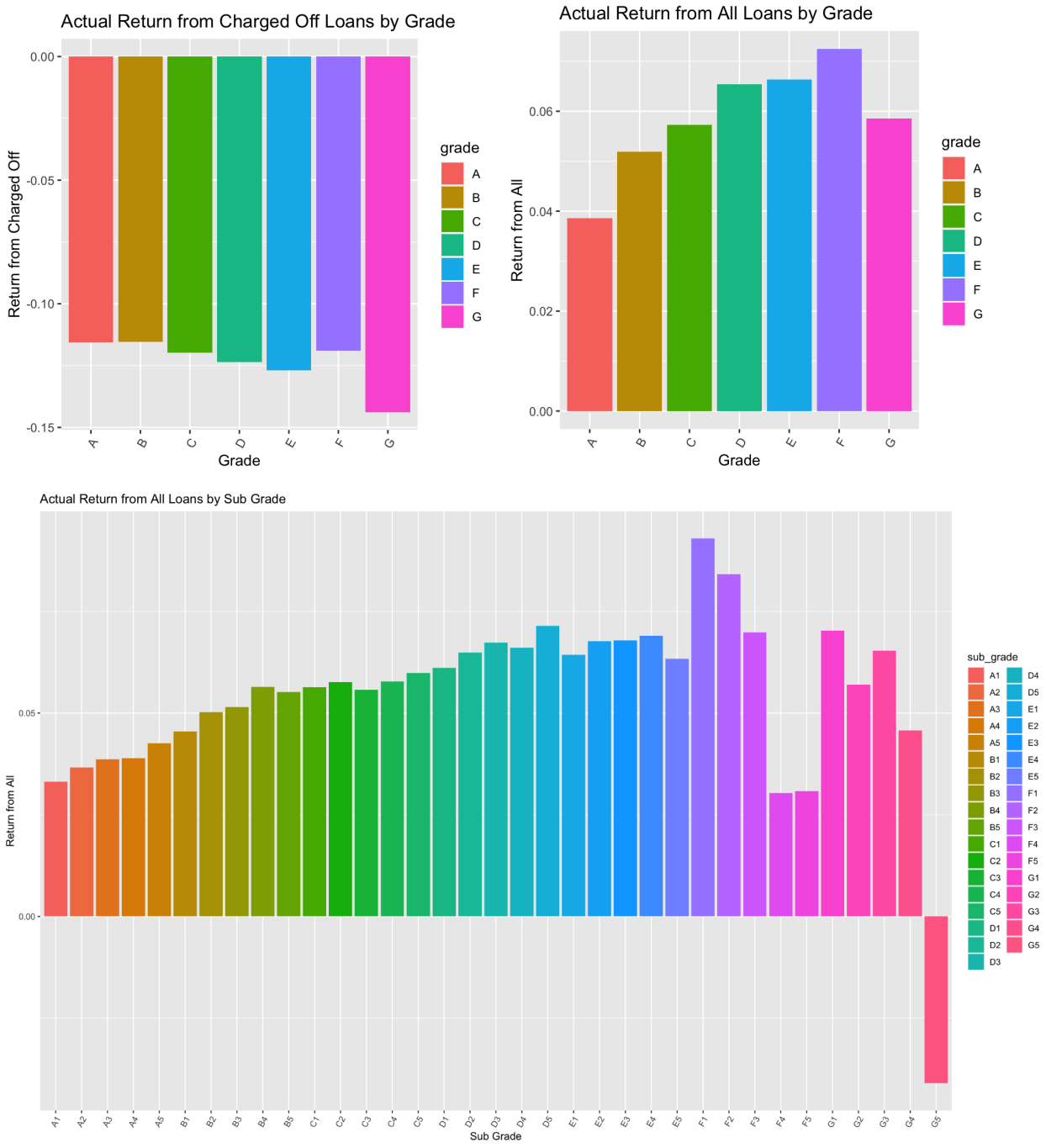
Based on the bar chart below, we see that the negative returns vary by a little by grade, which is observed from the normal distribution trend. The negative returns range between 10-12%.

Compare the average return values with the average interest-rate on loans – do you notice any differences, and how do you explain this? How do returns vary by grade, and by sub-grade.

From the bar chart below, the highest average return is seen on grade F, because even though the grade has a very high default risk, at the same time the interest rate for the grade is very high such that investors can possibly achieve higher return on these loans than of other grades. By sub grade, return also increases as subgrade goes worse from A1-F1, and decreases in trend afterwards. The highest average return is found on F4.

If you wanted to invest in loans based on this data exploration, which loans would you invest in?

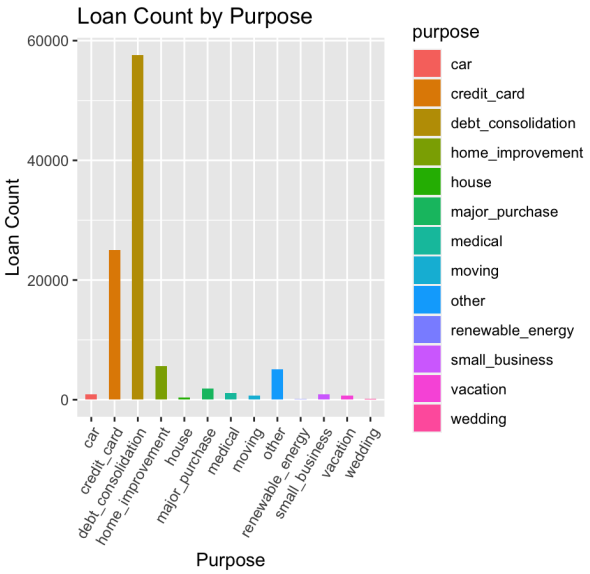
To achieve the highest return, we suggest investing in F1. If safety is the priority, we suggest taking A1 that has the lowest default risk.

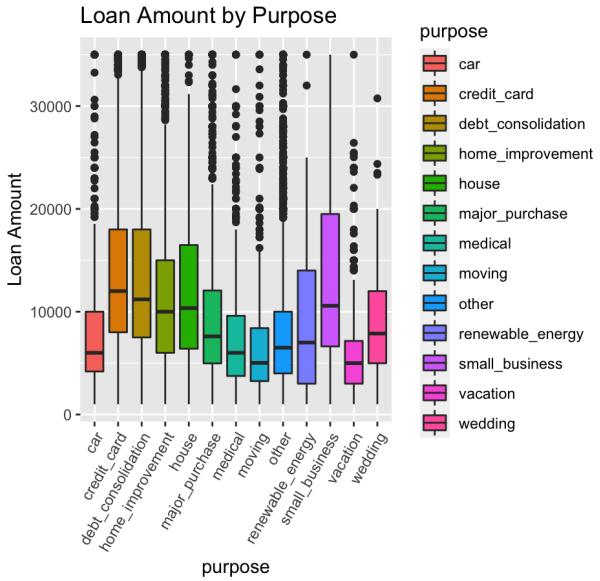


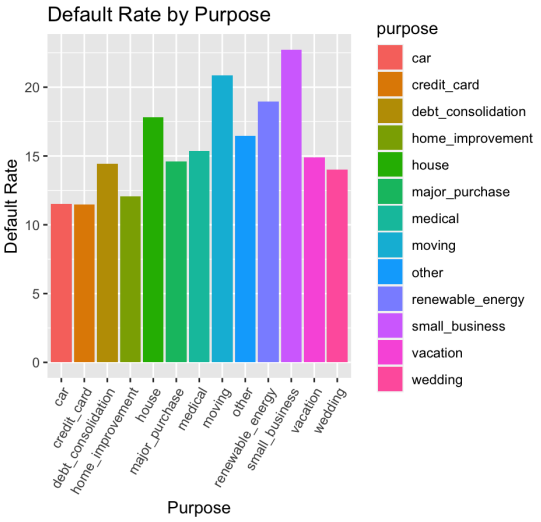
(v) What are people borrowing money for (purpose)? Examine how many loans, average amounts, etc. by purpose? Do loan amounts vary by purpose? Do defaults vary by purpose? Does loan-grade assigned by Lending Club vary by purpose?

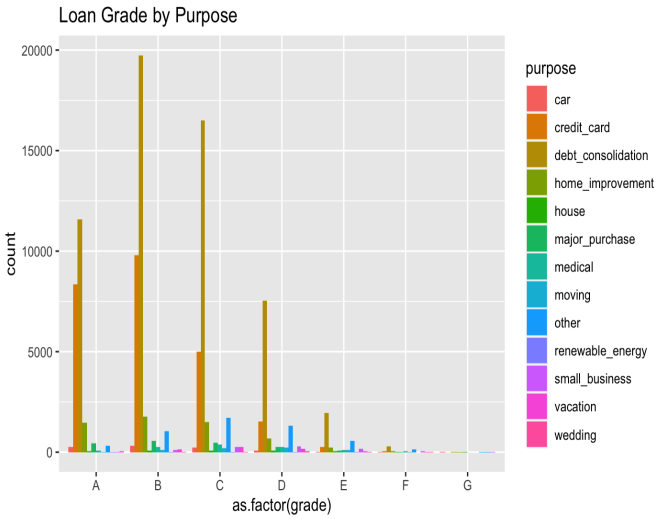
People are borrowing money for purposes such as payments for car, credit card, medical, house, etc with debt consolidation as the most common. We see from the bar chart below, that loan amounts vary by purpose. From the boxplot, we see that the average amount of loans vary by purpose. Also default rates vary by purpose, with small businesses having the

highest default rate. Furthermore, loan grade vary by purpose, with debt-consolidation being the most common purpose and grade B being the most common loan grade.





