IDS 572 Assignment 2
Case – Loan default prediction and investment strategies in online lending

Part A

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

Objective: The primary goal is to assist clients in making informed investment decisions regarding peer-to-peer (P2P) loans by identifying potentially profitable loans and assessing default risks.

Evaluating Decisions:

- **Default Rate**: Percentage of loans charged off; a lower rate indicates better decisions.
- **Return on Investment (ROI)**: Calculated from total payments received minus principal; higher ROI signifies better investment.
- **Predictive Accuracy**: Evaluate loan classifications through model accuracy (precision, recall).
- **Investment Diversification**: Analyze risks and diversify investments across grades and purposes to minimize risk.

Goal of Predictive Models:

- Identify patterns in borrower attributes linked to default rates.
- Classify loans by risk level to guide safer investment choices.
- Provide actionable, data-driven insights for decision support.

Potential Target Variables:

- **Loan Status**: Categories such as "Fully Paid," "Charged Off," or "Current."
- **Default Rate**: Likelihood of a loan defaulting based on various factors.
- **Return on Investment (ROI)**: Percentage return expected from loans.
- **Grade**: Assigned risk grade (A to G) for assessing risk and potential return.

(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 maybe some of the important attributes to consider for your decision task?

Categorization of Data Attributes:

1. **Loan Information**:
    - loan_amnt, term, int_rate, installment, purpose, grade, sub_grade, loan_status.
2.
3. **Borrower Information**:annual_inc, emp_length, home_ownership, dti, fico_range_low, fico_range_high, verification_status.
4. **Payment and Default Information**:
    - total_pymnt, total_rec_prncp, total_rec_int, recoveries, collection_recovery_fee.
5. **Timeline Information**:
    - issue_d, earliest_cr_line, last_pymnt_d, next_pymnt_d, last_credit_pull_d.
6. **Derived or Calculated Metrics**:
    - out_prncp, out_prncp_inv, total_acc.

Important Attributes: Key attributes for decision-making likely include loan_amnt, int_rate, loan_status, annual_inc, dti, and fico_score, as they are critical for predicting defaults and assessing investment viability.

2. Data exploration – examine the data for basic insights.
(a) some questions to consider:

(i) What is the proportion of defaults (‘charged off’ vs ‘fully paid’ loans) in the data?
How does default rate vary with loan grade? Does it vary with sub-grade? Do you think loan grade and sub-grade convey useful information on riskiness of different loans?

Answer : The outcomes mentioned above were as expected. An A grade indicates the safest loan, while a G grade indicates the riskiest loan with a higher chance of default. Therefore, it is logical to see an increase in the loan default rate as we move from grade A to G. Similar trends can be observed with subgrades; as shown in the table above, the loan default rate increases from subgrade 1 to 5.

loan_status <chr>	count <int>	proportion <dbl>
Charged Off	13783	0.1378921
Fully Paid	86172	0.8621079

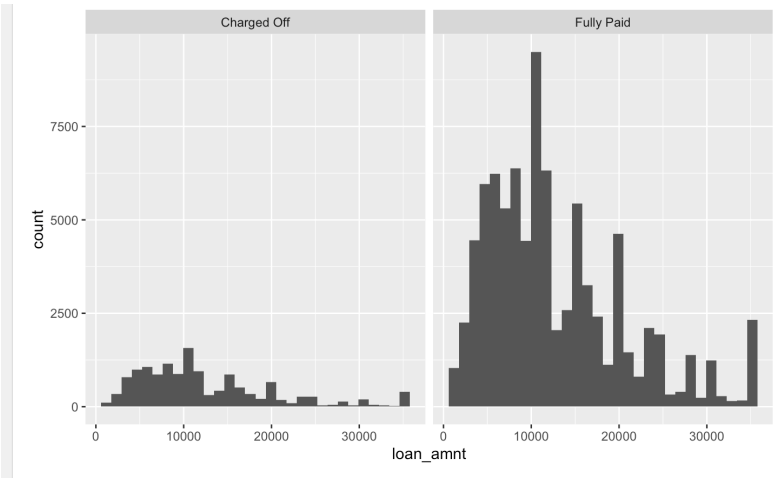
(ii) How many loans are there in each grade?
Do loan amounts vary by grade? And how does this vary by loan_status? Provide suitable plots to show this, and summarize your conclusions.
Answer :
There's a clear connection between loan grades and the number of loans and loan amounts. The higher-grade loans (A and B) have more total loans and larger amounts, which means they're less risky and have lower default rates. On the other hand, lower-grade loans (C, D, E, F, G) have fewer loans and smaller amounts, indicating higher risk. When it comes to loan status, a higher proportion of loans labeled "Fully Paid" are in the higher-grade categories, while lower-grade loans have more "Charged Off" statuses. This data can help you make smarter investment choices by focusing on higher-grade loans to reduce risk and increase repayment potential.

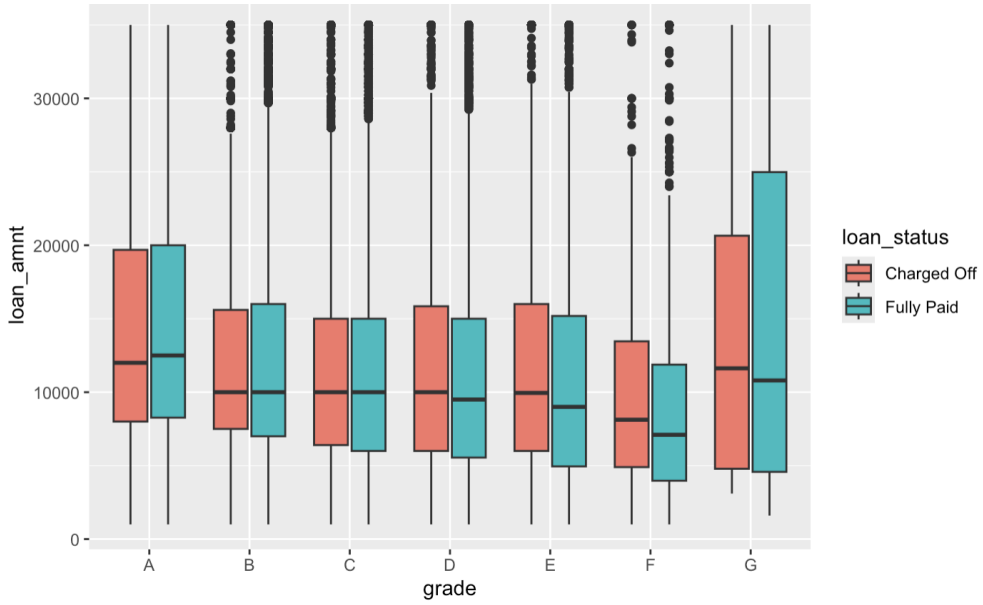
grade <chr>	total_loans <int>
A	22602
B	34457
C	26246
D	12337
E	3536
F	704
G	73

7 rows

grade <chr>	loan_status <chr>	sum(loan_amnt) <dbl>
A	Charged Off	16623500
A	Fully Paid	306134750
B	Charged Off	46194150
B	Fully Paid	384606575
C	Charged Off	56497075
C	Fully Paid	257912725
D	Charged Off	34365050
D	Fully Paid	111516350
E	Charged Off	12754000
E	Fully Paid	29389200

grade <chr>	loan_status <chr>	sum(loan_amnt) <dbl>
F	Charged Off	2384675
F	Fully Paid	4622850
G	Charged Off	446750
G	Fully Paid	641975





(iii) 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 analysis shows that interest rates rise with loan grade riskiness, both at the grade and subgrade levels. Additionally, loans with higher interest rates are more likely to default, as reflected by the higher proportion of charged-off loans at higher interest rates. These findings align with the expectations of higher risk being associated with higher interest rates in peer-to-peer lending markets.

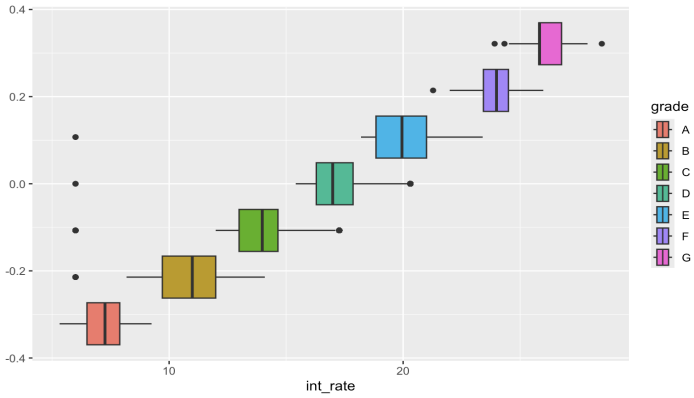
loan_status <chr>	mean(int_rate) <dbl>
Charged Off	13.86160
Fully Paid	11.76765

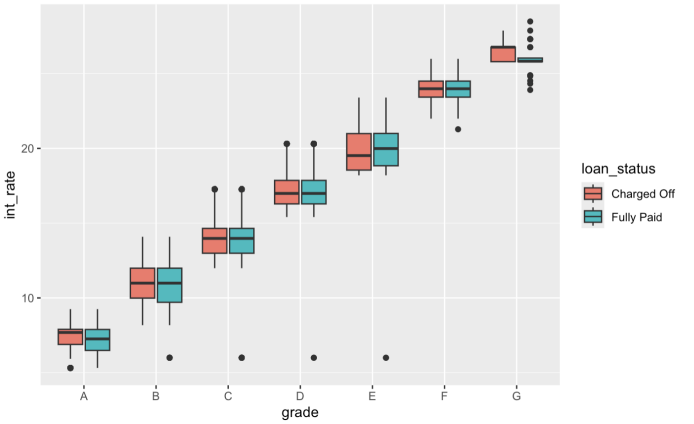
2 rows

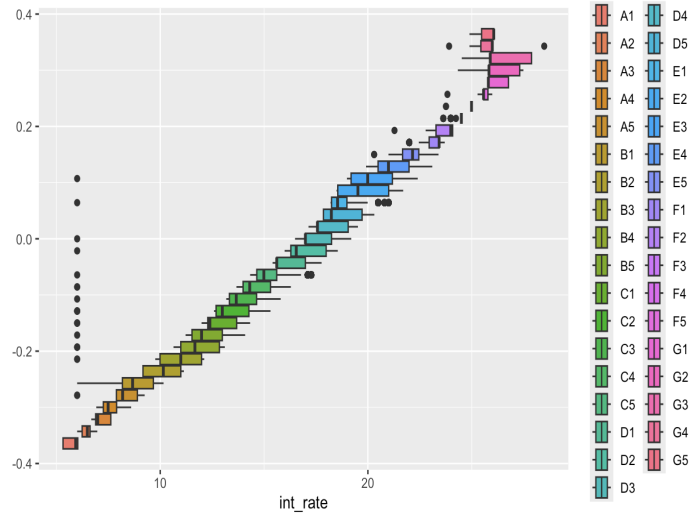
grade <chr>	mean(int_rate) <dbl>
A	7.215668
B	10.852160
C	13.950170
D	17.225323
E	19.987814
F	23.919119
G	26.215479

sub_grade <chr>	mean(int_rate) <dbl>
A1	5.708628
A2	6.429465
A3	7.134888
A4	7.518571
A5	8.275672
B1	8.965699
B2	10.036217
B3	10.983030
B4	11.836819
B5	12.347138
C1	12.960988
C2	13.463975
C3	14.053313
C4	14.641219
C5	15.347776
D1	16.173396
D2	17.021050
D3	17.478817

sub_grade <chr>	mean(int_rate) <dbl>
D4	18.087495
D5	18.608622
E1	18.961866
E2	19.635567
E3	20.224829
E4	21.097568
E5	22.083325
F1	23.058366
F2	23.717532
F3	24.382081
F4	24.960361
F5	25.632456
G1	26.142353
G2	26.424286
G3	26.465000
G4	25.892857
G5	25.670000

