LOAN DATA SUMMARY

By TrungHPB

1. CLEANING THE DATA

```sql
-- CONVERT months_since_last_deliquent to Integer -> Update table column
SELECT
    CASE
    WHEN [Months_since_last_delinquent] <> 'NA' THEN CAST([Months_since_last_delinquent] AS INT)
    WHEN [Months_since_last_delinquent] = 'NA' THEN NULL
    END
FROM PortfolioProject.dbo.credit_train$
ORDER BY [Months_since_last_delinquent] ASC

UPDATE PortfolioProject.dbo.credit_train$
SET [Months_since_last_delinquent] =
    CASE
    WHEN [Months_since_last_delinquent] <> 'NA' THEN CAST([Months_since_last_delinquent] AS INT)
    WHEN [Months_since_last_delinquent] = 'NA' THEN NULL
    END
FROM PortfolioProject.dbo.credit_train$
--CONVERT from nvarchar to float
ALTER TABLE PortfolioProject.dbo.credit_train$
ALTER COLUMN [Months_since_last_delinquent] FLOAT
```

Converting the **months_since_last_deliquent** from string to float

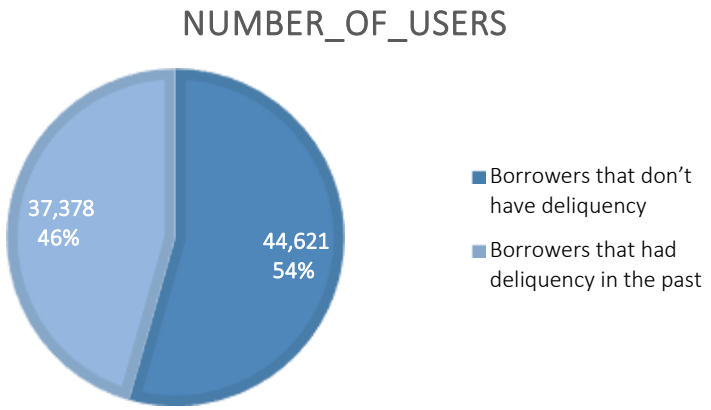
```sql
--REMOVE DUPLICATES DATA TABLE
WITH CTE AS (
SELECT *,
    ROW_NUMBER() OVER (
        PARTITION BY  Loan_ID,
                      Customer_ID,
                      Loan_Status,
                      Term,
                      Annual_Income,
                      Years_in_current_job,
                      Home_Ownership,
                      Purpose
                      ORDER BY
                        Loan_ID
        ) row_num
FROM PortfolioProject.dbo.credit_train$
)
SELECT *
FROM CTE
WHERE row_num > 1
--there are 13.794 rows that are duplicates -> REMOVING them
```

Finding the duplicate entries by the following columns: Loan_ID, customer_ID, Loan_status, Term, Annual_Income, Years_in_current_job, Home_Ownership, Purpose…

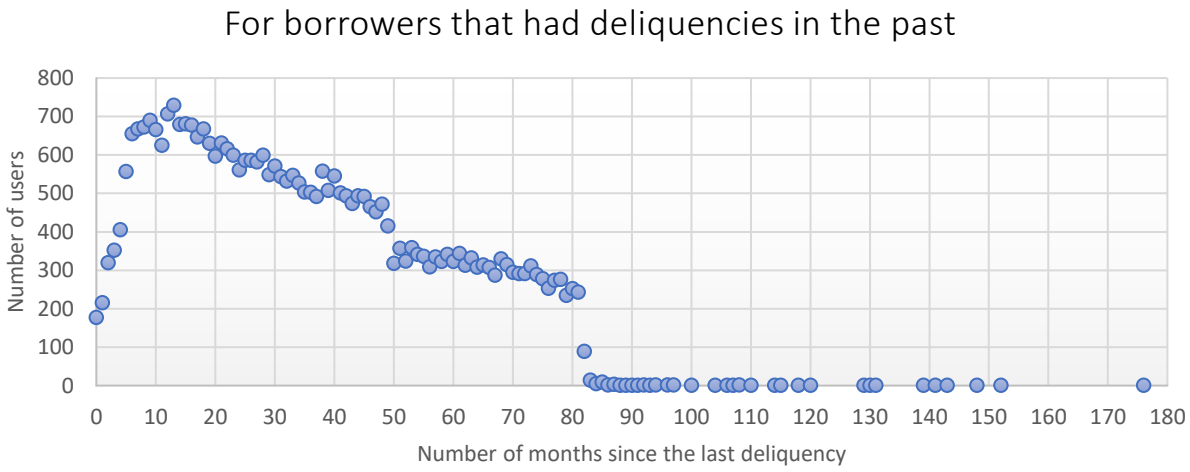
```sql
WITH CTE AS (
SELECT *,
    ROW_NUMBER() OVER (
        PARTITION BY  Loan_ID,
                      Customer_ID,
                      Loan_Status,
                      Term,
                      Annual_Income,
                      Years_in_current_job,
                      Home_Ownership,
                      Purpose
                      ORDER BY
                        Loan_ID
        ) row_num
FROM PortfolioProject.dbo.credit_train$
)
DELETE
FROM CTE
WHERE row_num > 1
```