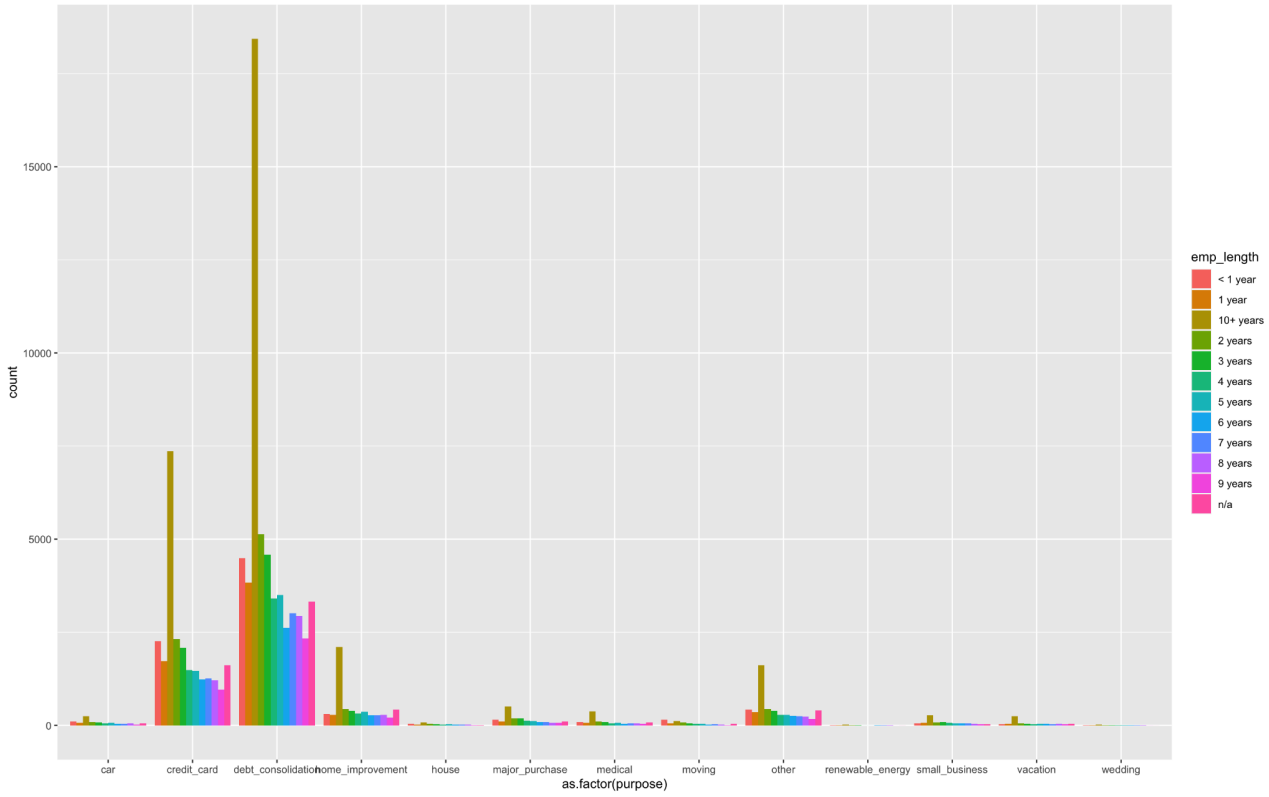




(vi) Consider some borrower characteristics like employment-length, annual-income, fico-scores (low, high). How do these relate to loan attributes like, for example, loan_amout, loan_status, grade, purpose, actual return, etc.

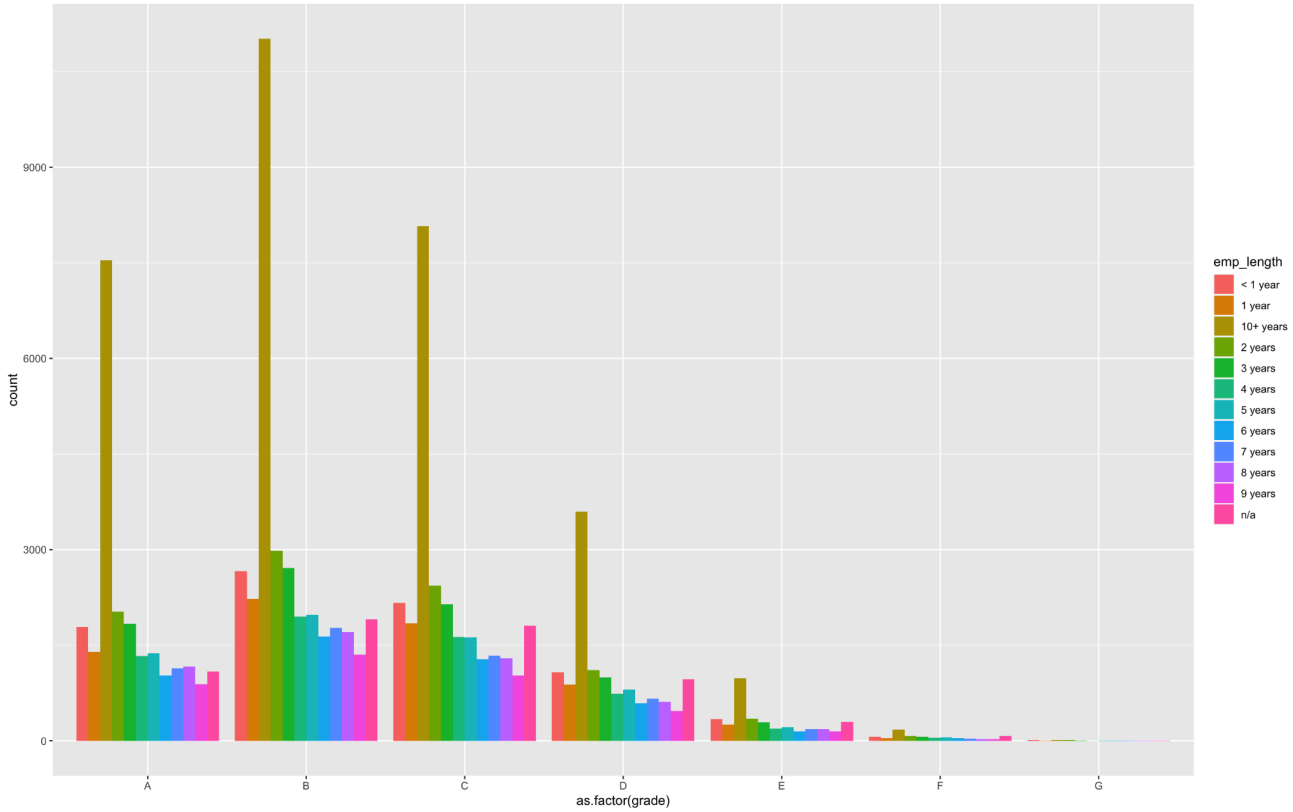
In this case, we consider a few borrower's profile and connect with some loan attributes

Employment length and loan purpose:



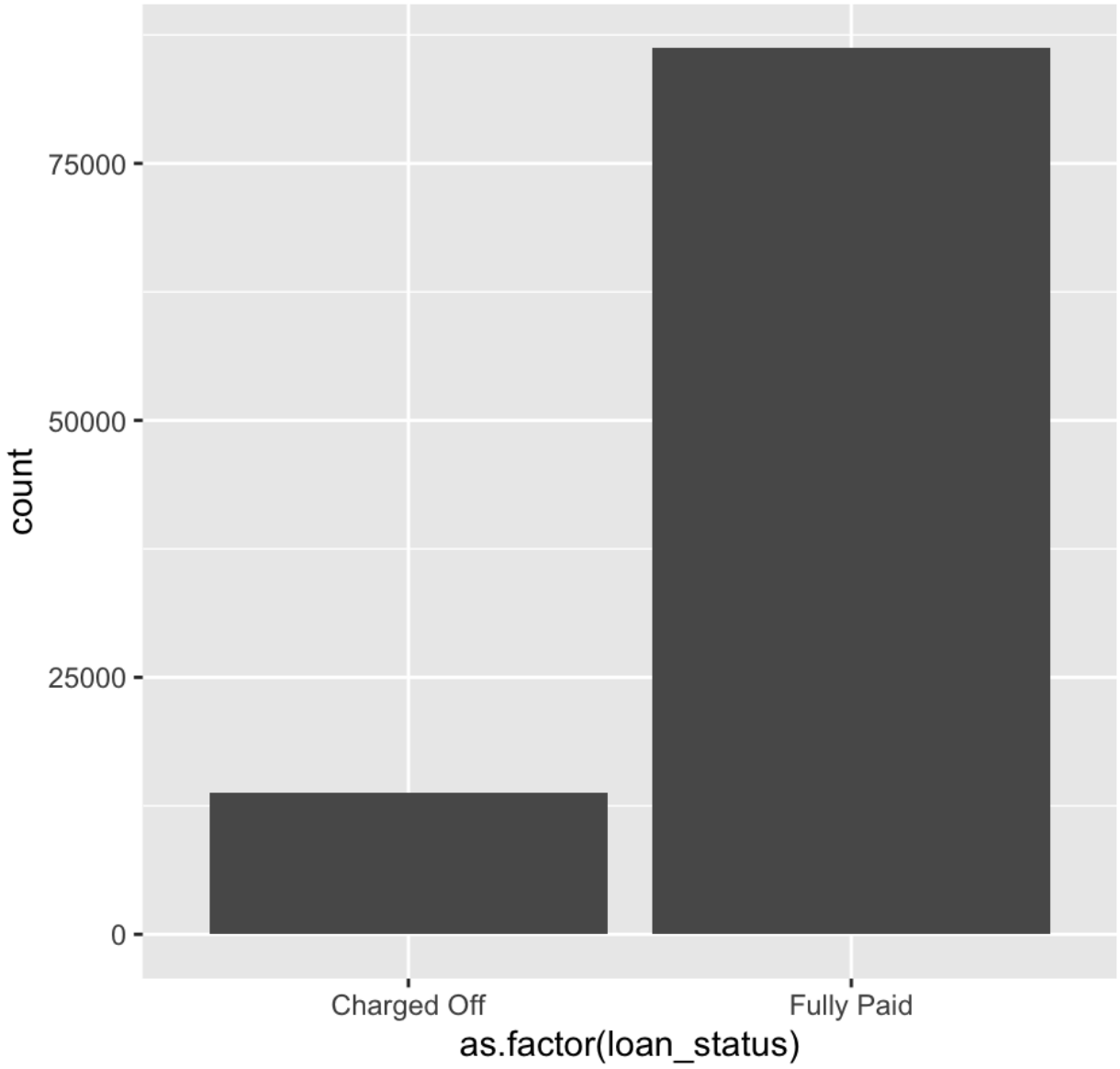
Analysis: In this bar chart, the highest number of loan amounts taken has an employment length that is higher 10 years. Furthermore, debt consolidation is the most common for the purpose of loan.

Employment length and grade:



Analysis: This bar chart shows that grade B is the most common loan taken, followed by A. The least is for loan grade G.

Total payment and loan status:



Analysis: The bar chart above shows that total payment is higher for fully paid loans when compared to charged off loans. This is because charged off loans are usually not being paid in full and therefore it has a higher number of recoveries as well to get the money back and to recover from bad debt.