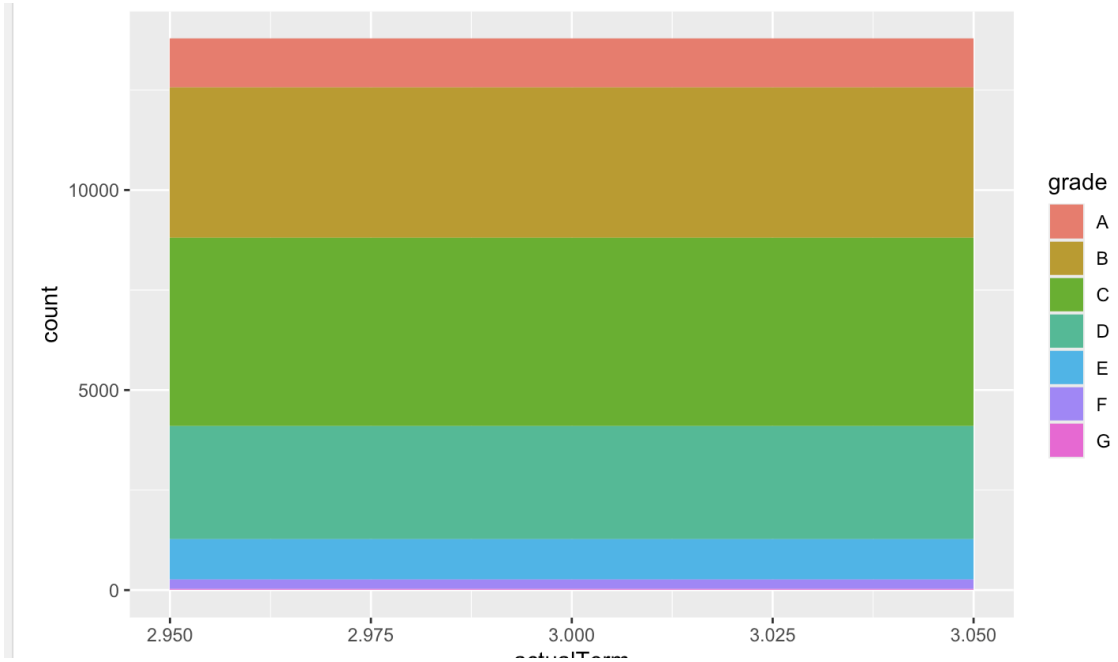

0.4
0.2
0.0
-0.2
-0.4
10
20
int_rate
grade
A
B
C
D
E
F
G


20
10
int_rate
A
B
C
D
E
F
G
grade
loan_status
Charged Off
Fully Paid


0.4
0.2
0.0
-0.2
-0.4
10
20
int_rate
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
G1
G2
G3
G4
G5

(iv) 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).

Answer : For loans that have been fully paid back, the time it takes to fully repay them varies from 0 to approximately 3.6 years. However, the average actual repayment period for fully paid back loans is 2.1 years. It is observed that the average repayment period increases with the loan grades.



(v) Calculate the annual return for a loan. Explain how you calculate the percentage annual return.
Is there any return from loans which are ‘charged off’? Explain. How does return from charged -off loans vary by loan grade?
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.
If you decide to invest in loans based on this data exploration, which loans would you prefer to invest in?

Answer : To calculate the annual percentage return for a loan, you first need to find the difference between the total payment made by the borrower and the original funded loan amount. Then, divide this difference by the funded amount to express the return as a percentage. To annualize the return, take into account the loan term by multiplying the result by the ratio of 12 months to the loan term (for example, for a 36-month loan, multiply by 12/36). The final formula is:

Annual Return = ((total payment - funded amount) / funded amount) × 12 / loan term × 100

This formula provides the percentage return on an annual basis, allowing for easier comparison of loans with different durations.

Loans classified as 'charged off' yield negative returns, indicating a loss, as the funded amount exceeds the total payment.

Charged off loans display negative returns, as shown in the table below. While the average interest rate on loans increases from grade A to G, the returns (losses) from charged-off loans show little variation, despite the increase in defaults from grade A to G.

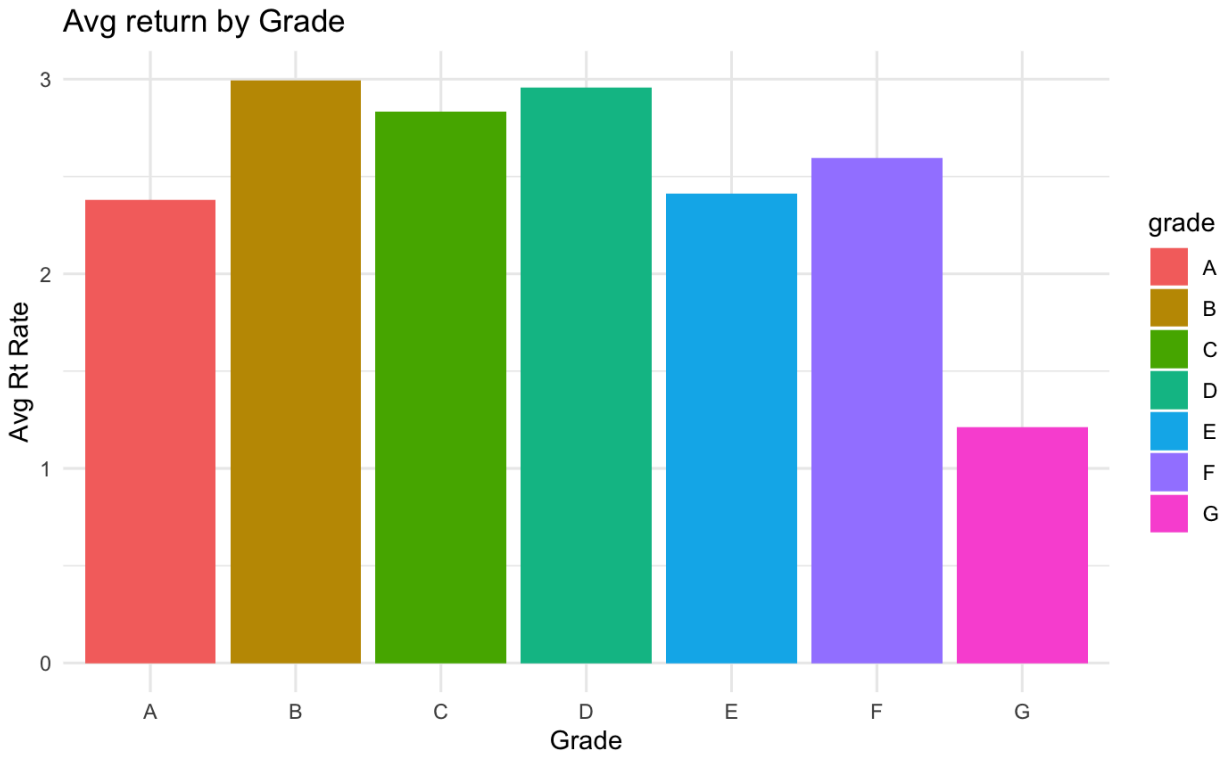
Average returns increase from grade A to B and then decrease from B to G. Within the sub-grades, B4 appears to have the highest return. Based on this data exploration, if I were to invest in loans, I would choose to invest in B4.

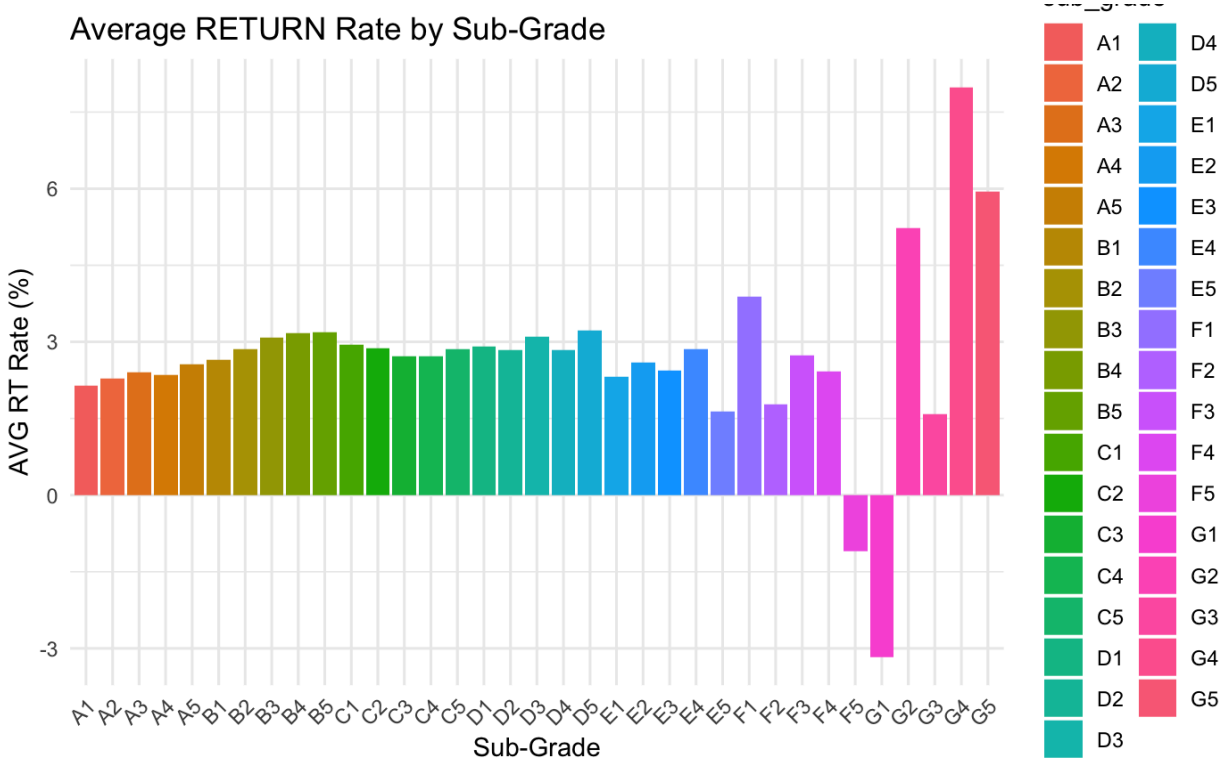
loan_status <chr>	nLoans <int>	avgInterest <dbl>	avgLoanAmt <dbl>	avgActualRet <dbl>	avgActualTerm <dbl>	minActualRet <dbl>	maxActualRet <dbl>
Charged Off	13783	13.86160	12280.72	-1210.5482	3.000000	-3333.333	1380.878
Fully Paid	86172	11.76765	12705.11	803.7233	2.138382	0.000	4081.542

grade <chr>	nLoans <int>	defaults <int>	defaultrate <dbl>	avgInterest <dbl>	stdInterest <dbl>	avgLoanAmt <dbl>	avgPmnt <dbl>	avgRet <dbl>	stdRet <dbl>
A	22602	1210	0.05353509	7.215668	0.9638845	14280.075	15320.92	2.379819	4.067155
B	34457	3756	0.10900543	10.852160	1.4760585	12502.560	13639.64	2.995801	6.119150
C	26246	4707	0.17934161	13.950170	1.2349027	11979.342	13012.33	2.834538	8.236685
D	12337	2839	0.23012077	17.225323	1.2197827	11824.706	12851.39	2.957377	9.858577
E	3536	1013	0.28648190	19.987814	1.4341964	11918.326	12623.59	2.410828	11.542508
F	704	227	0.32244318	23.919119	0.9483656	9953.871	10552.82	2.593148	13.052398
G	73	31	0.42465753	26.215479	0.8544833	14914.041	16396.25	1.212423	14.716045

7 rows | 1-10 of 13 columns

sub_grade <chr>	nLoans <int>	defaults <int>	defaultrate <dbl>	avgInterest <dbl>	stdInterest <dbl>	avgLoanAmt <dbl>	avgPmnt <dbl>	avgRet <dbl>
A1	3614	102	0.02822357	5.708628	0.3462865	14076.681	14990.626	2.147418
A2	3455	125	0.03617945	6.429465	0.1686541	13826.889	14783.287	2.274063
A3	3668	183	0.04989095	7.134888	0.3427884	14350.409	15404.877	2.412943
A4	5436	333	0.06125828	7.518839	0.3628423	14501.766	15571.240	2.357618
A5	6425	467	0.07268482	8.275654	0.4379961	14407.195	15532.628	2.567195
B1	6329	525	0.08295149	8.965699	0.7579684	12709.050	13720.742	2.643641
B2	7082	724	0.10223101	10.036217	0.8307024	12951.101	14095.298	2.859775
B3	7396	792	0.10708491	10.983030	0.9266443	12645.051	13821.986	3.075730
B4	7165	876	0.12226099	11.836819	0.8921302	12258.405	13450.605	3.178556
B5	6485	839	0.12937548	12.347138	0.9369392	11918.454	13063.776	3.194965





(vi) What are people borrowing money for (loan 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? Summarize what you find.

