Loans in India

Laura Carralero and Ana Polo

13/10/2024

Contents

0.1 Introduction

India is a country with a very large population (the probability of being born in India is 17,7%) and with significant social and economic disparities among its people. Due to this, we found it interesting to observe which variables influence the ability to obtain a financial loan within its population. We focused on variables such as the loan amount, the annual income of the applicant or the education to conduct this analysis. Since a considerate part of the population is not even allowed to request a loan due to the caste system imposed in India, our population only represent a small portion of the country.

The caste system continues to play a significant role in determining access to economic resources and opportunities in India. While legal frameworks and government initiatives aim to bridge the gap, caste-based economic exclusion remains a major issue, particularly in terms of access to loans. This topic will arise several times on the analysis since there is a huge difference between their values on different assets, their education, etcetera, which might affect the result of the loan status.

To make this analysis more complete and focus on the actual situation of this country, we have relied on the following articles: (Bandyopadhyay 2016), (Tiwari 2013) and (Abhiman 2007).

0.2 Our Loans Dataset

Our dataset is about the parameters that made a loan be approved or denied, in particular this data set contains data from 4269 applicants from India. We have chosen this dataset because, taking into consideration our own situations and concerns, we would like to investigate how easy it is to get a loan from a bank account in that country. Just to compare if the Indian population face the same problems as the Spanish youth, which are the need of a huge amount of money to even ask for a loan (in our case to be able to buy a house). So let's study some parameters on this applicants and what are the main factors that affect if a loan is approved or not.

Out dataset is from kaggle and can be found here: https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset

The table consists on *11 columns* and we have created two more variables that might be of interest for some analysis. The variables are the next ones:

- **loan_id** (id): a number to identify each applicant and protect his/her privacy
- **education** (char): if the applicant has graduated from university or not
- **no_of_dependents** (num): the number of dependents from each applicant
- **income_an** (num): the annual income of the borrower
- **loan_amount** (num): the amount of asked from the bank
- **loan_term** (num): the duration for which the loan is taken in years
- **cibil_score** (num): the cibil score, which is a specific type of credit score used in India reflecting their ability to repay loans. The credit risk is the possibility of loss due to a borrower's defaulting on a loan or not meeting contractual obligations. This cibil score range from 300 to 900, where a low CIBIL score typically indicates that an individual has poor creditworthiness or is considered a higher credit risk by lenders.
- **residential_assets_value** (num): the value of the borrower's residential assets, for example his/her properties, house, etcetera.
- **commercial_assets_value** (num): The value of any commercial assets owned by the borrower (e.g., business properties). This can also indicate financial stability.

- **luxury__assets__value** (num): The value of luxury items owned by the borrower (e.g., expensive cars, jewelry). While not essential, this can reflect the borrower's lifestyle and potential financial strain.
- **bank__asset__value** (num): The total value of assets held in the bank by the borrower. This provides insight into the borrower's liquid assets.
- **loan__status** (char.): Indicates the status of the loan (approved or rejected). This is crucial for understanding the outcome of loan applications.

We also have created these two variables:

- **Cibil__score__char**: we have created a variable from the cibil score that group the applicants based on their risk of non-payment. We have follow this table to create the categories:

Cibil Score	Category
300-549	Poor credit risk
550-649	Fair credit risk
650-749	Good credit risk
750-900	Excellent credit risk

- **Total assets**: the sum of residential__assets__value, commercial__assets__value , bank__asset__value and luxury__assets__value.

Note that **we haven't deleted the variables that sum up the total assets** because each of them may have different impact on the situation of the loan, so it may be interesting to study them one by one.

The first code that may be interesting are the ones that enable as to have a basic notion on the table, its columns plus the ones we have created:

```
#Let call our table as loans.
loans <- read.csv("C:/Users/laull/Programación R/loan_approval_dataset.csv")

library(dplyr)
library(pander)
library(ggplot2)

#We create the columns described before
loans$total_assets = rowSums (cbind(loans$residential_assets_value, loans$commercial_assets_value, loans$bank_asset

loans <- loans %>%
  mutate(cibil_score_char = case_when(
    cibil_score >= 300 & cibil_score <= 549 ~ "Poor credit risk",
    cibil_score >= 550 & cibil_score <= 649 ~ "Fair credit risk",
    cibil_score >= 650 & cibil_score <= 749 ~ "Good credit risk",
    cibil_score >= 750 & cibil_score <= 900 ~ "Excellent credit risk",
    TRUE ~ NA_character_

  ))
# View the first few rows
pander(head(loans))
```

Table 2: Table continues below

loan_id	no_of_dependents	education	self_employed	income_annum
1	2	Graduate	No	9600000
2	0	Not Graduate	Yes	4100000
3	3	Graduate	No	9100000
4	3	Graduate	No	8200000
5	5	Not Graduate	Yes	9800000
6	0	Graduate	Yes	4800000

Table 3: Table continues below

loan_amount	loan_term	cibil_score	residential_assets_value
29900000	12	778	2400000
12200000	8	417	2700000
29700000	20	506	7100000
30700000	8	467	18200000
24200000	20	382	12400000
13500000	10	319	6800000

Table 4: Table continues below

commercial_assets_value	luxury_assets_value	bank_asset_value	loan_status
17600000	22700000	8e+06	Approved
2200000	8800000	3300000	Rejected
4500000	33300000	12800000	Rejected
3300000	23300000	7900000	Rejected
8200000	29400000	5e+06	Rejected
8300000	13700000	5100000	Rejected

total_assets	cibil_score_char
50700000	Excellent credit risk
1.7e+07	Poor credit risk
57700000	Poor credit risk
52700000	Poor credit risk
5.5e+07	Poor credit risk
33900000	Poor credit risk

```r
# Check the type of data
sapply(loans, class)
```

```
                 loan_id         no_of_dependents                education 
               "integer"                "integer"              "character" 
           self_employed             income_annum              loan_amount 
             "character"                "integer"                "integer" 
               loan_term              cibil_score residential_assets_value 
               "integer"                "integer"                "integer" 
 commercial_assets_value      luxury_assets_value         bank_asset_value 
               "integer"                "integer"                "integer" 
             loan_status             total_assets         cibil_score_char 
             "character"                "numeric"              "character" 
```

We have also checked if the data was balanced, this means that the number of loans approved are around half of the sample:

```
nrows = nrow(loans)
approved = nrow(loans[trimws(loans$loan_status) == "Approved",
    ])
print(approved/nrows)
```

```
[1] 0.6221598
```

Since around 62% of the applicants has their loan approved, we can consider that the data is balanced and therefore easier to work with in the second part of the assignment.

Apart from that, we should make other important steps before beginning the analysis:

1. Check if there is any missings on the data

```
missings = colSums(is.na(loans))
cat("The missings are:\n")
```

```
The missings are:
```

```
print(missings)
```

```
                 loan_id         no_of_dependents                education 
                       0                        0                        0 
           self_employed             income_annum              loan_amount 
                       0                        0                        0 
               loan_term              cibil_score residential_assets_value 
                       0                        0                        0 
 commercial_assets_value      luxury_assets_value         bank_asset_value 
                       0                        0                        0 
             loan_status             total_assets         cibil_score_char 
                       0                        0                        0 
```

2. Check for duplicates:

```
# Check for rows duplicated but with different ID

cols_to_check <- loans[, !names(loans) %in% "loan_id"]
num_duplicados <- sum(duplicated(cols_to_check))
print(paste("There are", num_duplicados, "rows with the same values"))
```

```
[1] "There are 0 rows with the same values"
```

```
# Check for duplicate Loan_ids
dup_ids = sum(duplicated(loans$Loan_id))
print(paste("There are", dup_ids, "rows with the same ID"))
```

```
[1] "There are 0 rows with the same ID"
```

3. Change the character columns into factors, so we can group afterwards by their categories.

```
loans$education <- as.factor(loans$education)
loans$self_employed <- as.factor(loans$self_employed)
loans$loan_status <- as.factor(loans$loan_status)
loans$cibil_score_char <- as.factor(loans$cibil_score_char)
```

This change has several benefits since converting columns to factors in R improves memory efficiency by storing categorical data as integers rather than strings, making it ideal for large datasets. It also ensures that categorical variables are treated correctly in statistical models, enabling proper contrasts and groupings.

Additionally, factors help with data visualization, allowing better control over category ordering and labeling, while maintaining data integrity by limiting values to predefined levels.

0.3 Analysis with frequency tables

The frequency tables allow as to study how the data is distributed in different categories, if the variable is categorical or in different bins if we study a continuous variable.

For this section we are going to use the **fdth** library in R, that allow as to create these frequency tables for categorical and numerical data. In particular, we find useful to separate the ranges in the same name of intervals, that way we can study also the variability of the continuous variables by the length of the bins. To choose how many bins there are I will follow the **Sturge's rule**, that calculates the number of intervals based on the number of observations following the next formula:

$$\text{Number of intervals} = \lceil Log(N) \rceil + 1,$$

where N is the number of observations (sample size), and we are using the ceiling function (round up) over the logarithm of N. In our case, since we have a small dataset this is a rule that fits well with our data and the result are 14 intervals.

```
library(fdth)

# Step 1: Define the bounds of the range
income_min <- min(loans$income_annum, na.rm = TRUE)
income_max <- max(loans$income_annum, na.rm = TRUE)

# Define number of bins based on the Sturge´s rule
n_bins <- ceiling(log(nrows, base = 2)) + 1
bin_width <- (income_max - income_min)/n_bins
print(paste("The bin width is", bin_width))
```

```
[1] "The bin width is 692857.142857143"
```

```
# Define a vector with breakpoints between bins
breaks_custom <- seq(income_min, income_max, by = bin_width)

# Step 2: Use the cut() function to bin the data
loans$income_binned <- cut(loans$income_annum, breaks = breaks_custom,
    include.lowest = TRUE, right = FALSE)

# Step 3: Create a frequency table
freq_table <- as.data.frame(table(loans$income_binned))

# Rename the columns for clarity
colnames(freq_table) <- c("Income_Range", "Frequency")
```

```r
# Step 4: Calculate midpoints (class marks), density, and
# percentage Lower bounds of the bins
freq_table$Lower.bound <- breaks_custom[-length(breaks_custom)]
# Upper bounds of the bins
freq_table$Upper.bound <- breaks_custom[-1]
freq_table$Midpoints <- (freq_table$Lower.bound + freq_table$Upper.bound)/2
freq_table$Cumulative_Frequency <- cumsum(freq_table$Frequency)

# Calculate percentage of total
total_count <- sum(freq_table$Frequency)
freq_table$Percentage <- (freq_table$Frequency/total_count) *
    100

freq_table = freq_table %>%
    mutate(across(where(is.numeric), round, 2)) %>%
    mutate(Percentage = paste0(Percentage, "%"))


# Step 5: View the final frequency table
pander(freq_table)
```

Table 6: Table continues below

Income_Range	Frequency	Lower.bound	Upper.bound	Midpoints	Cumulative_Frequency	Percentage
[2e+05,8.93e+05)	309	2e+05	892857	546429	309	7.24%
[8.93e+05,1.59e+06)	279	892857	1585714	1239286	588	6.54%
[1.59e+06,2.28e+06)	306	1585714	2278571	1932143	894	7.17%
[2.28e+06,2.97e+06)	305	2278571	2971429	2625000	1199	7.14%
[2.97e+06,3.66e+06)	292	2971429	3664286	3317857	1491	6.84%
[3.66e+06,4.36e+06)	324	3664286	4357143	4010714	1815	7.59%
[4.36e+06,5.05e+06)	309	4357143	5050000	4703571	2124	7.24%
[5.05e+06,5.74e+06)	322	5050000	5742857	5396429	2446	7.54%
[5.74e+06,6.44e+06)	294	5742857	6435714	6089286	2740	6.89%
[6.44e+06,7.13e+06)	325	6435714	7128571	6782143	3065	7.61%
[7.13e+06,7.82e+06)	301	7128571	7821429	7475000	3366	7.05%
[7.82e+06,8.51e+06)	309	7821429	8514286	8167857	3675	7.24%
[8.51e+06,9.21e+06)	312	8514286	9207143	8860714	3987	7.31%
[9.21e+06,9.9e+06]	282	9207143	9900000	9553571	4269	6.61%

As we can see, the minimum annual income is 200.000 and the maximum is 9.900.000, moreover we can confirm that the annual income of our applicants is distributed smoothly between this range, since all the 14 intervals have around 6-7% of the total sample. From this we can conclude that the people that asks for a loan does nos represent the overall population of India, since the annual income of a population usually follows a right-skewed distribution, because most of the people have a lower annual income compare to the salaries of the upper class, which are a minority. This supports the idea we mentioned at the introduction that only a small percentage of the whole population ask for loans.

Let's see the frequency table of the other variables:

```r
# Step 1: Define the bounds of the range
tot_assets_min <- min(loans$total_assets, na.rm = TRUE)
tot_assets_max <- max(loans$total_assets, na.rm = TRUE)

# Define the bins width and a vector with breakpoints
# between bins
bin_width2 <- (tot_assets_max - tot_assets_min)/n_bins
print(paste("The bin width is", bin_width2))
```

```
[1] "The bin width is 6450000"
```

```r
breaks_custom2 <- seq(tot_assets_min, tot_assets_max, by = bin_width2)

# Step 2: Use the cut() function to bin the data
loans$tot_assets_binned <- cut(loans$total_assets, breaks = breaks_custom2,
    include.lowest = TRUE, right = FALSE)

# Step 3: Create a frequency table
freq_table2 <- as.data.frame(table(loans$tot_assets_binned))

# Rename the columns for clarity
colnames(freq_table2) <- c("Income_Range", "Frequency")

# Step 4: Calculate midpoints (class marks), density, and
# percentage Lower bounds of the bins
freq_table2$Lower.bound <- breaks_custom2[-length(breaks_custom2)]
# Upper bounds of the bins
freq_table2$Upper.bound <- breaks_custom2[-1]
freq_table2$Midpoints <- (freq_table2$Lower.bound + freq_table2$Upper.bound)/2
freq_table2$Bin_Width <- freq_table2$Upper.bound - freq_table2$Lower.bound
freq_table2$Cumulative_Frequency <- cumsum(freq_table2$Frequency)
# Calculate percentage of total
total_count2 <- sum(freq_table2$Frequency)
freq_table2$Percentage <- (freq_table2$Frequency/total_count2) *
    100

freq_table2 = freq_table2 %>%
    mutate(across(where(is.numeric), round, 2)) %>%
    mutate(Percentage = paste0(Percentage, "%"))

# Step 5: View the final frequency table
pander(freq_table2)
```