(vii) Generate some (at least 3) new derived attributes which you think may be useful for predicting default., and explain what these are. For these, do an analyses as in the questions above (as reasonable based on the derived variables).

I. Proportion of satisfactory bankcard accounts - This attribute is basically the number of credit bank accounts a borrower holds. We assume that a higher proportion of satisfactory

bankcard accounts means that the borrower has a lot of credit on him, which inturn means that he/she is more likely to default.

Code:
lcdf$propSatisBankcardAccts <- ifelse(lcdf$num_bc_tl>0, lcdf$num_bc_sats/lcdf$num_bc_tl, 0)

II. The length of borrower's history with Lending Club - This attribute is basically checking the borrower's history/relationship with Lending Club. The longer the history of the person, the more likely this person is going to pay for his debts due to the longstanding relationship with Lending Club which shows that this person has a proven track record.

Code:
lcdf$earliest_cr_line<-paste(lcdf$earliest_cr_line, "-01", sep = "")
lcdf$earliest_cr_line<-parse_date_time(lcdf$earliest_cr_line, "myd")
lcdf$borrHistory <- as.duration(lcdf$earliest_cr_line %--% lcdf$issue_d) / dyears(1)

III. **Loan amount to annual income** - The **loan-to-annual income** ratio is a measure comparing the amount of loan with the annual income. We find that the borrowers who have high annual income do not default on loan, as they have the higher capacity to pay back the borrowed amount.

Code:
lcdf$propLoanAmt_to_AnnInc <- lcdf$loan_amnt/lcdf$annual_inc

(b) Are there missing values?

Yes there are missing values.

What is the proportion of missing values in different variables?
Explain how you will handle missing values for different variables. You should consider what the variable is about, and what missing values may arise from – for example, a variable monthsSinceLastDeliquency may have no value for someone who has not yet had a delinquency; what is a sensible value to replace the missing values in this case?
Are there some variables you will exclude from your model due to missing values?

In this part, we checked out the missing value, and we dropped the variables with all NAs. Basically, we remove variables which have more than 60% missing value because the data available is insufficient to predict missing values. First of all, we impute missing values with '-', to get the columns with missing values. For example, mths_since_last_delinq: has 48% missings, these pertain to non delinquency, so replace by a value higher than the max (500), we would try this out and put results in a temporary dataset lcx, with the attributes that have missing value.

Below information is the specific explanation for the variables we changed:

- bc_open_to_buy, use means because the number of NA is really low (1.2%).
- Replace na in last_credit_pull_d with a date older than 1 year(valid period). In this case we chose 1 Jan 2015.
- mo_sin_old_il_acct, use mean because number of NA is really low (3.8%)
- For mths_since_recent_bc, use max because NA because it means the person has never opened a bankcard acc before. So we assign a number that is the longest, or way above the max. #mths_since_recent_inq, use max because NA because it means no inquiry has been made. So we assign a number that is the longest, or way above the max.
- bc_util, use mean because number of NA is really low (1.2%)
- listnum_tl_120dpd_2m, use mean because number of NA is really low (2.6%)
- percent_bc_gt_75, use mean because number of NA is really low (1.2%)
- revol_util, use mean because number of NA is really low (.04%)
- emp_length, NA means 0 experience, so replace it with < 1 year
- After trying this out on the temporary dataframe lcx, if we are sure this is what we want, we can now replace the missing values on the lcdf dataset

#CHECK FOR NAs AGAIN # The last payment date missing are from 'Charger Off' where they didn't pay at all.

```
           emp_title                title mths_since_last_delinq            revol_util          last_pymnt_d     last_credit_pull_d
             0.06705              0.00012               0.49919               0.00041               0.00064               0.00004
         avg_cur_bal        bc_open_to_buy              bc_util    mo_sin_old_il_acct  mths_since_recent_bc  mths_since_recent_inq
             0.00002              0.00964               0.01044               0.03620               0.00911               0.10612
       num_rev_accts      num_tl_120dpd_2m        pct_tl_nvr_dlq      percent_bc_gt_75
             0.00001              0.03824               0.00016               0.01034
```

```
 emp_title           title           mths_since_last_delinq   revol_util      last_pymnt_d                  last_credit_pull_d  avg_cur_bal
Length:100000      Length:100000      Min.   :  0.00         Min.   :  0.00   Min.   :2013-02-01 00:00:00.00   Length:100000      Min.   :     0
Class :character   Class :character   1st Qu.: 15.00         1st Qu.: 36.20   1st Qu.:2016-01-01 00:00:00.00   Class :character   1st Qu.:  2791
Mode  :character   Mode  :character   Median : 31.00         Median : 54.10   Median :2016-12-01 00:00:00.00   Mode  :character   Median :  6312
                                      Mean   : 33.94         Mean   : 53.75   Mean   :2016-11-10 01:42:09.68                      Mean   : 12470
                                      3rd Qu.: 50.00         3rd Qu.: 71.80   3rd Qu.:2017-11-01 00:00:00.00                      3rd Qu.: 17267
                                      Max.   :188.00         Max.   :153.70   Max.   :2019-02-01 00:00:00.00                      Max.   :395953
                                      NA's   :49919          NA's   :41       NA's   :64                                         NA's   :2
bc_open_to_buy      bc_util       mo_sin_old_il_acct mths_since_recent_bc mths_since_recent_inq num_rev_accts    num_tl_120dpd_2m pct_tl_nvr_dlq
Min.   :     0   Min.   :  0.00   Min.   :  0        Min.   :  0.0        Min.   : 0.000       Min.   : 2.00   Min.   :0.000    Min.   : 20.00
1st Qu.:  1203   1st Qu.: 42.30   1st Qu.: 96        1st Qu.:  6.0        1st Qu.: 2.000       1st Qu.: 9.00   1st Qu.:0.000    1st Qu.: 91.00
Median :  3893   Median : 66.20   Median :128        Median : 14.0        Median : 5.000       Median :13.00   Median :0.000    Median : 97.80
Mean   :  9046   Mean   : 62.45   Mean   :125        Mean   : 24.5        Mean   : 6.908       Mean   :14.76   Mean   :0.001    Mean   : 94.05
3rd Qu.: 10602   3rd Qu.: 86.20   3rd Qu.:152        3rd Qu.: 29.0        3rd Qu.:10.000       3rd Qu.:19.00   3rd Qu.:0.000    3rd Qu.:100.00
Max.   :332178   Max.   :188.80   Max.   :640        Max.   :616.0        Max.   :24.000       Max.   :87.00   Max.   :2.000    Max.   :100.00
NA's   :964      NA's   :1044     NA's   :3620       NA's   :911          NA's   :10612        NA's   :1       NA's   :3824     NA's   :16
percent_bc_gt_75
Min.   :  0.00
1st Qu.: 16.70
Median : 50.00
Mean   : 48.03
3rd Qu.: 75.00
Max.   :100.00
NA's   :1034
```

(c) Consider the potential for data leakage. You do not want to include variables in your model which may not be available when applying the model; that is, some data may not be available for new loans before they are funded. Leakage may also arise from variables in the data which

may have been updated during the loan period (ie., after the loan is funded). Identify and explain which variables will you exclude from the model.

We considered the potential data leakage in the next variables because they have a highly correlated data, specially the actual return with 0.98 .So, we should avoid to work with this variables and the others such as last payment amount, total payment , actual term , interested rate, Number of revolving trades opened in past 24 months or open_rv_24m and Number of revolving trades opened in past 12 months or open_rv_12m.Because leakage will cause our model to become very inaccurate and that is why we should avoid it.

3. Do a univariate analyses to determine which variables (from amongst those you decide to consider for the next stage prediction task) will be individually useful for predicting the dependent variable (loan_status). For this, you need a measure of relationship between the dependent variable and each of the potential predictor variables. Given loan-status as a binary dependent variable, which measure will you use?
From your analyses using this measure, which variables do you think will be useful for predicting loan_status?
(Note – if certain variables on their own are highly predictive of the outcome, it is good to ask if this variable has a leakage issue).
We will next develop predictive models for loan_status.

To measure the relationship between the binary dependent variable and the potential predictor variable, we need to use AUC analysis as the dependent variable. The following measures may be essential for predicting loan status.

First of all, we know that in the AUC function, the predictor variable has to be numeric. Then, we determine which variables yield an AUC > 0.5. Finally, we sort them in descending order to see which variables are better suitable for our predictive model.

The below table shows the AUC values for the potential variables versus loan status. The higher the AUC, the better the variable can predict the loan status. Based on the below table, we see that the variables actualReturn and totReturn have the highest AUC values, however these variables won't be available at the beginning of the credit analysis evaluation, so we cannot use them as predictors as they might cause leakage. Therefore, variables such as actualTerm, int_rate and annual_inc and among others will be useful as predictors.

	names	x
1	actualReturn	0.9859487

2	totReturn	0.9659158
3	actualTerm	0.6639332
4	int_rate	0.6581483
5	annual_inc	0.5767804
6	tot_hi_cred_lim	0.5735512
7	total_bc_limit	0.5730079
8	avg_cur_bal	0.5691553
9	dti	0.5682696
10	tot_cur_bal	0.5611950
11	mort_acc	0.5583196
12	propLoanAmt_to_AnnInc	0.5506494
13	bc_util	0.5431050
14	borrHistory	0.5396895

15	revol_bal	0.5367332
16	emp_length	0.5344576
17	revol_util	0.5314717
18	propSatisBankcardAccts	0.5253335
19	loan_amnt	0.5211402
20	total_acc	0.5184907
21	num_op_rev_tl	0.5176556
22	total_bal_ex_mort	0.5169192
23	mths_since_last_delinq	0.5121309
24	num_rev_accts	0.5078333
25	num_sats	0.5077449
26	installment	0.5071865