Here is the revised text:

Loan Amount Distribution by Purpose:
A significant majority, approximately 62% of the total loan amount, was borrowed for Debt Consolidation, indicating a prevalent trend of borrowers seeking to manage existing debts. The second largest category, Credit Card loans, accounted for around 25% of the total loan amount, suggesting many borrowers use loans to pay off high-interest credit card debts. The remaining 13% of the total loan amount was allocated to other purposes, including Home Improvement, Major Purchases, Small Business, and Medical expenses.

Default Rates by Purpose:
Loans for Renewable Energy, Small Business, Moving, and House purposes exhibited the highest default rates respectively. These elevated default rates suggest that borrowers in these

categories may face particular challenges in repayment. Correspondingly, these loan purposes are associated with high average interest rates.This pattern indicates a potential risk assessment by lenders, as higher interest rates often correlate with higher perceived risks.

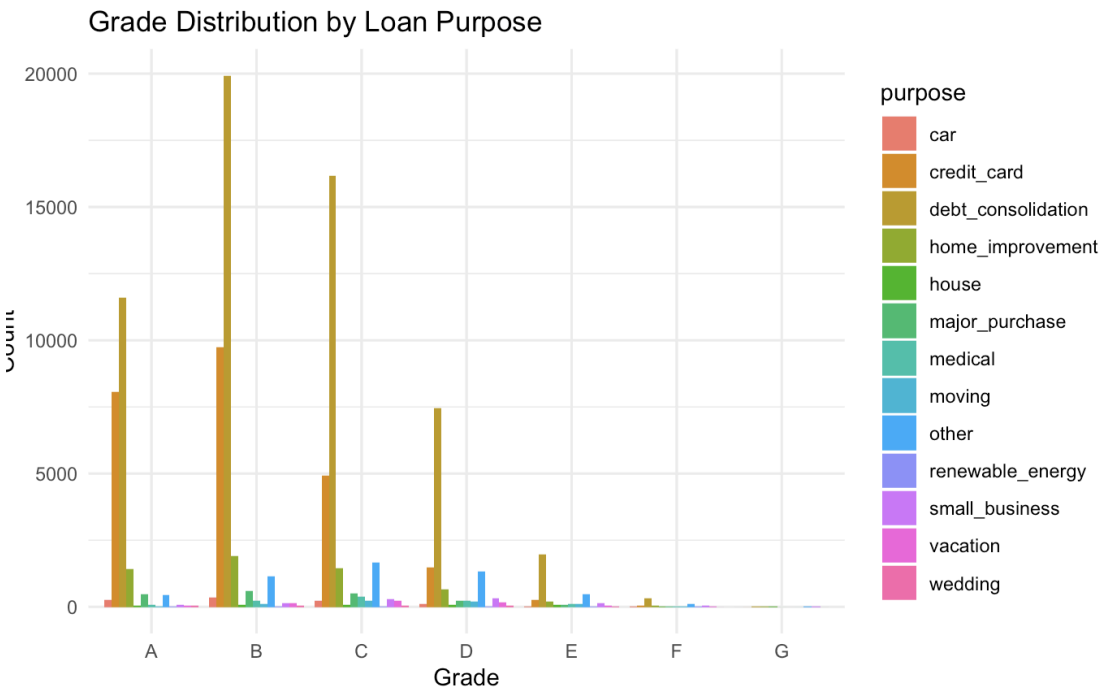
Loan Grades and Default Rates:
Analysis of loan grades reveals that most loan amounts are concentrated in Credit Card and Debt Consolidation categories, generally associated with lower grades (B, C, and D). Most loans with higher default rates are predominantly found in these lower grades, suggesting that borrowers in these categories may be less creditworthy or face financial difficulties.

Interest Rates and Repayment Outcomes:
For fully paid loans, we see interest rates such as 15.59% for a $3,000 loan and 11.99% for a $15,950 loan, indicating that borrowers can manage these payments successfully despite relatively high interest rates. On the other hand, charged-off loans show higher interest rates, such as 20.20% for a $7,150 loan and 17.77% for a $10,000 loan, which illustrates the financial strain these borrowers face.

Summary Insights:
The data highlights the significant reliance on loans for Debt Consolidation and Credit Card purposes, with the associated challenges of higher default rates in other categories. Understanding the relationship between loan purpose, default rates, and interest rates is crucial for lenders to manage risk and for borrowers to make informed decisions.

Recommendations:
Lenders may benefit from offering additional support or financial education for borrowers in higher-risk categories, particularly those with higher default rates. Tailored financial products that address borrowers' specific needs and repayment capabilities could improve repayment outcomes and reduce default rates in vulnerable loan categories.

purpose <chr>	defaults <int>	total_loans <int>	defaultrate <dbl>	total_loan_amount <dbl>	avg_loan_amount <dbl>	min_interest_rate <dbl>
car	110	995	11.05528	8063175	8103.693	5.32
credit_card	2825	24526	11.51839	331236300	13505.517	5.32
debt_consolidation	8290	57481	14.42216	757843000	13184.235	5.32
home_improvement	742	5721	12.96976	67169975	11740.950	5.32
house	59	404	14.60396	5292050	13099.134	6.03
major_purchase	253	1926	13.13603	17991925	9341.602	5.32
medical	195	1075	18.13953	8164625	7595.000	5.32
moving	131	689	19.01306	4714225	6842.126	6.03
other	814	5166	15.75687	42853375	8295.272	5.32
renewable_energy	11	62	17.74194	550500	8879.032	6.03

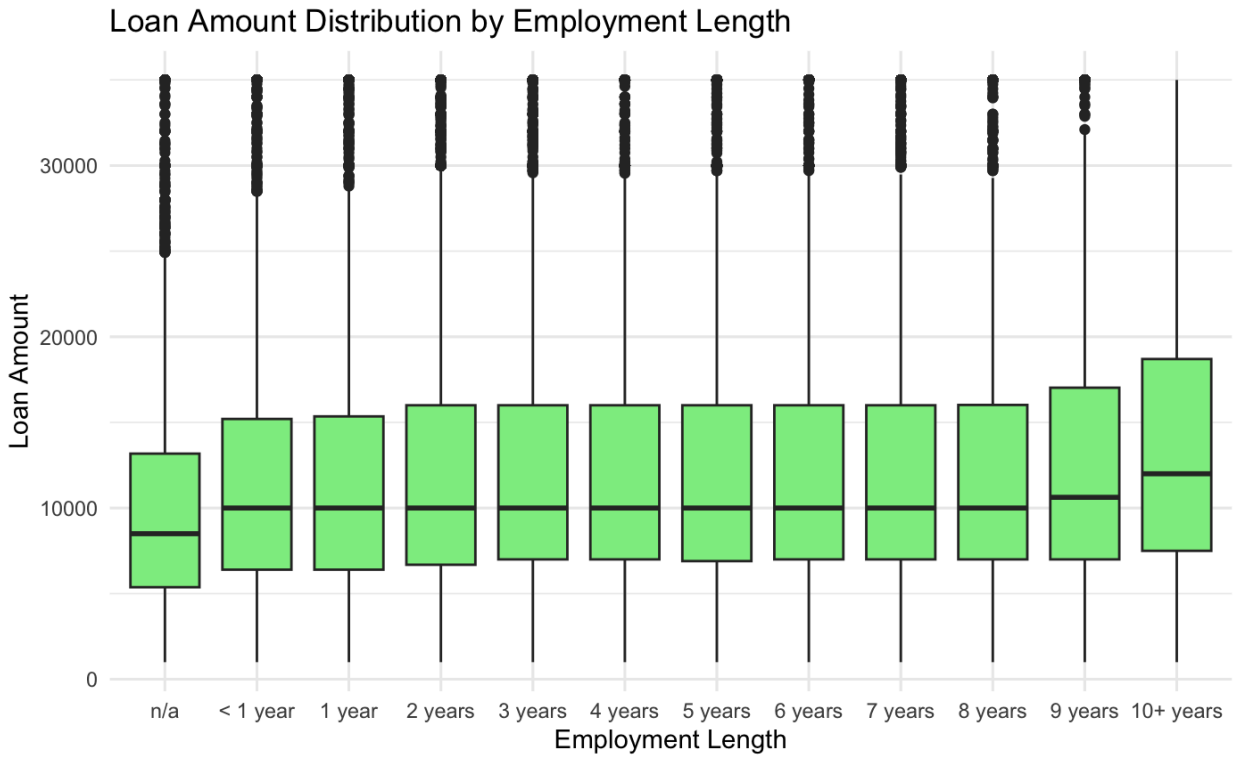


(vii) Consider some borrower characteristics like employment-length, annual-income, fico-scores
(low, high). How do these relate to loan attributes like, for example, loan_amout, loan_status,

grade, purpose, actual return, etc.

Answer: The analysis reveals a strong connection between borrower characteristics, particularly employment length, and various loan attributes such as loan status, grade, and amounts. Borrowers with longer employment lengths generally have lower default rates, indicating better financial stability. Longer employment is associated with higher loan grades and larger average loan amounts, suggesting that lenders view these borrowers as lower risk.

Additionally, while actual returns are negative across different employment lengths, borrowers with stable employment exhibit less negative returns than those with shorter employment durations. This indicates that employment stability impacts loan performance and leads to better repayment behavior. Overall, these findings can improve lenders' risk assessment strategies and help borrowers make informed financial decisions, underscoring the significance of employment stability in the lending process.



(viii) 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).

