# Bank loan performance analysis in Power BI

By

V. Hemapriya.

# Contents:

**Table of Contents**                                               **Page No**

# 1.Problem Statement:

- In today's data-driven world, understanding how borrower details and loan characteristics impact loan performance is very important for banking institutions.
- This project seeks to delve deep into a lending loan dataset to uncover the relationship between borrower behaviour (such as employment length, income, and debt-to-income ratio) and loan characteristics (including amount, term, and interest rate) to unearth critical insights into loan performance metrics.
- By examining patterns in loan statuses such as fully paid, charged off, or late payments, this analysis aims to empower banking institutions with actionable insights to optimize loan lending strategies, mitigate credit risk, and enhance overall portfolio performance.

# 2.Data Dictionary:

| Table Name | Field Name | Description |
| --- | --- | --- |
| LoanDetails | id | Unique identifier for each loan. |
| | loan_amnt | The amount of money requested by the borrower. |
| | funded_amnt | The actual amount of money funded for the loan. |
| | term | The duration of the loan in months. |
| | int_rate | The interest rate of the loan. |
| | installment | The monthly payment owed by the borrower. |
| | grade | The loan grade assigned by the lending company. |
| | sub_grade | The loan subgrade assigned by the lending company. |
| | issue_d | The month in which the loan was funded. |
| | purpose | The reason provided by the borrower for the loan. |

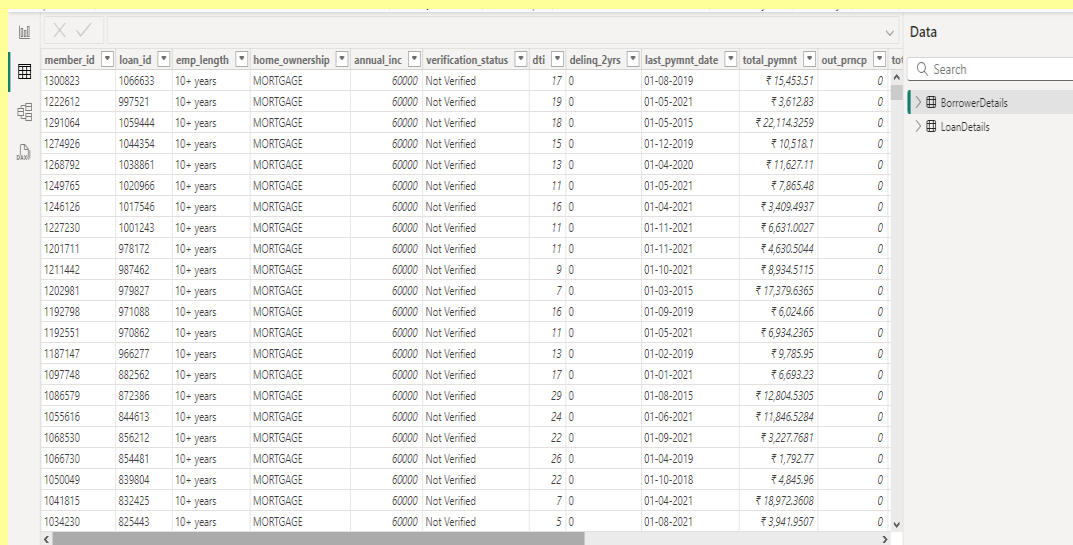| | id | Unique identifier for each loan. |
| --- | --- | --- |
| | member_id | Unique identifier for each borrower. |
| | emp_length | Employment length in years. |
| | home_ownership | The status of home ownership reported by the borrower. |
| | annual_inc | The annual income reported by the borrower. |
| BorrowerDetails | verification_status | Indicates if the borrower's income was verified. |
| | dti | The debt-to-income ratio of the borrower. |
| | delinq_2yrs | The number of past-due incidences in the borrower's credit file. |
| | last_pymnt_d | The month of the last payment received. |
| | total_pymnt | The total amount received in payments. |
| | out_prncp | The remaining outstanding principal amount of the loan. |

# 3.Project Steps and Objectives:

## 3.1 Importing Data:

- Import the "Loan Details" and "Borrower Details" sheets from the "bank loan.xlsx" file into Power BI.

## 3.2 Transformation Using Power Query:

### 3.2.1 Data Cleaning:

- Handling Missing Values and Duplicates:
- Replace missing values (null) in the 'emp_ length' column of the "Borrower Details" table with '0 year'.
- Remove rows with missing values in the 'last_ payment_ d' and 'delinq_2yrs' columns.
- Remove duplicate rows in the 'id' column of the "Loan Details" table.

| member_id | loan_id | emp_length | home_ownership | annual_inc | verification_status | dti | delinq_2yrs | last_pymnt_date | total_pymnt | out_prncp | tot |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1300823 | 1066633 | 10+ years | MORTGAGE | 60000 | Not Verified | 17 | 0 | 01-08-2019 | ₹ 15,453.51 | 0 | |
| 1222612 | 997521 | 10+ years | MORTGAGE | 60000 | Not Verified | 19 | 0 | 01-05-2021 | ₹ 3,612.83 | 0 | |
| 1291064 | 1059444 | 10+ years | MORTGAGE | 60000 | Not Verified | 18 | 0 | 01-05-2015 | ₹ 22,114.3259 | 0 | |
| 1274926 | 1044354 | 10+ years | MORTGAGE | 60000 | Not Verified | 15 | 0 | 01-12-2019 | ₹ 10,518.1 | 0 | |
| 1268792 | 1038861 | 10+ years | MORTGAGE | 60000 | Not Verified | 13 | 0 | 01-04-2020 | ₹ 11,627.11 | 0 | |
| 1249765 | 1020966 | 10+ years | MORTGAGE | 60000 | Not Verified | 11 | 0 | 01-05-2021 | ₹ 7,865.48 | 0 | |
| 1246126 | 1017546 | 10+ years | MORTGAGE | 60000 | Not Verified | 16 | 0 | 01-04-2021 | ₹ 3,409.4937 | 0 | |
| 1227230 | 1001243 | 10+ years | MORTGAGE | 60000 | Not Verified | 11 | 0 | 01-11-2021 | ₹ 6,631.0027 | 0 | |
| 1201711 | 978172 | 10+ years | MORTGAGE | 60000 | Not Verified | 11 | 0 | 01-11-2021 | ₹ 4,630.5044 | 0 | |
| 1211442 | 987462 | 10+ years | MORTGAGE | 60000 | Not Verified | 9 | 0 | 01-10-2021 | ₹ 8,934.5115 | 0 | |
| 1202981 | 979827 | 10+ years | MORTGAGE | 60000 | Not Verified | 7 | 0 | 01-03-2015 | ₹ 17,379.6365 | 0 | |
| 1192798 | 971088 | 10+ years | MORTGAGE | 60000 | Not Verified | 16 | 0 | 01-09-2019 | ₹ 6,024.66 | 0 | |
| 1192551 | 970862 | 10+ years | MORTGAGE | 60000 | Not Verified | 11 | 0 | 01-05-2021 | ₹ 6,934.2365 | 0 | |
| 1187147 | 966277 | 10+ years | MORTGAGE | 60000 | Not Verified | 13 | 0 | 01-02-2019 | ₹ 9,785.95 | 0 | |
| 1097748 | 882562 | 10+ years | MORTGAGE | 60000 | Not Verified | 17 | 0 | 01-01-2021 | ₹ 6,693.23 | 0 | |
| 1086579 | 872386 | 10+ years | MORTGAGE | 60000 | Not Verified | 29 | 0 | 01-08-2015 | ₹ 12,804.5305 | 0 | |
| 1055616 | 844613 | 10+ years | MORTGAGE | 60000 | Not Verified | 24 | 0 | 01-06-2021 | ₹ 11,846.5284 | 0 | |
| 1068530 | 856212 | 10+ years | MORTGAGE | 60000 | Not Verified | 22 | 0 | 01-09-2021 | ₹ 3,227.7681 | 0 | |
| 1066730 | 854481 | 10+ years | MORTGAGE | 60000 | Not Verified | 26 | 0 | 01-04-2019 | ₹ 1,792.77 | 0 | |
| 1050049 | 839804 | 10+ years | MORTGAGE | 60000 | Not Verified | 22 | 0 | 01-10-2018 | ₹ 4,845.96 | 0 | |
| 1041815 | 832425 | 10+ years | MORTGAGE | 60000 | Not Verified | 7 | 0 | 01-04-2021 | ₹ 18,972.3608 | 0 | |
| 1034230 | 825443 | 10+ years | MORTGAGE | 60000 | Not Verified | 5 | 0 | 01-08-2021 | ₹ 3,941.9507 | 0 | |