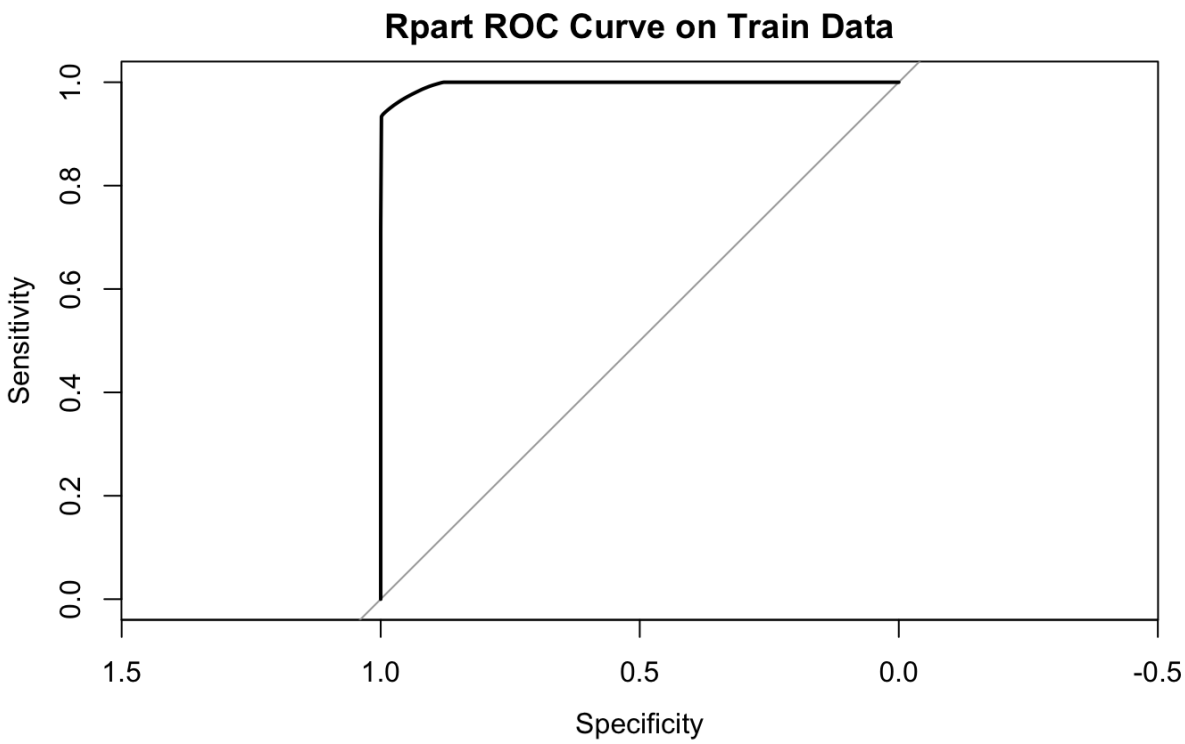
Answers below are for Q4. and Q5.
4(a) Split the data into training and validation sets. What proportions do you consider, why?
4(b) How will you evaluate performance – which measure do you consider, and why?

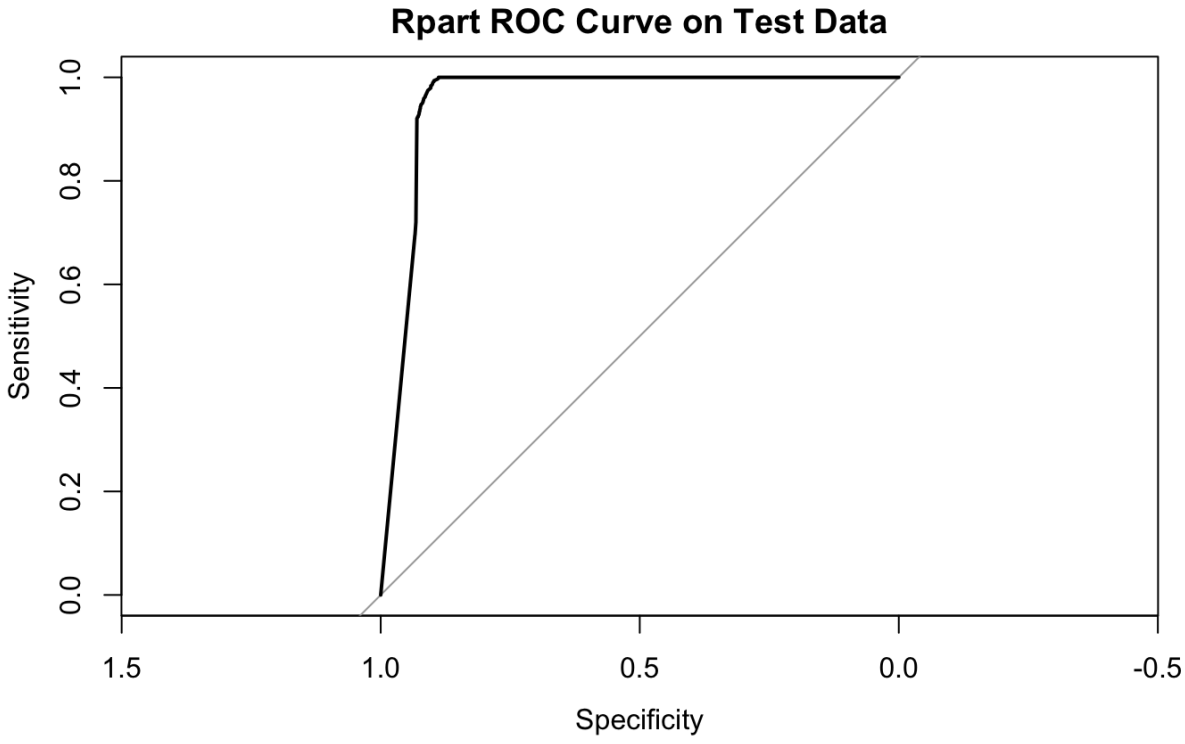
For evaluation of models, you should include confusion matrix related measures, as well as ROC analyses and lifts. Explain which performance measures you focus on, and why.
5. Develop a decision tree model to predict default.
Train decision tree models (use either rpart or c50)
What parameters do you experiment with, and what performance do you obtain (on training and validation sets)? Clearly tabulate your results and briefly describe your findings.
[If something looks too good, it may be due to leakage – make sure you address this]
Identify the best tree model. Why do you consider it best?
Describe this model – in terms of complexity (size).
Examine variable importance. How does this relate to your univariate analyses in Question 3 above?
Briefly describe how variable importance is obtained (the process used in the decision tree learning algorithm you use(rpart or c50).

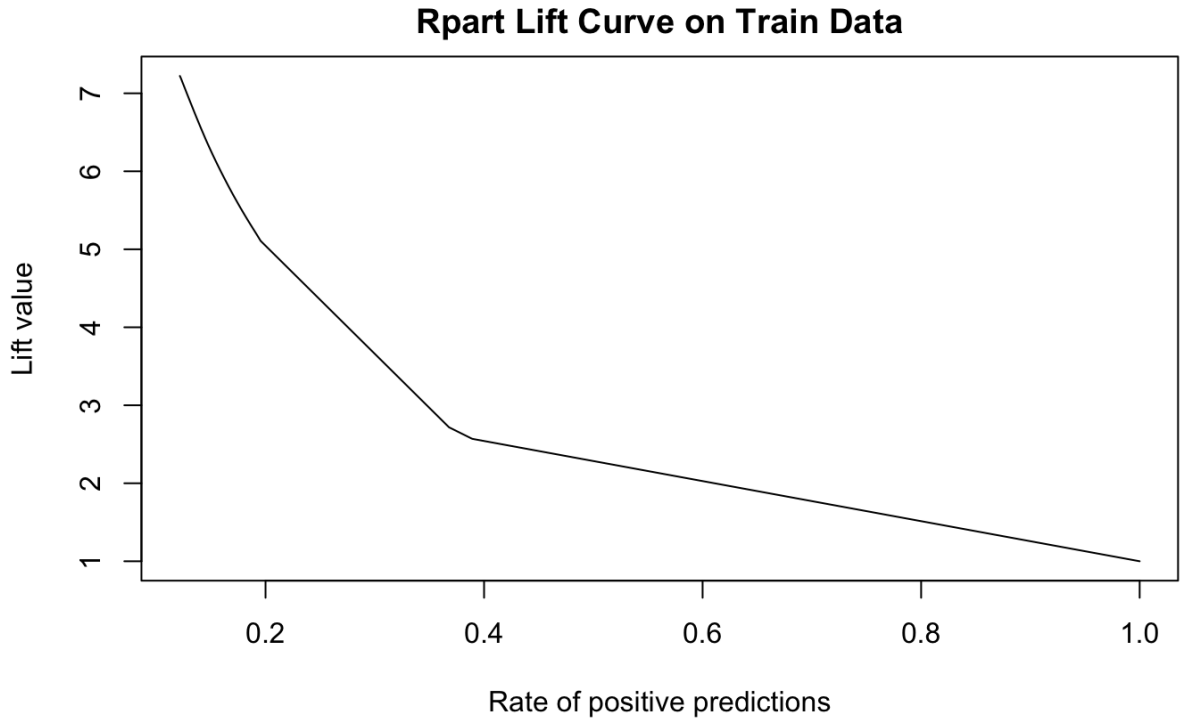
We thought the most suitable proportions for training and validation sets would be 70% and 30%. It is not ideal for using a low proportion of training data because a model should have enough training sets to train itself in order to increase the accuracy and predictive power of the model. However, the validation sets can't be too much as well, or it might disturb the results of training and have unreasonable outcomes.