2. QUERIES

```sql
-- Find out how many users have last deliquent in specified number of months
WITH CTE
AS
(
SELECT [Months_since_last_delinquent], COUNT(DISTINCT(Customer_ID)) AS number_of_users
FROM PortfolioProject.dbo.credit_train$
GROUP BY [Months_since_last_delinquent]
--ORDER BY [Months_since_last_delinquent] ASC
)
SELECT *, (CAST(number_of_users AS float) / 37378 * 100) AS Percentage_
FROM CTE
ORDER BY [Months_since_last_delinquent] ASC
```

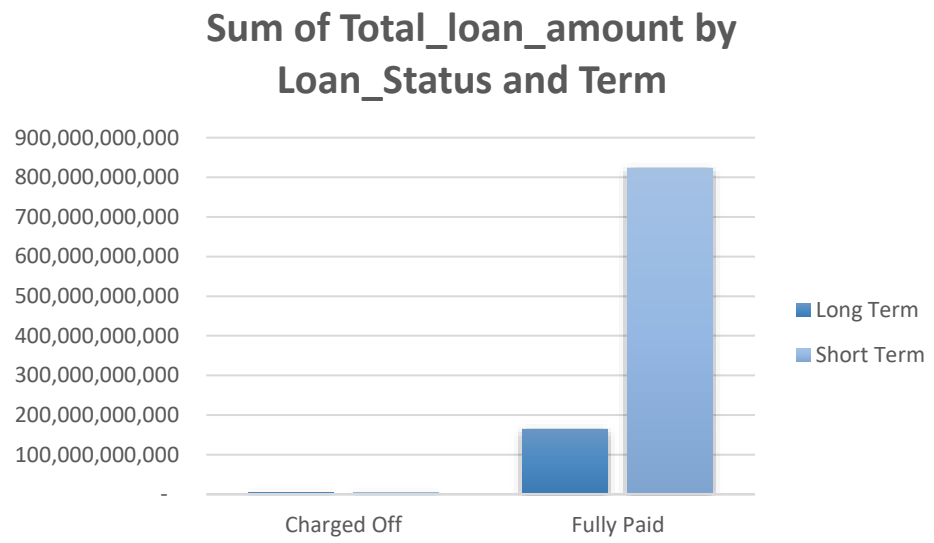


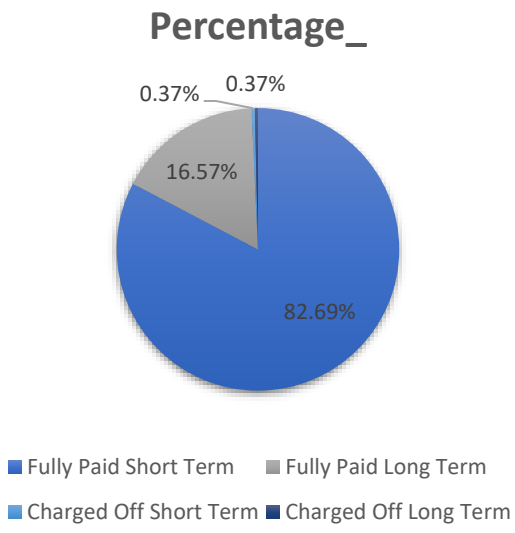
There is 54% out of all of the borrowers had delinquencies in the past.