Table 8: Table continues below

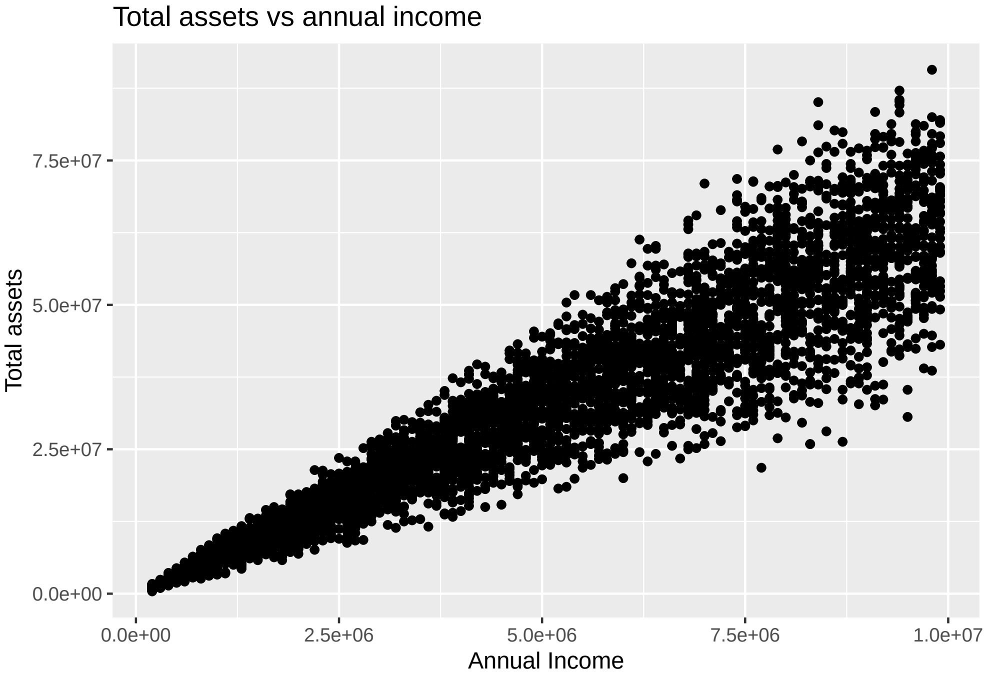
Income_Range	Frequency	Lower.bound	Upper.bound	Midpoints	Bin_Width	Cumulative_Frequency	Percentage
[4e+05,6.85e+06)	416	4e+05	6850000	3625000	6450000	416	9.74%
[6.85e+06,1.33e+07)	441	6850000	13300000	10075000	6450000	857	10.33%
[1.33e+07,1.98e+07)	449	13300000	19750000	16525000	6450000	1306	10.52%
[1.98e+07,2.62e+07)	455	19750000	26200000	22975000	6450000	1761	10.66%
[2.62e+07,3.26e+07)	461	26200000	32650000	29425000	6450000	2222	10.8%
[3.26e+07,3.91e+07)	468	32650000	39100000	35875000	6450000	2690	10.96%
[3.91e+07,4.55e+07)	426	39100000	45550000	42325000	6450000	3116	9.98%
[4.55e+07,5.2e+07)	367	45550000	5.2e+07	48775000	6450000	3483	8.6%
[5.2e+07,5.84e+07)	310	5.2e+07	58450000	55225000	6450000	3793	7.26%
[5.84e+07,6.49e+07)	222	58450000	64900000	61675000	6450000	4015	5.2%
[6.49e+07,7.13e+07)	150	64900000	71350000	68125000	6450000	4165	3.51%
[7.13e+07,7.78e+07)	70	71350000	77800000	74575000	6450000	4235	1.64%
[7.78e+07,8.42e+07)	28	77800000	84250000	81025000	6450000	4263	0.66%
[8.42e+07,9.07e+07]	6	84250000	90700000	87475000	6450000	4269	0.14%

We can see from this table what we discussed earlier: as expected, the total assets of the applicants are concentrated in the lower values. We can calculate that 81.5% of the entire sample falls below the ninth interval. This indicates that a minority of applicants possess significantly higher total assets compared to the rest. Additionally, we notice that the bandwidth is slightly less than 10 times the previous one, which means there is considerably more variance in total assets than in annual incomes. In other words, individuals in different categories of total assets can exhibit significant variation in the values of this variable.

Why does this happen? Several factors might contribute to these differences between the two variables. One possible explanation is that while salaries don't differ significantly, the applicants' properties or assets may fluctuate more. This could be due to the fact that certain types of assets are not accessible to most of the population—perhaps due to high costs, lack of knowledge about assets, or limited use of banks in India from the majority of the population. For example, luxury assets might not be available to many people because of their price or the castes system mentioned in the introduction.

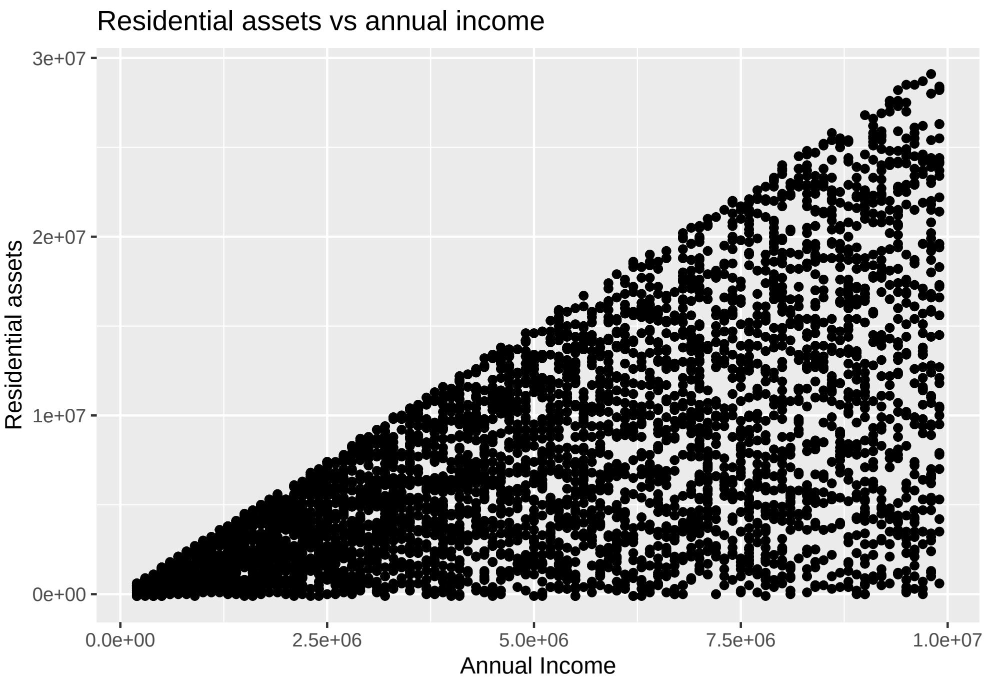
This statement can be supported by watching the scatter plot that contains both variables. In this plot we should see that only people with higher values have access to higher assets. Moreover, as we have said these assets will be more fluctuating, in other words, they will have a considerable variance even in the same range of salaries, since the assets have more dispersion than the salary. Let's check out the plot:

```
library(ggplot2)
plot = ggplot(loans, aes(x = income_annum, y = total_assets))
plot + geom_point() + labs(title = "Total assets vs annual income",
    x = "Annual Income", y = "Total assets")
```



Furthermore, if we look into some particular asset, we are going to see that the possesion of this asset change a lot depending on the annual income. In particular, we are going to study the residential asset, this is the output:

```
plot = ggplot(loans, aes(x = income_annum, y = residential_assets_value))
plot + geom_point() + labs(title = "Residential assets vs annual income",
    x = "Annual Income", y = "Residential assets")
```



This indicates that once a certain income threshold is crossed, access to housing becomes much more attainable. Not only does it become easier, but it also increases exponentially. This can occur either because the homes of higher-income individuals are significantly more expensive than average, or because the number of properties they own rises considerably—whether for personal use, such as vacation homes, or as a secondary source of income.

This give us no hope for two students that may start in the low range of the annual income.

Now we will study the last variable on this section the **loan amount**:

```
# Step 1: Define the bounds of the range
loan_amount_min <- min(loans$loan_amount, na.rm = TRUE)
loan_amount_max <- max(loans$loan_amount, na.rm = TRUE)

# Define a vector with breakpoints between bins
bin_width3 <- (loan_amount_max - loan_amount_min)/n_bins
print(paste("The bin width is", bin_width3))
```

```
[1] "The bin width is 2800000"
```

```
breaks_custom3 <- seq(loan_amount_min, loan_amount_max, by = bin_width3)

# Step 2: Use the cut() function to bin the data
loans$loan_amount_binned <- cut(loans$loan_amount, breaks = breaks_custom3,
    include.lowest = TRUE, right = FALSE)

# Step 3: Create a frequency table
freq_table3 <- as.data.frame(table(loans$loan_amount_binned))

# Rename the columns for clarity
colnames(freq_table3) <- c("Income_Range", "Frequency")

# Step 4: Calculate midpoints (class marks), density, and
# percentage Lower bounds of the bins
freq_table3$Lower.bound <- breaks_custom3[-length(breaks_custom)]
# Upper bounds of the bins
freq_table3$Upper.bound <- breaks_custom3[-1]
freq_table3$Midpoints <- (freq_table3$Lower.bound + freq_table3$Upper.bound)/2
freq_table3$Cumulative_Frequency <- cumsum(freq_table3$Frequency)

# Calculate percentage of total
total_count3 <- sum(freq_table3$Frequency)
freq_table3$Percentage <- (freq_table3$Frequency/total_count3) *
    100

freq_table3 = freq_table3 %>%
    mutate(across(where(is.numeric), round, 2)) %>%
    mutate(Percentage = paste0(Percentage, "%"))

# Step 5: View the final frequency table
pander(freq_table3)
```

Table 10: Table continues below

Income_Range	Frequency	Lower.bound	Upper.bound	Midpoints	Cumulative_Frequency	Percentage
[3e+05,3.1e+06)	398	3e+05	3100000	1700000	398	9.32%
[3.1e+06,5.9e+06)	410	3100000	5900000	4500000	808	9.6%
[5.9e+06,8.7e+06)	410	5900000	8700000	7300000	1218	9.6%
[8.7e+06,1.15e+07)	432	8700000	11500000	10100000	1650	10.12%
[1.15e+07,1.43e+07)	433	11500000	14300000	12900000	2083	10.14%
[1.43e+07,1.71e+07)	430	14300000	17100000	15700000	2513	10.07%
[1.71e+07,1.99e+07)	432	17100000	19900000	18500000	2945	10.12%
[1.99e+07,2.27e+07)	382	19900000	22700000	21300000	3327	8.95%
[2.27e+07,2.55e+07)	305	22700000	25500000	24100000	3632	7.14%
[2.55e+07,2.83e+07)	231	25500000	28300000	26900000	3863	5.41%
[2.83e+07,3.11e+07)	200	28300000	31100000	29700000	4063	4.68%
[3.11e+07,3.39e+07)	106	31100000	33900000	32500000	4169	2.48%
[3.39e+07,3.67e+07)	72	33900000	36700000	35300000	4241	1.69%
[3.67e+07,3.95e+07]	28	36700000	39500000	38100000	4269	0.66%

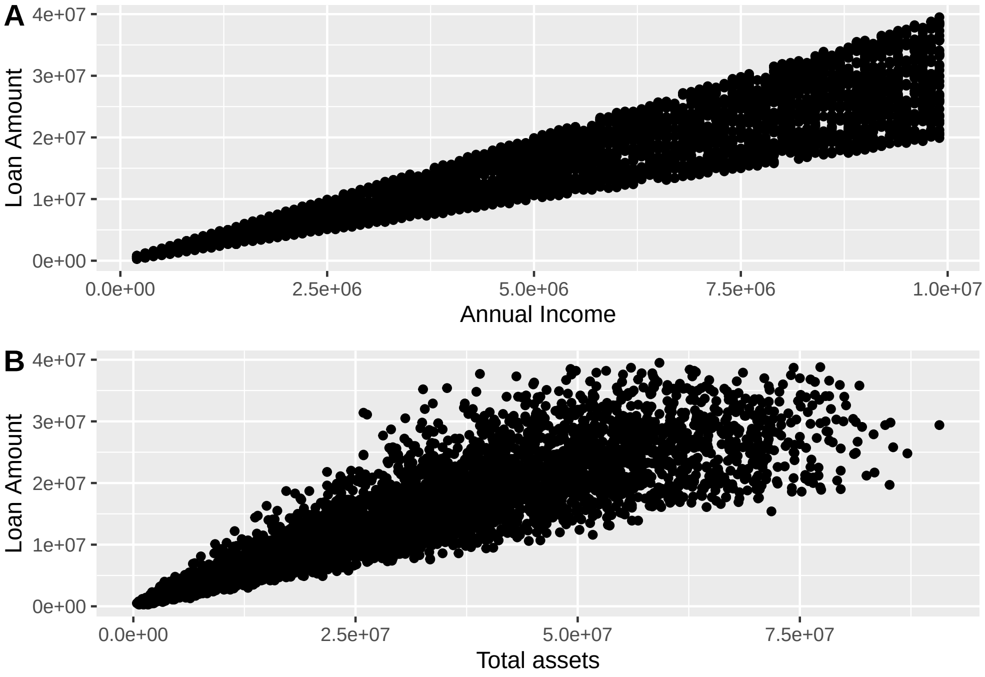
From this table we can see that the distribution of the loans is homogeneus in the intervals with lower values and then decrease rapidly in the last six intervals. This means that while most of the people ask for low and medium loans, there is a minority whose loans are much higher compare to the others. This does not come as a surprise since most of the people has lower values on their assets, and it is expected that the loan each applicant ask for depends on the assets they own and their salaries. Furthermore, only those who have higher values of assets will be able to obtain the highest loans, so people are supposed to ask for loans which are proportionate to their assets.

Let's check that the loan amount increases at the same time as the total assets and the salary amount.

```r
library(cowplot)
# Salary vs loan amount
plot1 = ggplot(loans, aes(x = income_annum, y = loan_amount)) +
    geom_point() + labs(x = "Annual Income", y = "Loan Amount")

# Total_assets vs loan_amount
plot2 = ggplot(loans, aes(x = total_assets, y = loan_amount)) +
    geom_point() + labs(x = "Total assets", y = "Loan Amount")

# show both toguether
plot_grid(plot1, plot2, labels = c("A", "B"), ncol = 1, nrow = 2) +
    labs(title = "Distribution of Loan amount through annual income vs total assets")
```



We can state that while loan amount increases, both in value and variance, while the other variables increases; there is much more **heteroscedasticity** in the total assets. Meaning that there is a more linear dependecy on the annual income and the loan amount; while having a great value of total assets would tell as that the loan asked is probably higher than the average but we wouldn't be able to approximate a small range for it.

Another analysis of interest that we may approach after seeing this graphics is to compute the The Pearson correlation coefficient between the Loan amount and these two variables.

0.3.0.1 The Pearson correlation coefficient The Pearson correlation coefficient is a statistical measure that quantifies the linear relationship between two variables. It ranges from -1 to 1:

- -1: Indicates a perfect negative correlation, meaning that as one variable increases, the other decreases in a perfectly linear fashion.
- 0: Indicates no correlation between the variables.
- 1: Indicates a perfect positive correlation, meaning that as one variable increases, the other increases in a perfectly linear fashion.

So watching the plots we expect both pearson correlation coefficients being positive but the coefficient between the annual income and the loan amount would be closer to 1 than the coefficient between the total assets and the loan amount, since these two last variables seems to have a less linear dependence between each other than the first pair. If we compute this coefficient in R we would have the result we have explained:

```
correlation_coefficient1 <- cor(loans$loan_amount,
    loans$income_annum)
correlation_coefficient2 <- cor(loans$loan_amount,
    loans$total_assets)

print(paste("The correlation coefficient between loan amount and annual income is",
    correlation_coefficient1))
```

```
[1] "The correlation coefficient between loan amount and annual income is 0.927469910987149"
```

```
print(paste("The correlation coefficient between loan amount and total assets is",
    correlation_coefficient2))
```

```
[1] "The correlation coefficient between loan amount and total assets is 0.867067020777455"
```

0.4 Analysis of measures of centrality, variability, and shape

Now we are going to make some analysis more in detail for some continuous variables, in particular, we are going to analyze the continuos variables of the last section and maybe look more into detail of one of the assets.

For this section, we are going to use the **psych** package in R , which provides tools for performing various statistical analyses, including descriptive statistics, factor analysis, reliability analysis (e.g., Cronbach's alpha), and principal component analysis (PCA). It's widely used for analyzing survey data and understanding the relationships between different variables. The package also includes functions for handling missing data and data visualization, making it highly versatile for complex datasets. Additionally, the package is highly flexible, allowing for easy customization of analyses and integration with other R packages for comprehensive data exploration and modeling.

The first analysis we are going to make are focused on understanding the distribution of two of the variables analyzed before: the anual income and the loan amount.