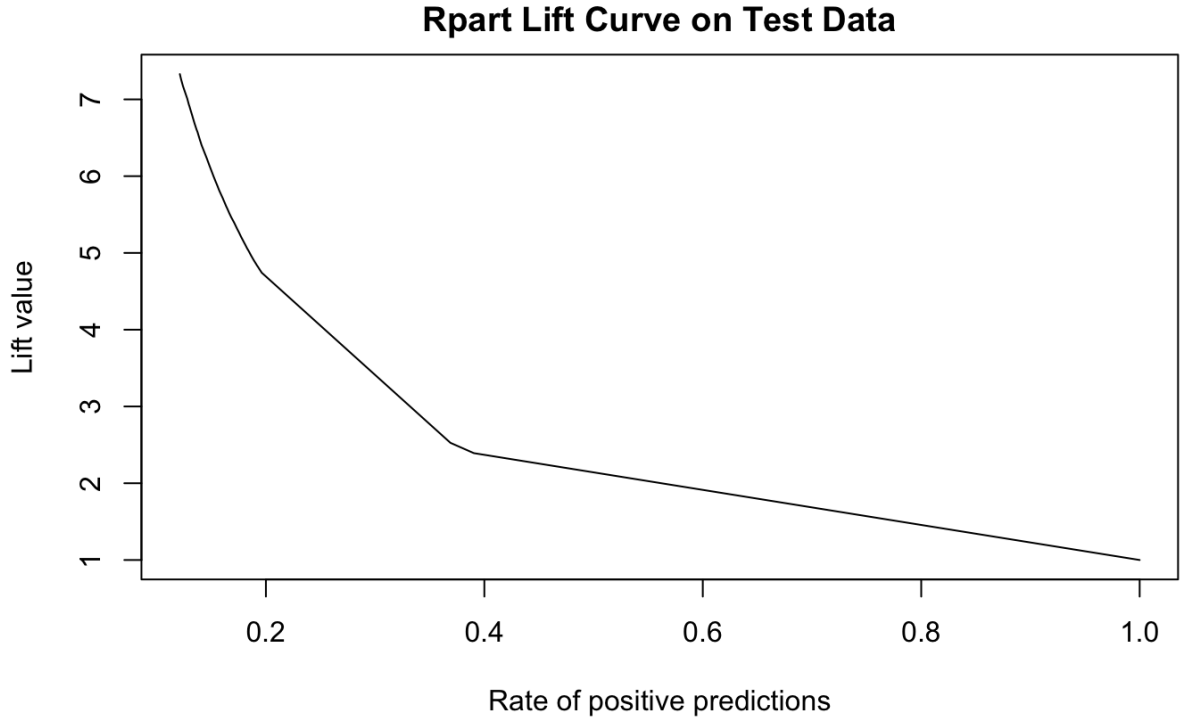
```
[1] 70000   148
[1] 30000   148
```

For our analysis, we decided to use the rpart package in developing the decision tree. To develop the decision tree, we decided to set various values such as the CP, prior, minsplit, and split (gini/information). We found out that the data is unbalanced (82.9% for Fully Paid, 17.1% for Charged Off on Training set), and this is a problem as we try to develop the tree. We couldn't have the tree to split, as describing every tree as "Fully Paid" will result in a 17.1% error rate. Therefore, we tried to tweak the CP parameter in order to split it, but as a result, a very odd CP table occurred; xerror increased instead of decreased. Therefore, we decided to set the 'prior' to 0.5 and 0.5; as a result, xerror is going down. Regardless, we weren't able to lower the overall relative error to less than 0.50358. Also, we can't reduce xerror lower than 0.79532 even with the model having a high accuracy), which in this case is the model with 'prior' (0.5, 0.5) in the name rpDT1. For model 2, which is the unbalanced data, we set 'prior' to 1-0.171=0.829, 0.171. As a result, a similar occurred when the resulting CP went back to another odd CP table, in which xerror increases as relative error decreases. But one thing we found out is that the accuracy from this model that is saved in rpDT2 is the highest, which is 70% on training data and 63% on test data. Therefore, rpDT2 (Decision Tree 2) is the better model. To provide more context on rpDT2, the ROC Curve and the Lift Curve of Decision Tree Model 2, for Train and Test data is stated below.


Rpart ROC Curve on Train Data
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity


Rpart ROC Curve on Test Data
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.5
1.0
0.5
0.0
-0.5
Specificity


Rpart Lift Curve on Train Data
Lift value
1
2
3
4
5
6
7
0.2
0.4
0.6
0.8
1.0
Rate of positive predictions

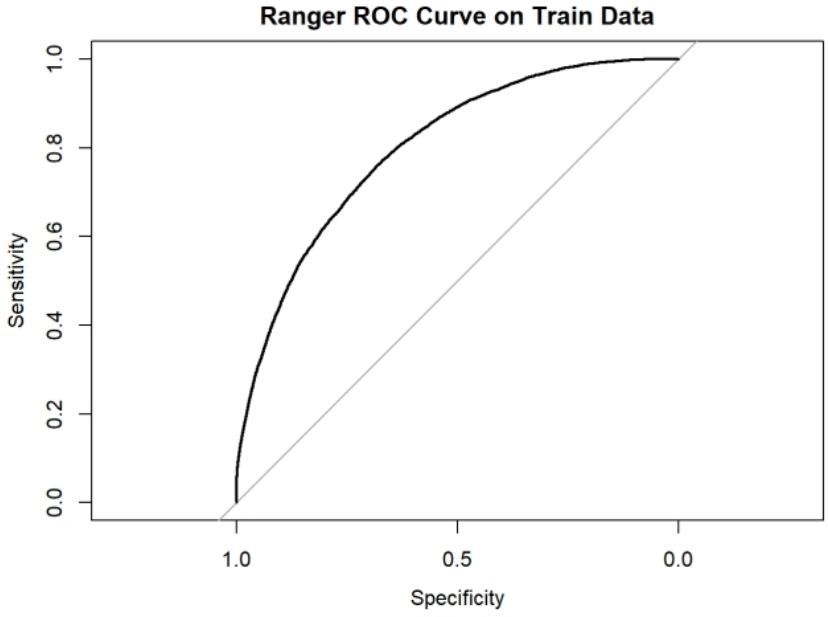

Rpart Lift Curve on Test Data
Lift value
1
2
3
4
5
6
7
0.2
0.4
0.6
0.8
1.0
Rate of positive predictions

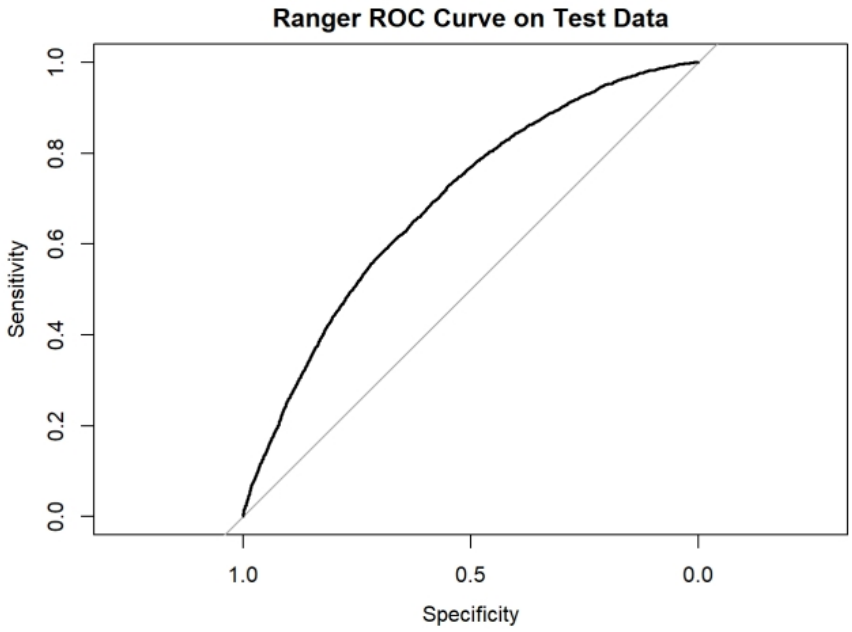
6. (a) Develop random forest and boosted tree model (using gbm or xgb)
Note the ‘ranger’ library and xgb can give faster computations.

What parameters do you experiment with, and how does this affect performance?
Describe the best random forest and boosted tree model in terms of number of trees, performance, variable importance.

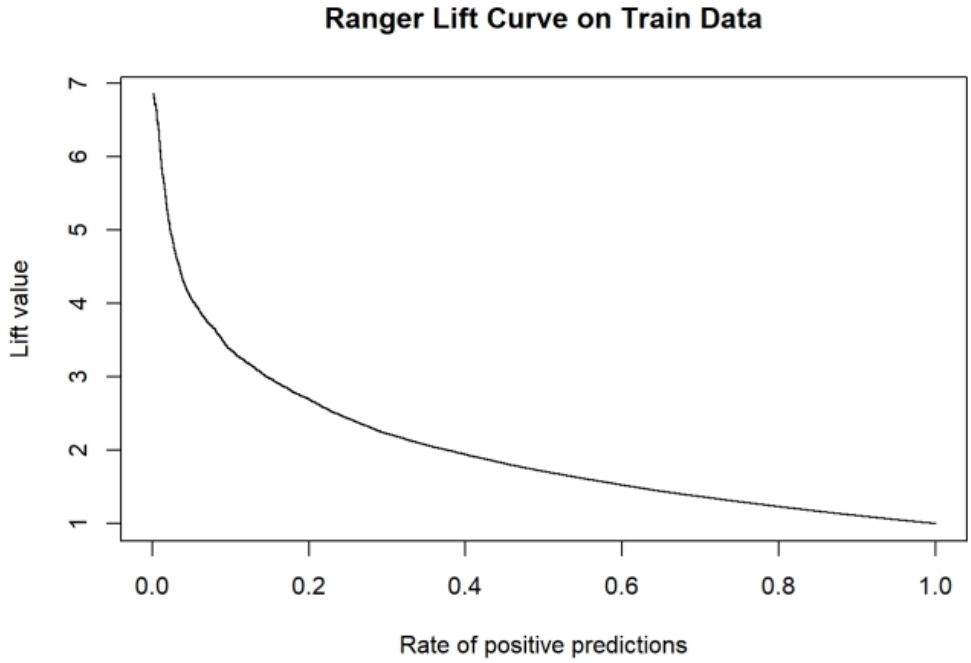
(b)Compare the performance of random forest, boosted tree and decision tree model from Q 5 above. Do you find the importance of variables to be different ?
Which model would you prefer, and why ?

