# Lending Club Loan

## Team: 8

**Lending Club** is a US peer-to-peer lending company, headquartered in San Francisco, California.It was the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission (SEC), and to offer loan trading on a secondary market.

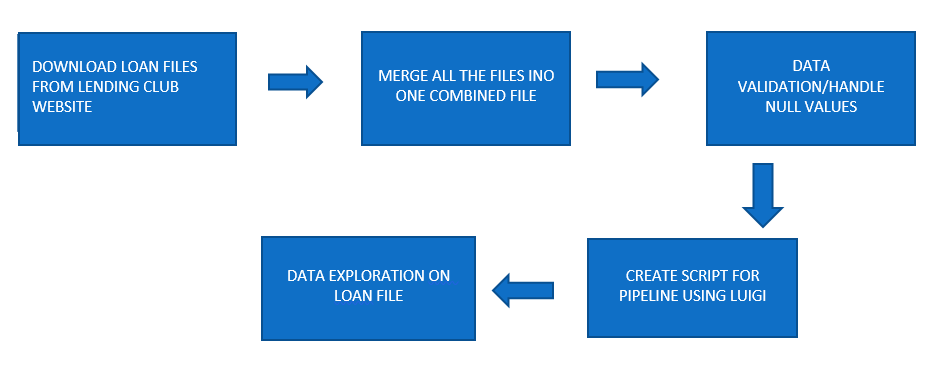
Lending Club enables borrowers to create unsecured personal loans between $1,000 and $40,000. The standard loan period is three years. Investors can search and browse the loan listings on Lending Club website and select loans that they want to invest in based on the information supplied about the borrower, amount of loan, loan grade, and loan purpose. Investors make money from interest. Lending Club makes money by charging borrowers an origination fee and investors a service fee.

**Problem Statement:**

You are working at a bank and you are considering investing in Lending club. Since there are no standard models, you are expected to build prediction models that will help you predict the interest rates based on various parameters users would input.