Let's begin analyzing the annual income:

```
library(psych)
descr_anual_income <- loans %>%
    select(income_annum) %>%
    describe()

# Round the decimals to two digits
descr_anual_income = descr_anual_income %>%
    mutate(across(where(is.numeric), round, 2))

pander(descr_anual_income)
```

Table 12: Table continues below

	vars	n	mean	sd	median	trimmed
income_annum	1	4269	5059124	2806840	5100000	5065467

	mad	min	max	range	skew	kurtosis	se
income_annum	3558240	2e+05	9900000	9700000	-0.01	-1.18	42959

These are the caractheristics that we learn from this data:

- **Data symmetrically distributed**: The first notable observation is that the mean (5.059.124) and the median (5.100.000), which is more robust to outliers, are very close. This suggests that the data is relatively symmetrically distributed and there are not many large outliers—something that could have been expected given the nature of the variable. This conclusion is further supported by the negative kurtosis value (-1.184), indicating that the distribution has lighter tails and fewer extreme outliers, with the data more concentrated around the center. Since the skewness value (-0.013) is slightly negative, we could say that there are more low-income values than high-income values. However, since the skewness coefficient is so small, it does not strongly affect the overall homogeneity of the data.
- **Large variability**: The standard deviation (2.806.840) is quite large, indicating a significant amount of variability in annual income. This means that incomes are spread out across a wide range of values, suggesting potential diversity in the population.Furthermore, the range (9.700.000) is nearly twice the value of the mean and the median. This large range suggests that while the central tendency metrics are close, the dataset still includes a wide spread of income values. Although it is a large range, we must state that the relatively moderate standard deviation and MAD indicate that most of the data is concentrated within a narrower band around the mean.

  Another measure that upholds this large variability is the median absolute deviation (MAD) of 3.558.240, which is more robust to outliers that the deviation and is also a large number.
- **Few outliers**: as we have said before, all the metrics indicate that in this variable there are not many outliers. This is also supported by the “trimmed” mean (5.065.467), since this measure removes extreme values from the calculation of the mean and it is fact very close to the mean and median.

Now let’s analyze the loan amounts:

```r
descr_loan_amount <- loans %>%
    select(loan_amount) %>%
    describe()

# Round the decimals to two digits
descr_loan_amount = descr_loan_amount %>%
    mutate(across(where(is.numeric), round, 2))


pander(descr_loan_amount)
```

Table 14: Table continues below

	vars	n	mean	sd	median	trimmed
loan_amount	1	4269	15133450	9043363	14500000	14742230

	mad	min	max	range	skew	kurtosis	se
loan_amount	10229940	3e+05	39500000	39200000	0.31	-0.75	138410

For this analysis the conclusions are similar to the one before, we´ve got that:

- The **mean** loan amount is 15.133.450 while the median is slightly lower at 14.500.000. The fact that the median is close to the mean suggests that the data might be **relatively symmetrically distributed**, with a few potential outliers not greatly influencing the central tendency. As it happened before, the trimmed mean reinforced this statement, since it is close to both the mean and the median.
- Variability in the loan amount data is indicated by the standard deviation of 9.043.363. This high standard deviation means that there is **substantial variability**, with loans being dispersed over a wide range of values. Additionally, the median absolute deviation (MAD), a more robust measure of variability, is 10.229.940, which confirms that the middle of the data also exhibits considerable spread.
- The range of loan amounts is 39.200.000 with the smallest loan being 300.000 and the largest being 3.500.000. This wide range highlights that while many loans might cluster around the central values, the data includes both very small and very large loan amounts.
- Skewness is 0,31, indicating a slight positive skew, which means that there are a few higher loan amounts that might stretch the distribution to the right. However, the skewness value is still relatively low, suggesting that the data is close to symmetric.
- The kurtosis is -0,745, indicating that the distribution has lighter tails compared to a normal distribution. This suggests that extreme values (very high or very low loan amounts) are less frequent than in a normal distribution. Thus, the data is more concentrated around the mean, with fewer extreme outliers.
- The standard error (SE) of the mean is 138.409,80, which gives an estimate of the accuracy of the mean. This small value relative to the mean indicates that the mean estimate is fairly precise despite the variability in the data.

In this section, I will analyze both total assets and residential assets. We have chosen to focus on residential assets because this variable has distinct characteristics that differ from the others, and the analysis of the results will reflect these differences:

```r
descr_assets <- loans %>%
    select(c(residential_assets_value, total_assets)) %>%
    describe()

# Round the decimals to two digits
descr_assets = descr_assets %>%
    mutate(across(where(is.numeric), round, 2))

pander(descr_assets)
```

Table 16: Table continues below

	vars	n	mean	sd	median
residential_assets_value	1	4269	7472617	6503637	5600000
total_assets	2	4269	32548770	19506563	31500000

Table 17: Table continues below

	trimmed	mad	min	max
residential__assets__value	6630085	6078660	-1e+05	29100000
total__assets	31739069	22832040	4e+05	90700000

	range	skew	kurtosis	se
residential__assets__value	29200000	0.98	0.18	99539
total__assets	90300000	0.3	-0.77	298550

When examining total assets, we observe that it shares similar characteristics with the variables previously analyzed. Therefore, we will concentrate on the residential assets variable:

- The mean is approximately 2.000.000 higher than the median, indicating the presence of outliers. This is further supported by the trimmed mean being lower than the mean, which suggests that extreme outliers on the higher end are influencing the average.
- The kurtosis is positive, meaning that the distribution has heavier tails, indicating more extreme outliers compared to a normal distribution.
- The positive skewness of $0,978$ suggests that the distribution of residential assets is right-skewed, meaning that most values cluster on the lower end, but there are some very high values pulling the distribution to the right.
- The range is large, spanning from -100.000 to 29.100.000, a total range of 29.200.000. This wide spread indicates that the data includes both very low and very high values of residential assets, suggesting a diverse population.
- Also, we have detected that the minimum value appears to be negative (-100.000), which might warrant further investigation, because in real-world scenarios, negative asset values are unusual, and this could indicate a data entry error or a specific scenario where debt exceeds the value of assets.

0.5 Comparison of continuos variables in groups

We aim to find if the probability of not paying back a loan is related to the economic position of the applicant or even the amount of money asked. This section is where we are going to take advantage from the variable created **Cibil score**, since it gives each applicant a category depending on their risk of unpaying debts. In this section we are going to answer some questions about **fraud tendencies**, which are the ones that intrigue us, and maybe the reader, the most.

Let's the variable Cibil Score in general:

```
cibil_score_data = loans %>%
    group_by(cibil_score_char) %>%
    summarise(n = n(), percentage = (n()/nrows * 100))

pander(cibil_score_data)
```

cibil__score__char	n	percentage
Excellent credit risk	1056	24.74
Fair credit risk	683	16

cibil_score_char	n	percentage
Good credit risk	745	17.45
Poor credit risk	1785	41.81

From this table we can conclude that almost half of the population have a poor credit risk, meaning that in this sample of applicants, most of them have higher credit risk and they are more likely to default on loans or delay payments, thus requiring more scrutiny or stringent terms in financial transactions. On the contrary, only around 25% of the sample are considered to have an excellent credit risk, which might be an advantage for their loans to be approved.

Now we are going to study the distribution of the other variables based on these categories:

0.5.1 Do applicants who have more assets or a higher income don't pay their debts?

Let's begin first by studying how is the annual income distributed in the different categories:

```
annual_inc_score = loans %>%
    group_by(cibil_score_char) %>%
    summarise(n = n(), percentage = n()/nrows * 100, mean = mean(income_annum),
        median = median(income_annum), sd = sd(income_annum),
        cv = sd(income_annum)/mean(income_annum), q25 = quantile(income_annum,
            0.25), q75 = quantile(income_annum, 0.75), min = min(income_annum),
        max = max(income_annum), iqr = IQR(income_annum)) %>%
    mutate(across(where(is.numeric), round, 2))


annual_inc_score = annual_inc_score %>%
    mutate(percentage = paste0(percentage, "%"))

pander(annual_inc_score)
```

Table 20: Table continues below

cibil_score_char	n	percentage	mean	median	sd	cv
Excellent credit risk	1056	24.74%	4897633	4800000	2838838	0.58
Fair credit risk	683	16%	5036750	5e+06	2840188	0.56
Good credit risk	745	17.45%	5177987	5300000	2788807	0.54
Poor credit risk	1785	41.81%	5113613	5100000	2780352	0.54

q25	q75	min	max	iqr
2400000	7300000	2e+05	9900000	4900000
2500000	7500000	2e+05	9900000	5e+06
2700000	7600000	2e+05	9900000	4900000
2800000	7500000	2e+05	9900000	4700000

From this data we can state, unexpectedly (or not), that the applicants that have a Excellent credit risk tends to have a lower annual income compare to the other categories. This is supported by both the mean and the median. Apart from that, the quantiles confirm that this tendency is present in most of the sample but that the different categories have applicants from all the salary ranges. This last point is also implicit in the IQR column, where we can conclude that there is considerate range on annual incomes in each category.

We will study this in more detail in the section of the graphics.

It also happens the same in the total assets table:

```r
total_assets_score = loans %>%
    group_by(cibil_score_char) %>%
    summarise(n = n(), percentage = n()/nrows * 100, mean = mean(total_assets),
        median = median(total_assets), sd = sd(total_assets),
        cv = sd(total_assets)/mean(total_assets), q25 = quantile(total_assets,
            0.25), q75 = quantile(total_assets, 0.75), min = min(total_assets),
        max = max(total_assets), iqr = IQR(total_assets)) %>%
    mutate(across(where(is.numeric), round, 2))


total_assets_score = total_assets_score %>%
    mutate(percentage = paste0(percentage, "%"))

pander(total_assets_score)
```

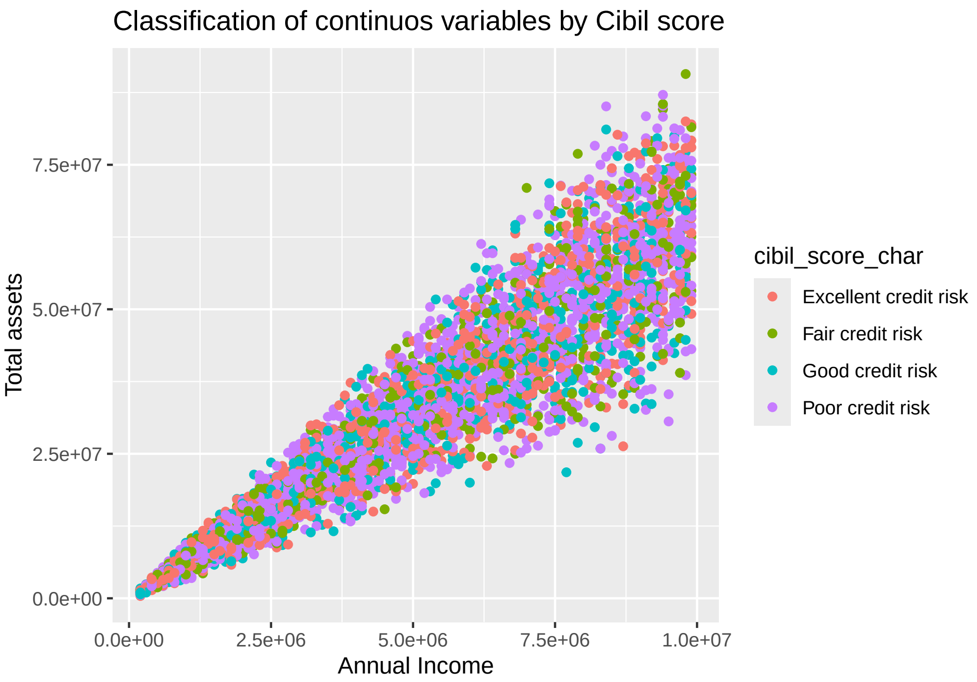
Table 22: Table continues below

cibil_score_char	n	percentage	mean	median	sd	cv	q25	q75	min	max	iqr
Excellent credit risk	1056	24.74%	31635133	29550000	19907127	0.63	1.5e+07	46325000	4e+05	82500000	31325000
Fair credit risk	683	16%	32304100	31300000	19563242	0.61	15700000	45600000	6e+05	90700000	29900000
Good credit risk	745	17.45%	33029530	33200000	19075107	0.58	16500000	47900000	7e+05	81100000	31400000
Poor credit risk	1785	41.81%	32982241	32200000	19419155	0.59	17200000	47900000	5e+05	87100000	30700000

From this data we would make the same conclussions as the prior table.

Something that may be of interest is to compare the scatter plot made before to study the total assets based on the salary divided by groups:

```r
plot = ggplot(loans, aes(x = income_annum, y = total_assets,
    color = cibil_score_char))
plot + geom_point() + labs(title = "Classification of continuos variables by Cibil score",
    x = "Annual Income", y = "Total assets")
```



The data clearly shows that it doesn't exist a trend in colors on the plot, indicating that the risk of loans going unpaid is consistent across different ranges of annual income and total assets.

0.5.2 Are defaulters also the ones that ask for a higher money loan?

```
loan_amount_score=
  loans %>%
  group_by(cibil_score_char) %>%
  summarise(n = n(),
            percentage = n() / nrows * 100,
            mean = mean(loan_amount),
            median=median(loan_amount),
            sd = sd(loan_amount),
            cv = sd(loan_amount) / mean(loan_amount),
            q25= quantile(loan_amount,0.25),
            q75 = quantile(loan_amount,0.75),
            min = min(loan_amount),
            max = max(loan_amount),
            iqr = IQR(loan_amount))%>%
            mutate(across(where(is.numeric), round, 2))


loan_amount_score = loan_amount_score %>%
  mutate(percentage = paste0(percentage, "%"))


pander(loan_amount_score)
```

Table 24: Table continues below

cibil_score_char	n	percentage	mean	median	sd	cv	q25	q75	min	max	iqr
Excellent credit risk	1056	24.74%	14655871	13900000	9145162	0.62	7e+06	20900000	4e+05	39500000	13900000
Fair credit risk	683	16%	15294729	14500000	9194741	0.6	7750000	22300000	4e+05	38700000	14550000
Good credit risk	745	17.45%	15454094	14800000	9061278	0.59	8200000	21900000	3e+05	38800000	13700000
Poor credit risk	1785	41.81%	15220448	14800000	8912845	0.59	8e+06	21300000	3e+05	38200000	13300000

Although there is a tendency of asking a lower loan from the applicants who have an excellent credit risk, from this last table we conclude, disappointingly, that there is a lot of variety on the loan amounts in the different categories of cibil score. We are going to do further analysis to conclude so.

In other words, we would like to prove that there is no significant difference in the loan amount between different cibil scores. The first approach would be to make an ANOVA analysis, but since our data has a heavy right tail, it probably does not follow a normal distribution (this will be analyzed in detail in the graphics section). So our second alternative is to compute an **Kruskal-Wallis test**.

0.5.2.1 Kruskal-Wallis test

This is a non-parametric statistical test used to determine if there are statistically significant differences between the medians of three or more independent groups. The test ranks all the data points from all groups together and compares the sum of ranks between groups to assess whether they come from the same distribution.

The for this test the hypothesis are:

- **Null Hypothesis** (H_0): The distributions of all groups are identical, implying that the population medians are equal across groups.
- **Alternative Hypothesis** (H_1): At least one group has a different distribution, indicating that the population medians differ across the groups.

The code is the next one:

```r
# Kruskal-Wallis test for loan amount by CIBIL score
kruskal_test_loan <- kruskal.test(loan_amount ~ cibil_score_char, data = loans)

# View the result
kruskal_test_loan
```

```

    Kruskal-Wallis rank sum test

data:  loan_amount by cibil_score_char
Kruskal-Wallis chi-squared = 4.6264, df = 3, p-value = 0.2013
```

Since the p-value is 0,2013, we cannot reject the null hypothesis. This means that there is no statistically significant difference in the distributions of loan amounts across the different CIBIL score categories. In other words, based on this test, there is no evidence to suggest that the median loan amounts vary significantly between the CIBIL score categories.