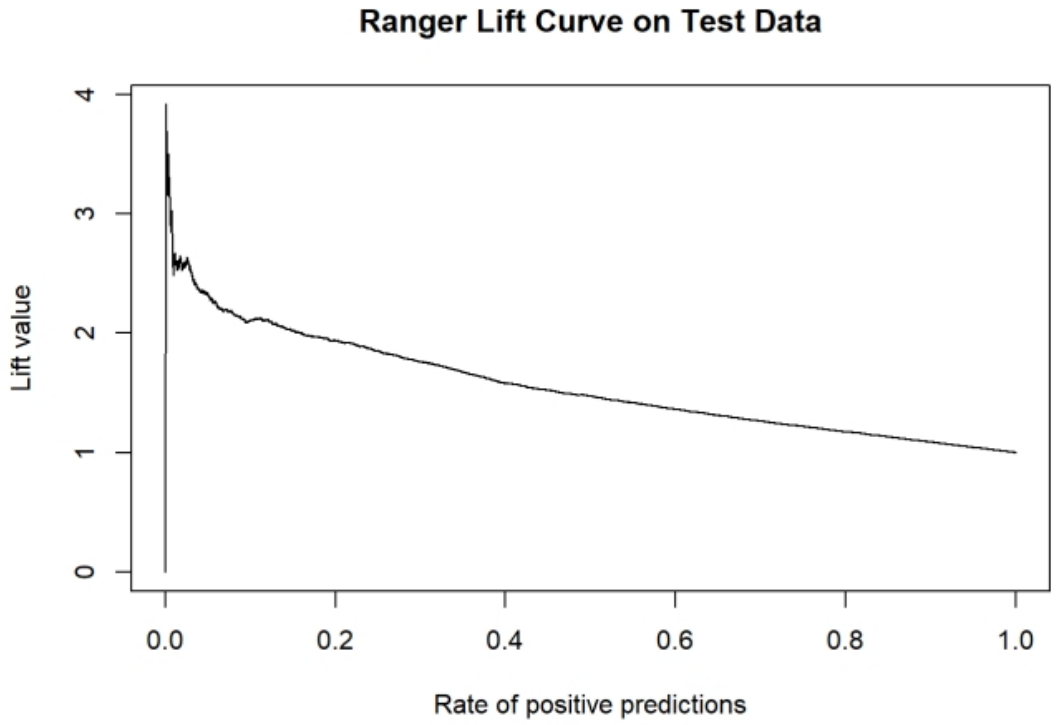
In this question, we found out that our first random forest model is overfitting. As a result, we decreased our number of trees from 1000 to 500, trying to reduce the overfitting rate. After trying many times to build some reasonably random forest, we found that the most suitable number of trees is around 200~400. Moreover, we also try to decrease the max depth and increase the min node. Finally, we pick one best model from our priority three models, because it’s simplest and has the same testing accuracy. The final model is shown in rgModel1. The evaluation of rgModel1 is done with AUC, ROC, confusion matrices, and lift curve. Also, we found out some rules we can follow to reduce the overfitting problem. That is, the number of trees should be around ten times the number of variables. Moreover, we set a higher mtry, in order to reduce the noisy predictors. Lastly, we think that we should build easier trees for looking at higher values of mtry. To be more specific, we limit our trees' depth and increase the size of nodes.





The lift curve shows that our ranger model has a predicting value, although the results are not exponentially better than no model scenario. This is consistent with the accuracy depicted in the ROC curve.





7.The purpose of the model is to help make investment decisions on loans. How will you evaluate the models on this business objective? Consider a simplified scenario - for example, that you have $100 to invest in each loan, based on the model's prediction. So, you will invest in all loans that are predicted to be 'Fully Paid'. Key questions here are: how much, on average, can you expect to earn after 3 years from a loan that is paid off, and what is your potential loss from a loan that has to be charged off ?

One can consider the average interest rate on loans for expected profit – is this a good estimate of your profit from a loan? For example, suppose the average int_rate in the data is 11.2%; so after 3 years, the $100 will be worth (100 + 3*11.2) = 133.6, i.e a profit of $33.6. Now, is 11.2% a reasonable value to expect – what is the return you calculate from the data? Explain what value of profit you use.

For a loan that is charged off, will the loss be the entire invested amount of $100? The data shows that such loans have do show some partial returned amount. Looking at the returned amount for charged off loans, what proportion of invested amount can you expect to recover? Is this overly optimistic? Explain which value of loss you use.

You should also consider the alternate option of investing in, say in bank CDs (certificate of deposit); let's assume that this provides an interest rate of 2%. Then, if you invest $100, you will receive $106 after 3 years (not considering reinvestments, etc), for a profit of $6.

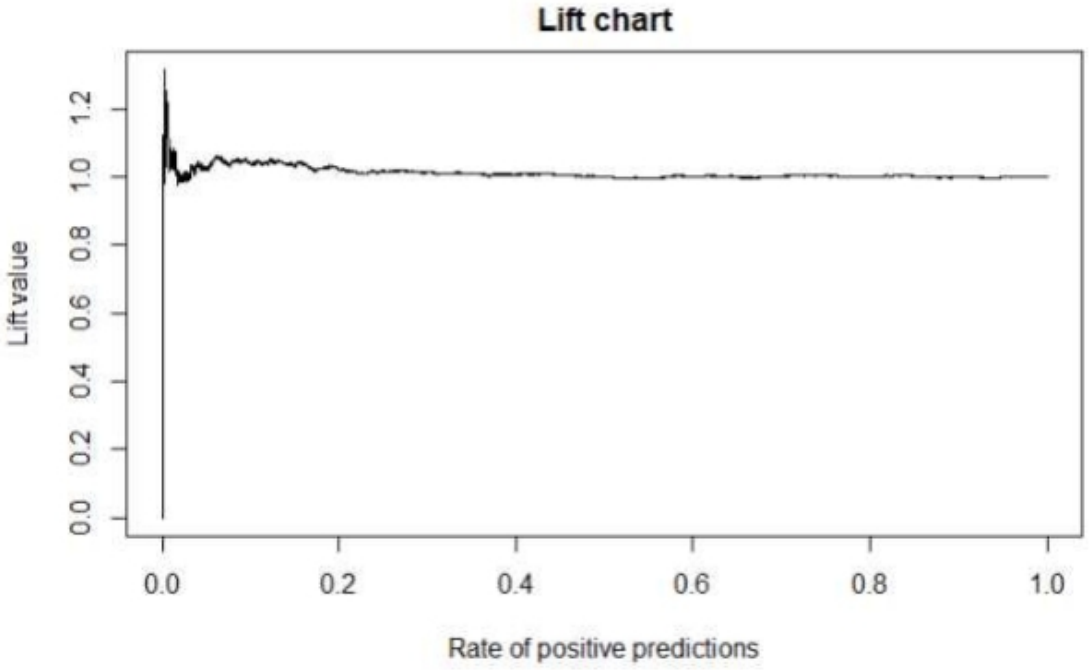
Considering a confusion matrix, we can then have profit/loss amounts with each cell, as follows:

		Predicted	
		FullyPaid	ChargedOff
Actual	FullyPaid	*profitValue*	$6
	ChargedOff	*lossValue*	$6

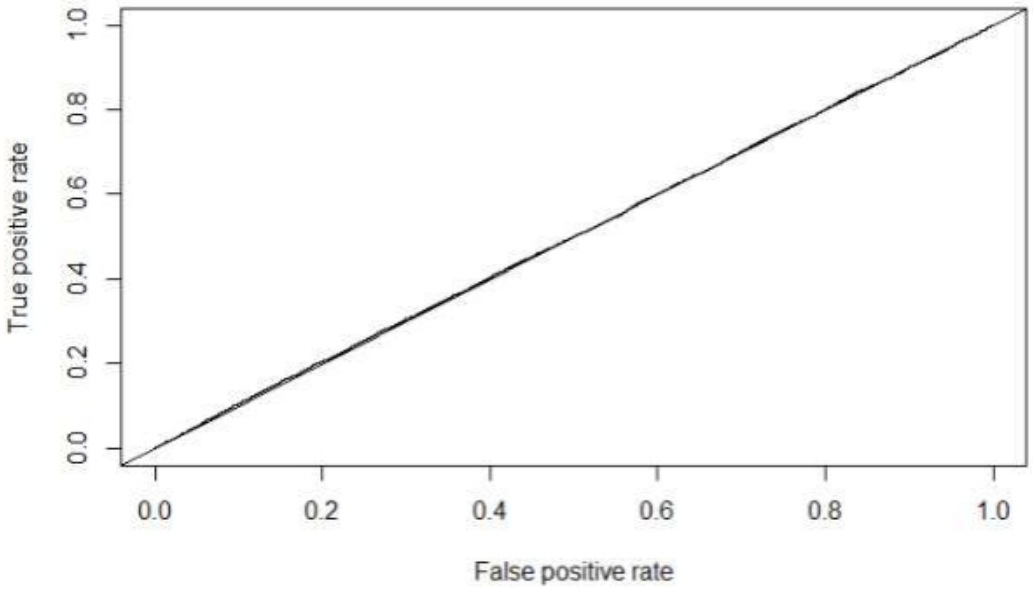
(a) Compare the performance of your models from Questions 5, 6 above based on this. Note that the confusion matrix depends on the classification threshold/cutoff you use. Which model do you think will be best, and why.
(b) Another approach is to directly consider how the model will be used – you can order the loans in descending order of prob(fully-paid). Then, you can consider starting with the loans which are most likely to be fully-paid and go down this list till the point where overall profits begin to decline (as discussed in class). Conduct an analyses to determine what threshold/cutoff value of prob(fully-paid) you will use and what is the total profit from different models – decision tree, random forest, boosted trees. Also compare the total profits from using a model to that from investing in the safe CDs. Explain your analyses and calculations.
Which model do you find to be best and why. And how does this compare with what you found to be best in part (a) above.

In this question, we found out that the fully paid loan has an actual term of around 2 years and a charged off loan has a term of 3 years. With the actual term, we can easily calculate the actual return. In “Fully Paid ” scenario, the actual term is around 2 years and the avg actual interest rate is around 8. We can assume that the amount of investment is $100, then the profit in the investment of the loan is $ 16. Moreover, if we consider the risk free return is 2% for the remaining 1 year, the amount turns into $ 2. As a result, the total profit becomes $16 + $2 = $18, in each fully paid situation. Similarly, the loss value by investing in a loan which could be default comes out to be equal to 11.7 times. So we can know the actual return is 11.7, and the actual term for default loans is 3. This makes our loss value equal to $35 for each loan. We have used random forest as the selected model for this analysis on account of its ability to predict the default loans correctly. The graph for cumulative profit is as follows:

loan_status <chr>	avgInt <dbl>	avgActInt <dbl>
Charged Off	13.85188	-11.962532
Fully Paid	11.71333	8.020985



AUC



The best model will give predictions of loan status fully paid > 0.7 , in a confusion matrix and also because we omitted variables that might cause data leakage .