Data
Search
> BorrowerDetails
> LoanDetails

### 3.2.2 Dealing with Inconsistencies:

- Ensure words in the 'purpose' column are separated by spaces instead of underscores (e.g., "credit card" instead of "credit_ card").
- Format the 'purpose' and 'home_ ownership' columns to proper case.

## 3.2.2 Data Transformation:

Column Transformation:

- Change the data type of the 'total_ pymnt' column to 'Fixed decimal number'.
- Round off the numbers in the 'funded_ amnt' column to 2 decimal places.

| | 2yrs | ABC 123 last_pymnt_date | $ total_pymnt |
|---|---|---|---|
| 1 | 0 | 01-01-2015 | 5,861.07 |
| 2 | 0 | 01-04-2020 | 1,008.71 |
| 3 | 0 | 01-06-2021 | 3,003.65 |
| 4 | 0 | 01-01-2015 | 12,226.30 |
| 5 | 0 | 01-01-2016 | 3,242.17 |
| 6 | 0 | 01-01-2015 | 5,631.38 |
| 7 | 0 | 01-01-2016 | 8,136.84 |
| 8 | 0 | 01-01-2015 | 3,938.14 |
| 9 | 0 | 01-04-2019 | 646.02 |
| 10 | 0 | 01-11-2019 | 1,476.19 |
| 11 | 0 | 01-06-2020 | 7,677.52 |
| 12 | 0 | 01-09-2020 | 13,943.08 |
| 13 | 0 | 01-07-2019 | 2,270.70 |
| 14 | 0 | 01-01-2015 | 3,478.98 |
| 15 | 0 | 01-10-2020 | 7,471.99 |
| 16 | 0 | 01-01-2015 | 1,270.17 |
| 17 | 0 | 01-01-2015 | 12,519.26 |
| 18 | 0 | 01-05-2020 | 3,785.02 |
| 19 | 2 | 01-02-2015 | 7,164.50 |
| 20 | 0 | 01-07-2019 | 9,459.96 |
| 21 | 0 | 01-08-2015 | 27,663.04 |
| 22 | 0 | 01-09-2020 | 14,025.40 |
| 23 | 0 | 01-01-2015 | 11,902.56 |

| ABC 123 id | 1²₃ loan_amnt | $ funded_amnt |
| --- | --- | --- |
| 1077501 | 5000 | 4,975.00 |
| 1077430 | 2500 | 2,500.00 |
| 1077175 | 2400 | 2,400.00 |
| 1076863 | 10000 | 10,000.00 |
| 1075358 | 3000 | 3,000.00 |
| 1075269 | 5000 | 5,000.00 |
| 1069639 | 7000 | 7,000.00 |
| 1072053 | 3000 | 3,000.00 |
| 1071795 | 5600 | 5,600.00 |
| 1071570 | 5375 | 5,350.00 |
| 1070078 | 6500 | 6,500.00 |
| 1069908 | 12000 | 12,000.00 |
| 1064687 | 9000 | 9,000.00 |
| 1069866 | 3000 | 3,000.00 |
| 1069057 | 10000 | 10,000.00 |
| 1069759 | 1000 | 1,000.00 |
| 1065775 | 10000 | 10,000.00 |
| 1069971 | 3600 | 3,600.00 |
| 1062474 | 6000 | 6,000.00 |
| 1069742 | 9200 | 9,200.00 |
| 1069740 | 20250 | 19,142.16 |
| 1039153 | 21000 | 21,000.00 |
| 1069710 | 10000 | 10,000.00 |

## Column Renaming:

- Rename the column 'issue_ d' to 'issue_ date'.
- Rename the column 'last_ pymnt_ d' to 'last_ pymnt_ date'.

| ABC 123 issue_date | ABC 123 last_pymnt_date |
| --- | --- |
| 01-12-2018 | 01-01-2015 |
| 01-12-2018 | 01-04-2020 |
| 01-12-2018 | 01-06-2021 |
| 01-12-2018 | 01-01-2015 |
| 01-12-2018 | 01-01-2016 |
| 01-12-2018 | 01-01-2015 |
| 01-12-2018 | 01-01-2016 |
| 01-12-2018 | 01-01-2015 |
| 01-12-2018 | 01-04-2019 |
| 01-12-2018 | 01-11-2019 |

## Creating New Columns:

- Create a new custom column named 'total_ amount_ paid' to calculate the total amount paid by each borrower by subtracting 'out_ prncp' from 'total_ pymnt'.
- Add a new conditional column named 'delinquency_ status' to identify if the borrower has any delinquencies. If the number of delinquencies in 'delinq_2yrs' is greater than 0, the status should be "Delinquent", otherwise "Not Delinquent".



## Column Dropping:

- Remove the 'sub_ grade' column as that does not significantly contribute to the analysis.