As we have expected, the test supports our conclusions. To make this analysis more complete we are going to use some graphics in 0.7.1 Histograms and density plots.

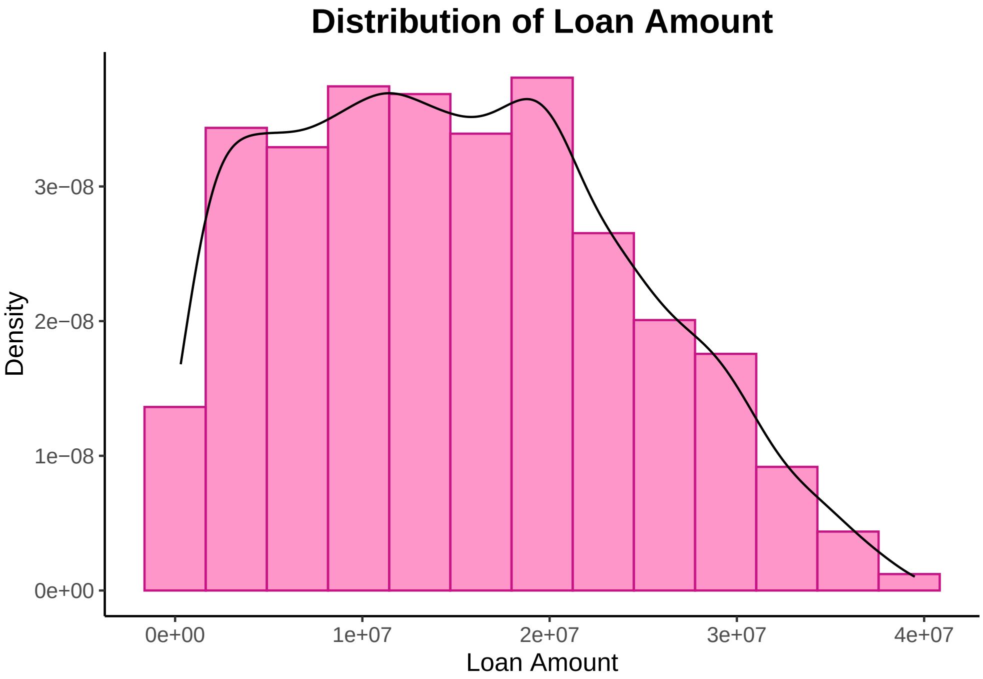
0.6 Plots.

For this section, we need to use the package **ggplot2** to make the graphs.

0.6.1 Histograms and density plots.

```
ggplot(loans, aes(loan_amount)) +
  geom_histogram(aes(y = after_stat(density)),
                 color = "mediumvioletred", fill = "hotpink", bins = 13, alpha = 0.7) +
  geom_density(color = "black") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "Distribution of Loan Amount",
       x = "Loan Amount",
       y = "Density")
```

0.6.1.1 Loan amount.

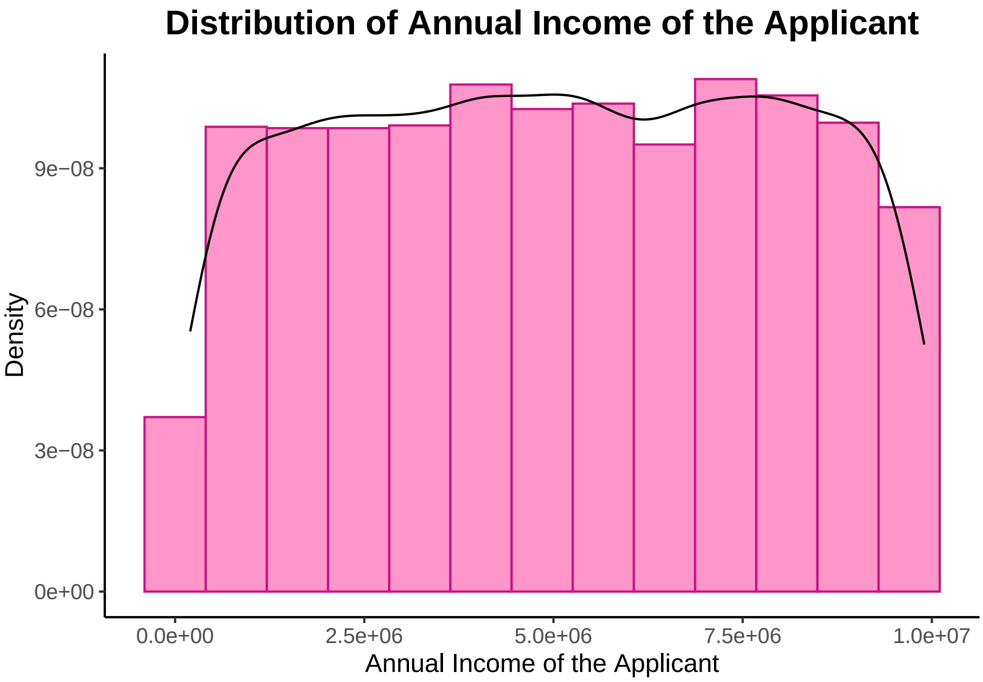


This graph visualizes the loan amount distribution, combining both discrete (histogram) and continuous (density plot) representations to give a clearer idea of the spread and peaks in the data. The histogram gives a view of the frequency distribution of loan amounts, while the density plot provides a smoothed version, giving an idea of the underlying distribution.

The graph reflects loan distribution in India, showing the varying loan amounts provided. The majority of loans in India tend to be small to mid-sized, aimed at individuals and small businesses, reflected in the concentration of loans in the lower to mid-range amounts in the graph. Larger loans, which represent a smaller proportion of the distribution, are typically availed by big corporations or high-value projects. The distribution aligns with the credit structure and economic goals of India, particularly in promoting financial inclusion and supporting economic growth.

```
ggplot(loans, aes(income_annum)) + geom_histogram(aes(y = after_stat(density)),
    color = "mediumvioletred", fill = "hotpink",
    bins = 13, alpha = 0.7) + geom_density(color = "black") +
    theme_classic() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) +
    labs(title = "Distribution of Annual Income of the Applicant",
        x = "Annual Income of the Applicant",
        y = "Density")
```

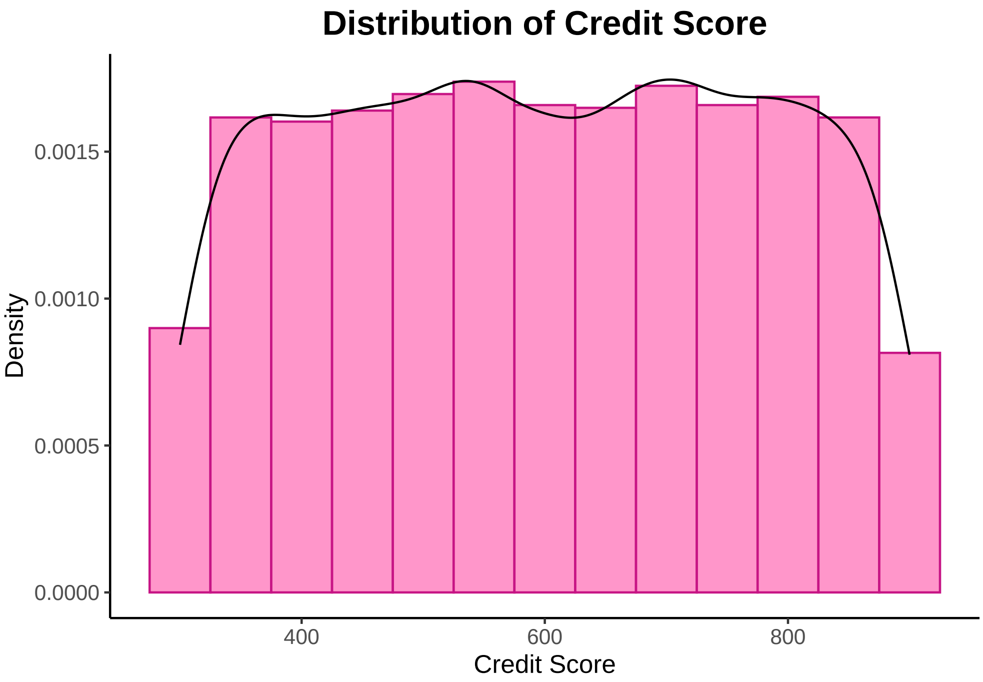
0.6.1.2 Annual Income of the Applicant.



This graph, as the last one, gives a view of the frequency distribution of the annual income of the applicant, in this case. This graph illustrates that the sample of applicants is composed of individuals with relatively high incomes, and the distribution is very uniform across this income range in the Indian context. In India, the income distribution and wealth are shaped by the country's large population, economic diversity, and income inequality. India has a wide income gap; a small percentage of the population holds a large portion of the nation's wealth, while a vast majority have relatively low incomes.

```
ggplot(loans, aes(cibil_score)) + geom_histogram(aes(y = after_stat(density)),
    color = "mediumvioletred", fill = "hotpink",
    bins = 13, alpha = 0.7) + geom_density(color = "black") +
    theme_classic() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) +
    labs(title = "Distribution of Credit Score",
        x = "Credit Score", y = "Density")
```

0.6.1.3 Credit Score.



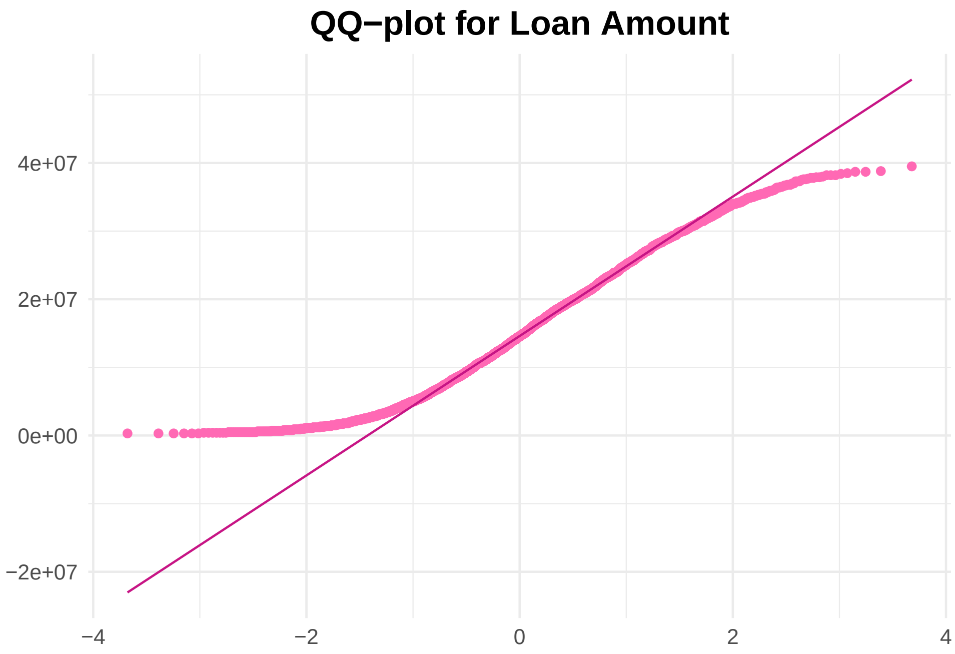
This graph represents the distribution of credit scores among applicants. The frequency distribution of credit scores range from 300 to 900, which is typical in India and many other countries where credit score systems like CIBIL or similar models are used. The heights of the bars suggest that credit scores are distributed across all ranges quite evenly, without major concentration in any specific range. In India, credit scores are typically assessed by agencies like CIBIL (Credit Information Bureau India Limited). A credit score of 750 and above is often considered excellent, making individuals eligible for loans with favorable terms. Scores in the 600-750 range are still decent but may result in stricter loan conditions or slightly higher interest rates. Scores below 600 could be considered risky, and individuals might face challenges in securing credit. The density curve suggests that the distribution is relatively uniform, although there is a slight peak in the middle, around the 600 to 700 range.

0.6.2 QQ-plots.

To contrast the normality of these distributions, we will examine the normal QQ-plots.

```r
ggplot(loans, aes(sample = loan_amount)) + stat_qq(color = "hotpink") +
    stat_qq_line(color = "mediumvioletred") + theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12), axis.text = element_text(size = 10)) +
    labs(title = "QQ-plot for Loan Amount", x = "", y = "")
```

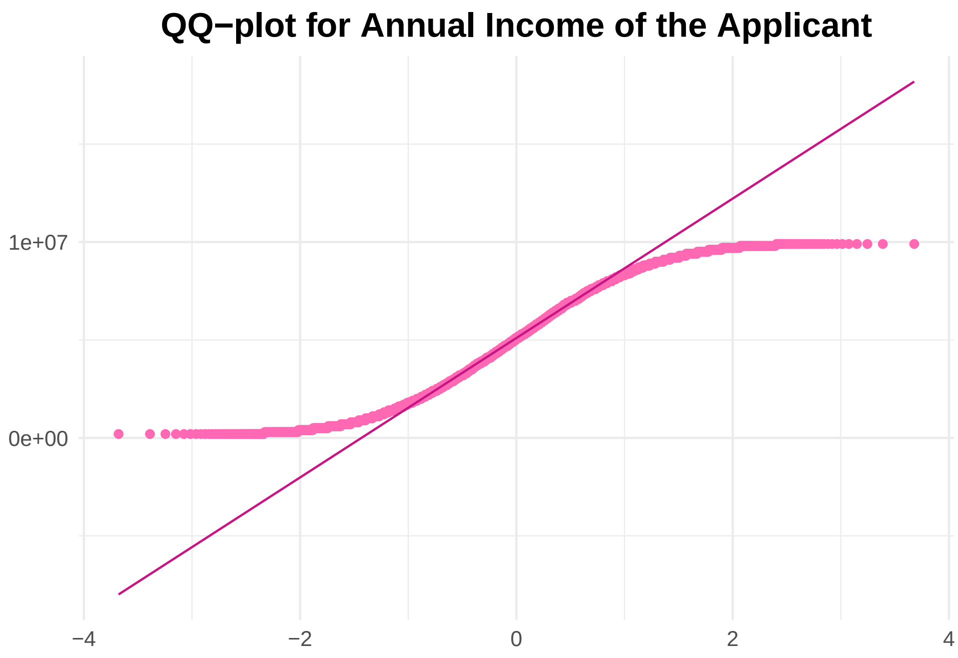
0.6.2.1 QQ-plot for Loan Amount.