Answer:

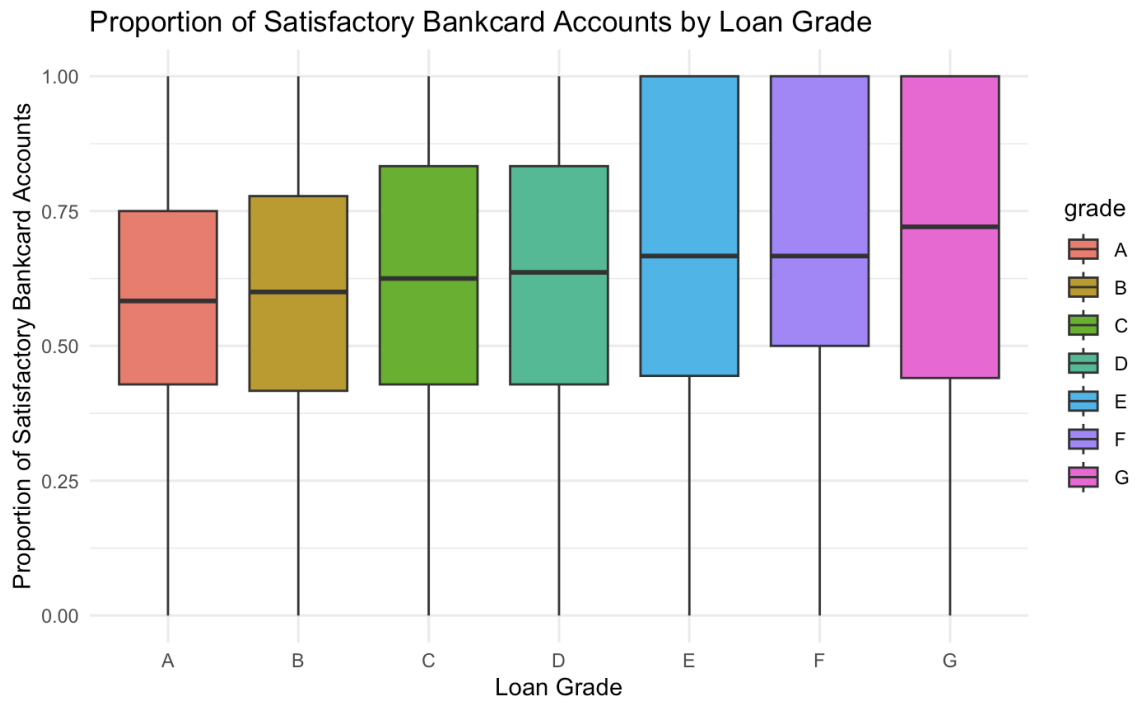
Ability to Pay Back: This variable measures the borrower's income relative to their monthly payments, providing critical insights into their repayment capability. A higher ratio indicates a stronger ability to repay, suggesting that the borrower is more likely to meet their obligations.

Proportion of Satisfactory Bankcard Accounts: This metric reflects the number of bankcard accounts in good standing, indicating the borrower's credit management skills. A higher proportion suggests that the borrower has successfully managed their credit accounts, which can correlate with lower default rates.

First-Year Payment Ability: This variable assesses the monthly payment's total outstanding balance (excluding mortgages), indicating whether the borrower can handle their initial

payment obligations. A favorable ratio implies that the borrower is likely to maintain timely payments in the early stages of the loan.
Credit Utilization Ratio: This metric represents the proportion of available credit being utilized. A lower credit utilization ratio generally signals better financial health and responsible credit management, while a higher ratio may indicate potential financial distress.
Analysis and Implications

Examining these derived variables will provide valuable insights into how they relate to loan default rates across different loan grades. By analyzing the correlation between these metrics and default rates, lenders can enhance their risk assessment processes and make more informed decisions regarding loan approvals.



defaults <int>	defaultRate <dbl>	avgAbilityToPayback <dbl>	avgPropSatisBC_Accnts <dbl>	avgFirstYrPaymentAbility <dbl>	avgCreditUtilization <dbl>
1210	0.05354456	20.23343	0.5947878	12.05134	0.4139979
3756	0.10900543	19.19586	0.6003454	12.03321	0.3322541
4706	0.17934451	18.76981	0.6294445	12.75120	0.2889848
2838	0.23009567	18.25837	0.6310905	12.72023	0.2656464
1013	0.28648190	17.23033	0.6512535	12.60188	0.2609391
227	0.32244318	17.91436	0.6617172	13.81794	0.2665977
31	0.42465753	13.74465	0.6692285	10.37390	0.2221783

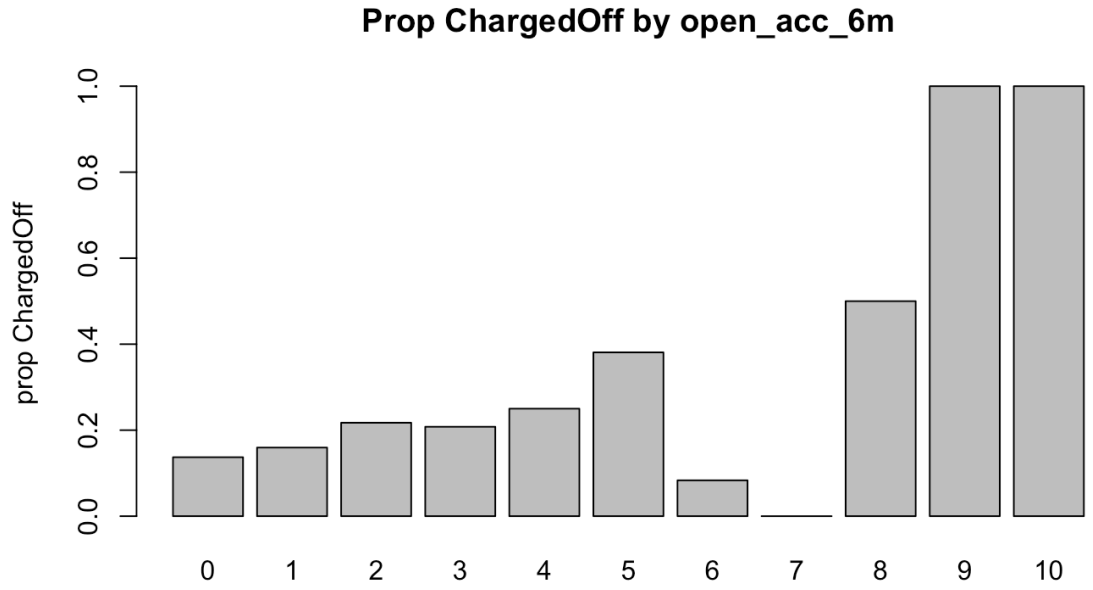
2(b) Are there missing values in the data ? What is the proportion of missing values in different variables?
Explain how you will handle missing values for different variables. You should consider what he variable is about, and what missing values may arise from – for example, a variable monthsSinceLastDeliquency 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?

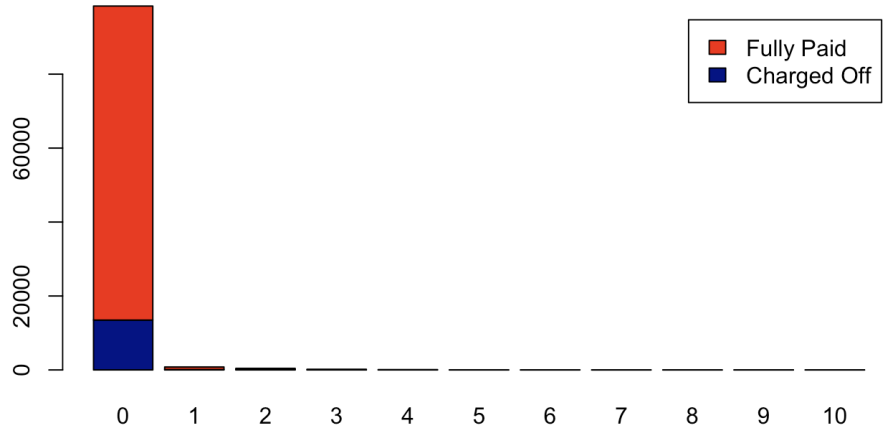
Answer:

In analyzing the dataset, it is essential to identify columns that contain sensitive information or have excessive missing values. Variables such as id, member_id, address, name, and phone_number should be removed from the dataset as they do not contribute meaningful insights and pose potential privacy risks. Additionally, variables with a high proportion of missing values, such as mths_since_last_major_derog, should be evaluated for their relevance. For instance, if these missing values indicate the absence of derogatory marks, replacing NA with 0 would make sense. Similarly, for open_acc, substituting missing values with 0 can represent borrowers without open accounts, ensuring data consistency.

When it comes to handling missing values, the strategy must align with the nature of each variable. For instance, variables like mths_since_last_delinquency can be retained if their missing values are reasonably imputable, while those with over 60% missing data may need to be excluded from the analysis altogether. By carefully considering which variables to keep and how to treat missing values, the dataset can remain robust and relevant for analytical purposes, enabling more accurate predictions and insights while ensuring privacy is upheld.

```
> # Final dimensions check
> dim(lcdf)  # how many variables left
[1] 99942    68
```



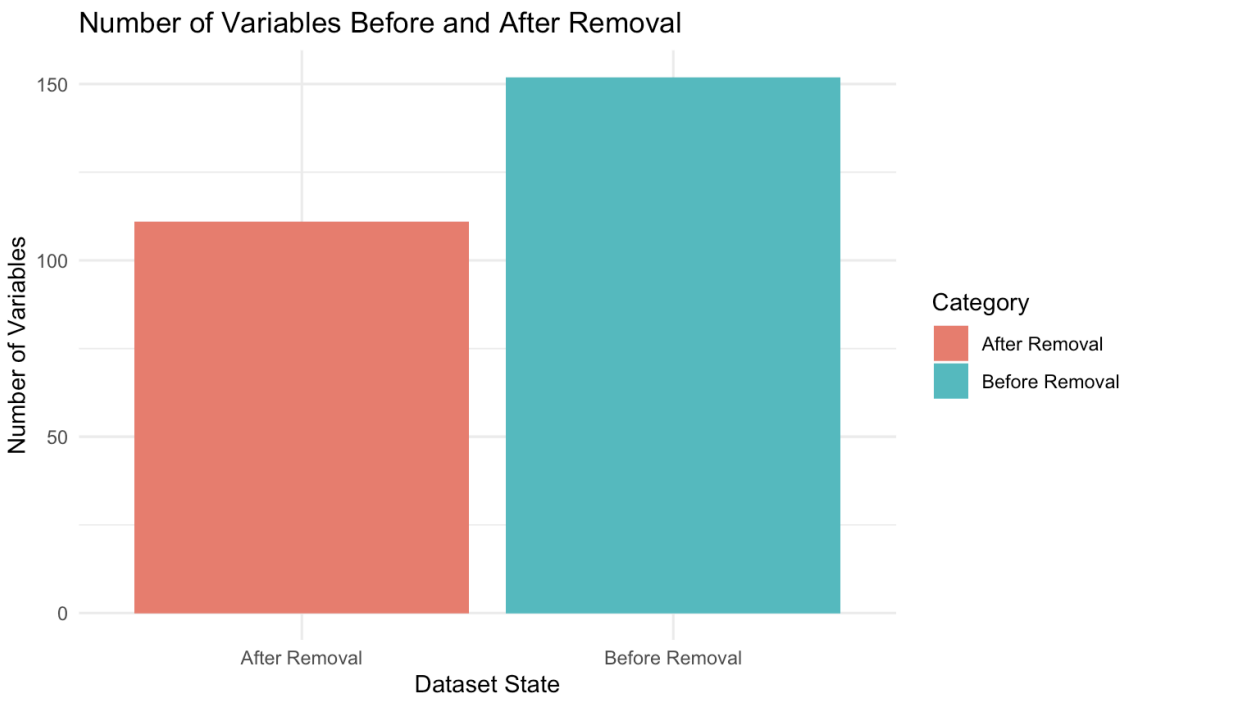


2(c) Consider the potential for data leakage. You do not want to include variables in your model which may not be available when applying the model; that is, some data may not be available for new loans before they are funded. Leakage may also arise from variables in the data which may have been updated during the loan period (ie., after the loan is funded). Identify and explain which variables you will exclude in developing predictive models.

Answer:

In developing predictive models, it's crucial to avoid data leakage, which can lead to overfitting and diminished performance on unseen data. To mitigate this risk, certain variables should be excluded from the model. Specifically, identifying information such as `id` and `member_id` should be removed, as they do not provide meaningful insights into loan performance. Additionally, variables like `last_pymnt_d` and `next_pymnt_d` offer information that is only available after the loan has been funded, creating a bias in the model. Other variables related to borrower hardships and settlements, such as those starting with `hardship` or `settlement`, also reflect borrower behavior during repayment and should be omitted.