## 3.2.4 Data Modelling:

Identify the common column between both the tables and establish relationships between the two tables. Ensure the cross-filter direction is set to "Both". This step is crucial for enabling cross-table analysis and ensuring data integrity within the dataset.
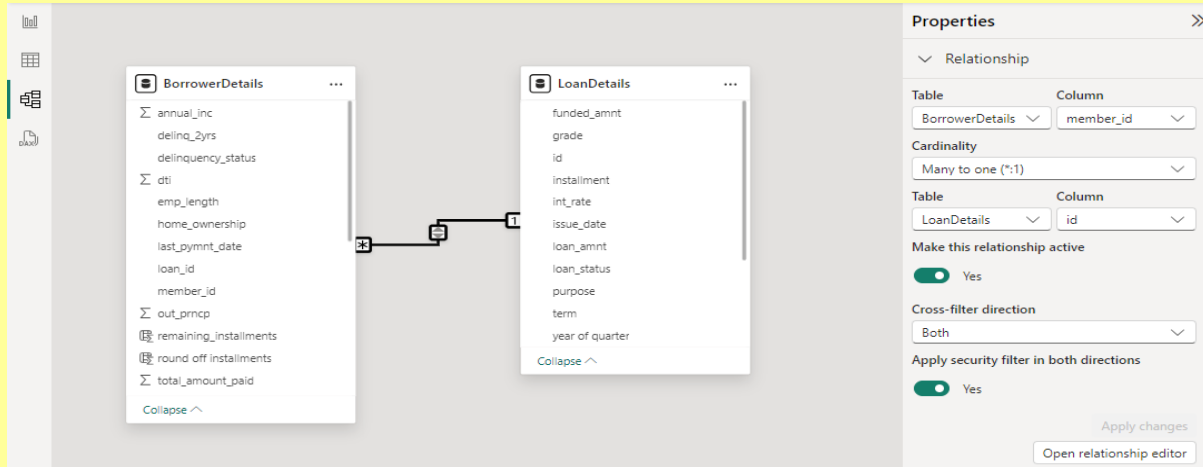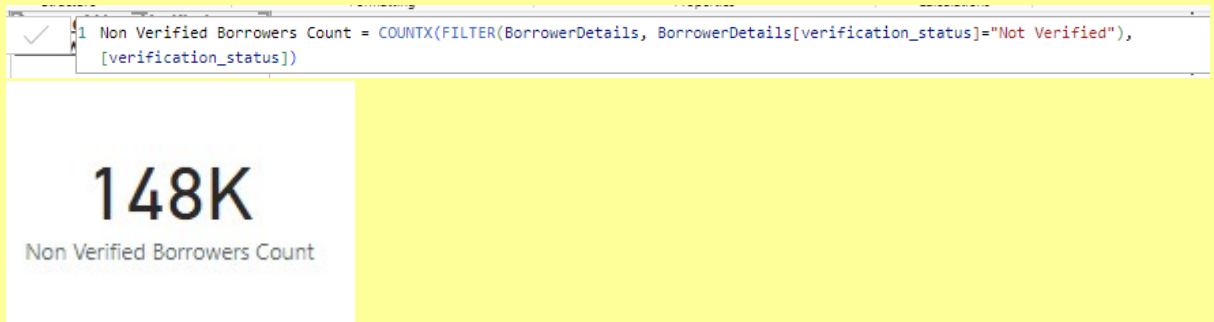


## 3.2.5 Creating Measures and Calculated Columns using DAX

- Create a new calculated column named 'remaining_ instalments' using DAX in the "Borrower Details" table to calculate the number of remaining instalments by dividing the remaining principal amount ('out_ prncp') by the monthly instalment amount ('instalment') and round up the result using the CEILING() function to account for any partial payments.

```
1 round off installments = CEILING(1,BorrowerDetails[remaining_installments])
```

| dti | delinq_2yrs | last_pymnt_date | total_pymnt | out_prncp | total_amount_paid | delinquency_status | remaining_installments | round off installments |
|---|---|---|---|---|---|---|---|---|
| 17 | 0 | 01-08-2019 | ₹ 15,453.51 | 0 | ₹ 15,453.51 | "Not Delinquent" | 0 | 0 |
| 19 | 0 | 01-05-2021 | ₹ 3,612.83 | 0 | ₹ 3,612.83 | "Not Delinquent" | 0 | 0 |
| 18 | 0 | 01-05-2015 | ₹ 22,114.3259 | 0 | ₹ 22,114.3259 | "Not Delinquent" | 0 | 0 |
| 15 | 0 | 01-12-2019 | ₹ 10,518.1 | 0 | ₹ 10,518.1 | "Not Delinquent" | 0 | 0 |
| 13 | 0 | 01-04-2020 | ₹ 11,627.11 | 0 | ₹ 11,627.11 | "Not Delinquent" | 0 | 0 |
| 11 | 0 | 01-05-2021 | ₹ 7,865.48 | 0 | ₹ 7,865.48 | "Not Delinquent" | 0 | 0 |
| 16 | 0 | 01-04-2021 | ₹ 3,409.4937 | 0 | ₹ 3,409.4937 | "Not Delinquent" | 0 | 0 |
| 11 | 0 | 01-11-2021 | ₹ 6,631.0027 | 0 | ₹ 6,631.0027 | "Not Delinquent" | 0 | 0 |
| 11 | 0 | 01-11-2021 | ₹ 4,630.5044 | 0 | ₹ 4,630.5044 | "Not Delinquent" | 0 | 0 |
| 9 | 0 | 01-10-2021 | ₹ 8,934.5115 | 0 | ₹ 8,934.5115 | "Not Delinquent" | 0 | 0 |
| 7 | 0 | 01-03-2015 | ₹ 17,379.6365 | 0 | ₹ 17,379.6365 | "Not Delinquent" | 0 | 0 |
| 16 | 0 | 01-09-2019 | ₹ 6,024.66 | 0 | ₹ 6,024.66 | "Not Delinquent" | 0 | 0 |
| 11 | 0 | 01-05-2021 | ₹ 6,934.2365 | 0 | ₹ 6,934.2365 | "Not Delinquent" | 0 | 0 |
| 13 | 0 | 01-02-2019 | ₹ 9,785.95 | 0 | ₹ 9,785.95 | "Not Delinquent" | 0 | 0 |
| 17 | 0 | 01-01-2021 | ₹ 6,693.23 | 0 | ₹ 6,693.23 | "Not Delinquent" | 0 | 0 |
| 29 | 0 | 01-08-2015 | ₹ 12,804.5305 | 0 | ₹ 12,804.5305 | "Not Delinquent" | 0 | 0 |
| 24 | 0 | 01-06-2021 | ₹ 11,846.5284 | 0 | ₹ 11,846.5284 | "Not Delinquent" | 0 | 0 |
| 22 | 0 | 01-09-2021 | ₹ 3,227.7681 | 0 | ₹ 3,227.7681 | "Not Delinquent" | 0 | 0 |
| 26 | 0 | 01-04-2019 | ₹ 1,792.77 | 0 | ₹ 1,792.77 | "Not Delinquent" | 0 | 0 |
| 22 | 0 | 01-10-2018 | ₹ 4,845.96 | 0 | ₹ 4,845.96 | "Not Delinquent" | 0 | 0 |
| 7 | 0 | 01-04-2021 | ₹ 18,972.3608 | 0 | ₹ 18,972.3608 | "Not Delinquent" | 0 | 0 |
| 5 | 0 | 01-08-2021 | ₹ 3,941.9507 | 0 | ₹ 3,941.9507 | "Not Delinquent" | 0 | 0 |

- Create a measure named 'Non-Verified Borrowers Count' using DAX to count the number of loans that have been 'Not Verified'.

```
1 Non Verified Borrowers Count = COUNTX(FILTER(BorrowerDetails, BorrowerDetails[verification_status]="Not Verified"),
    [verification_status])
```

## 148K
Non Verified Borrowers Count

- Create a measure named 'Fully Paid Loan Percentage' to calculate the percentage of fully paid loans. Divide the number of loans with a "Fully Paid" loan status by the total number of loans and then format this measure as Percentage.

```
1 Fully Paid Loan Percentage = [Total no of Loans]/[Fully Paid]
```

```
1 % of Fully Paid = [Fully Paid Loan Percentage]/100
```