This QQ-plot suggests that the loan amounts are not normally distributed. Instead, they likely have a positive tendency, meaning there may be many smaller loan amounts and a few very large ones. This is common in loan data sets, where the majority of applicants request moderate loans, while a small portion seeks much larger amounts. This pattern may be seen in cases where some individuals or businesses apply for very large loans, while most others apply for smaller or more typical amounts.

```r
ggplot(loans, aes(sample = income_annum)) + stat_qq(color = "hotpink") +
    stat_qq_line(color = "mediumvioletred") + theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12), axis.text = element_text(size = 10)) +
    labs(title = "QQ-plot for Annual Income of the Applicant",
        x = "", y = "")
```

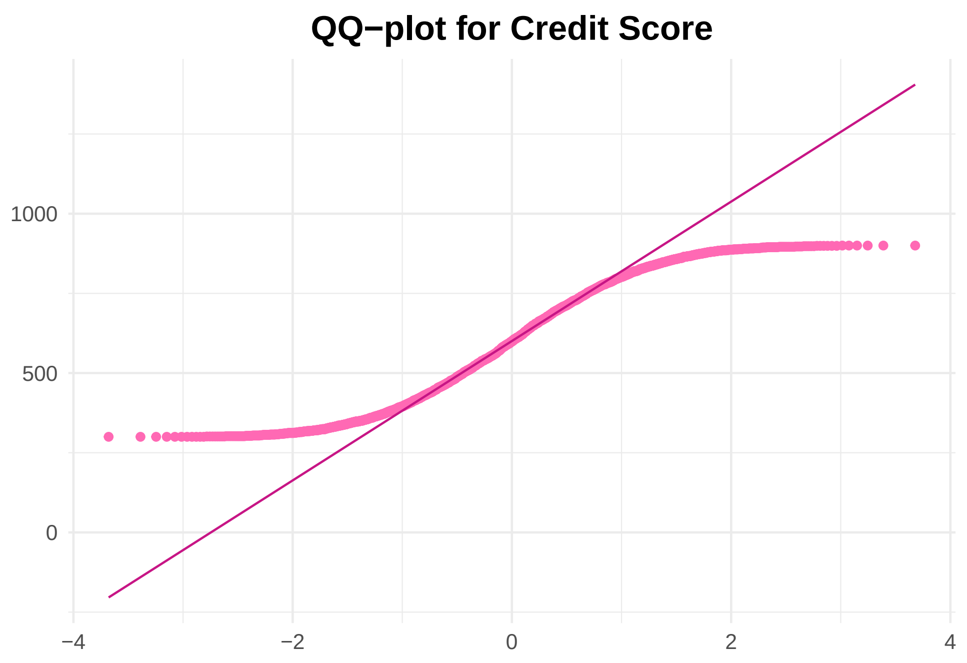
0.6.2.2 QQ-plot for Annual Income of the Applicant.



This plot shows the distribution of incomes among loan applicants in India and suggests that traditional statistical models might not adequately capture the income variations present in the country, which could impact loan approval and risk models.

```r
ggplot(loans, aes(sample = cibil_score)) + stat_qq(color = "hotpink") +
    stat_qq_line(color = "mediumvioletred") + theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12), axis.text = element_text(size = 10)) +
    labs(title = "QQ-plot for Credit Score", x = "", y = "")
```

0.6.2.3 QQ-plot for Credit Score.



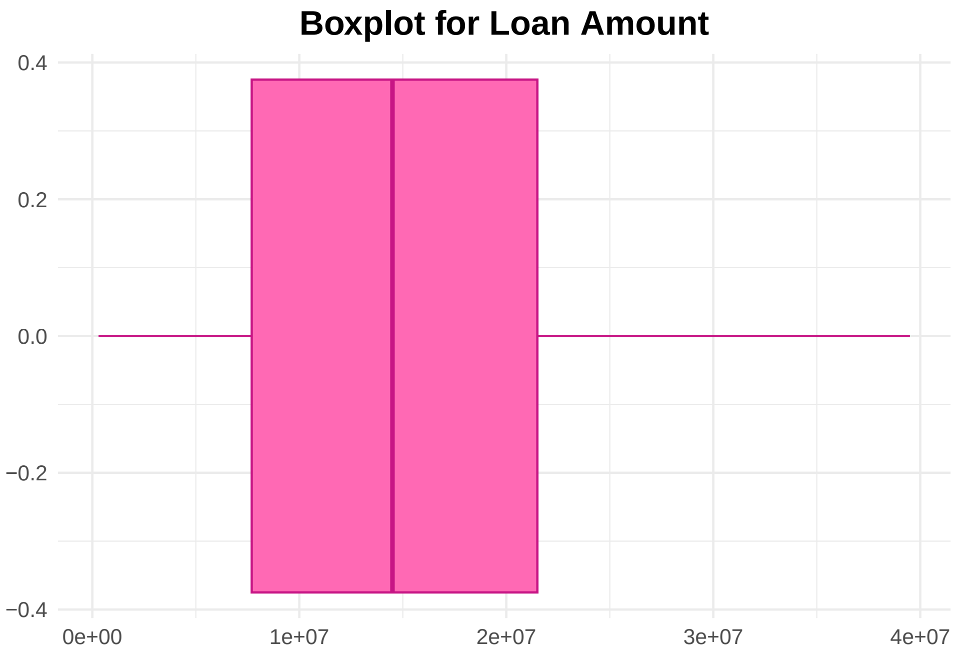
Following the arguments on the last two graphs, the distribution of the credit score neither follows a normal distribution.

0.6.3 Box plots.

We are going to use the box plots to show the centrality measures and the range of the data values.

```
ggplot(loans, aes(loan_amount)) + geom_boxplot(fill = "hotpink",
    color = "mediumvioletred") + theme_minimal() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) + labs(title = "Boxplot for Loan Amount",
    x = "", y = "")
```

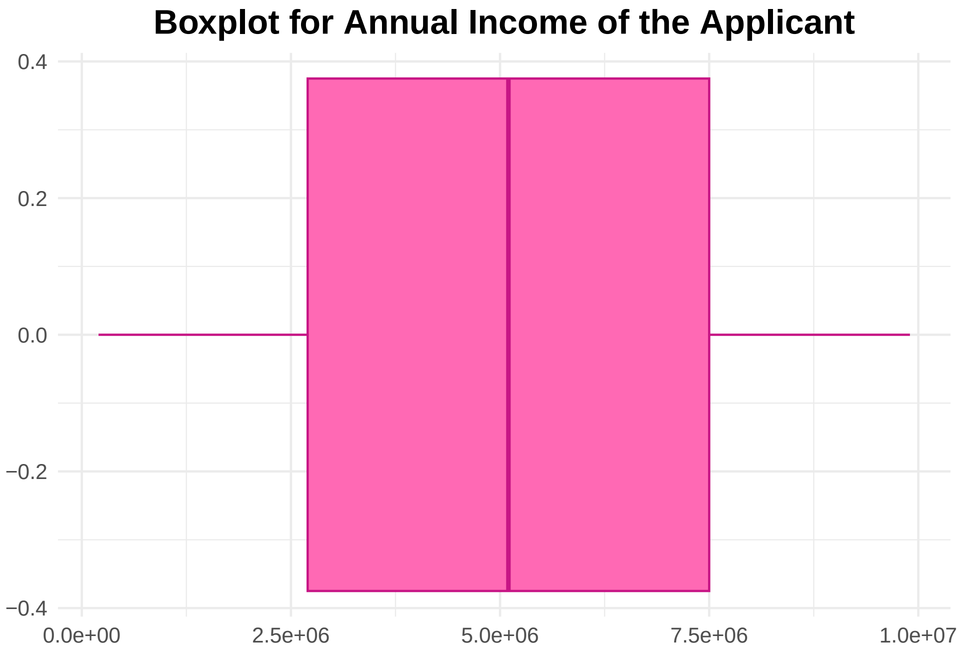
0.6.3.1 Box plot for Loan Amount.



This box plot visualizes the distribution of loan amounts for applicants. The box plot shows a concentrated and symmetric distribution of loan amounts; suggesting that most applicants fall within this loan bracket. This support the distribution seen in the histogram.

```r
ggplot(loans, aes(income_annum)) + geom_boxplot(fill = "hotpink",
    color = "mediumvioletred") + theme_minimal() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) + labs(title = "Boxplot for Annual Income of the Applicant",
    x = "", y = "")
```

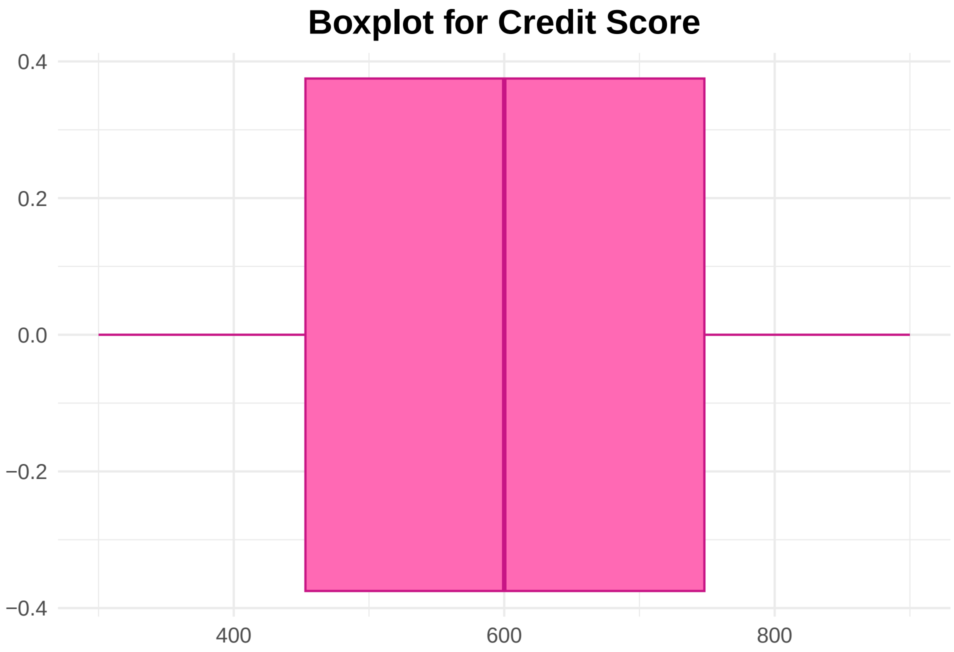
0.6.3.2 Box plot for Annual Income of the Applicant.



This box plot also shows a concentrated and symmetric distribution of the Annual Income of the Applicant. As we said before, in India a very small percentage of the population owns the majority of the national wealth. This is a consequence of the social class stratification. Let's delve a little deeper into the society of this country: in Hinduism, the term Varna refers to a social class within the hierarchical structure of traditional Hindu society. This system is outlined in texts like the Manusmriti, which categorizes and ranks four main varnas and specifies their occupations, duties, and moral obligations, or Dharma. The main varnas are: Brahmins (scholars, priests, or teachers), Kshatriyas: (rulers, administrators, or warriors), Vaishyas (farmers, merchants, or agriculturalists) and Shudras (artisans, laborers, or servants). This makes less probables that Vaishyas and Shudras ask for a loan. This fourfold classification represents a form of social stratification distinct from the more complex system of Jatis, often referred to as "caste" in European terminology.

```
ggplot(loans, aes(cibil_score)) + geom_boxplot(fill = "hotpink",
    color = "mediumvioletred") + theme_minimal() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) + labs(title = "Boxplot for Credit Score",
    x = "", y = "")
```

0.6.3.3 Box plot for Credit Score.

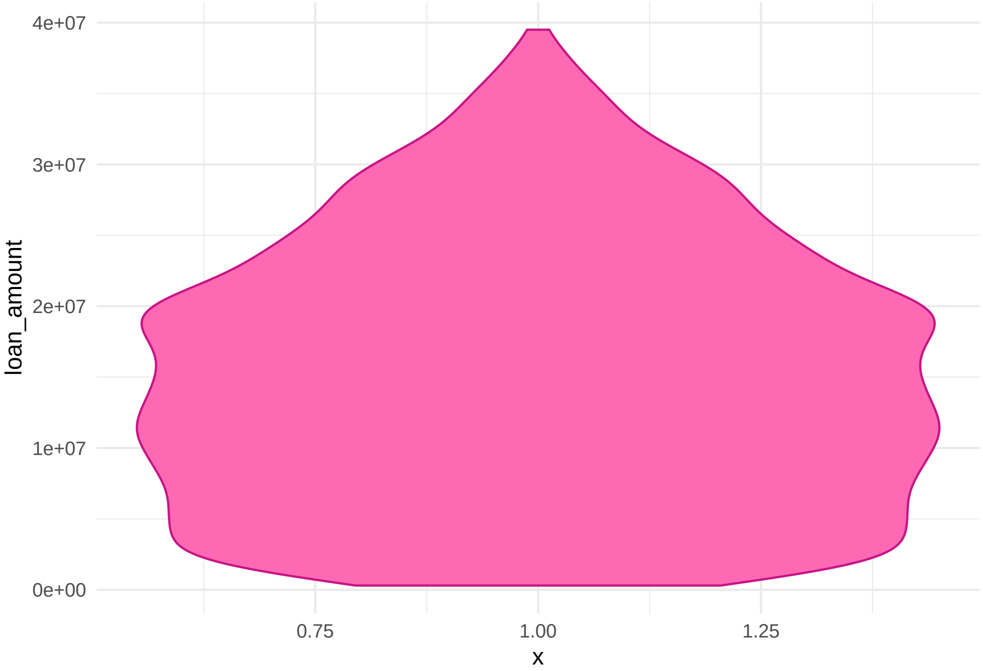


In this third box plot, we see something similar to the other ones, a symmetric and concentrated distribution.

0.6.4 Violin plots.

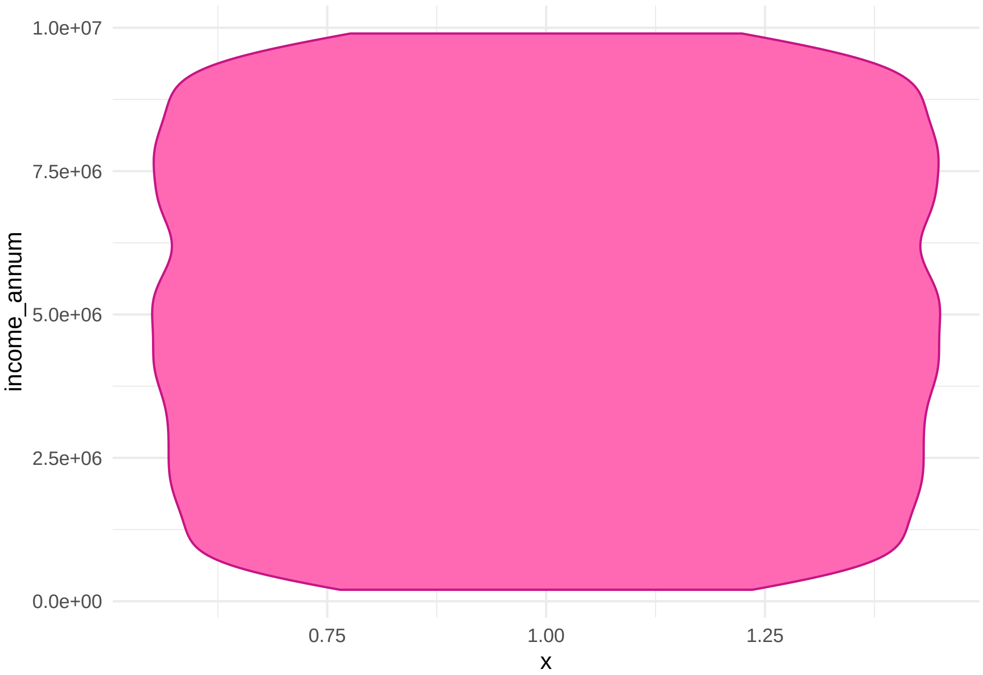
A violin plot is a combination of a density and a box plot, so we can see the distribution and the probability density of our chosen variables.

```
ggplot(loans, aes(x = 1, y = loan_amount))+
  geom_violin(color = "mediumvioletred", fill = "hotpink") + theme_minimal()
```