Furthermore, variables that describe the loan's financial state post-funding—such as `funded_amnt_inv`, `out_prncp`, and `total_rec_int`—are contingent on the loan's history and are inappropriate for modeling new loan applications. Similarly, variables like `term`, `disbursement_method`, and `application_type` may not provide consistent predictive power, as they can vary with market conditions. By excluding these variables, the model can focus on relevant features that accurately represent borrower characteristics and their likelihood to repay, thereby enhancing the robustness and reliability of the predictive analysis.



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).

The dependent variable, **loan_status**, is a binary variable with two possible outcomes. To determine the relationship between this dependent variable and each potential predictor, we can apply statistical methods such as the **point-biserial correlation coefficient** or the **phi coefficient**. These methods measure the association between a binary variable (loan_status) and continuous or categorical predictor variables. Alternatively, we can also use the **AUC (Area**

Under the Curve) approach to assess the predictive power of each variable by calculating the AUC value for each predictor.

	variable	auc_value
loan_status	loan_status	1.0000000
annRet	annRet	0.9676530
total_pymnt	total_pymnt	0.7540075
sub_grade	sub_grade	0.6639991
int_rate	int_rate	0.6546702
grade	grade	0.6542020
acc_open_past_24mths	acc_open_past_24mths	0.5808674
annual_inc	annual_inc	0.5764603
bc_open_to_buy	bc_open_to_buy	0.5733377
tot_hi_cred_lim	tot_hi_cred_lim	0.5731866
total_bc_limit	total_bc_limit	0.5729599
avg_cur_bal	avg_cur_bal	0.5693621
ability_to_payback	ability_to_payback	0.5686064
dti	dti	0.5676090
total_rev_hi_lim	total_rev_hi_lim	0.5639888
tot_cur_bal	tot_cur_bal	0.5615251
mo_sin_rcnt_tl	mo_sin_rcnt_tl	0.5612214
mths_since_recent_inq	mths_since_recent_inq	0.5588659
mort_acc	mort_acc	0.5577655
mo_sin_rcnt_rev_tl_op	mo_sin_rcnt_rev_tl_op	0.5573777
mths_since_recent_bc	mths_since_recent_bc	0.5535080
home_ownership	home_ownership	0.5519604
inq_last_6mths	inq_last_6mths	0.5504949
mo_sin_old_rev_tl_op	mo_sin_old_rev_tl_op	0.5503813
credit_utilization	credit_utilization	0.5495383

```
bc_open_to_buy
                0.5639888                 0.5808674                 0.5693621
0.5733377
                  bc_util       mo_sin_old_il_acct      mo_sin_old_rev_tl_op
mo_sin_rcnt_rev_tl_op
                0.5431869                 0.5282933                 0.5503813
0.5573777
           mo_sin_rcnt_tl                mort_acc      mths_since_recent_bc
mths_since_recent_inq
                0.5612214                 0.5577655                 0.5535080
0.5588659
                num_bc_tl                num_il_tl            num_rev_accts
num_tl_120dpd_2m
                0.5132073                 0.5096878                 0.5047725
0.5000826
           pct_tl_nvr_dlq          tot_hi_cred_lim        total_bal_ex_mort
total_bc_limit
                0.5131406                 0.5731866                 0.5205033
0.5729599
total_il_high_credit_limit                    annRet       ability_to_payback
firstyr_pymnt_ability
                0.5130619                 0.9676530                 0.5686064
0.5171462
        credit_utilization
                0.5495383
```

By analyzing the relationship between **loan_status** and each predictor, we can identify which variables are useful for prediction. However, we must be cautious of **data leakage**, which occurs when a predictor includes information that wouldn't be available at the time of prediction.

In this context, variables such as **annRet**, **loan_status**, and **total_pymnt** are highly correlated with the target variable and are likely to represent post-loan outcomes, meaning they would only be known after the loan has been issued. Including these variables in the model can result in overly optimistic predictions and hinder the model's ability to generalize to new data.

To avoid data leakage, it is essential to exclude variables like **annRet**, **loan_status**, and **total_pymnt** from the predictive model. Doing so will enhance the model's accuracy and ensure it generalizes effectively to unseen data.

4(a) Split the data into training and validation sets: What proportions do you consider, and why?

We will split the data into two subsets, with **70%** allocated to the training set and **30%** to the validation set. This **70/30 split** strikes an effective balance by providing the model with a substantial portion of data for training while reserving enough data to evaluate its performance. The 70% training data allows the model to learn from a broad set of patterns and relationships, which is essential for more complex models that need significant data to capture relevant

trends. The remaining 30% is sufficient for testing the model's performance and ensures a robust evaluation without the risk of overfitting to the test set. This split helps ensure that the model generalizes well to unseen data by maintaining diversity in the training set while keeping a reasonable portion for evaluation.

4(b) How will you evaluate performance – which measures do you consider, and why?

Choosing the right performance metrics is essential for understanding how well the model is performing. For binary classification problems like predicting whether a loan will be charged off, the following metrics are particularly relevant:

- **Confusion Matrix**: This gives a detailed overview of the model's accuracy by showing the number of true positives, false positives, true negatives, and false negatives. It helps us understand the types of errors the model makes, which is crucial when predicting loan outcomes.
- **ROC Curve and AUC (Area Under the Curve)**: These metrics assess the model's ability to distinguish between classes (i.e., whether a loan will be paid off or charged off) at various thresholds. The AUC quantifies how well the model discriminates between positive and negative cases.
- **Lift**: In loan default prediction, the goal is to identify potential defaults as effectively as possible. Lift analysis evaluates how much better the model performs in identifying defaults compared to random guessing, making it particularly useful for targeting high-risk cases in a practical setting.

These measures provide a comprehensive evaluation of the model's effectiveness, highlighting its classification accuracy, its discriminative power, and its ability to prioritize high-risk cases.

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 uni-variate analyses in Question 3 above?

Answer:
We will use the rpart decision tree to predict default.

```
Classification tree:
rpart(formula = loan_status ~ ., data = lcdfTrn %>% select(-all_of(varsOmit)),
    method = "class", parms = list(split = "information"), control = rpart.control(minsplit = 30))

Variables actually used in tree construction:
character(0)

Root node error: 6900/49971 = 0.13808

n= 49971

  CP nsplit rel error xerror xstd
1  0      0         1      0    0
```

The output above shows the result of using the rpart function to train the decision tree. We can see from above that the root node error is 0.13808 which implies that 13.80% of the observations are misclassified by the decision tree at the root node. We can also observe that the tree did not grow beyond the root node i.e. no variables were used in tree construction. This
indicates that the model has to be improved. In the context of decision trees, the "cp" (complexity parameter) is a tuning parameter that controls the trade-off between the complexity
of the tree and its goodness of fit to the training data. A smaller value of "cp" results in a more complex tree, while a larger value encourages simpler trees by penalizing additional splits. If the tree does not grow beyond the root node, it suggests that the algorithm is being conservative
in splitting the data. Additionally, we may consider other parameters in the rpart function, such as minsplit (the minimum number of observations that must exist in a node for a split to beattempted). Adjusting these parameters can also influence the growth of the tree. To promote
the growth of the tree and encourage more splits, we can try decreasing the complexity Parameter.

```
lcDT1 <- rpart(loan_status ~., data=lcdfTrn %>% select(-all_of(varsOmit)), method="class", parms = list(split =
"information"), control = rpart.control(cp=0.0001, minsplit = 50))
```

The cp argument sets the complexity parameter at 0.001 and the minsplit sets the number of minimum observation required for a split to occur at 50.

```
Classification tree:
rpart(formula = loan_status ~ ., data = lcdfTrn %>% select(-all_of(varsOmit)),
    method = "class", parms = list(split = "information"), control = rpart.control(cp = 1e-04,
        minsplit = 50))

Variables actually used in tree construction:
 [1] ability_to_payback         acc_open_past_24mths       annual_inc                 avg_cur_bal
 [5] bc_open_to_buy             bc_util                    credit_utilization         dti
 [9] emp_length                 firstyr_pymnt_ability      home_ownership             initial_list_status
[13] inq_last_6mths             installment                int_rate                   loan_amnt
[17] mo_sin_old_il_acct         mo_sin_old_rev_tl_op       mo_sin_rcnt_rev_tl_op      mo_sin_rcnt_tl
[21] mort_acc                   mths_since_last_delinq     mths_since_recent_bc       mths_since_recent_inq
[25] num_accts_ever_120_pd      num_actv_bc_tl             num_actv_rev_tl            num_bc_sats
[29] num_bc_tl                  num_il_tl                  num_op_rev_tl              num_rev_accts
[33] num_rev_tl_bal_gt_0        num_tl_op_past_12m         open_acc                   pct_tl_nvr_dlq
[37] percent_bc_gt_75           prop_SatisBC_Accnts        purpose                    revol_bal
[41] revol_util                 sub_grade                  tot_coll_amt               tot_cur_bal
[45] tot_hi_cred_lim            total_acc                  total_bal_ex_mort          total_bc_limit
[49] total_il_high_credit_limit total_rev_hi_lim           verification_status

Root node error: 6900/49971 = 0.13808

n= 49971

           CP nsplit rel error xerror     xstd
1  0.00038043      0   1.00000 1.0000 0.011177
2  0.00037681     47   0.97145 1.0265 0.011300
3  0.00032609     58   0.96725 1.0370 0.011347
4  0.00030303     63   0.96507 1.0432 0.011376
5  0.00028986     81   0.95797 1.0572 0.011439
6  0.00027053    101   0.95101 1.0613 0.011457
7  0.00026570    132   0.93768 1.0672 0.011484
8  0.00024155    138   0.93609 1.0742 0.011515
9  0.00023188    145   0.93391 1.0806 0.011543
10 0.00019928    150   0.93275 1.0839 0.011558
11 0.00019324    171   0.92739 1.1001 0.011628
12 0.00018116    174   0.92681 1.1093 0.011668
13 0.00017391    204   0.91957 1.1164 0.011698
14 0.00016304    214   0.91783 1.1194 0.011711
15 0.00014493    226   0.91493 1.1381 0.011791
16 0.00010870    277   0.90710 1.1522 0.011850
17 0.00010000    285   0.90623 1.1657 0.011906
```

The output above shows that a tree with 285 splits and a cross validation error of 1.1657 was created using 51 variables and an optimal cp value of 0.0001. The results can be used to assess the tree's effectiveness and pinpoint possible areas for development.

Choosing the optimal CP value involves experimenting with different values and assessing the model's performance using cross-validation. The value that results in the lowest cross-validation error rate is often selected. This is a strategy to find a balance between a model that fits the training data well and a model that generalizes effectively to new data.
In this instance, the optimal cp value is 0.0003804 which results in the lowest cross validation error of 1.

```
> optCP
[1] 0.0003804348
```

The above figure shows the pruned tree.