**Part 1: Data wrangling and exploratory data analysis:**

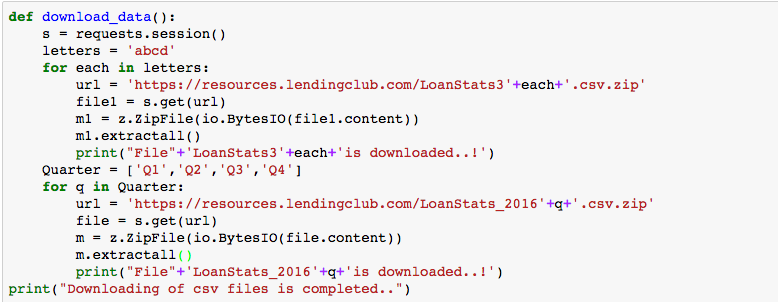
**Downloading and Merging the Files:**



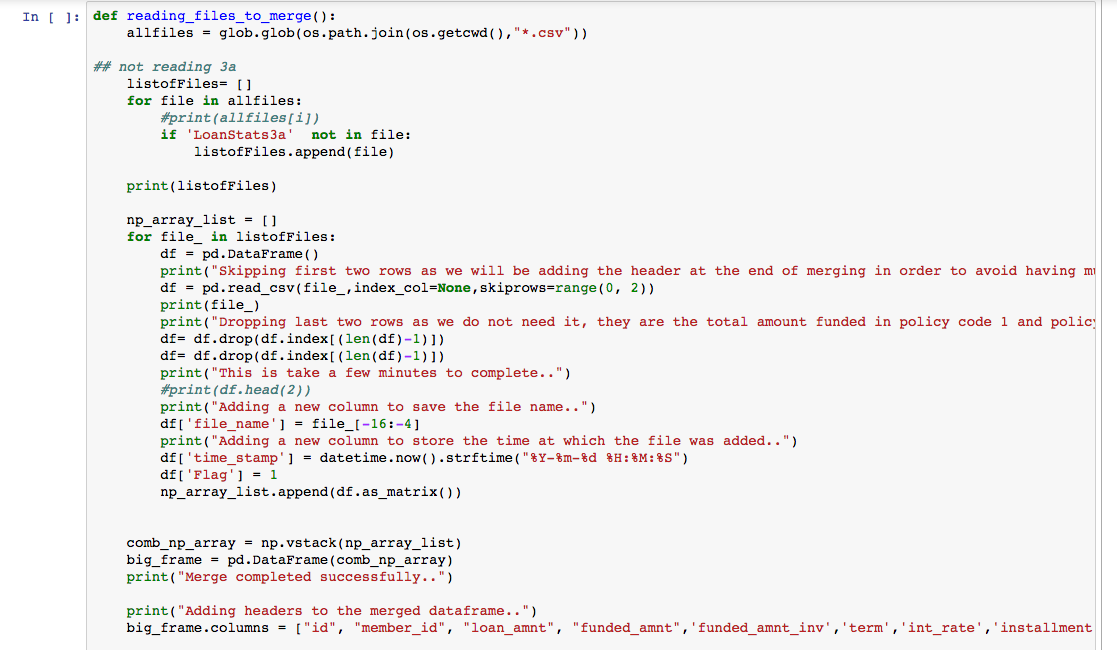
* Programmatically downloaded all the loan data csv files from <https://www.lendingclub.com/info/download-data.action>

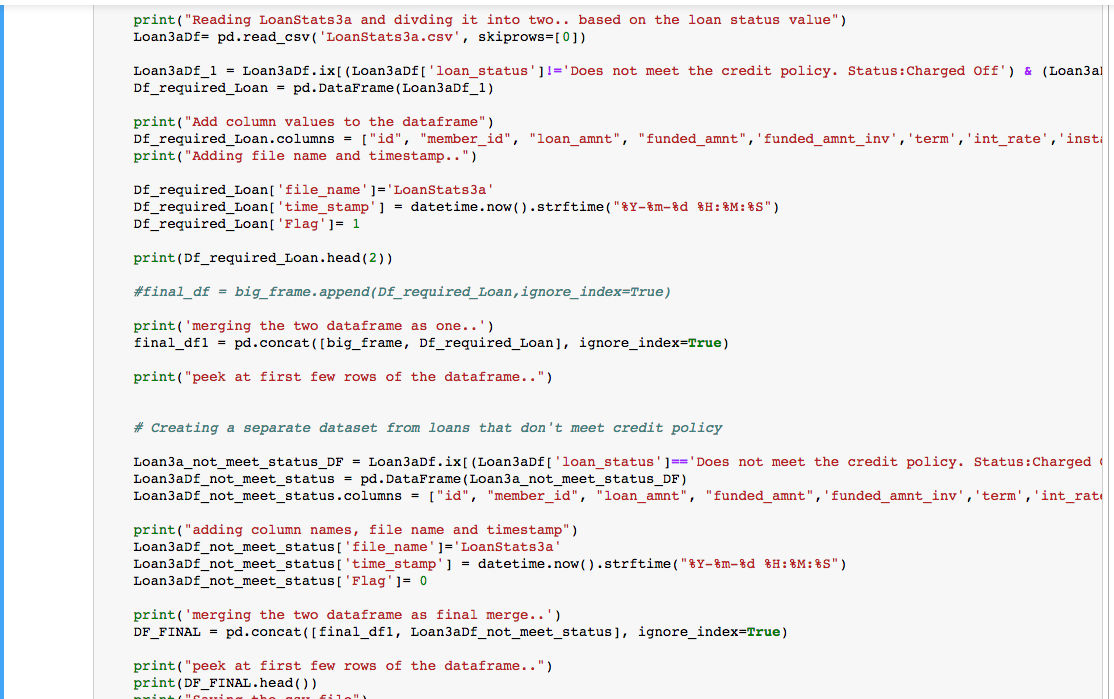
inside the docker container.