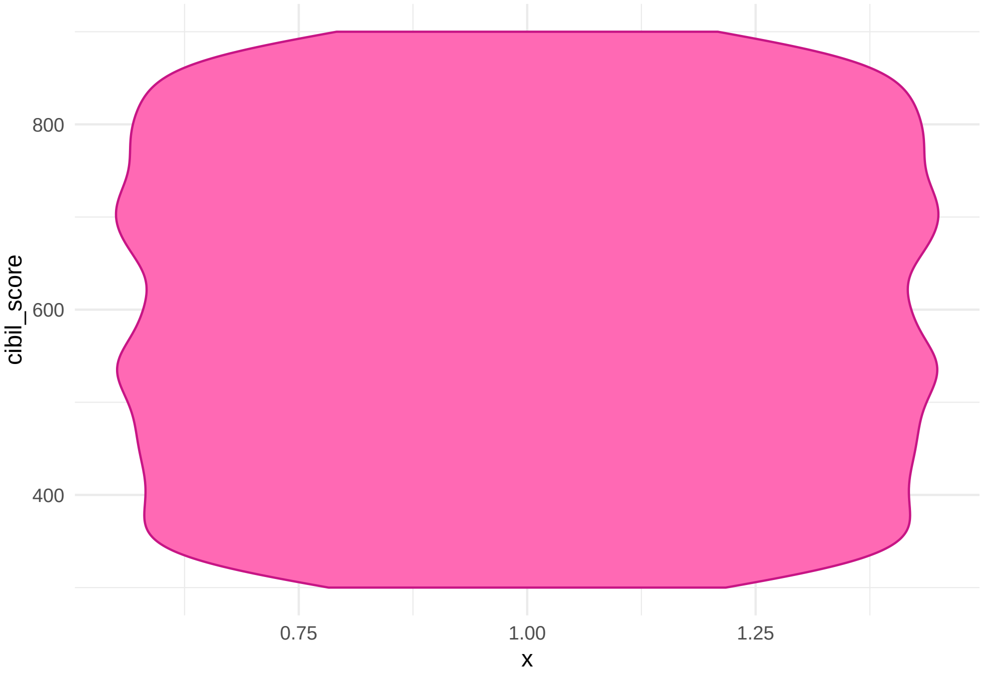
This violin plot shows that the most common loan amounts are concentrated around 20 million, with fewer requests for amounts significantly higher or lower than this. This pattern could inform lenders about the typical loan demands in the Indian market, focusing on midsized loans for either personal or business purposes.

```
ggplot(loans, aes(x = 1, y = income_annum))+
  geom_violin(color = "mediumvioletred", fill = "hotpink") + theme_minimal()
```



This violin plot shows that Annual Income of the Applicant of a loan in India have a wide distribution, which could mean that the economy of the applicants are not similar.

```
ggplot(loans, aes(x = 1, y = cibil_score))+
  geom_violin(color = "mediumvioletred", fill = "hotpink") + theme_minimal()
```

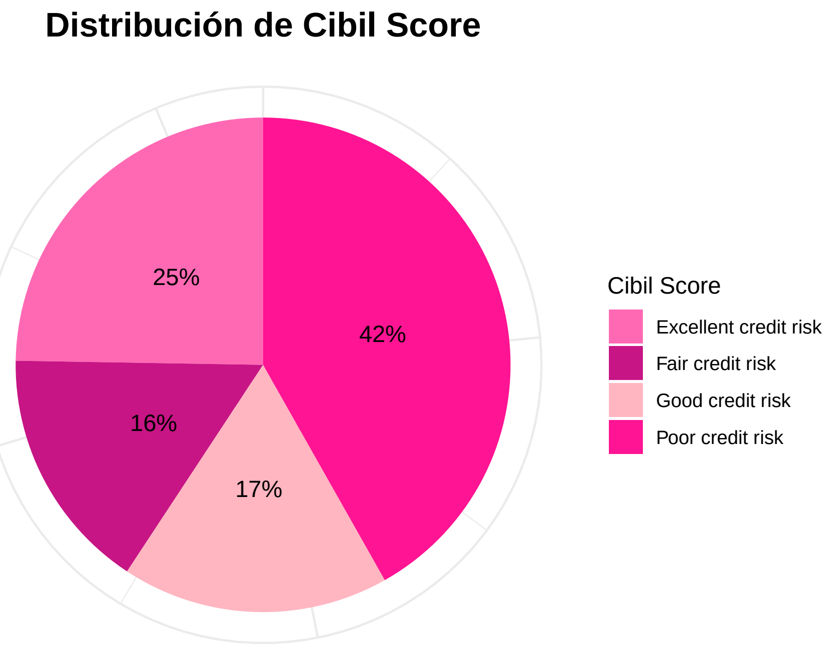


This plot shows that the Credit score is very diverse in India.

0.7 Plots for grouped data.

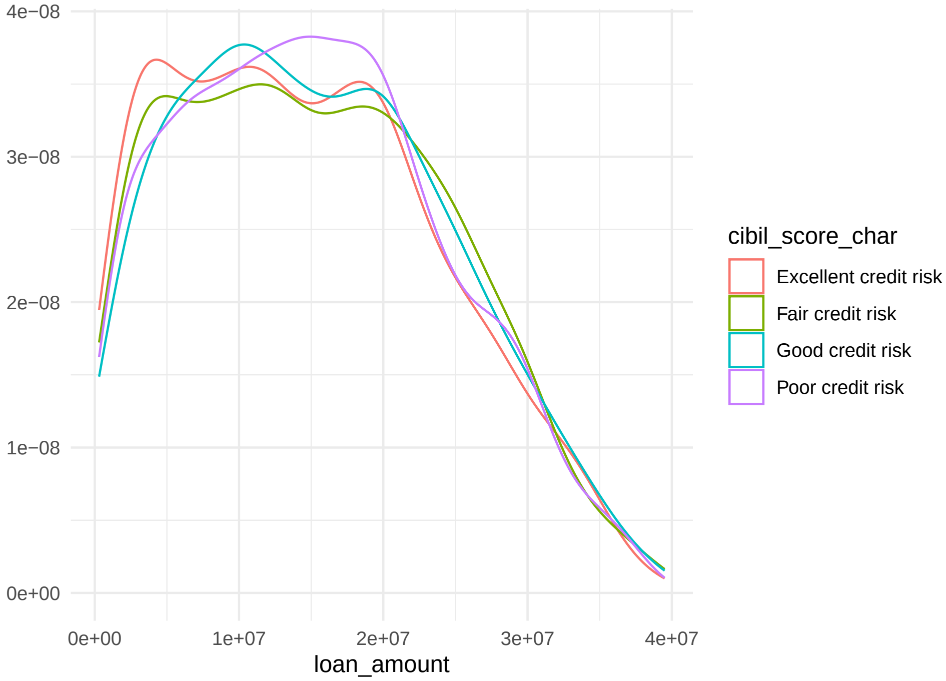
```
library(scales)

ggplot(loans, aes(x = "", fill = cibil_score_char)) +
  geom_bar(width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Distribución de Cibil Score",
       fill = "Cibil Score",
       x = NULL,
       y = NULL) +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        legend.position = "right") +
  scale_fill_manual(values = c("hotpink", "mediumvioletred", "lightpink", "deeppink")) +
  geom_text(aes(label = scales::percent(..count../sum(..count..), accuracy = 1)),
            stat = "count", position = position_stack(vjust = 0.5))
```



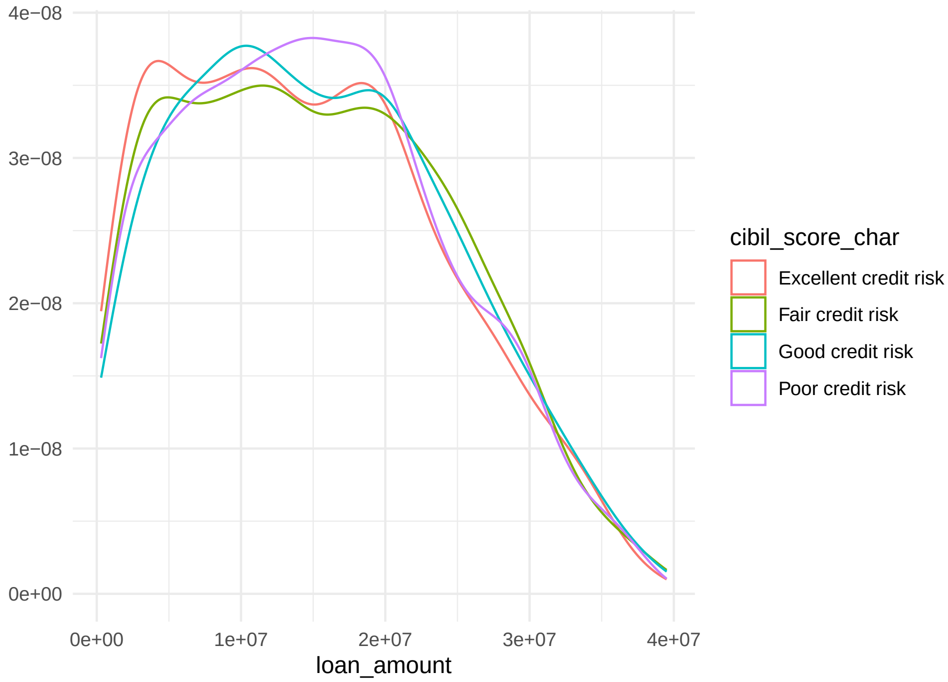
0.7.1 Histograms and density plots.

```
qplot(loan_amount, data = loans, geom = "density", color = cibil_score_char) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
```



In this graph we can see that there is not difference in the density of the loan amount between the different credit score values. This also happens for the annual income of the applicant, as we show below.

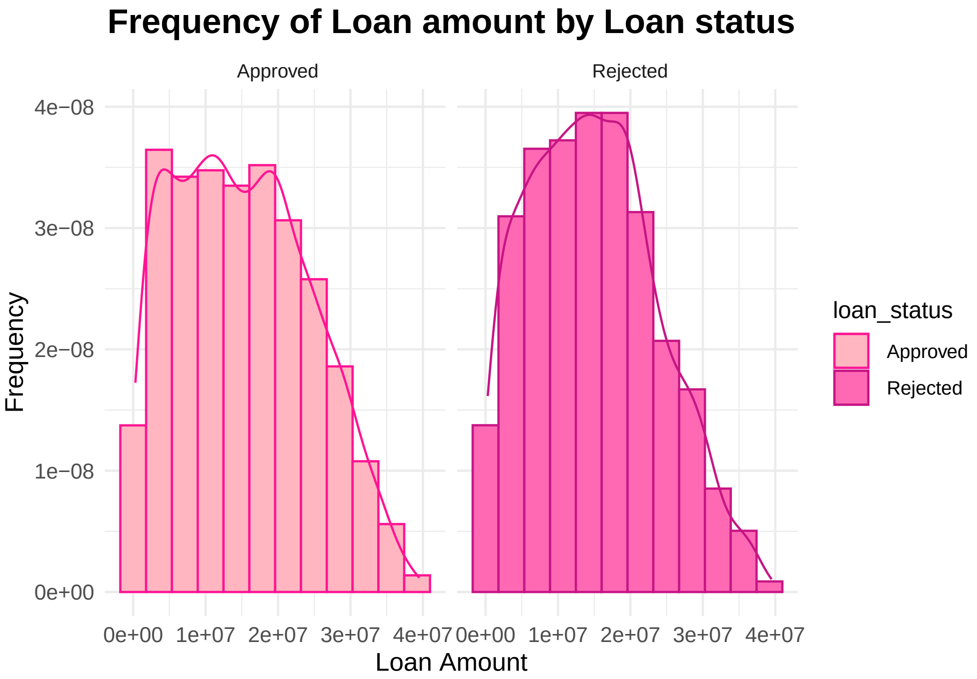
```
qplot(loan_amount, data = loans, geom = "density", color = cibil_score_char) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
```



From now on we are going to make these analysis using the categorical variable: loan status Although it is not the same as in section 0.5 Comparison of continuous variables, the civil_score_char did not give us much to analyze since it doesn't seem to affect the distribution of our continuous variables (the plots were also not too attractive); that is why we have chosen the columnLoan Status to analyze the following plots:

```
ggplot(loans, aes(loan_amount)) +
  geom_histogram(aes(y = ..density.., fill = loan_status, color = loan_status), bins = 12) +
  geom_density(aes(color = loan_status)) +
  scale_fill_manual(values = c("lightpink", "hotpink")) +
  scale_color_manual(values = c("deeppink", "mediumvioletred")) +
  facet_wrap(~ loan_status) +
  theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "Frequency of Loan amount by Loan status", x = "Loan Amount", y = "Frequency")
```

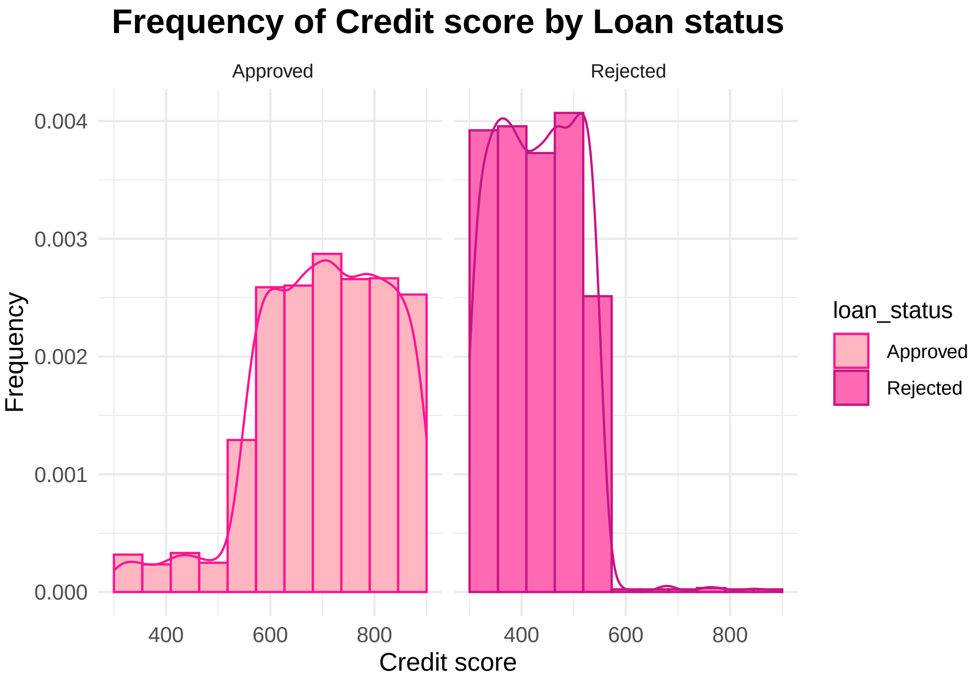
0.7.1.1 Frequency of Loan amount by Loan status.



The graph presents two density plots, one for approved loans and one for rejected loans, illustrating the distribution of loan amounts within each category. Both approved and rejected loans appear to fall within a similar range, with the majority of loans concentrated between 0 and 2 million. There is some overlap between the two distributions, suggesting that loan amounts alone may not be a strong predictor of loan approval or rejection.

```
ggplot(loans, aes(cibil_score)) +
  geom_histogram(aes(y = ..density.., fill = loan_status, color = loan_status), bins = 12) +
  geom_density(aes(color = loan_status)) +
  scale_fill_manual(values = c("lightpink", "hotpink")) +
  scale_color_manual(values = c("deeppink", "mediumvioletred")) +
  facet_wrap(~ loan_status) +
  theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "Frequency of Credit score by Loan status", x = "Credit score", y = "Frequency")
```

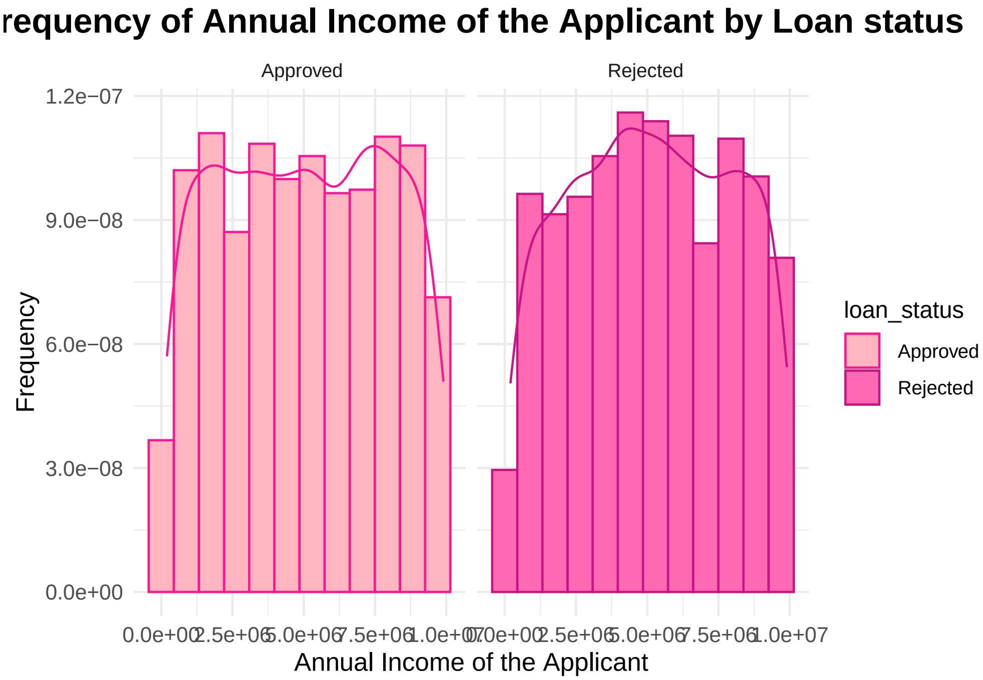
0.7.1.2 Frequency of Credit score by Loan status.