## 2.52
Fully Paid Loan Percentage

## 0.03
% of Fully Paid
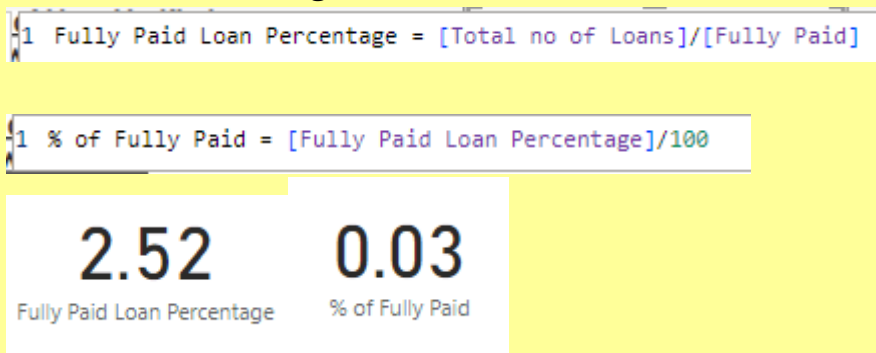
# 4.Creating Comprehensive Report:

## Report 1: Loan Performance Analysis

The Loan Performance Analysis report aims to provide insights into the performance of loans based on various factors such as loan amount, loan status, term, interest rate, and purpose.

➢ **Total Funded Amount**: Create a card visual to display the total funded amount.
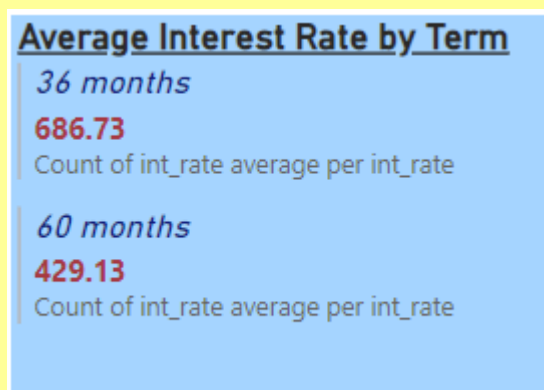
```
1 total funded amount = SUM(LoanDetails[funded_amnt])
```



**₹ 6.63bn**
total funded amount
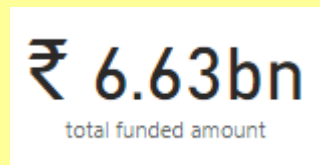
➢ **Fully Paid Loan Percentage**: Create a gauge chart to display the 'Fully Paid Loan Percentage' measure
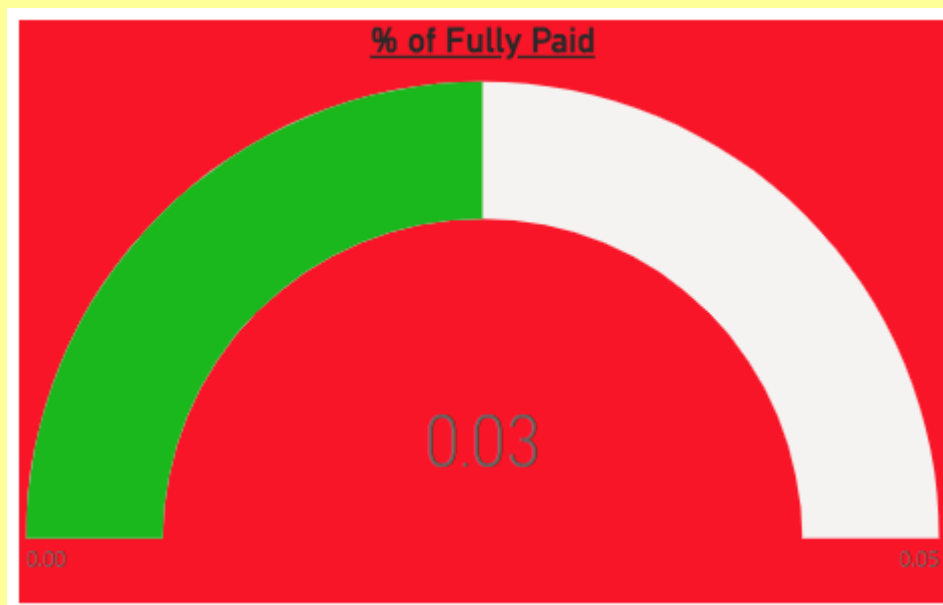
➤ **Average Interest Rate by Term:** Create a multi-row card to show the average interest rate for each term.

**Average Interest Rate by Term**

*36 months*

**686.73**

Count of int_rate average per int_rate

*60 months*

**429.13**

Count of int_rate average per int_rate

➤ **Loan Status Distribution:** Create a pie chart to visualize the sum of total payments by loan status.



Sum of installments by loan_status
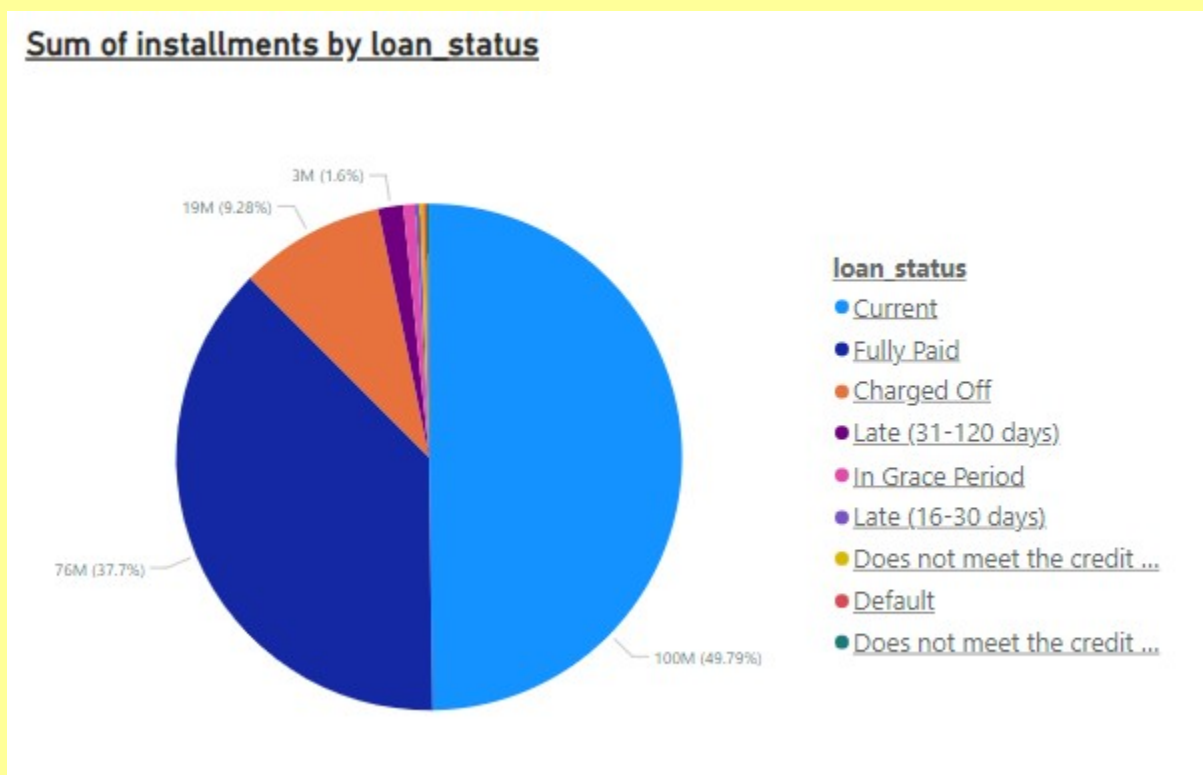
➢ **Loan Amount by Purpose**: Create a tree map to show the average loan amount by purpose.