File on GitHub: [Merging\_The\_file\_To\_one\_Dataset.py](https://github.com/bajajsweta/Lending-Club-Analysis/blob/master/Loan_Data/Luigi%20Files%20for%20Loan%20Data/Merging_The_file_To_one_Dataset.py)

****

* Once all the csv files are downloaded, read all the files into a list except LoanStats3a as it has loans that did not meet the credit policy. We will read that separately and then merge it with the rest of the loan data.
* We are skipping the headers, row: ‘Notes offered by Prospectus (https://www.lendingclub.com/info/prospectus.action)’and , last two rows as they are not required and we will add header column after merging the data in order to avoid having duplicate headers in the file.
* While reading the files into the data frame as one, we added a **timestamp**, **Flag** and also the **filename** in order to when the data was recorded and belongs to which loan data file.



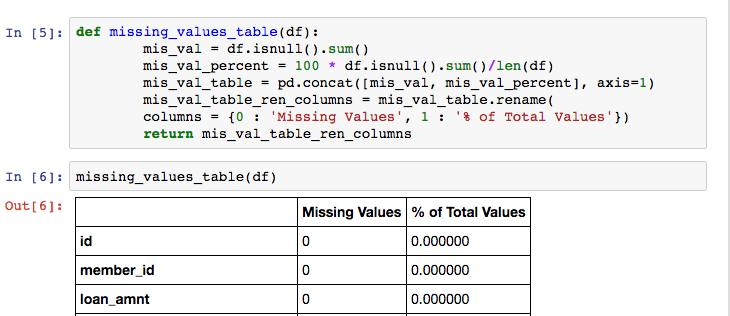
****

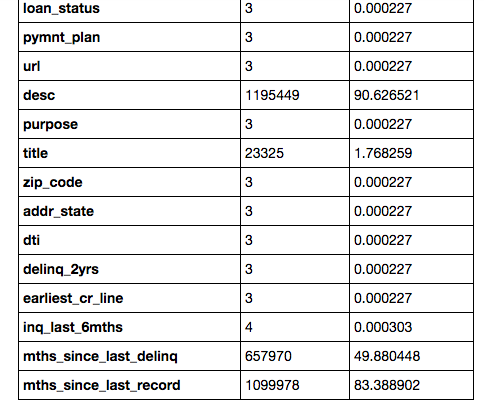
* Read LoanStats3a csv file and divided into by reading the loan status column. If the loan status column does not have values equal to **Does not meet the credit policy. Status: Charged Off** and **Does not meet the credit policy. Status: Fully Paid** then store the rows in a new data frame (Loan3aDf). And then we merge it with the rest of the files with flag as 1.
* And If the loan status column has values equal to **Does not meet the credit policy. Status: Charged Off** and **Does not meet the credit policy. Status: Fully Paid** then store the rows in a new data frame (Loan3a\_not\_meet\_status\_DF) and merged it with the rest of the data with flag as 0.
* At last we store it in a csv file(Combined\_data\_file.csv).

**Validation of dataset**

File in GitHub: [Validation\_Script.py](https://github.com/bajajsweta/Lending-Club-Analysis/blob/master/Loan_Data/Luigi%20Files%20for%20Loan%20Data/Validation_Script.py)

* Once a single dataset is created, we call the validate function to validate all the columns.
* After checking the percentage of null/nans in each column, we dropped a few columns and filled nulls with unknown in categorical columns and interpolation on a few columns.