In this graph, while the overall shapes are similar, there are differences in the density of loans within certain credit score ranges. For instance, there appears to be a higher density of approved loans around 600-700, while rejected loans might have a slightly higher density around 400-500. Credit score is likely a significant factor in loan approval decisions, but it’s not the sole determinant. Other factors, such as income, employment history, and debt-to-income ratio, are likely considered as well.

```
ggplot(loans, aes(income_annum)) + geom_histogram(aes(y = ..density..,
    fill = loan_status, color = loan_status), bins = 12) +
    geom_density(aes(color = loan_status)) + scale_fill_manual(values = c("lightpink",
    "hotpink")) + scale_color_manual(values = c("deeppink",
    "mediumvioletred")) + facet_wrap(~loan_status) +
    theme_minimal() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) + labs(title = "Frequency of Annual Income of the Applicant by Loan status
    x = "Annual Income of the Applicant", y = "Frequency")
```

0.7.1.3 Frequency of Annual Income of the Applicant by Loan status.



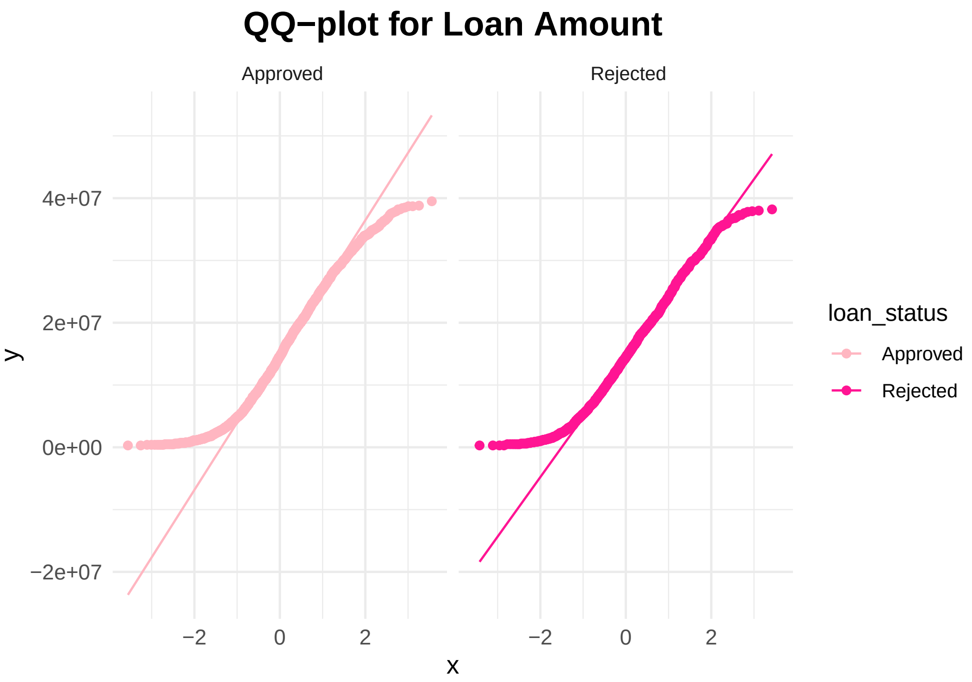
The graph presents two density distributions, one for approved loans and one for rejected loans, illustrating the distribution of annual incomes within each category. Both approved and rejected loans appear to fall within a similar range, with the majority of incomes concentrated between 0 and 10 million. While the overall shapes are similar, there are differences in the density of loans within certain annual income ranges.

0.7.2 QQ Plots.

This qq plots shows that non of this variables follows a normal distribution.

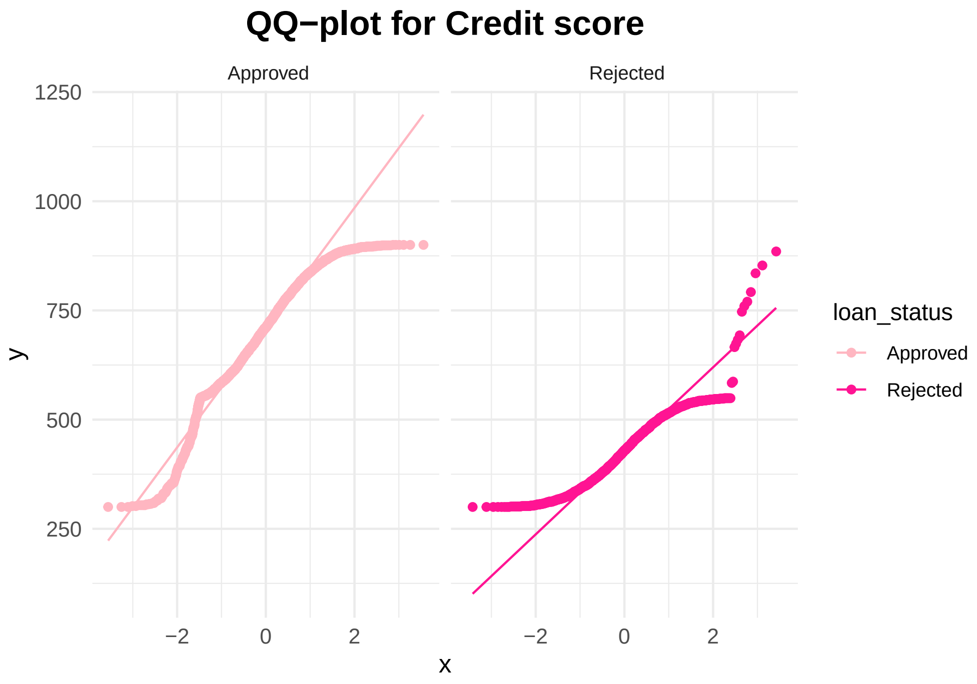
```
ggplot(loans, aes(sample = loan_amount)) +
  stat_qq(aes(color = loan_status)) +
  stat_qq_line(aes(color = loan_status)) +
  facet_wrap(~ loan_status) +
  theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "QQ-plot for Loan Amount") +
  scale_color_manual(values = c("lightpink", "deeppink"))
```

0.7.2.1 QQ-plot for Loan Amount.



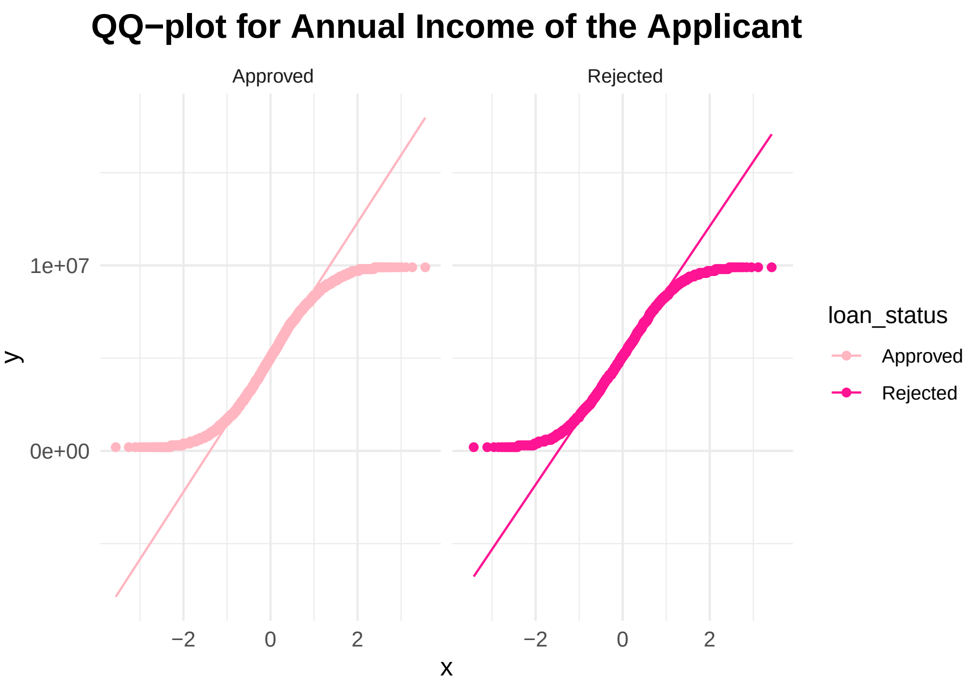
```
ggplot(loans, aes(sample = cibil_score)) +
  stat_qq(aes(color = loan_status)) +
  stat_qq_line(aes(color = loan_status)) +
  facet_wrap(~ loan_status) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "QQ-plot for Credit score") +
  scale_color_manual(values = c("lightpink", "deeppink"))
```

0.7.2.2 QQ-plot for Credit score.



```
ggplot(loans, aes(sample = income_annum)) +
  stat_qq(aes(color = loan_status)) +
  stat_qq_line(aes(color = loan_status)) +
  facet_wrap(~ loan_status) +
  theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "QQ-plot for Annual Income of the Applicant") +
  scale_color_manual(values = c("lightpink", "deeppink"))
```

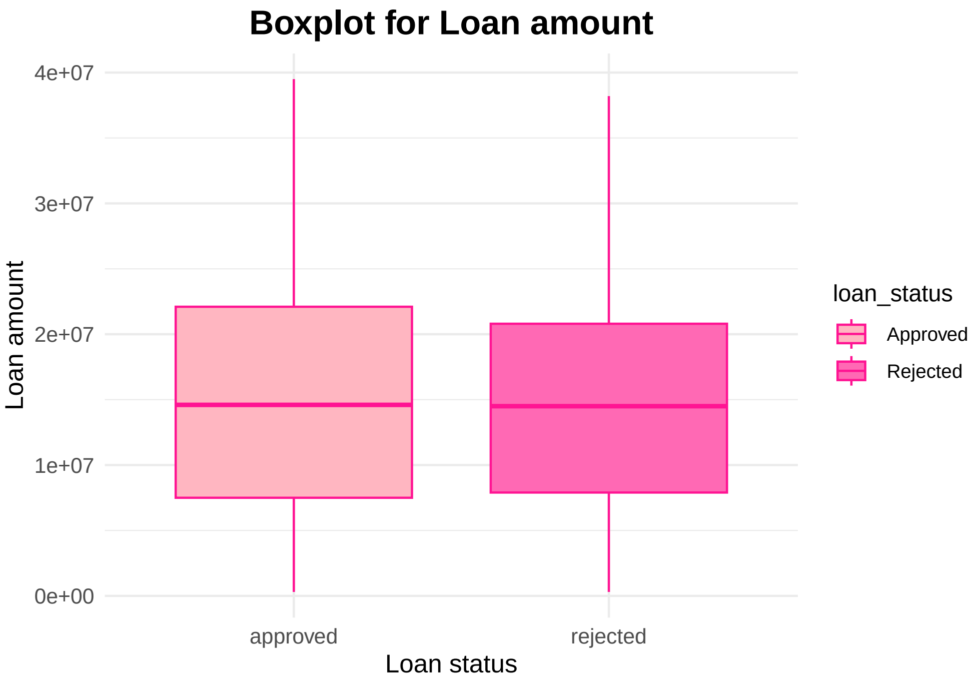
0.7.2.3 QQ-plot for Annual Income of the Applicant.



0.7.3 Box plots.

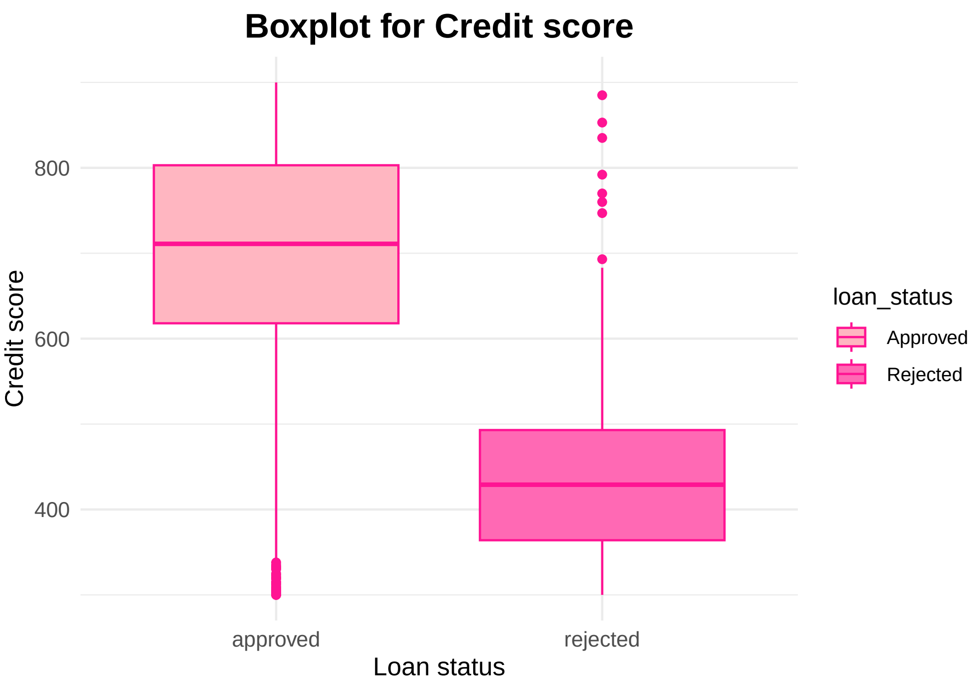
```
ggplot(loans, aes(x = loan_status, y = loan_amount)) +
  geom_boxplot(aes(fill = loan_status), color = "deeppink") +
  scale_fill_manual(values = c("lightpink", "hotpink")) +
  theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "Boxplot for Loan amount", x = "Loan status", y = "Loan amount") +
  scale_x_discrete(labels = c("approved", "rejected"))
```