Avg loan amount by purpose:
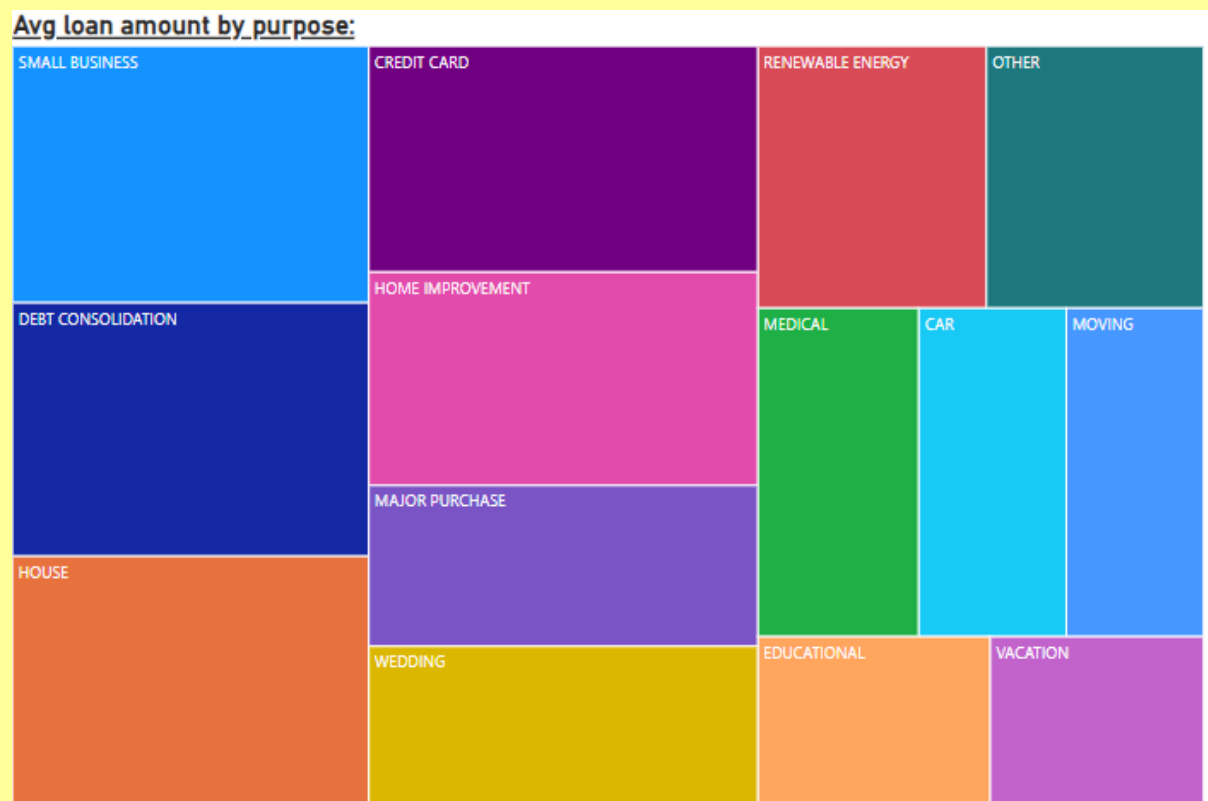
➢ **Instalment Over Time:** Create a line chart to visualize the sum of instalments by Year and Quarter of the issue date.



Sum of installments by Year of quarter:

➢ **Maximum Total Amount Paid by Loan Status**: Create a column chart to display the maximum total amount paid by loan status.

**Maximum total amount by Loan-status:**
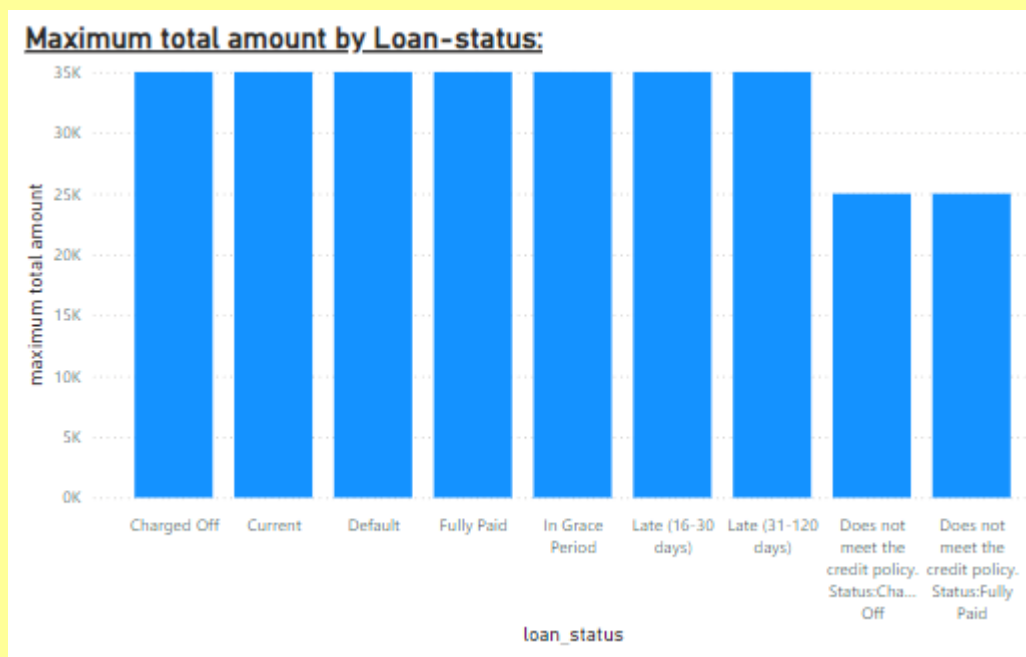


➢ **Minimum Annual Income by Grade**: Create a funnel chart to show the minimum annual income by grade.