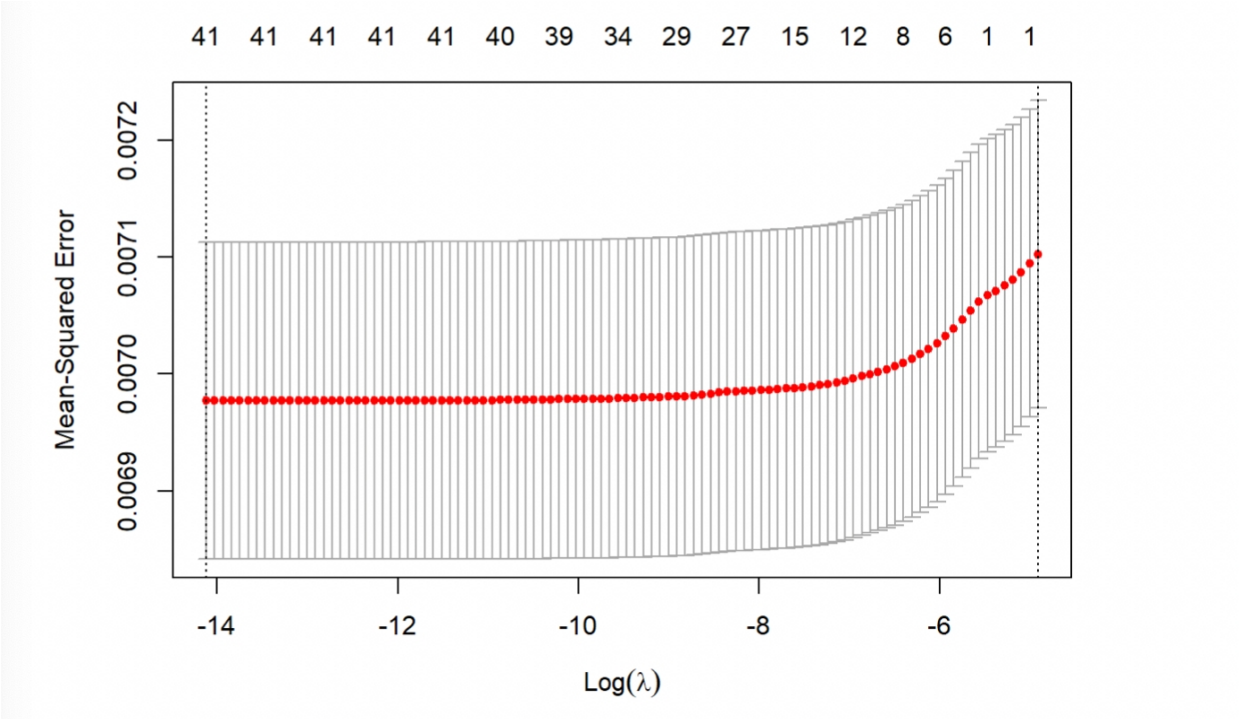
Part B: predictive models for loans with high returns

8. Develop models to identify loans which provide the best returns. Explain how you define returns? Does it include Lending Club's service costs?
Develop glm, rf, gbm (xgb) models for this. Show how you systematically experiment with different parameters to find the best models. Compare model performance.

We think that "Return" is the profit earned by the investors. The formula is calculated as total payment minus the investment money, and then divided by the total funded amount. However, we still think that it was not that clear, so we multiply it by the actual term of loan. Yet, it has one condition which needs to be followed. That is, the loan must be paid back. Otherwise, the calculation doesn't have any meaning. Also, we found out that "Return" does not account for Lending Club's service fees. "Lending Club makes money by charging borrowers an origination fee and investors a service fee.

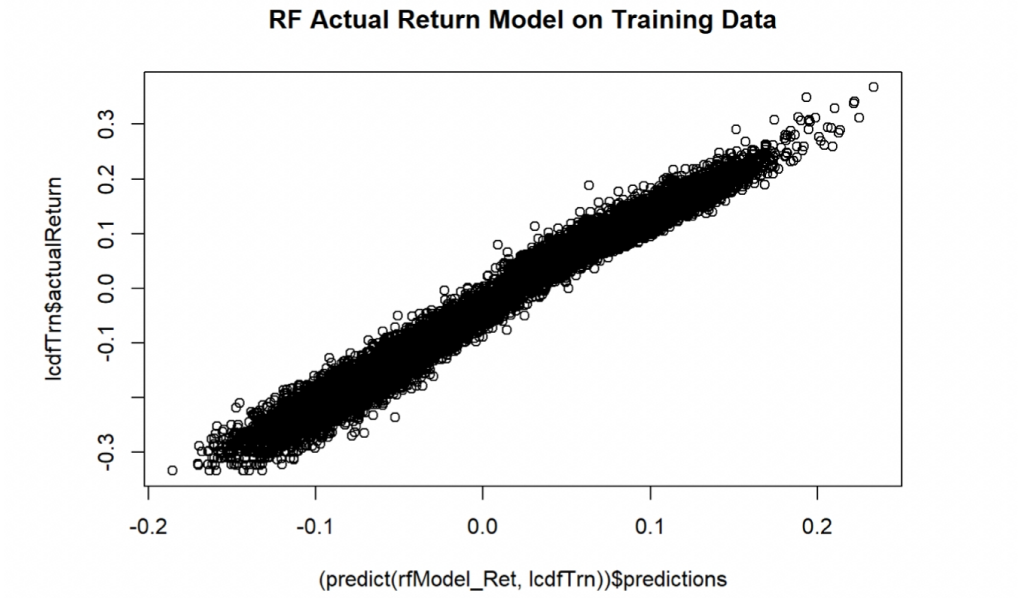
1. glm model

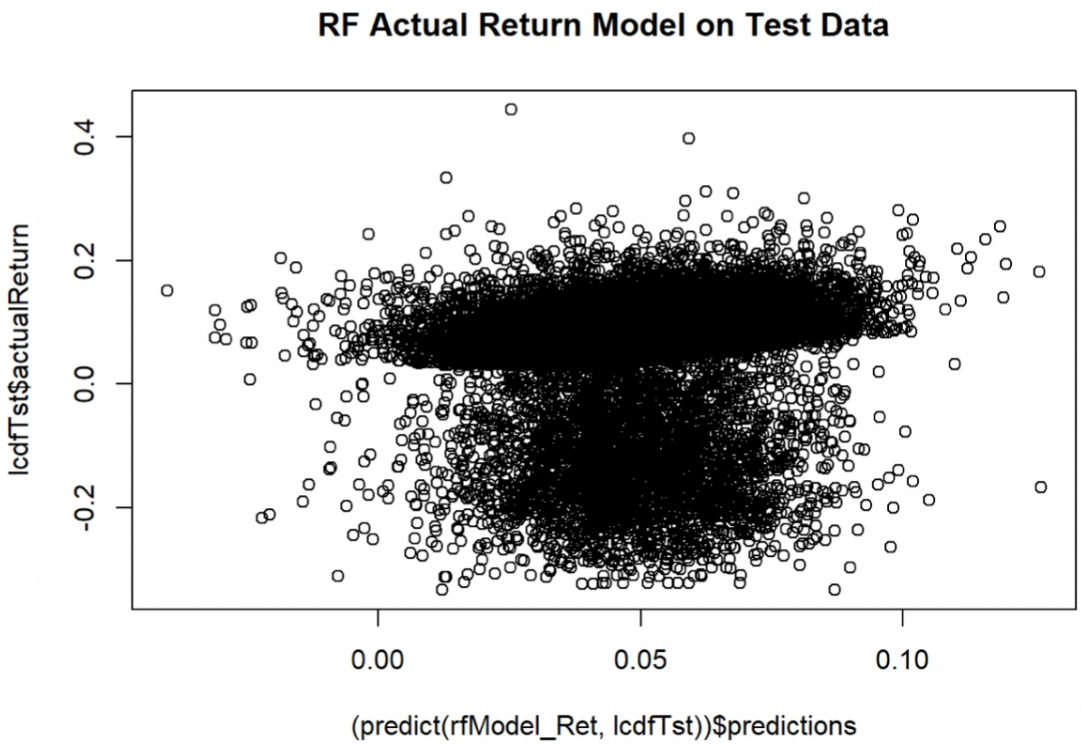
We created the decile table for Training and Testing data using the glm model. To be more specific, our model is able to distinguish the loans with the highest actualReturn, which are concentrated on the top deciles. As we go down the deciles, the average predicted return decreases. However, this model does not avoid defaults very well. Even though the Average Actual Return for the top deciles are the highest, there are too many defaults. Combining the Actual Return and Loan Status model is necessary.



2. Random Forest

We found out that it performs well with Training Data, but now with Testing Data. It has overfitting problems with Testing Data. As a result, try to change all kinds of variables, max.depth, num.trees, mtry,sample.fraction,...etc. However, we still can't fix the problems of overfitting with Testing Data. Fortunately, we figured out one variable," Total Payment" which can increase performance with Testing Data significantly. Sadly, this variable is leakage value, which we can not use at all. Moreover, we used a ranger model to create a decile table for Training and Testing Data. To be more specific, our model is able to distinguish the loans with the highest actualReturn, which are concentrated on the top deciles. As we go down the deciles, the average predicted return decreases. However, this model does not avoid defaults very well. Even though the Average Actual Return for the top deciles are the highest, there are too many defaults. Combining the Actual Return and Loan Status model is necessary.