0.7.3.1 Box plot for Loan amount.



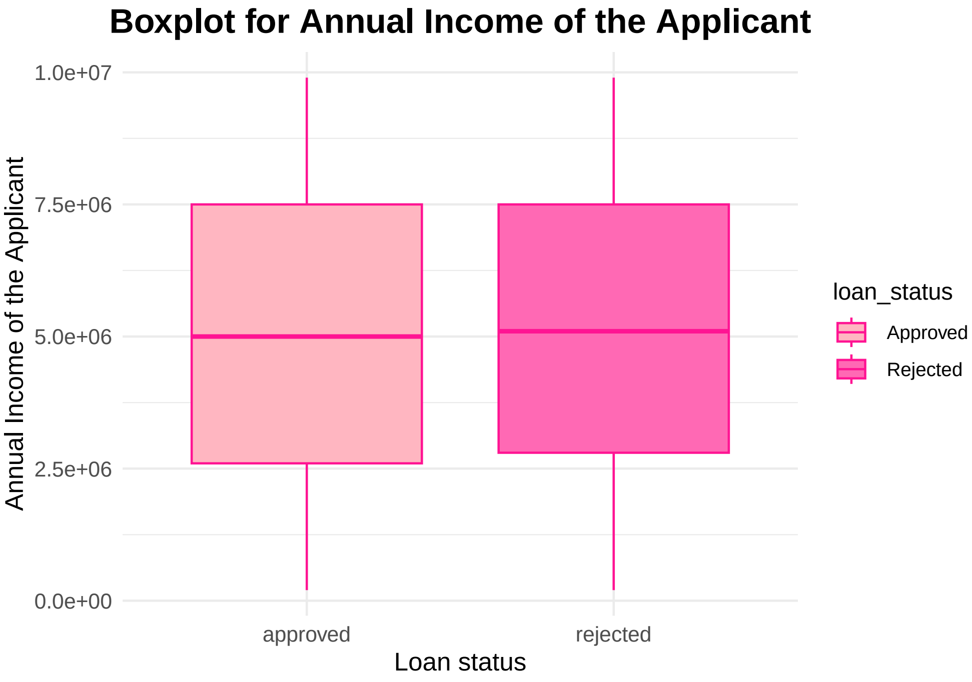
```
ggplot(loans, aes(x = loan_status, y = cibil_score)) +
  geom_boxplot(aes(fill = loan_status), color = "deeppink") +
  scale_fill_manual(values = c("lightpink", "hotpink")) +
  theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = "Boxplot for Credit score", x = "Loan status", y = "Credit score")+
  scale_x_discrete(labels = c("approved", "rejected"))
```

0.7.3.2 Box plot for Credit score.



```r
ggplot(loans, aes(x = loan_status, y = income_annum)) +
    geom_boxplot(aes(fill = loan_status),
        color = "deeppink") + scale_fill_manual(values = c("lightpink",
    "hotpink")) + theme_minimal() + theme(plot.title = element_text(hjust = 0.5,
    size = 16, face = "bold"), axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)) +
    labs(title = "Boxplot for Annual Income of the Applicant",
        x = "Loan status", y = "Annual Income of the Applicant") +
    scale_x_discrete(labels = c("approved",
        "rejected"))
```

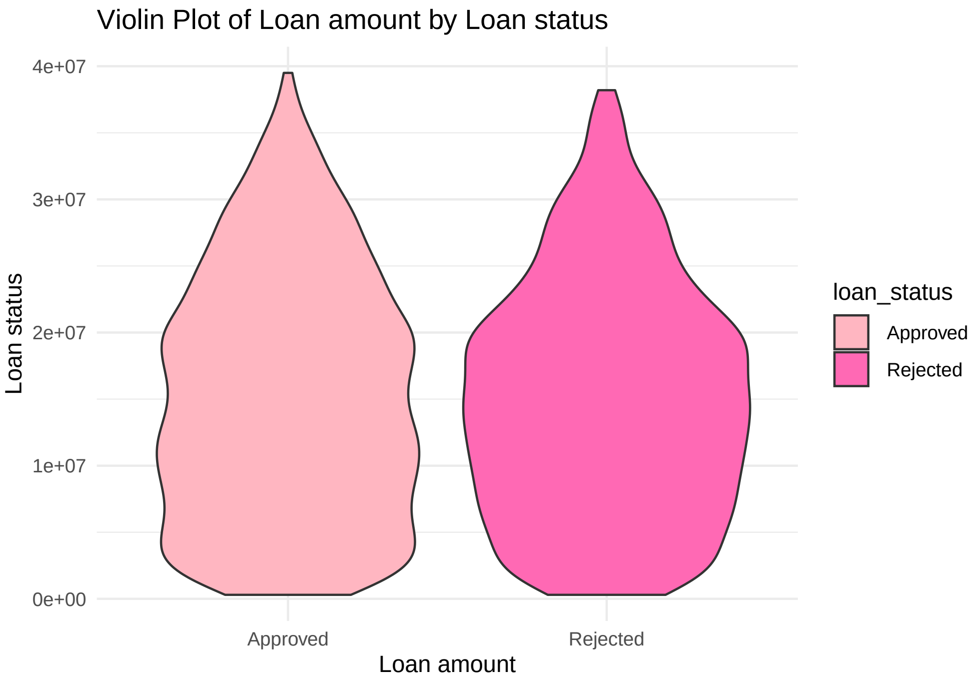
0.7.3.3 Box plot for Annual Income of the Applicant.



0.7.4 Violin plots.

```
ggplot(loans, aes(x = loan_status, y = loan_amount)) +
  geom_violin(aes(fill = loan_status)) +
  scale_fill_manual(values = c("lightpink", "hotpink")) +
  theme_minimal() +
  labs(title = "Violin Plot of Loan amount by Loan status", x = "Loan amount", y = "Loan status") +
  scale_x_discrete(labels = c("Approved", "Rejected"))
```

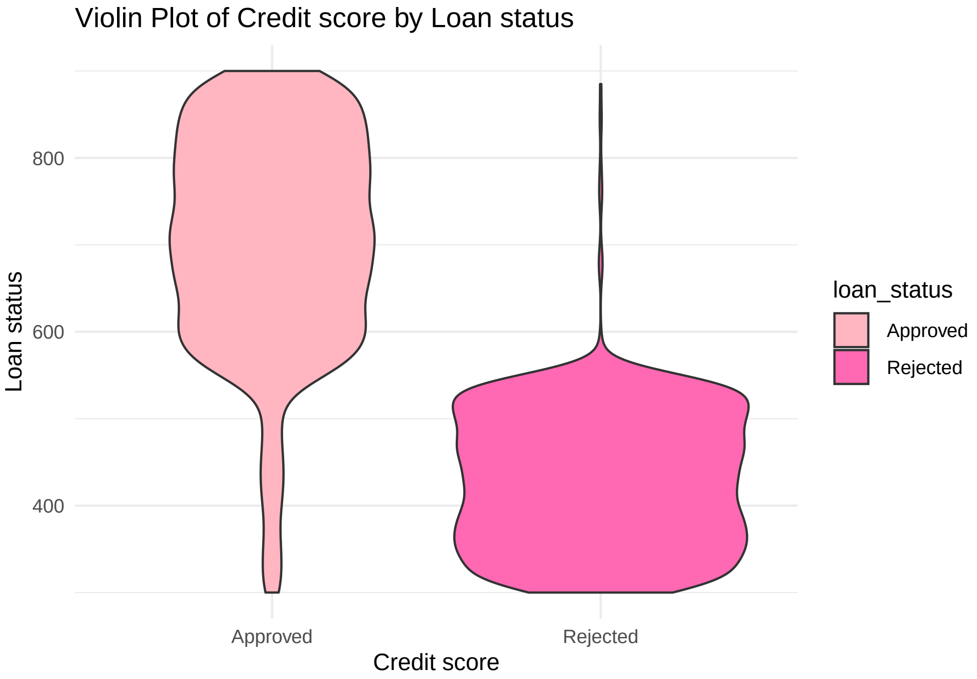
0.7.4.1 Violin Plot of Loan amount by Loan status.



The violin plot illustrates the distribution of loan amounts based on whether they were approved or rejected. Both approved and rejected loans exhibit similar distributions, with a concentration of lower loan amounts and a gradual decrease towards higher amounts. This suggests that most loan applications tend to be for smaller amounts. The violin for approved loans appears to have a slightly higher density peak at the lower end of the range, suggesting that most approved loans tend to be for smaller amounts.

```
ggplot(loans, aes(x = loan_status, y = cibil_score)) +
    geom_violin(aes(fill = loan_status)) + scale_fill_manual(values = c("lightpink",
    "hotpink")) + theme_minimal() + labs(title = "Violin Plot of Credit score by Loan status",
    x = "Credit score", y = "Loan status") + scale_x_discrete(labels = c("Approved",
    "Rejected"))
```

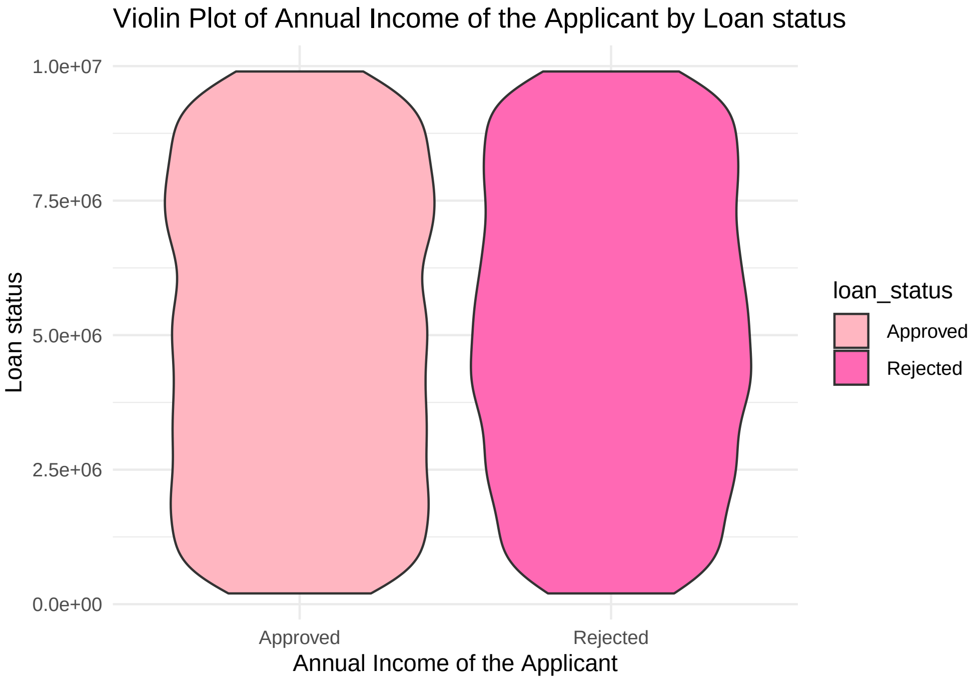
0.7.4.2 Violin Plot of Credit score by Loan status.



The violin plot clearly shows that credit score is a key factor in the loan approval decision. Applicants with higher credit scores have a higher probability of obtaining approval. However, it's important to note that credit score is not the sole factor considered, and other factors such as payment history, income, and debt-to-income ratio also play a significant role.

```r
ggplot(loans, aes(x = loan_status, y = income_annum)) +
    geom_violin(aes(fill = loan_status)) +
    scale_fill_manual(values = c("lightpink",
        "hotpink")) + theme_minimal() + labs(title = "Violin Plot of Annual Income of the Applicant by Loan status"
    x = "Annual Income of the Applicant",
    y = "Loan status") + scale_x_discrete(labels = c("Approved",
    "Rejected"))
```

0.7.4.3 Violin Plot of Annual Income of the Applicant by Loan status.



In this case, both approved and rejected loans exhibit similar distributions, with a concentration of lower annual incomes and a gradual decrease towards higher incomes. This suggests that a majority of loan applicants tend to have lower to moderate annual incomes. The results suggest that annual income is a factor considered by lenders in loan approval decisions. Applicants with higher incomes may be perceived as having a lower risk of default.

0.8 Categorical variables analysis.

0.8.1 Frequency table.

Here we take the categorical variable Education and obtain its frequency table.

```r
library(knitr)

frequency_table <- table(loans$education)

frequency_df <- as.data.frame(frequency_table)

# rename the columns
colnames(frequency_df) <- c("Education", "Frequency")

# Campute the percentage
total_count <- sum(frequency_df$Frequency)
frequency_df$Percentage <- (frequency_df$Frequency / total_count) * 100

frequency_df <- frequency_df %>%
              mutate(across(where(is.numeric), round, 2)) %>%
              mutate(Percentage = paste0(Percentage, "%"))


kable(frequency_df,
      col.names = c("Education", "Frequency", "Percentage (%)"),
      caption = "Frequency table of Education",
      align = "c")
```

Table 26: Frequency table of Education

Education	Frequency	Percentage (%)
Graduate	2144	50.22%
Not Graduate	2125	49.78%

According to data published by UNESCO in 2022, India has a literacy rate of 76.32%. Therefore, as shown in this table, where 50,22% of our sample has graduated, it confirms that our sample is not representative of the entire population of India, and only a minority has the opportunity to apply for a loan from a bank.

0.8.2 Descriptive contingency table.

In this one, we take the variables Loan status and Education (both categorical) and obtain their contingency table.

```r
contingency_table <- table(loans$loan_status, loans$education)
kable(contingency_table)
```

	Graduate	Not Graduate
Approved	1339	1317
Rejected	805	808

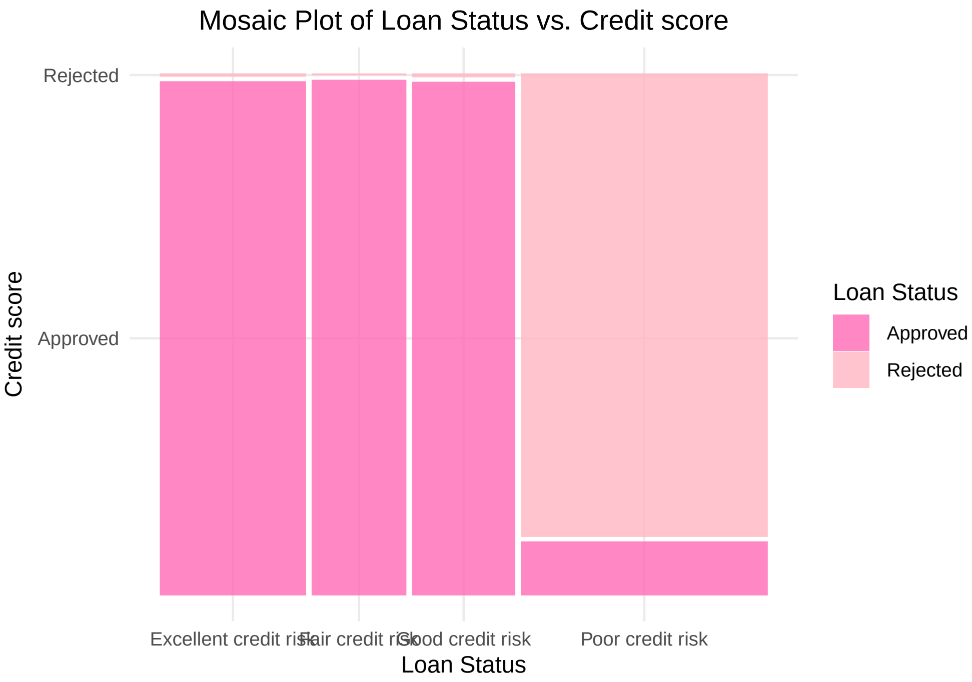
From this table it seems that the education does not affect on the probability of having the loan approved or rejected.

0.8.3 Mosaic plots.

This mosaic plot might not be visually appealing, but it provides a wealth of information. As we can see, being rejected is quite uncommon when the credit score is excellent, fair, or good. However, this changes when the credit score is poor— in that case, rejection becomes much more likely.

```
library(ggmosaic)

ggplot(loans) +
  geom_mosaic(aes(x = product(loan_status, cibil_score_char), fill = loan_status)) +
  labs(title = "Mosaic Plot of Loan Status vs. Credit score",
       x = "Loan Status",
       y = "Credit score",
       fill = "Loan Status") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("hotpink", "lightpink"))
```



From this we can conclude that the cibil score, or risk of non-payment, does influence significantly in the final status of the loan.

Bibliography

Abhiman, Das. 2007. “Determinants of Credit Risk in Indian Satate-Owned Banks: An Empirical Investigation.” *Economic Issues* 12 (2): 48–66.

Bandyopadhyay, Arindam. 2016. “Studying Borrowed Level Risk Characteristics of Education Loan in India.” *IIMB Management Review* 28 (1): 126–35.

Tiwari, Rajesh. 2013. “Role of Education Loan in Indian Higher Education.” *International Interdisciplinary Research Journal* 1 (2): 89–98.