**Minimum annual income by grade:**
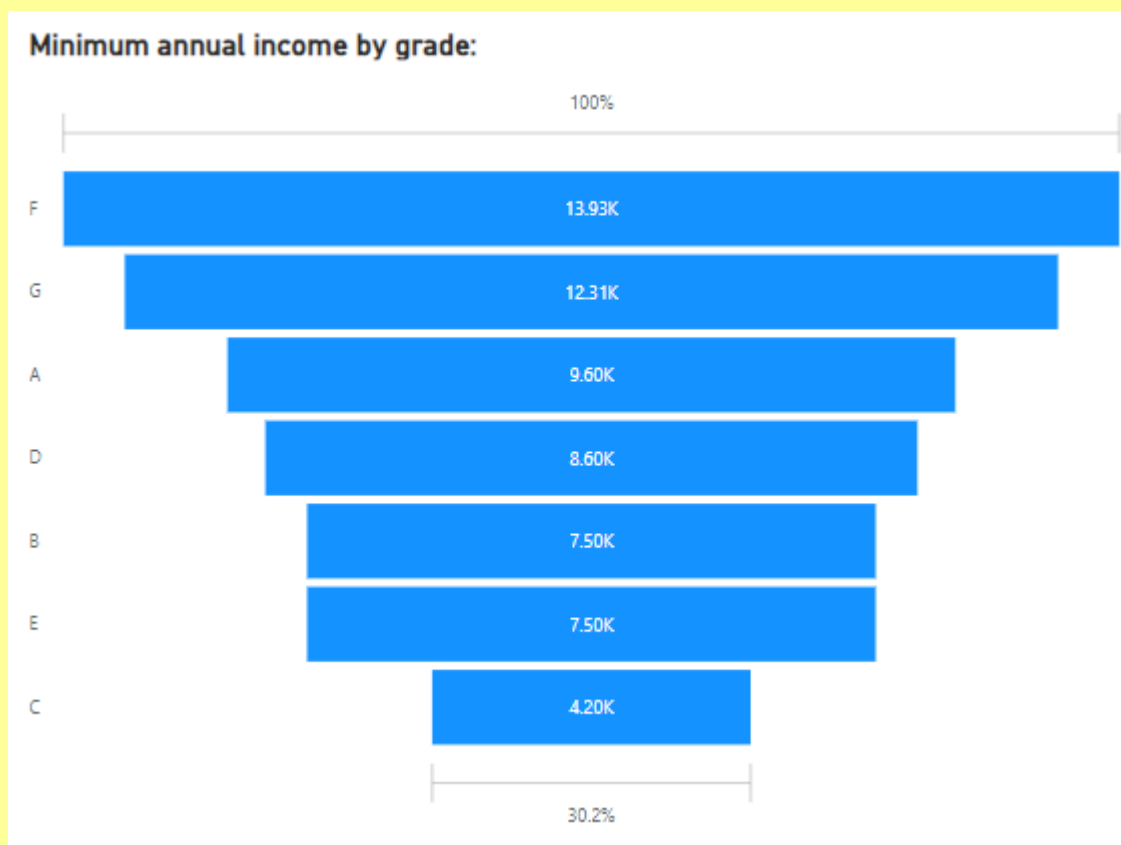
➢ **Issue Date Slicer**: Add a slicer for the Month of the issue date to enable dynamic data exploration
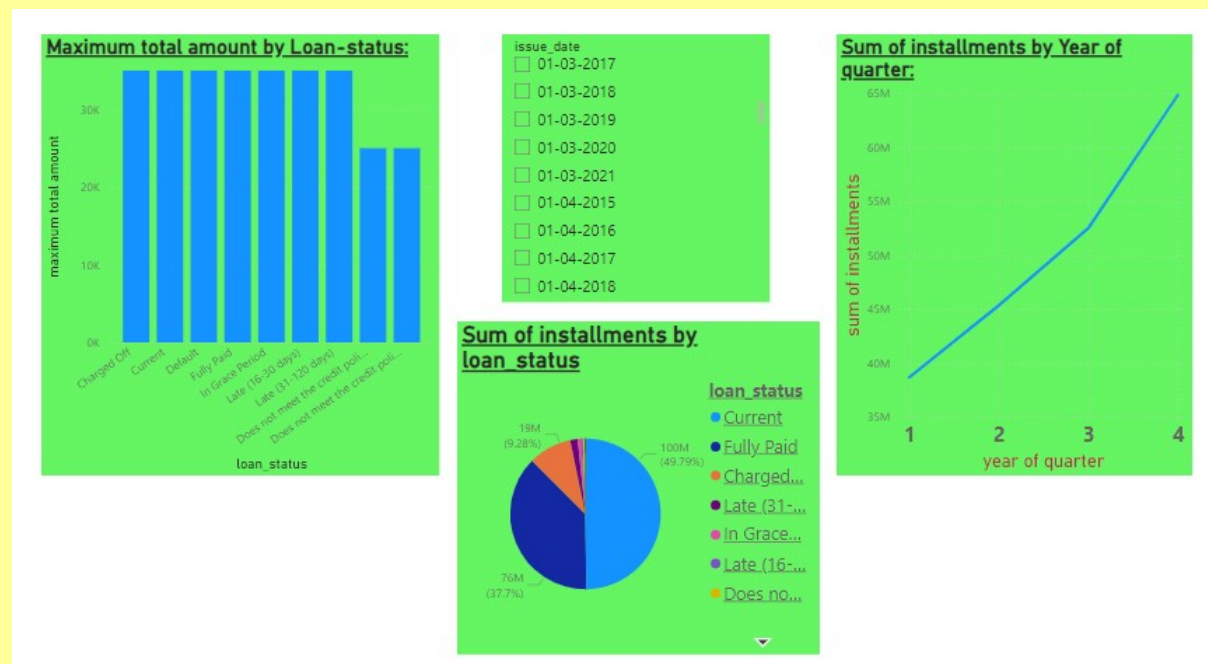
## Report 2: Borrower Profile Analysis

The Borrower Profile Analysis report aims to provide insights into the characteristics of borrowers such as home ownership, annual income, employment length, verification status, debt-to-income ratio, and delinquency history.

➤ **KPI Visual**: Create a KPI visual with the sum of total payment as the value, the year of last payment date as the trend axis, and the sum of loan amount as the target.

Sum of total payment and Total no of Loans by last_pymnt_date

₹ 97.0548912061M

Goal: 208 (+46660905.39%)

➤ **Average of Annual Income**: Display the average of annual income using a card visual.