When dealing with imbalanced datasets, where one class significantly outnumbers the other, it's essential to address the imbalance to prevent the model from being biased toward the majority class. The prior parameter in the rpart function can be a useful tool in such scenarios. The prior parameter allows us to specify the prior probabilities for each class. By default, the prior probabilities are taken from the dataset, reflecting the class distribution in the training data. However, if the dataset is imbalanced, using the default priors may result in a model that is overly biased toward the majority class.
To address this, we can manually set the prior probabilities to be more balanced, giving equal weight to each class. This helps the model learn patterns from both classes more effectively.

```
Root node error: 24986/49971 = 0.5

n= 49971

            CP nsplit rel error  xerror      xstd
1   2.3829e-01      0   1.00000 1.00993 0.0072637
2   4.7554e-03      1   0.76171 0.76203 0.0071030
3   2.1460e-03      4   0.74745 0.75636 0.0064432
4   1.7382e-03     11   0.72950 0.75851 0.0068365
5   1.6693e-03     15   0.72097 0.75584 0.0069428
6   1.5102e-03     16   0.71930 0.75482 0.0069599
7   1.4903e-03     17   0.71779 0.75356 0.0069681
8   1.1924e-03     18   0.71630 0.75413 0.0070658
9   1.1827e-03     20   0.71391 0.75347 0.0071196
```

The below figure shows the confusion matrix for the pruned tree lcDT1.

```
> table(pred = predTrn, true=lcdfTrn$loan_status)
            true
pred         Fully Paid Charged Off
  Fully Paid       33821         386
  Charged Off       9250        6514
>
> mean(predTrn == lcdfTrn$loan_status)
[1] 0.8071682
```

```
> table(pred = predict(lcDT1,lcdfTst, type='class'), true=lcdfTst$loan_status)
             true
pred          Fully Paid Charged Off
  Fully Paid       41354        6299
  Charged Off       1736         582
```

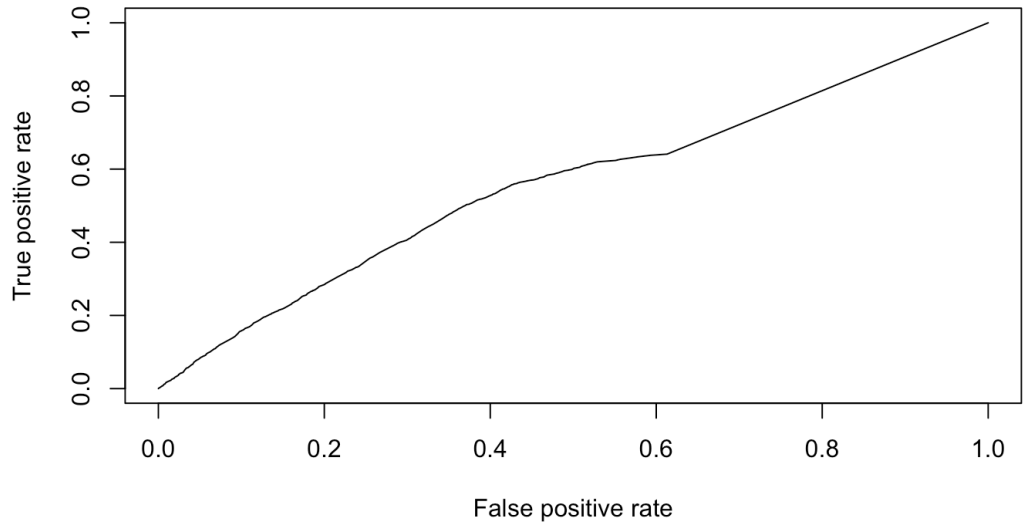
```
> mean(predict(lcDT1,lcdfTst, type='class') ==lcdfTst$loan_status)
[1] 0.8392067
```

The below figure shows the confusion matrix for the pruned tree lcDT1 with Ctresh=0.3

```
> CTHRESH=0.3
> predProbTrn=predict(lcDT1,lcdfTrn, type='prob')
> predTrnCT = ifelse(predProbTrn[, 'Charged Off'] > CTHRESH, 'Charged Off', 'Fully Paid')
> table(predTrnCT , true=lcdfTrn$loan_status)
             true
predTrnCT     Fully Paid Charged Off
  Charged Off       1424        1814
  Fully Paid       41647        5086
```

```
> table(predTstCT , true=lcdfTst$loan_status)
             true
predTstCT     Fully Paid Charged Off
  Charged Off       2443         815
  Fully Paid       40647        6066
```

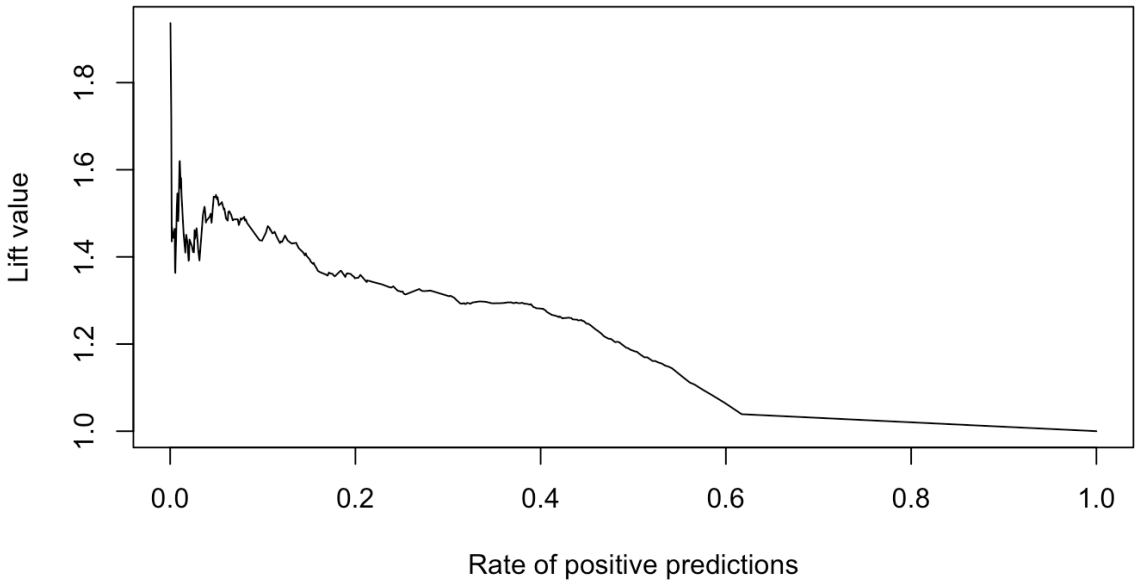
The below shows the ROC curve:



The below is AOC vallue:

```
> aucPerf@y.values
[[1]]
[1] 0.5573373
```

The below shows the lift curve:



The lift curve provides insights into how well a model is able to discriminate between positive and negative instances. A lift curve with a steep initial ascent indicates that the model is effectively distinguishing positive instances early in the ranked list. The steeper the lift curve, the more efficient the model is in identifying positive instances.

We can also examine variable importance for our decision tree. Variable importance scores indicate the contribution of each predictor variable to the model.

```
> lcDT1b$variable.importance
                sub_grade                 int_rate               emp_length            bc_open_to_buy
             2536.0129272             2116.3946374             1401.5207068             1367.1258340
              tot_cur_bal            total_bc_limit                   bc_util         total_bal_ex_mort
             1144.6621523             1122.4406663             1056.7976556             1052.3103976
         total_rev_hi_lim          tot_hi_cred_lim              avg_cur_bal                revol_bal
             1049.5190306             1019.1326282              967.6584970              932.9212066
  firstyr_pymnt_ability               revol_util        ability_to_payback                loan_amnt
              913.2012345              910.1118522              832.5550319              809.5119345
               installment                      dti total_il_high_credit_limit               annual_inc
              798.8840069              788.0517235              763.2725818              750.2678095
        credit_utilization              funded_amnt      mo_sin_old_rev_tl_op      mths_since_recent_bc
              749.6798675              739.5729832              614.7680807              606.5721296
                total_acc       mo_sin_old_il_acct     mo_sin_rcnt_rev_tl_op           num_op_rev_tl
              578.6356109              578.5342109              527.8631487              509.7396341
       percent_bc_gt_75                 open_acc             num_rev_accts                num_bc_tl
              495.9766782              488.3393160              485.8521643              456.5570631
                  purpose                    grade      acc_open_past_24mths           mo_sin_rcnt_tl
              454.4056329              440.4987830              438.5804815              429.5879624
         num_actv_rev_tl                 num_sats                num_il_tl                num_bc_sats
              418.0378701              415.4311306              398.9767382              387.5384586
      num_rev_tl_bal_gt_0      prop_SatisBC_Accnts    mths_since_last_delinq           num_actv_bc_tl
              374.1586658              366.7844047              364.2714403              344.3883409
          pct_tl_nvr_dlq     mths_since_recent_inq        num_tl_op_past_12m                 mort_acc
              337.1013123              333.0191375              305.7718978              253.1063798
         home_ownership       verification_status             tot_coll_amt     num_accts_ever_120_pd
              192.3448747              177.2972681              155.3544847              121.4102371
          inq_last_6mths                delinq_2yrs                  pub_rec      pub_rec_bankruptcies
              118.2327002               84.1211544               40.9347030               35.4461217
      initial_list_status                 tax_liens        num_tl_90g_dpd_24m collections_12_mths_ex_med
               33.2215533               17.2160776               15.0788429                7.2352746
         acc_now_delinq   chargeoff_within_12_mths             num_tl_30dpd               delinq_amnt
              1.9163490                1.7211945                0.7978301                0.5309132
```

The higher the value, the more important the feature was in the model's decision-making process.For our decision tree, the variable importance score for the variable ‘sub_grade’ is unusually high.

In our univariate analysis, we measured the correlation between our target variable ‘loan_status’ and other predictor variables. Correlation measures linear relationships. From the univariate analysis the correlation between sub grade and loan status was 0.664 which indicates a moderately strong positive linear relationship between the two variables. However, correlation might not capture complex, nonlinear relationships. The variable importance score of sub grade from our decision tree is very high which suggests that sub grade is classified as very important by the decision tree. Decision trees are capable of capturing nonlinear relationships that correlation might not fully reveal. If the relationship between "loan status" and "sub grade" is not strictly linear, the decision tree might assign high importance due to the variable's ability to capture nonlinear patterns. Decision trees can capture interaction effects between variables. If the predictive power of "sub grade" is enhanced when combined with other variables, the decision tree might assign higher importance. However, extremely high

variable importance scores might be a sign of overfitting,

6) Develop random forest and boosted tree model. 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

Let's look at the performance of the random forest model for different number of Trees.

n=200

```
> table(pred = scoreTrn$predictions[, "Fully Paid"] > 0.7, actual=lcdfTrn$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE          1        6900
  TRUE       43070           0

> table(pred = scoreTst$predictions[, "Fully Paid"] > 0.7, actual=lcdfTst$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE       2424        1113
  TRUE       40666        5768
```

n=500

```
> table(pred = scoreTrn$predictions[, "Fully Paid"] > 0.7, actual=lcdfTrn$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE        192        3102
  TRUE       42879        3798
>
> scoreTst <- predict(rfModel2,lcdfTst)
>
> table(pred = scoreTst$predictions[, "Fully Paid"] > 0.7, actual=lcdfTst$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE       1189         670
  TRUE       41901        6211
```

For different parameters

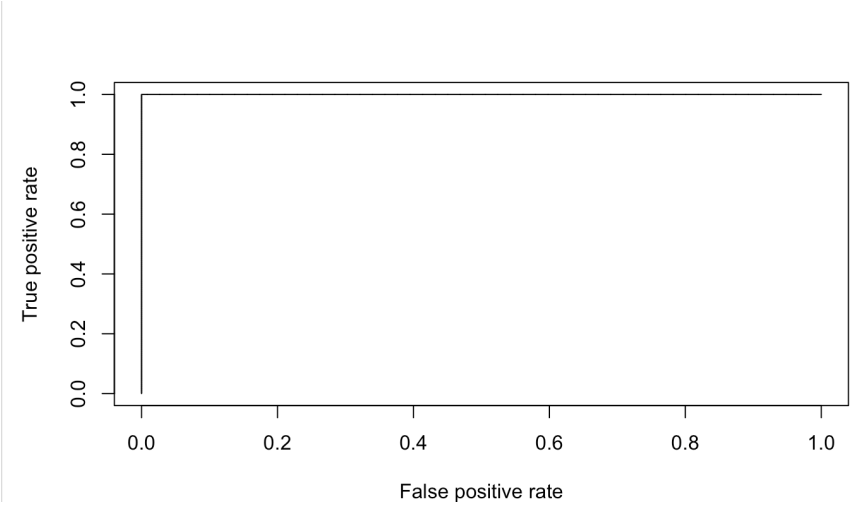
```
#Different parameters for random forest - for example, if the default model is seen to overfit
rfModel2 <- ranger(loan_status ~., data=lcdfTrn %>%  select(-all_of(varsOmit)),
                   num.trees =500, probability = TRUE, min.node.size = 50, max.depth = 15, importance='permutation')
#     min.node.size,  max.depth
```

```
> table(pred = scoreTrn$predictions[, "Fully Paid"] > 0.7, actual=lcdfTrn$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE        272        3009
  TRUE       42851        3839
```

```
> table(pred = scoreTst$predictions[, "Fully Paid"] > 0.7, actual=lcdfTst$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE       1320         644
  TRUE       41718        6289
```

```
> sprintf("AUC: %f", aucPerf@y.values)
[1] "AUC: 1.000000"
```

At N=500, the model identifies the highest number of true positives i.e. correctly identifies number of fully paid and charged off loans so we identify that as the best model.
The figure below shows the ROC curve for the random forest model performance on the test data. The AUC value for the below curve is 1.0000. A high AUC suggests that the model is good at distinguishing between the positive and negative classes overall. However, a model that is performing exceptionally well on test data could also mean that there is inadvertent data leakage, where information from the test set has unintentionally influenced the model during training.