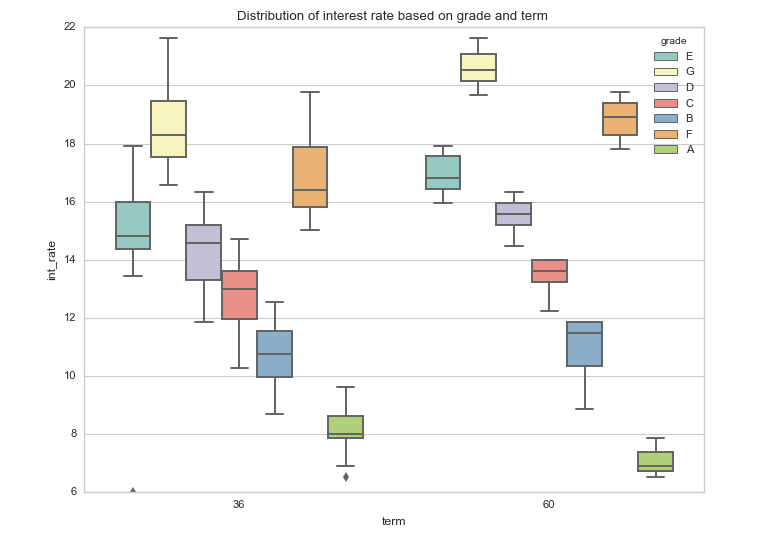


* We removed % from **interest rate** & **revolving utilization** and drop the word **months** and **year** from the columns term and employment respectively.

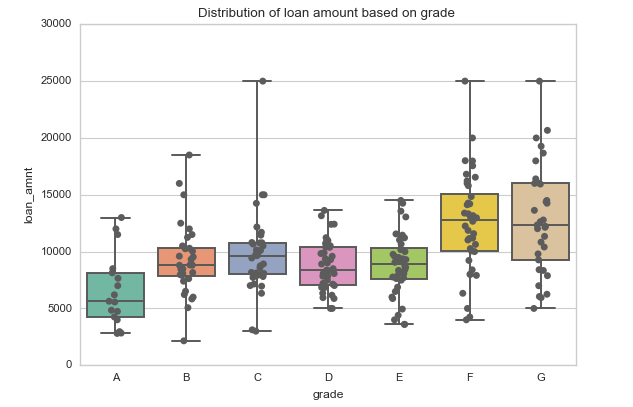


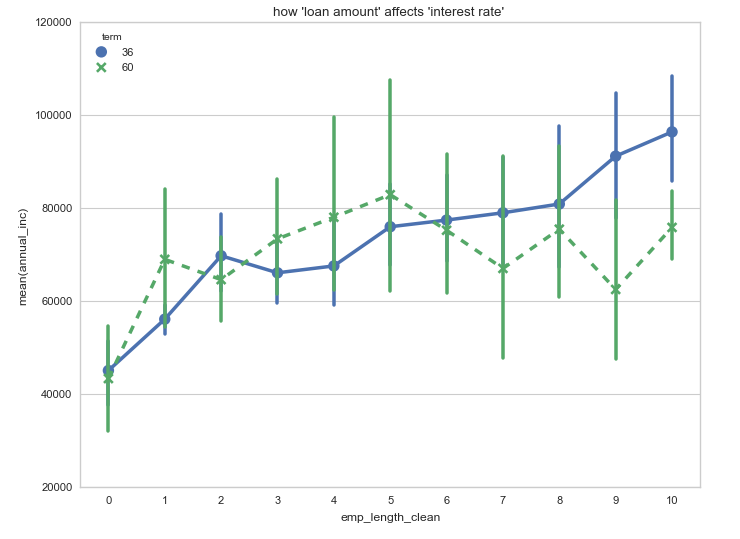
**Exploratory Analysis on loans data that did not meet credit policy**

**Interest rate plotted for all the grade type divided by term :**  
The interest rate is highest for grades (F, G) and lowest for A and B