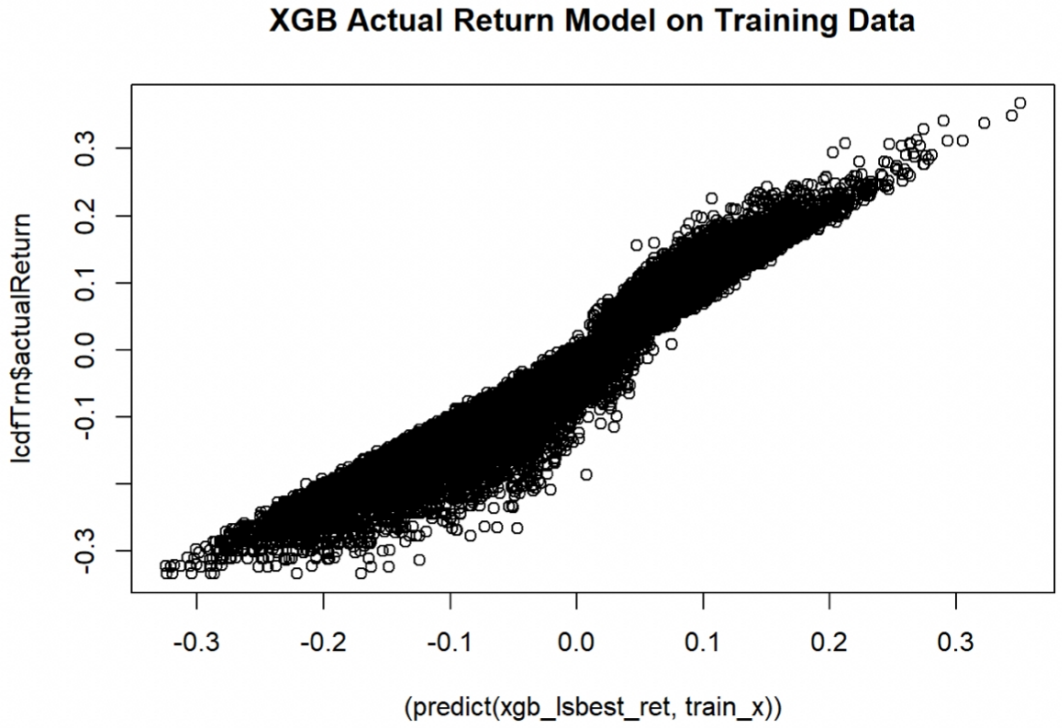
3. XGboost model

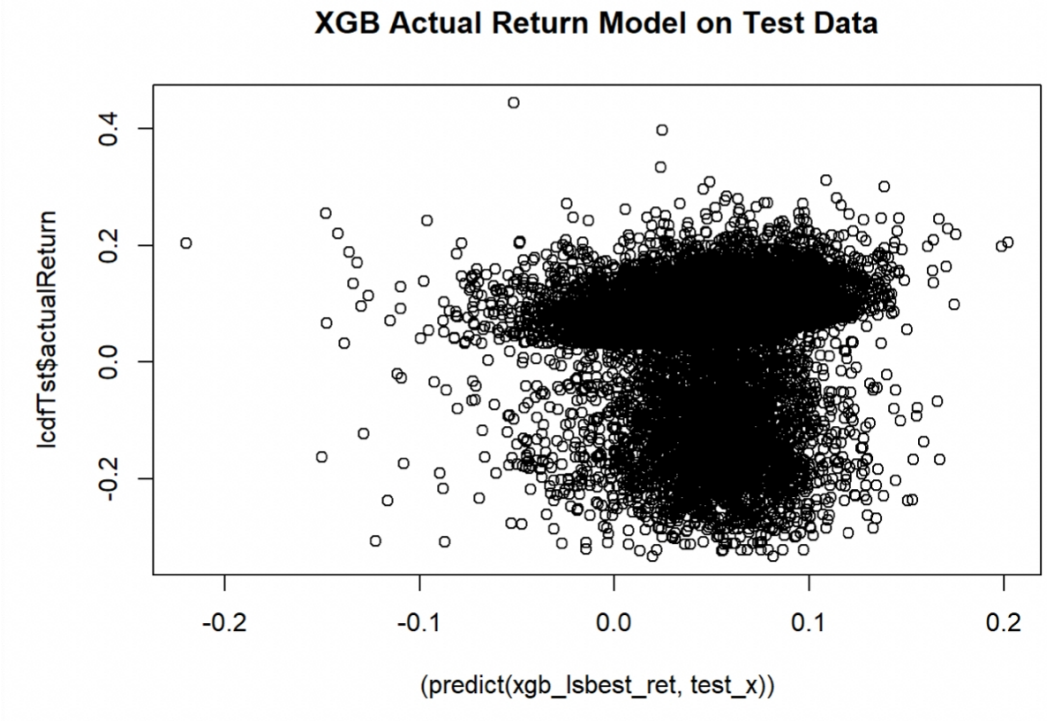
With the XGboost model, we tried the parameters max.depth and eta.

- eta - It makes the model more robust by shrinking the weights on each step. Typical values are 0.01-0.2 and the default value is 0.3. When experimenting with all the values and the parameters, we have troubles finding the correct combination to give better results on Test Data. The change in parameters did not result in notable changes in performance for Test, but it did for Train Data. We proceed with eta = 0.1, nrounds = 1000, max.depth = 6. We did this because we needed to counter the high bias even in Training data predictions. This improved our performance in Train Data, but not Test Data (before this, the model was weak on both).

We created the decile table for Training and Test data using the XGB model. Our model is able to distinguish the loans with the highest actualReturn, which are concentrated on the top deciles. As we go down the deciles, the average predicted return decreases.

However, this model does not avoid defaults very well. Even though the Average Actual Return for the top deciles are the highest, there are too many defaults. Combining the Actual Return and Loan Status model is necessary.


XGB Actual Return Model on Training Data
lcdfTrn$actualReturn
-0.3
-0.1
0.0
0.1
0.2
0.3
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
(predict(xgb_lsbest_ret, train_x))


XGB Actual Return Model on Test Data
lcdfTst$actualReturn
-0.2
0.0
0.2
0.4
-0.2
-0.1
0.0
0.1
0.2
(predict(xgb_lsbest_ret, test_x))

9.Considering results from the best model for predicting loan-status and that for predicting loan returns, how would you select loans for investment? There can be multiple approaches for combining information from the two models to make investment decisions (as discussed in class)– describe your approach, and show performance. How does performance here compare with use of single (i.e for predicting loan-status, or loan returns) models?

As we combine the loan status and loan returns prediction, we find out that the loan status prediction contains mostly higher grades' loans (A or B), which result in a lower average actual return. This makes sense because loans from higher grades are typically safer from getting defaults, but with the cost yielding lower interest rates.

To note, the predictions for the models are made by doing these steps:

- Finding the prediction for the actual return.
- Add the probability of a fully paid loan for each data point (derived from the score from the loan status model).
- Use the top decile of the results (we take the loans from the results that have the highest returns)
- We sort the top decile by the score from the loan status model, from highest probability of a fully paid to the lowest.

We first start by combining the forecasts with the Loan Status and Actual Returns models in our best Loan Status models for RF and XGB model, and then we take the top decile from the combined models and construct a new table from it. In actuality, the merged model outperforms the two original ones separately. The model was able to distinguish 2431 loans from the lower grades (mainly C-E) with the highest returns (averaged at 7%), which was significantly better than the loan status model's (4%). However, we must prevent the high number of defaults predicted by Actual Return in this top decile.

We can identify the loans with the highest likelihood of being paid in the top deciles by further dividing this decile into 20 deciles and then sorting them in ascending order of their loan status prediction score. With only 10 defaults, the resulting table enables us to discover 122 loans with average forecasted returns of roughly 7.4%. Furthermore,we can see that as we move down the deciles at the last table, the number of defaults considerably increases from 10 to 38 at the last deciles, despite the fact that our model does not boast the best possible accuracy. The similar process was used to merge the outcomes of the two XGB models.

With the fewest defaults, the combined XGB models were able to distinguish loans from those with lower grades. The combined models outperformed the individual models in the following ways:

- The Loan Status model by itself had a 3.8% average actual return but was able to reduce defaults on the top deciles.
- Despite a disproportionately large number of defaults, the Real Return model alone was able to collect an average actual return of 6.6% (far higher than the Loan Status projection) (464 loans).

We subsequently divided this top decile into 20 additional deciles and arranged them according to their likelihood of receiving full payment (score from the loan status model). Only 11 of the top 122 loans have defaulted, and the average real return is 6.2%. We were able to concentrate on the top 122 loans with the fewest defaults thanks to the combined deciles. These loans represent an investment opportunity with a relatively low risk and good actual returns that outperform what we saw in individual models. We can see that the number of defaults gradually increases to 42 out of 122 loans as we move down the 20 deciles. This indicates that, despite its flaws, our model is still capable of making predictions.

We essentially observed the same tendency in the case of GLM as we did in xgb and Ranger. We observed a low number of defaults (79) in the model where loan status was the target variable, but a relatively very low actual return (4.2%); in contrast, the model with actual return as the target variable had a higher average actual return of 7.3 percent, but it came at the cost of a higher number of defaults (449) that is actually in line with our expectations.

We now merge our GLM models using the techniques we used for RF and XGboost. Here are the outcomes from each of our unique models:
- With only 79 defaults out of 2431, the top decile of the loan status forecast was able to pinpoint the safest loans. The highest average actual return was predicted by our Real Return model at 7.3%, however it was unable to distinguish between defaults and non-defaults, as the average actual return was only 4.2%. (default of 449 loans).

Finally, the top decile of the Actual Return prediction was divided into 20 further deciles. Additionally, we included a score for predicting loan status, and we sorted the new 20 deciles according to this score. The resulting combination decile successfully identified the top 2 deciles (122 loans in each), with average actual returns of 7.1% and 7.9% and only 6 and 7 defaults, respectively. The glm model has so far performed the best for combination prediction.

10.As seen in data summaries and your work in the first assignment, higher grade loans are less likely to default, but also carry lower interest rates; many lower grad loans are fully paid, and these can yield higher returns. Considering this, one approach to making investment decisions may be to focus on lower grade loans (C and below), and try to identify those which are likely to be paid off. Develop models from the data on lower grade loans, and check if this can provide an effective investment approach. Compare performance of models from different methods (glm, gbm, rf).
Can this provide a useful approach for investment?
Compare performance with that in Q9 above?

Building a model to forecast loan status on lower grade loans is another strategy for maximizing returns from fully paid loans. With the predictions made above for the combination, we will compare the outcomes. With an average real return of 8.9%, our RF model for lower-grade loans can pinpoint the loans that are most likely to be fully paid (15 defaults out of 1112 loans).

When compared to our RF model for lower-grade loans, our GLM model performs worse. However, with only 125 defaults out of 1112 loans, its top decile has a good average real return of 7.3%. In comparison to our RF and GLM models on lower grade loans, our XGBoost model performs less well. With only 144 defaults out of 1112 loans, its top decile has a respectable average actual return of 6.6% overall. More defaults appear as we move down the deciles.

When comparing RF to RF, GLM to GLM, and XGB to XGB:
1) The Ranger Model appeared to be performing remarkably well with respect to lower quality loans data. The Lower Grade Loan Data Model appears to be performing better in all areas, including default rate and average actual return, if we compare the top deciles of the two separate Ranger models, one with Lower Grade Loans and the other that has All the Grades. If we use the ranger model, investing in lower rated loans appears to be the best option.

2) The comparison of the XGB model with data on lower-grade loans with that of its equivalent, which includes loans of every grade, is largely consistent with our assumptions. Greater default rates are seen in lower grade loan models, but so are higher real average returns.

3) Comparing the glm model with data on lower-grade loans to its equivalent, which includes loans of every grade, is about in line with our expectations. Greater default rates are seen in lower grade loan models, but so are higher real average returns.

###Profit analysis to choose the optimal investment strategy

We will now use the profit analysis to apply all the models. With $100 to invest in each loan, let's say we have to choose which top decile is the best to invest in (using what model?). A Charged Off debt costs $35 to invest in, as was discussed in our prior assignment. The

percentage of the actual average return for that decile will be used to determine the loan's return.

Due to their superior profit estimation, we would advise using the models listed below:
1. Use the RF model, which claims the biggest profit, on lower-grade loans.
2. Invest in the top 122 loans using the combined GLM models.

Please submit a pdf file with answers to the assignment questions, and supporting analyses. Also include a single Rmd file with your R code. Note – the code needs to be adequately commented and divided into sections in the Rmd file to help readability and ease understanding by others; arrange the Rmd file sections based on the assignment questions.