Average of Annual Income

73.29K

Average of Annual Income

➤ **Non-Verified Borrowers Count**: Display the count of non-verified borrowers using a card visual

```
Non Verified Borrowers Count = COUNTX(FILTER(BorrowerDetails, BorrowerDetails[verification_status]="Not Verified"),
[verification_status])
```
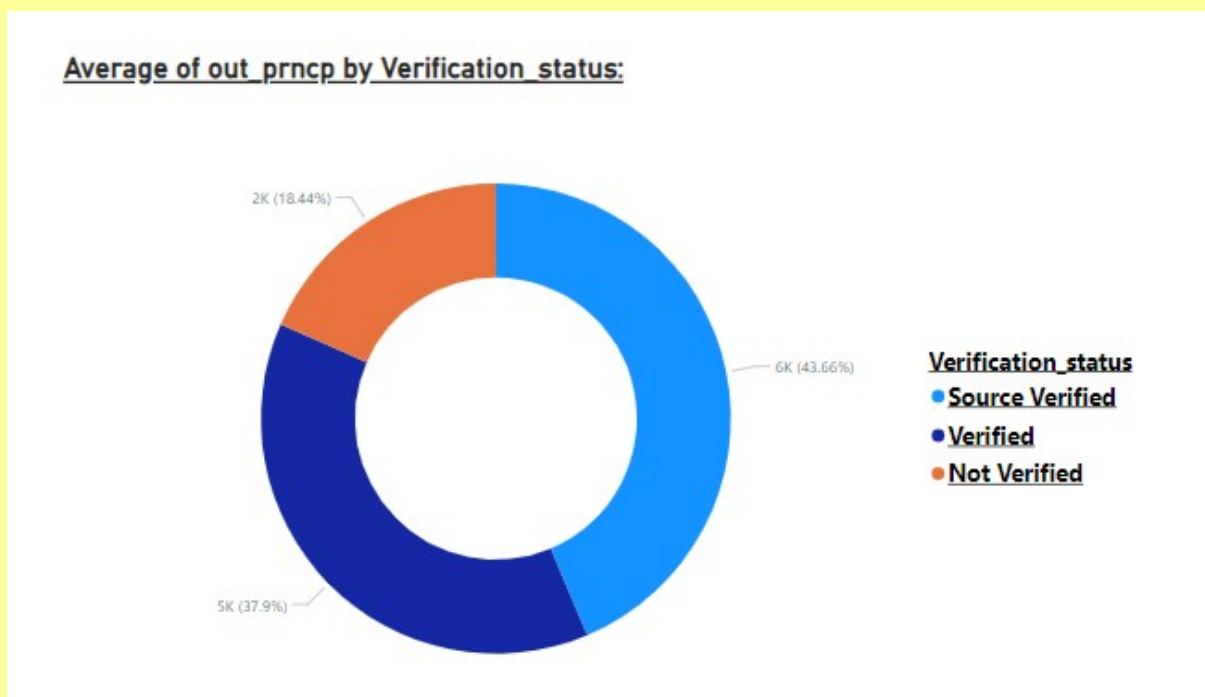
Count of Non-Verified Borrowers Count

148K

Non Verified Borrowers Count

➢ **Average Debt-to-Income by Delinquency Status**: Create a multi-row card to show the average debt-to-income ratio by delinquency status.

"Delinquent"
1427910
Sum of dti

"Not Delinquent"
6593269
Sum of dti

➢ **Sum of Loan Amount by Home Ownership:** Create a table to show the total loan amount by home ownership.

**Sum of Loan Amount by Home Ownership:**

| home_ownership | Total no of Loans |
|---|---|
| MORTGAGE | 5343 |
| OTHER | 11 |
| OWN | 934 |
| RENT | 4602 |
| **Total** | **466179** |

➢ **Average Remaining Principal by Verification Status**: Create a donut chart to display the average remaining outstanding principal by verification status.

**Average of out_prncp by Verification_status:**



2K (18.44%)

6K (43.66%)

5K (37.9%)

**Verification_status**
- Source Verified
- Verified
- Not Verified

➢ **Sum of Delinquencies by Home Ownership**: Create a bar chart to show the total number of delinquencies in the past 2 years by home ownership and filter the visual to display only Mortgage, Rent, and Own.
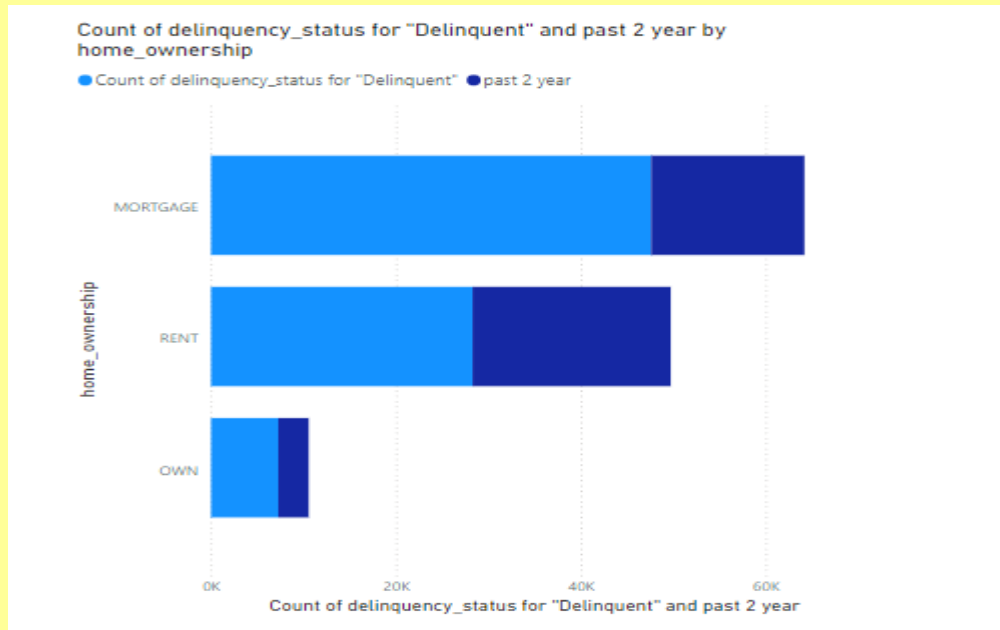