The customers are skewed towards the 10-40 months ago since their last delinquencies (with the highest proportion of 1.95% of total borrowers at 13 months since their last delinquencies), following by the 50-80 months group.

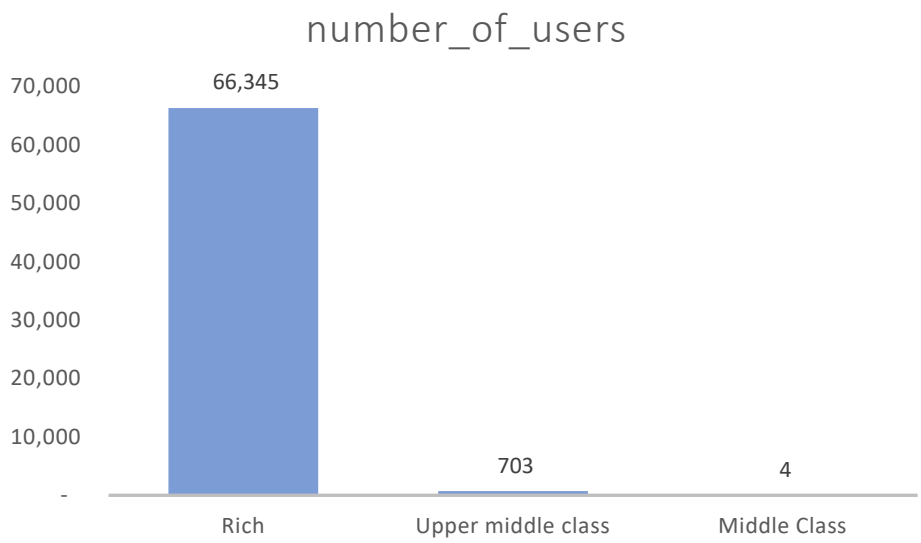
```sql
-- Find out total amount of loan filter by loan status and filter by term
DECLARE @loan float
SET @loan = (SELECT SUM(Current_Loan_Amount)
             FROM PortfolioProject.dbo.credit_train$);
WITH CTE AS
(
    SELECT Loan_Status, Term, SUM(Current_Loan_Amount) AS Total_loan_amount
    FROM PortfolioProject.dbo.credit_train$
    GROUP BY Loan_Status, Term
)
SELECT *, (Total_loan_amount / @loan) AS Percentage_
FROM CTE
ORDER BY Total_loan_amount DESC
```

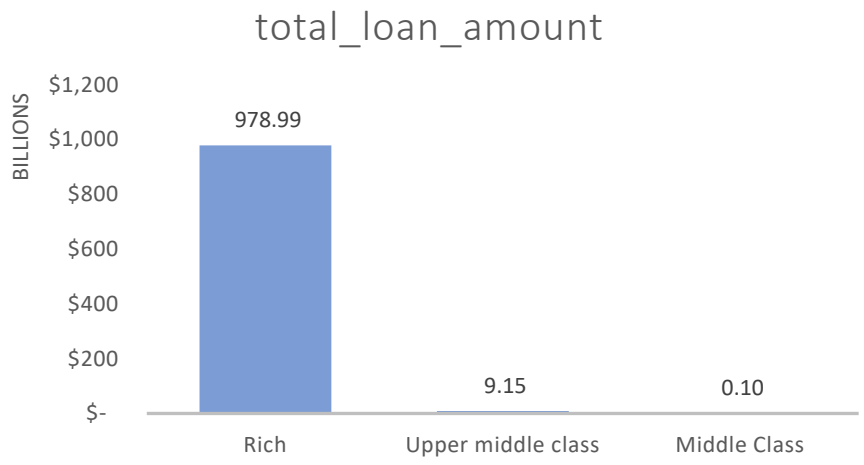




The written-off debt accounted for 0.74% of total loan amount, while the short-term fully paid accounted for 82.69%.