**Plotted loan amount based on different grades (we inferred presence of outliers from the below boxplot)**





Annual income based on employment length

**Pipeline:**

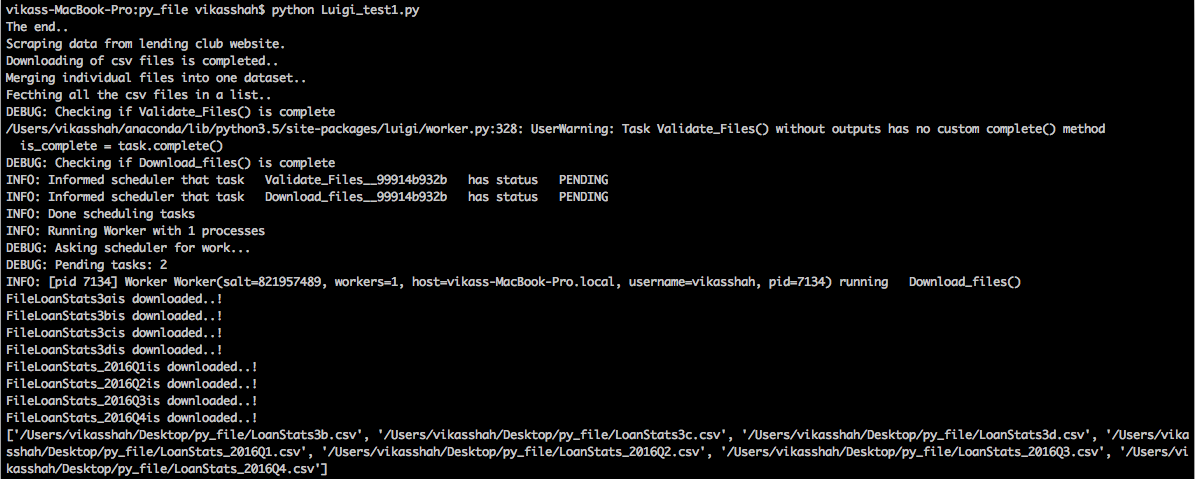
We used Luigi to schedule the task.

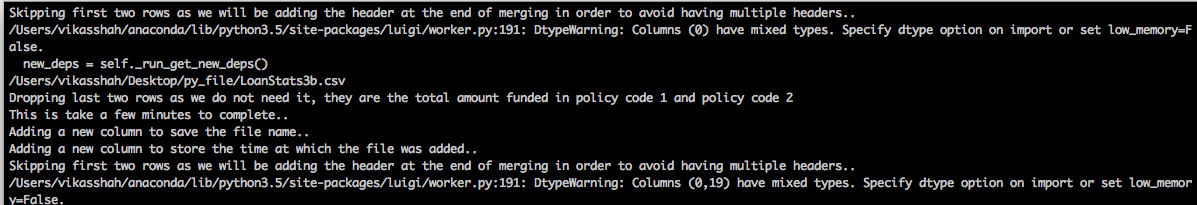
Steps to run Luigi:

To run loan data script: run the below command on the terminal:

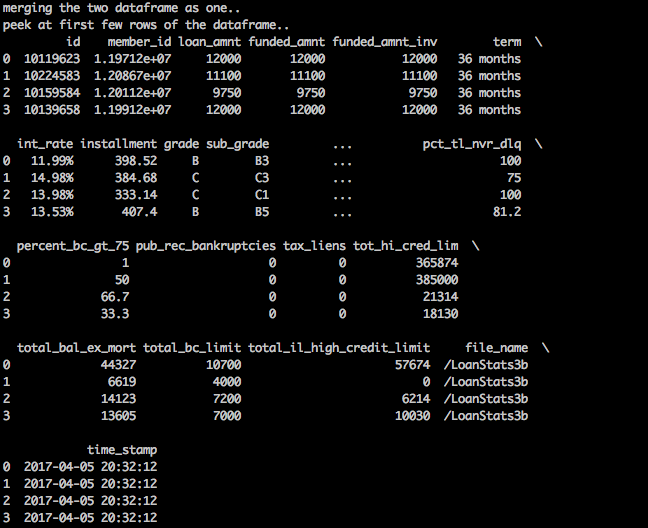
$ python [Luigi\_test1.py](https://github.com/bajajsweta/Lending-Club-Analysis/blob/master/Loan_Data/Luigi%20Files%20for%20Loan%20Data/Luigi_test1.py)