The table below shows the variable importance of our random forest model. The model classifies loan amount as the most significant variable when predicting the likelihood of default.
Differences in variable importance rankings between Random Forest and Recursive Partitioning models can arise because each model evaluates feature importance differently. Random Forest introduces randomness in feature selection during the construction of each tree.
This can lead to different subsets of features being considered important in different trees. In rpart decision tree, feature selection is deterministic and based on optimizing the splitting criterion at each node.
Another reason could be that Random Forest tends to capture complex relationships and interactions among variables due to the ensemble of trees. Rpart may capture simpler relationships in a single tree.

For Xgboost, we can see the performance by getting the auc value and the error as follows

```
[1]     train-error:0.134515    train-auc:0.674384      eval-error:0.138483     eval-auc:0.656492
Multiple eval metrics are present. Will use eval_auc for early stopping.
Will train until eval_auc hasn't improved in 10 rounds.

[2]     train-error:0.134635    train-auc:0.676716      eval-error:0.138323     eval-auc:0.657211
[3]     train-error:0.134595    train-auc:0.676808      eval-error:0.138803     eval-auc:0.658181
[4]     train-error:0.134715    train-auc:0.682450      eval-error:0.138643     eval-auc:0.663619
[5]     train-error:0.134755    train-auc:0.683670      eval-error:0.138683     eval-auc:0.663270
[6]     train-error:0.134715    train-auc:0.683646      eval-error:0.138643     eval-auc:0.663437
[7]     train-error:0.134715    train-auc:0.684815      eval-error:0.138643     eval-auc:0.664333
[8]     train-error:0.134715    train-auc:0.684928      eval-error:0.138643     eval-auc:0.664778
[9]     train-error:0.134715    train-auc:0.685816      eval-error:0.138643     eval-auc:0.664676
[10]    train-error:0.134715    train-auc:0.686976      eval-error:0.138643     eval-auc:0.664640
[11]    train-error:0.134715    train-auc:0.687186      eval-error:0.138723     eval-auc:0.664874
[12]    train-error:0.135076    train-auc:0.688962      eval-error:0.138563     eval-auc:0.666748
```

We can best iteration at 498,

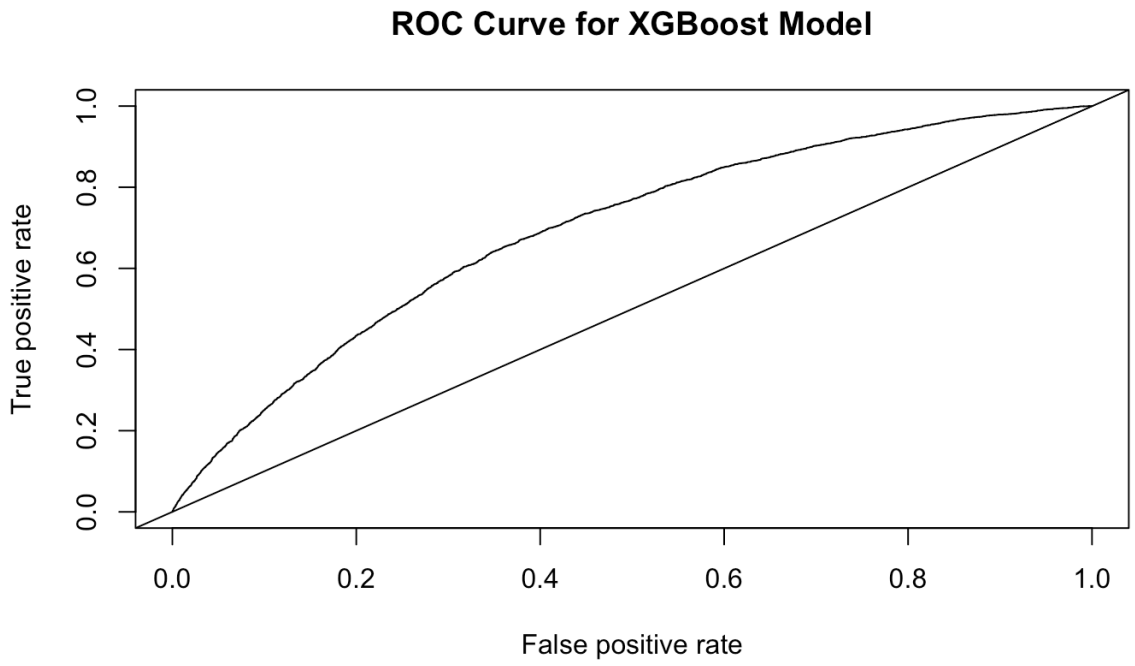
```
> cat("Best Iteration from Early Stopping:", best_iteration, "\n")
Best Iteration from Early Stopping: 498
```

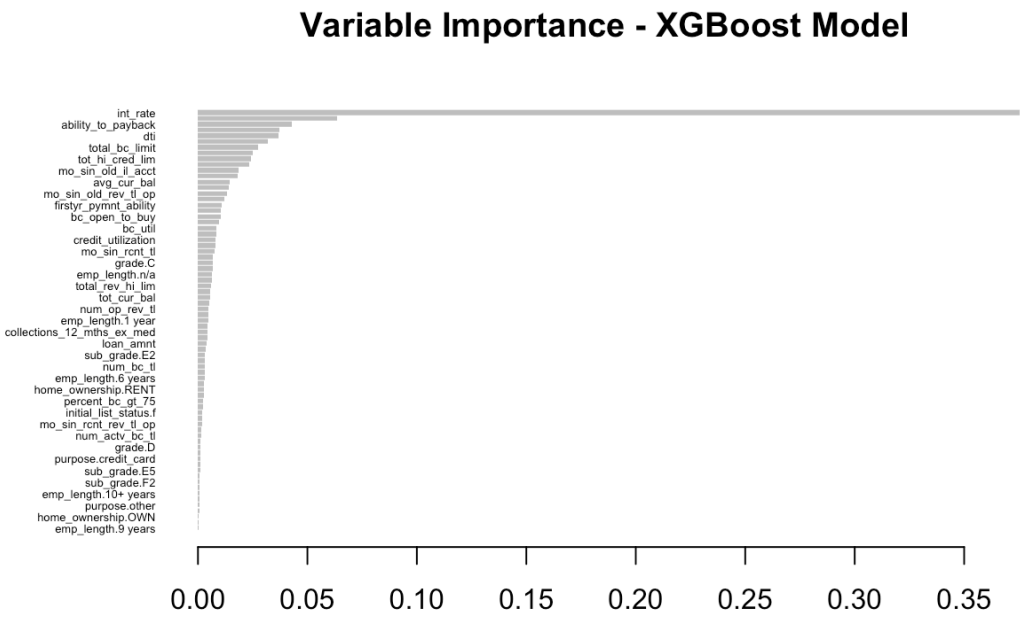
We can look at the confusion matrix of training data for fully

```
> table(pred = as.numeric(xpredTrg > 0.5), actual = colcdfTrn)
    actual
pred     0     1
   0 21591  3370
   1     0    25
>
```

Lets see the confusion matrix of test data

```
> table(pred = as.numeric(xpredTst > 0.5), actual = colcdfTst)
    actual
pred     0     1
   0 21540  3436
   1     4     5
>
```





The variable importance is as follows:

loan_amnt	funded_amnt	int_rate	installment
5.542983e-03	5.657843e-03	6.682707e-03	6.653317e-03
grade	sub_grade	emp_length	home_ownership
4.759397e-03	6.446562e-03	4.821882e-04	1.075477e-03
annual_inc	verification_status	purpose	dti
6.862099e-03	1.673727e-04	4.023680e-04	3.221585e-03
delinq_2yrs	inq_last_6mths	mths_since_last_delinq	open_acc
4.481761e-04	5.209675e-04	6.028578e-04	2.208016e-03
pub_rec	revol_bal	revol_util	total_acc
2.096445e-04	5.370952e-03	3.746935e-03	3.038395e-03
initial_list_status	collections_12_mths_ex_med	acc_now_delinq	tot_coll_amt
4.114037e-05	3.666745e-05	1.675235e-05	1.518232e-04
tot_cur_bal	total_rev_hi_lim	acc_open_past_24mths	avg_cur_bal
1.195504e-02	7.688940e-03	4.026179e-03	9.524096e-03
bc_open_to_buy	bc_util	chargeoff_within_12_mths	delinq_amnt
7.096555e-03	4.870975e-03	6.313571e-06	-5.670539e-06
mo_sin_old_il_acct	mo_sin_old_rev_tl_op	mo_sin_rcnt_rev_tl_op	mo_sin_rcnt_tl
1.327855e-03	1.878903e-03	1.555928e-03	1.693376e-03
mort_acc	mths_since_recent_bc	mths_since_recent_inq	num_accts_ever_120_pd
1.286173e-03	1.814288e-03	7.808865e-04	3.504654e-04
num_actv_bc_tl	num_actv_rev_tl	num_bc_sats	num_bc_tl
1.695050e-03	2.743258e-03	1.961169e-03	1.915516e-03
num_il_tl	num_op_rev_tl	num_rev_accts	num_rev_tl_bal_gt_0
1.686356e-03	2.774813e-03	2.830063e-03	2.970183e-03
num_sats	num_tl_120dpd_2m	num_tl_30dpd	num_tl_90g_dpd_24m
2.221457e-03	-1.289250e-06	5.407792e-06	9.558108e-05
num_tl_op_past_12m	pct_tl_nvr_dlq	percent_bc_gt_75	pub_rec_bankruptcies

The variable importance metric differs between models like **rpart** (decision trees), **random forest**, and **XGBoost** due to the unique ways in which each algorithm measures and interprets

the influence of features on predictions. Here's how each of these models evaluates variable importance:

1. rpart (Single Decision Tree)

- **Variable Importance Calculation**: In decision trees, variable importance is based on the total reduction in impurity (e.g., Gini impurity or information gain) that a variable achieves across all splits where it's used. This reduction reflects how much each feature contributes to improving classification or regression at each split in the tree.
- **Interpretation**: Since **rpart** only uses one tree, the importance is limited to the splits selected in this single structure, which may overlook other variables that might have been important in other potential splits if they had been selected. Therefore, variable importance in **rpart** is a direct result of the splits in one tree and can vary depending on the tree's specific structure.