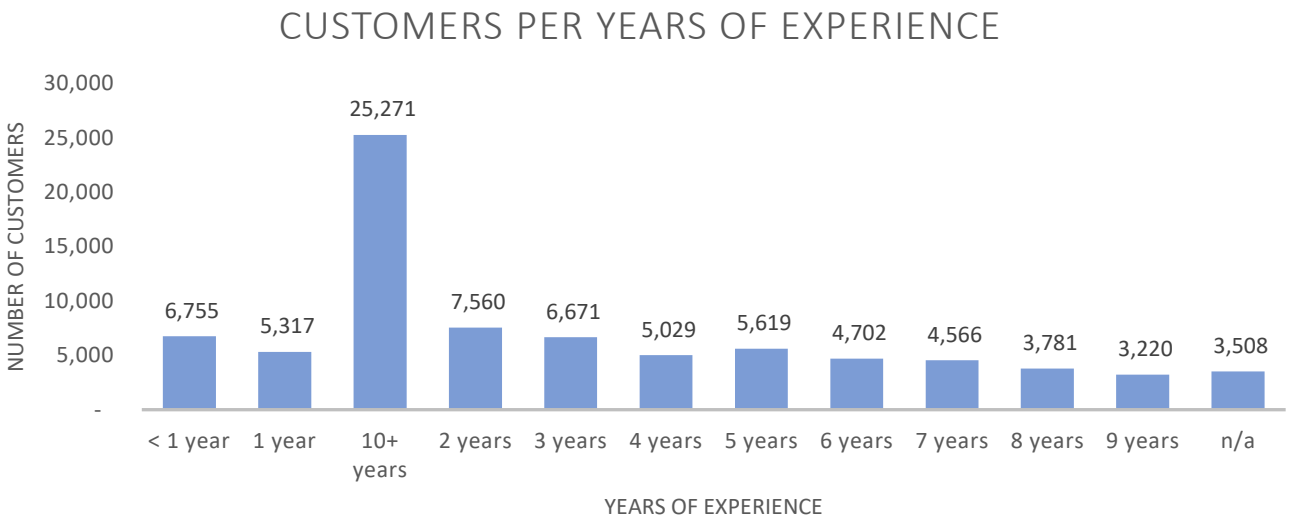
```sql
WITH CTE AS
(
SELECT Loan_ID, Customer_ID, Loan_Status, Current_Loan_Amount, Annual_Income, Monthly_Debt,
    CASE
        WHEN Annual_Income > 350000 THEN 'Rich'
        WHEN  Annual_Income > 100000 AND Annual_Income <=  350000  THEN 'Upper middle class'
        WHEN Annual_Income > 50000 AND Annual_Income <=  100000 THEN 'Middle Class'
        WHEN Annual_Income > 30000 AND Annual_Income <=  50000 THEN 'Lower Middle Class'
        WHEN Annual_Income <= 30000 THEN 'Poor'
    END AS SES_Breakdown,
     (Monthly_Debt / (Annual_Income / 12)) AS Debt_To_Income_Ratio
FROM PortfolioProject.dbo.credit_train$
WHERE Annual_Income <> 0 OR Annual_Income IS NOT NULL
)
SELECT SES_Breakdown, COUNT(DISTINCT(Customer_ID)) AS number_of_users, SUM(Current_Loan_Amount) total_loan_amount
FROM CTE
GROUP BY SES_Breakdown
ORDER BY SUM(Current_Loan_Amount) DESC
```