Screen%20Shot%202017-04-05%20at%208.36.24%20PM.png

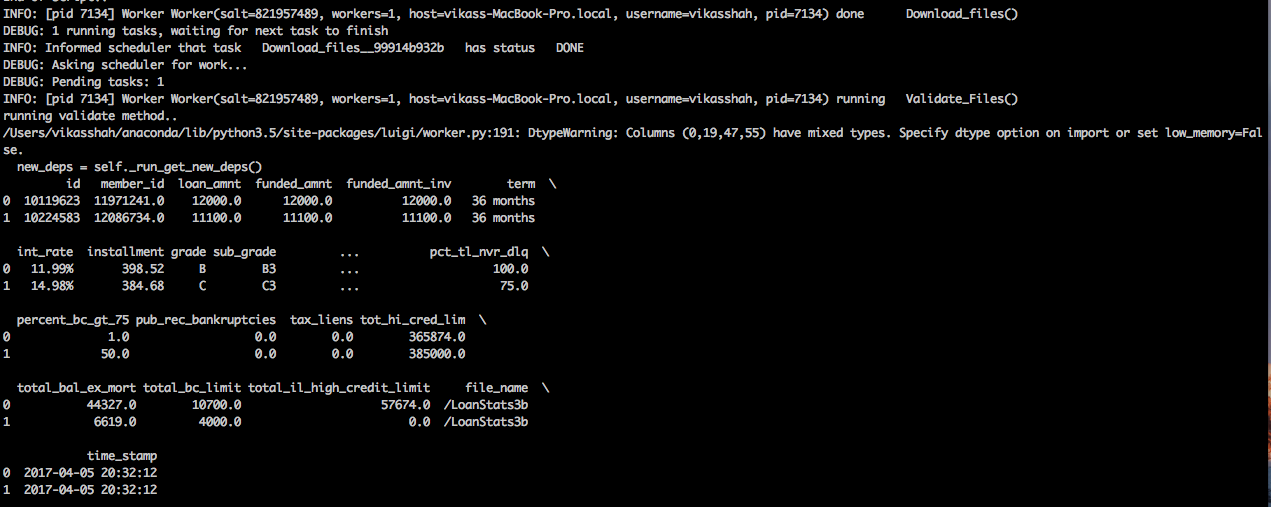




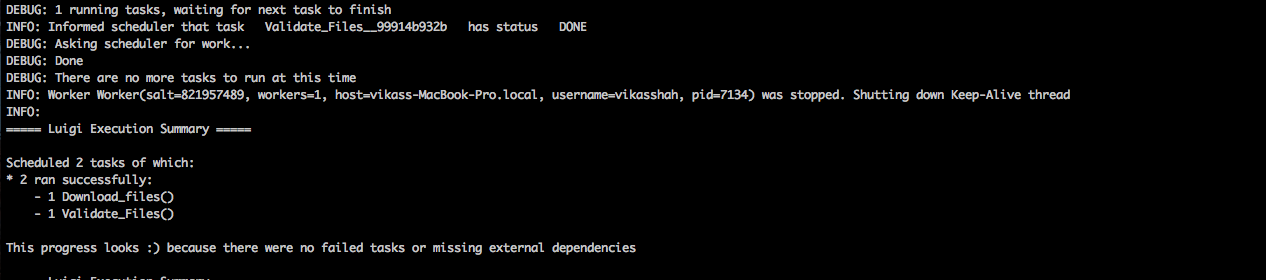
merging data with timestamp and the file name.



the job next runs the dependent job that is the validating the combined data file.



The task completes successfully!



**DECLINED LOAN FILE:**

**FLOW CHART OF THE STEPS PERFORMED:**

MERGE ALL THE FILES INO ONE COMBINED FILE

DATA VALIDATION/HANDLE NULL VALUES

DOWNLOAD FILES FROM LENDING CLUB WEBSITE



DATA EXPLORATION ON DECLINED LOAN FILE

CREATE SCRIPT FOR PIPELINE USING LUIGI

1. **DATA DOWNLOAD AND MERGING**:

**Data for declined loan set from lending club was downloaded for the following years**:

Path URL: https://www.lendingclub.com/info/download-data.action

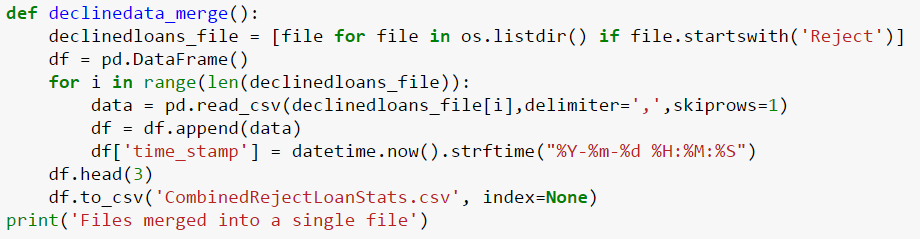
Total number of Files: 7

1. 2007 – 2012
2. 2013 – 2014
3. 2015
4. 2016Q1
5. 2016Q2
6. 2016Q3
7. 2016Q4

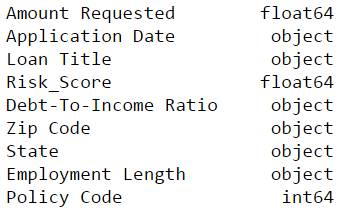


Session was created and all the files were downloaded into two steps.  
First, the zip files from 2007 to 2015 were downloaded using the above script and then extracted to get the required files.  
Same step was performed for files from 2016 for all the quarters.

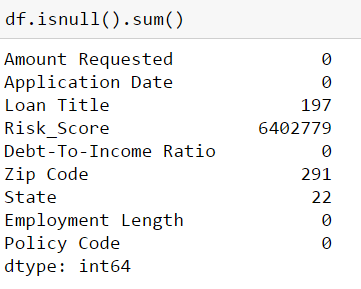
**Downloaded files from all years were then merged into single csv file:**



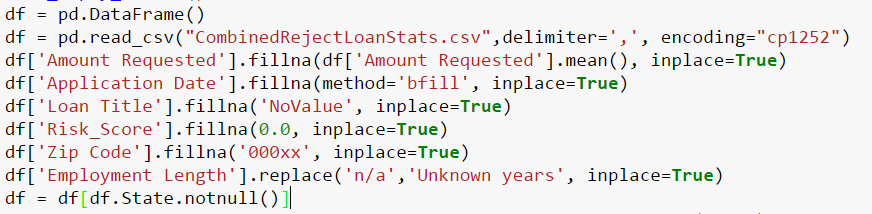
1. First row was removed from every file (which just contained the header file)
2. Timestamp was added for when the file was downloaded.
3. All the files were combined into a single csv file for further analysis.
4. **DATA VALIDATION/HANDLING MISSING VALUES:**
5. Load combined file created by the above step.
6. Check for the number and type of columns in the dataset  
     
   Number of columns: 9



1. Check the number of null values in each column



1. **Handle missing values:**

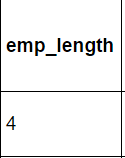
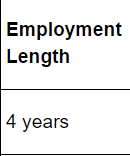


**Steps taken to handle missing values**:

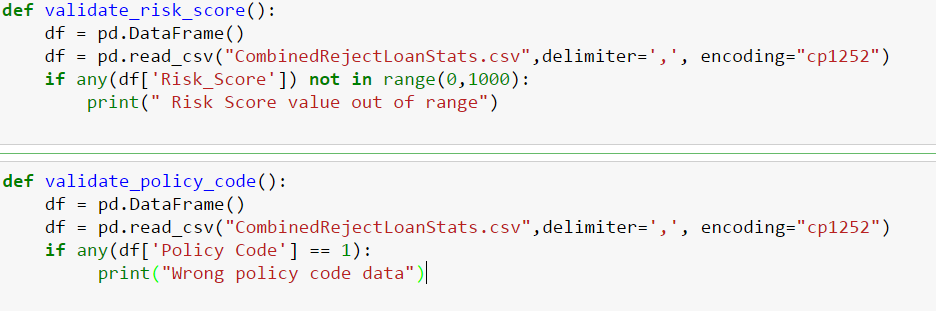
1. Replace empty values in numerical fields with mean of the value. (Amount Requested)
2. For continuous column like date use function like bfill to maintain continuous column (Application date)
3. For text column replaced empty values with text like “NoValue”. (Loan Title)
4. Replace N/A with unknown years in employment length
5. Deleted all the column where state was null (Number was 22), as the columns where state was null, didn’t have values in most of the other column. (State)
6. Deleted rows where more than 3 columns have null values.
7. **Cleaning the data**