➢ **Max Remaining Instalments by Employment Length**: Create a tree map to show the maximum remaining instalments by employment length.
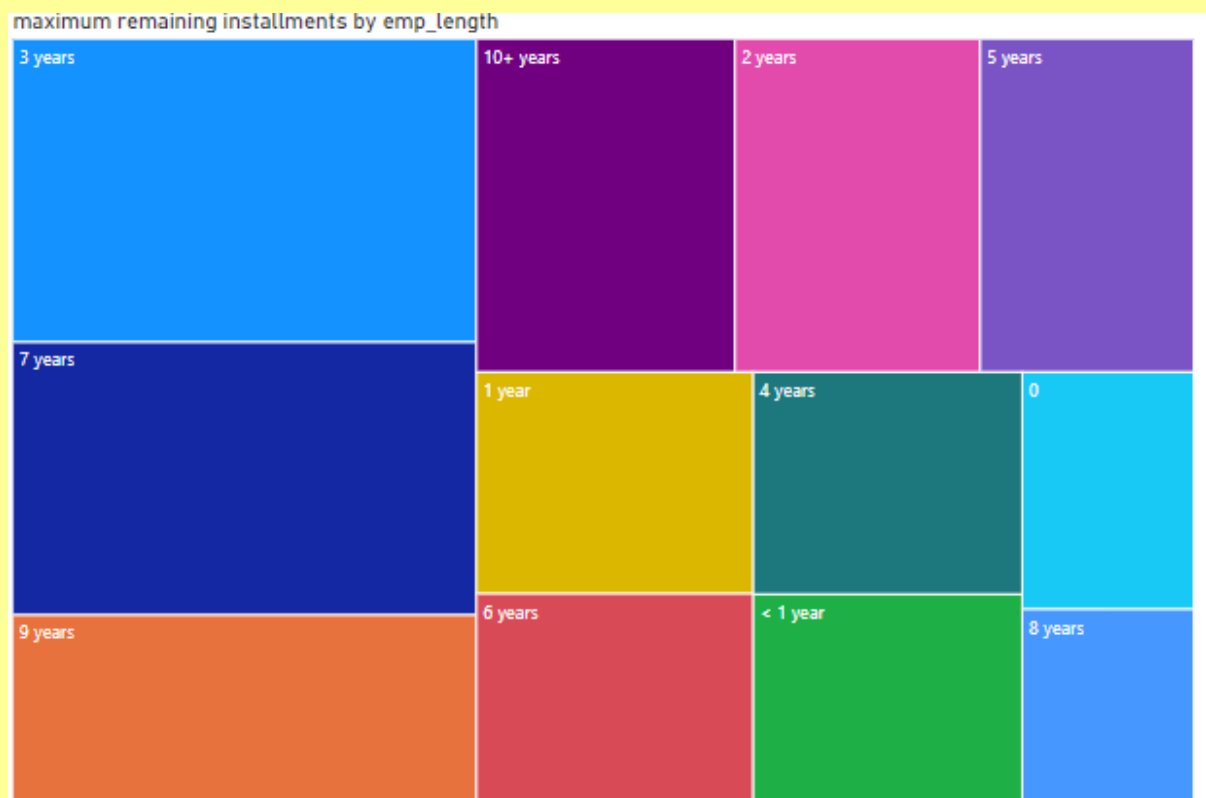
➢ **Total Amount Paid and Funded Amount Over Time:** Create a line chart to display the sum of total amount paid and the sum of funded amount by the year of last payment date.
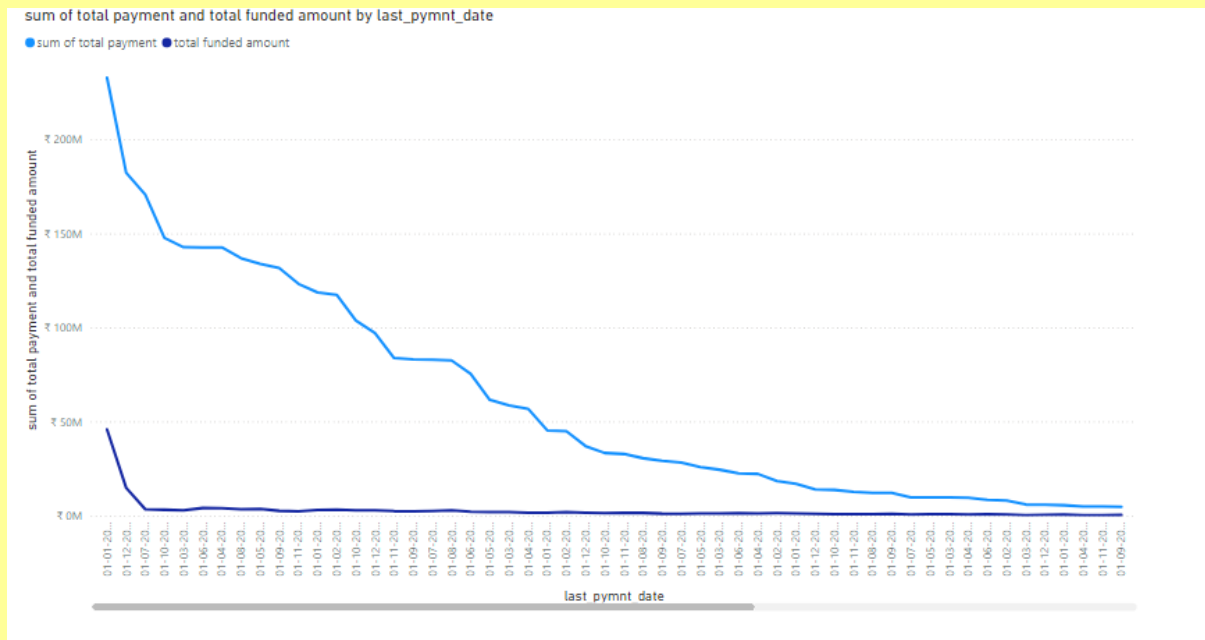


➢ **Purpose Slicer**: Add a slicer for loan purpose to enable dynamic data exploration
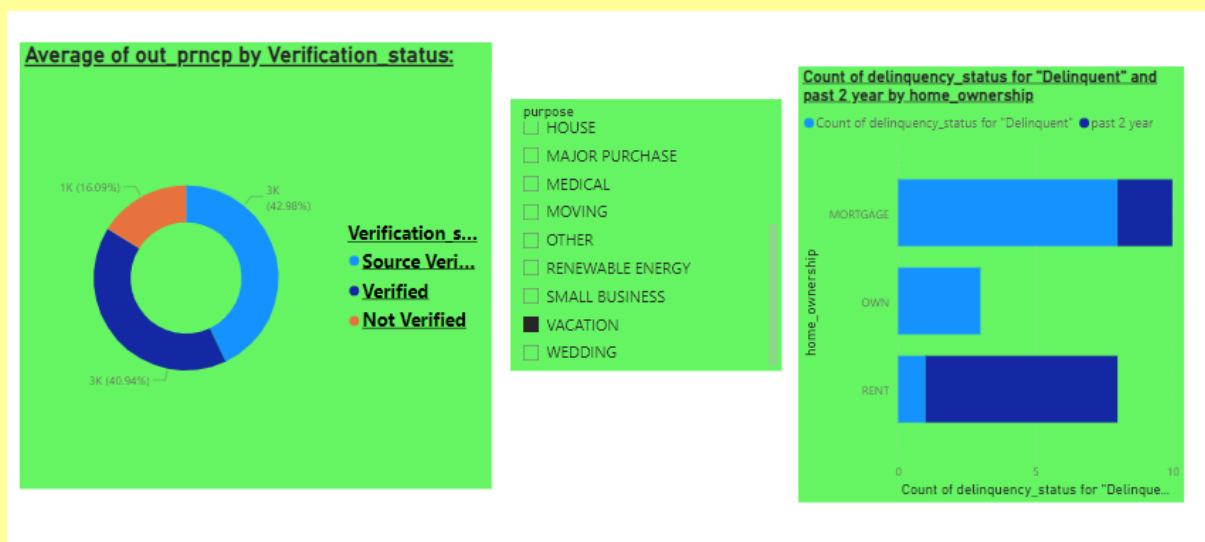
# Conclusion:

## Conclusion on Loan Performance Analysis:

- Loan performance analysis is a process that helps banks and other lenders assess the health of a loan portfolio. Loan performance analysis can help banks identify which loans are performing well and which loans may need additional attention.
- To conduct loan performance analysis, lenders typically gather data on a loans origination date, interest rate, and payment date. This information can help lenders determine whether a loan is performing as expected.
- Loan performance analysis can provide valuable insights for banks and other lenders. By identifying which loans are performing well and which loans may need additional attention, lenders can ensure that their portfolio remains healthy and meets customer needs.

## Conclusion on Borrowers Performance Analysis:

- In conclusion, borrower performance analysis serves as a vital tool for understanding the creditworthiness and repayment behaviour of borrowers. Key findings typically include trends in repayment rates, default risks, and overall financial health. By leveraging this analysis, lenders can make informed decisions about credit policies, risk management, and potential interventions to support borrowers.
- Moreover, consistent monitoring of borrower performance can help in identifying at-risk borrowers early, enabling proactive measures to mitigate losses. Overall, a thorough analysis not only enhances lender profitability but also promotes responsible lending practices, ultimately contributing to a more stable financial ecosystem.