2. Random Forest

- **Variable Importance Calculation**: Random forests calculate variable importance by averaging the impurity reduction across many trees. Each tree is built on a random subset of features, and each feature's importance is computed as the average decrease in impurity it causes across all trees in the forest. Another method, permutation importance, measures variable importance by calculating the difference in model accuracy before and after shuffling a feature's values.
- **Interpretation**: This method provides a more robust estimate of variable importance by averaging over multiple trees, which reduces the likelihood of overemphasizing any single feature. Variables that consistently contribute to splits across trees will have higher importance scores, resulting in an importance ranking that accounts for feature stability and interaction with others.

3. XGBoost

- **Variable Importance Calculation**: XGBoost measures variable importance in multiple ways, including:
    - **Gain**: The improvement in accuracy or reduction in error when a variable is used for a split.
    - **Frequency**: The number of times a feature appears in all trees (also known as "Cover").
    - **Total Cover**: The number of data points affected by splits on a given feature across all trees.

- **Interpretation**: In XGBoost, importance is based on cumulative gains across multiple trees in a sequential manner. This sequential boosting process means that importance is calculated in the context of already-built trees, capturing features that help correct previous errors. As such, variable importance in XGBoost often highlights features that are effective at reducing residual error.

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 ?

Assessing models for the business objective of making loan investment decisions involves considering both the expected profit and potential loss associated with the predicted outcomes ('Fully Paid' or 'Charged Off'). The actual profit may differ from the estimated profit due to borrower behaviour, economic conditions, and other factors. It's crucial to use historical data to refine and adjust profit estimates. For loans predicted to be 'Charged Off,' we need to assess the potential loss. This could be the total outstanding amount, including principal and interest, that may not be recovered.
The ideal model for making loan investment decisions should strike a balance between maximizing expected profit, minimizing potential loss, and considering the cost of capital. Continuous monitoring and refinement based on actual performance and evolving business conditions are essential for long-term success.
Let's see how we can choose value of profit and loss.

```
# A tibble: 2 × 3
  loan_status intRate avgActRet
  <fct>          <dbl>     <dbl>
1 Fully Paid      11.8      8.04
2 Charged Off     13.9    -12.1
```

The table above shows the average interest rate for the loans that are fully paid and the ones that were charged off. For Fully paid loans, one can consider the average interest rate on loans for expected profit. But we see that the average return on loans that are the fully paid is about 8%. Deductions for fees or operational costs can reduce the overall return. Miscellaneous factors, such as economic conditions or unexpected events, can impact returns. So the value of profit that we can consider here is 8% i.e. for every $100 invested, there is a profit of $8. The value of profit we can consider is $8.
We can see that for the charged off loans, the average return is -12% which means that for every $100 invested, the investor loses $12. So, the value of loss we can consider is $12.
For the alternate option of investing in CDs, our confusion matrix can look like,

	Predicted	
Actual	Fully Paid	Charged Off
Fully Paid	$8	$6
Charged Off	$12	$6

Ans: When assessing models for loan investment decisions, it's crucial to balance potential profit with the risk of loss. The expected profit from loans that are fully repaid can be estimated based on historical interest rates, though actual returns may be affected by operational costs, fees, and external factors like economic conditions. For loans that default (charged off), the potential loss involves the outstanding amount that is not recovered.

The model should aim to maximize profit while minimizing potential losses. Continuous monitoring and adjustment based on actual loan performance are important to maintain effective decision-making in the long term.

For the r part decision tree, the confusion matrix on test data was,

```
> table(predTstCT , true=lcdfTst$loan_status)
             true
predTstCT     Fully Paid Charged Off
  Charged Off       2443         815
  Fully Paid       40647        6066
```

Expected profit= $8(40,647)= $325,176
Expected Loss= $12(6,066)= $72792
Total Profit= $252,384

For the Random Forest, the confusion matrix was

```
> table(pred = scoreTst$predictions[, "Fully Paid"] > 0.7, actual=lcdfTst$loan_status)
       actual
pred    Fully Paid Charged Off
  FALSE       2424        1113
  TRUE       40666        5768
```

Expected profit= $8(40,666)+6(2424)= $339872
Expected Loss= $12(5768)+6(1113)= $75894
Total Profit= $263,978

For the xboost, the confusion matrix was

```
     actual
pred     0     1
   0 21540  3436
   1     4     5
```

Expected profit= $8(21540)+6(4)= $172344
Expected Loss= $12(3436)+6(5)= $41262
Total Profit= $131082

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; tabulate your results and summarize your findings. Compare performance of the best glm, rf abd gbm/xgb models. Explain what performance measures you use and why these are suitable for your task.

Answers:

Defining Returns: Returns are defined as the net profit percentage on the loan amount. Specifically:
Return=Total Payment−Funded AmountFunded AmountReturn=Funded AmountTotal Payment−Funded Amount
Ideally, this would also account for Lending Club's service costs if included in the data.

On Random Forest model, we experimented with number of trees.

- On GBM, we performed a grid search on minimum observations at each node, bag fraction values, number of trees to find which combination gave minimum root mean square values.

- On GLM, we performed ridge and lasso regression with multiple lambda values to get minimum RMSE.

- Actual return being a regression variable. Model performance is done basis RMSE, ie root of minimum square error.

- It takes the difference of squares between actual values and predicted values.

- The model that shows lowest RMSE is the best performing. Values are shown in the table below.

- In our case, GBM on existing data itself provides lowest RMSE on testing data.

Model <chr>	Train_RMSE <dbl>	Test_RMSE <dbl>
GLM	0.0901560128	0.0913407110
Random Forest	0.0303333211	0.0722468465
GBM	0.0003916733	0.0004554351

- GLM has the highest RMSE values among the three models, indicating it may not capture the complexity of the loan returns effectively.
- Random Forest shows significant improvement with lower RMSE on both training and testing data, suggesting it performs better than GLM at capturing non-linear relationships.

- GBM (Gradient Boosting Machine) has the lowest RMSE values for both training and testing, indicating it's the most accurate model for predicting loan returns in this context.

Comparing the three models and the parameters used for each one of it

Interpretation and Recommendation:

1. Best Model for Predicting Loan Returns: Given that GBM has the lowest RMSE on test data (0.0005), it's clearly the best model for predicting loan returns. The low RMSE indicates that the GBM model closely aligns predicted returns with actual returns.
2. Systematic Parameter Tuning:
    - Random Forest: Tuned by adjusting the number of trees.
    - GBM: Tuned by performing a grid search across parameters like interaction depth, number of trees, and shrinkage rate. This fine-tuning likely contributed to the very low RMSE achieved.
3. Performance Metric: RMSE was used as the performance metric, which is suitable for this regression task as it directly measures the difference between actual and predicted returns.

Investment Strategy Using Combined Models:

Since we have two objectives—predicting loan status (Fully Paid or Charged Off) and predicting loan returns—an effective investment strategy should combine both models:

1. Loan Selection Using Loan Status Model (GLM):
    - First, filter loans with a high probability of being "Fully Paid" based on the GLM model.
    - This initial filter helps reduce the risk of investing in loans that are likely to default.
2. Return Prioritization Using Return Prediction Model (GBM):
    - Among the loans predicted to be "Fully Paid," rank them by the expected returns predicted by the GBM model.
    - This ranking allows prioritizing loans that not only have a low risk of default but also offer the highest potential returns.
3. Consideration of Loan Grades:
    - Based on previous findings, higher-grade loans (e.g., A and B) have lower default risks but also lower returns. However, lower-grade loans (C, D, and E) can yield higher returns if they don't default.

    - This approach enables a balanced strategy by investing in lower-grade loans with high Fully Paid probabilities, thereby capitalizing on the potential for higher returns while managing risk.
4. Benchmarking Against Alternatives (e.g., CD Investment):
    - If investing in a 3-year CD at a 2% interest rate provides a $6 profit on a $100 investment, this benchmark can help evaluate the effectiveness of this loan-based investment strategy.
    - Loans selected with the combined GLM and GBM models should ideally yield a return exceeding this benchmark to justify the additional risk.

Conclusion:

The combined approach of using GLM for loan status prediction and GBM for return prediction is the most effective for making investment decisions. This strategy:

- Maximizes expected returns by prioritizing loans with high potential returns.
- Manages risk by filtering out loans likely to default.

This approach provides a better balance between risk and reward compared to using either model in isolation. Additionally, GBM outperforms other models for return prediction, making it the core model for guiding investment choices in loans.

9. Considering results from both the best model for predicting loan-status and the best model 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)– clearly describe your approach and the rationale, and show performance. How does performance here compare with use of single models (i.e for models predicting loan- status, or loan returns separately)?

Best Models for Loan-Status and Returns: For predicting loan-status (Fully Paid vs. Charged Off), the **GLM model**performed best, based on AUC. For predicting returns, **GBM** showed the lowest RMSE.

Investment Decision Approach: Since loan status is a classification variable, and return is a regression variable, comparing them directly for investment selection is challenging. To combine information from both models:

- **Decile Analysis**: We created decile charts for loan status and actual returns separately. This analysis highlighted that loans with grades **A and B** had higher probabilities of being Fully Paid (lower risk), but offered lower returns. Conversely, loans with grades **C, D, and E** showed higher returns, despite being slightly riskier.
- **Combining Information**: A practical approach is to prioritize loans with higher probabilities of being Fully Paid and then rank them based on their predicted returns. This dual approach helps maximize returns while managing risk. Thus, focusing on grades **C, D, and E** appears to be the most promising investment strategy, balancing potential returns with manageable risk.

10.As seen in data summaries and your work in Part A, 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? Which approach will you recommend to a client for making investment decisions? And, very importantly, why.

Answers:

Background: Higher-grade loans (A and B) are less likely to default but come with lower interest rates. Lower-grade loans, particularly **C and below**, show a greater likelihood of being fully paid while yielding higher returns on average.

Modeling Approach: We trained GLM, Random Forest, and GBM models specifically on lower-grade loans (C, D, E, and F). This subset allowed us to focus on loans that are riskier but have higher potential returns.

Model Comparison and Findings:

- The performance of each model was evaluated based on RMSE for return prediction and AUC for loan-status prediction.
- **GBM** again provided the lowest RMSE for predicting returns, and **GLM** showed the best AUC for loan-status.
- While lower-grade loans tend to yield higher returns, they also present a higher risk of default. Our analysis indicates that **focusing exclusively on lower-grade loans might not be ideal**, as these loans present a risk comparable to higher-grade loans (A and B), without a proportionate increase in returns.

Conclusion: Investing solely in lower-grade loans (C to F) does not seem to be a viable strategy despite their higher returns. A diversified approach, including high-grade loans, allows investors to benefit from stability in A and B grades while also capturing the higher returns of lower-grade loans that have been screened for lower default risk.

Overall Recommendation for Investment Decisions

- **Combined Strategy**: Based on the analysis, a **balanced investment strategy** is recommended:
  - Focus primarily on loans with grades **A and B** for stability, ensuring a lower risk of default.
  - Include select loans from grades **C, D, and E** that have high probabilities of being Fully Paid, based on the loan-status model, and high predicted returns, based on the return model.
- **Why This Strategy?**
  - **Risk Management**: High-grade loans provide a steady, predictable return, mitigating the risk associated with lower-grade loans.
  - **Return Optimization**: Lower-grade loans that meet the criteria for high return and low default probability can enhance portfolio returns without excessive risk exposure.
  - **Comparison with CDs**: This combined strategy provides returns higher than those achievable with a conservative investment in CDs, without exposing the investor to unnecessary risk from lower-grade loans exclusively.

This strategy offers a balanced approach, aligning with the business objective of maximizing returns while managing risk effectively. The combination of GLM for loan-status and GBM for returns enables investors to make informed decisions based on both the probability of repayment and potential profit from each loan.