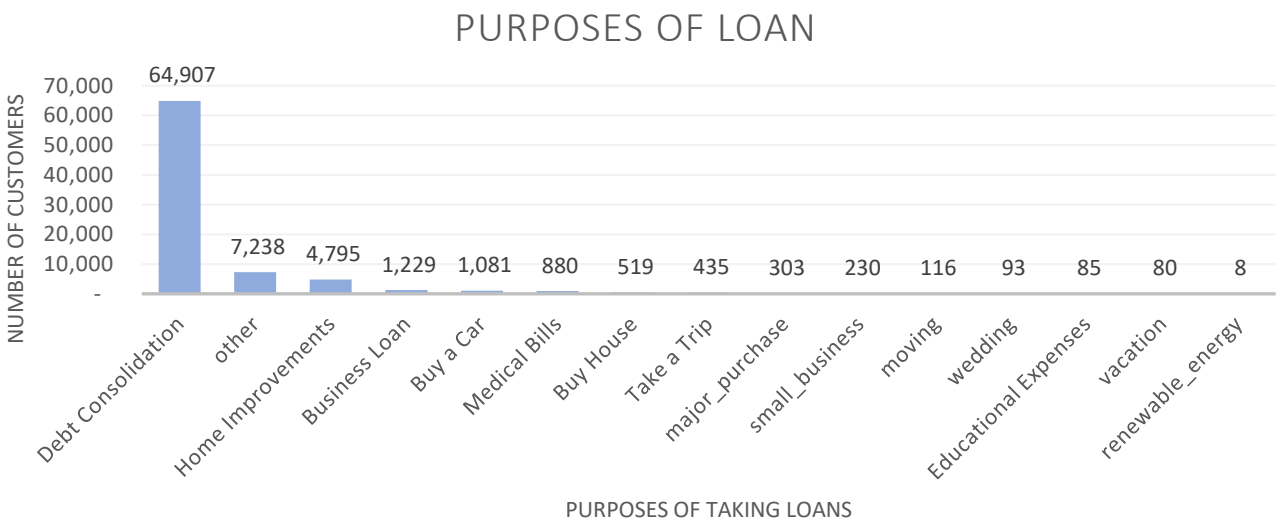


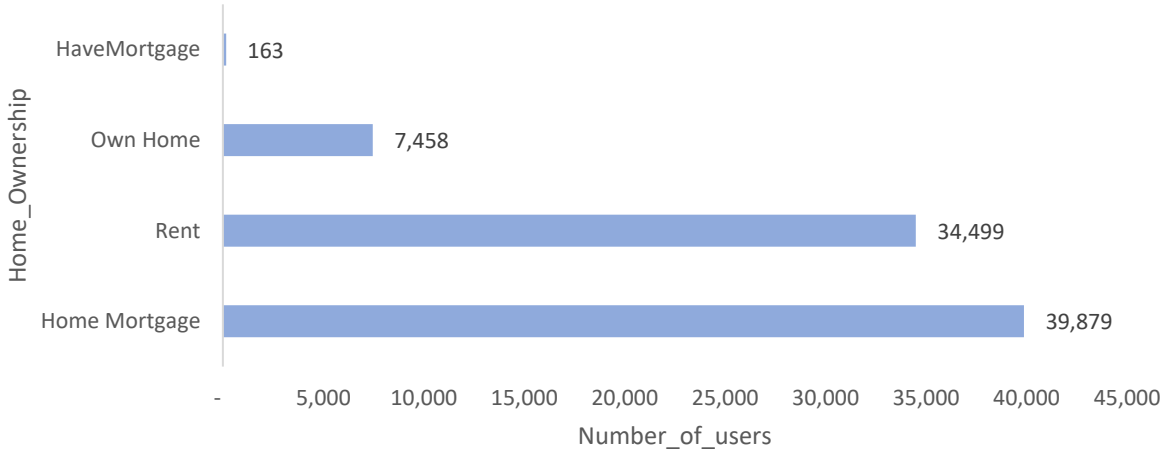
Number of borrowers are mainly come from the high end class, which has over $350K per year as their annual income.

```sql
--Years in current job group
SELECT Years_in_current_job, COUNT(DISTINCT(Customer_ID)) AS number_of_customers
FROM PortfolioProject.dbo.credit_train$
GROUP BY Years_in_current_job
```



Number of customers with years of experience in their current jobs.




HaveMortgage
163
Own Home
7,458
Rent
34,499
Home Mortgage
39,879
Home_Ownership
-
5,000
10,000
15,000
20,000
25,000
30,000
35,000
40,000
45,000
Number_of_users