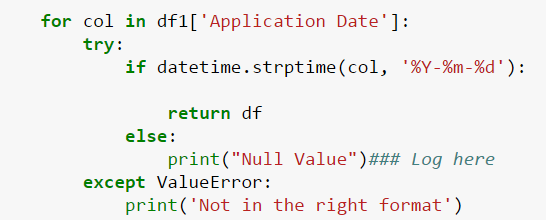


1. Created new column (emp\_length) having only numeric values by removing character value from the column.
2. Changed the data type of the column to numeric.
3. Created new column Application year by extracting only the year from the data column for ease of summary.
4. Removed % from debt to income ratio and changed to numeric for easier analysis and further model creation.

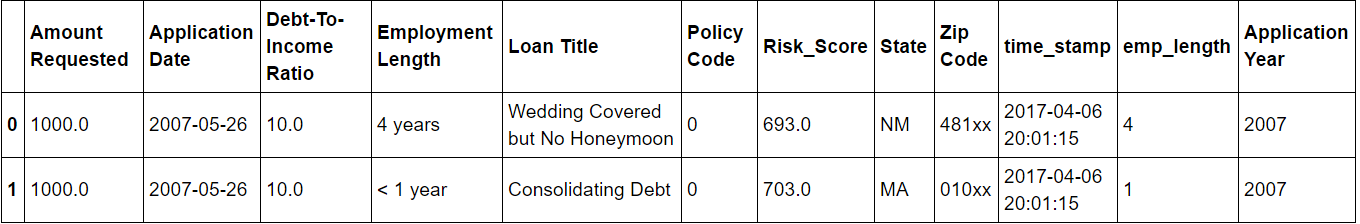


1. **Checking errors in data**





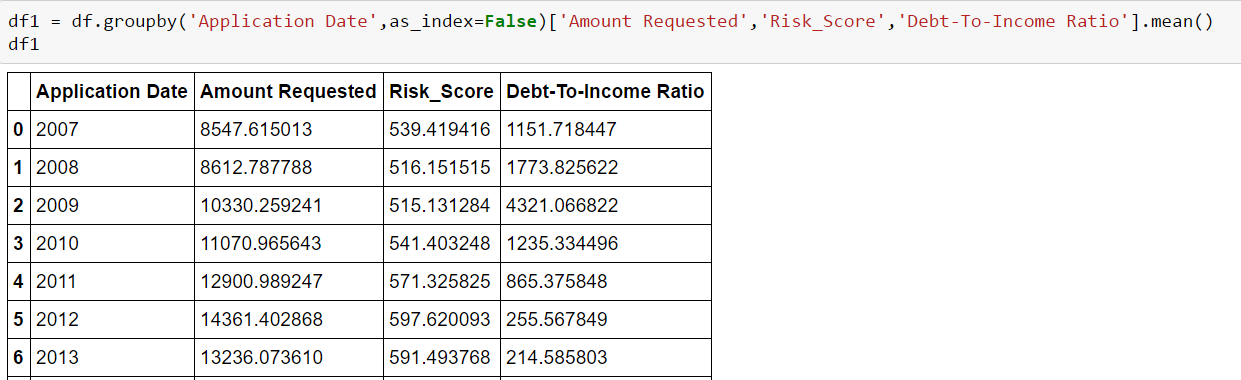
1. Check risk score is between 0 and 1000 so that no erroneous risk score is present
2. As it is declined loan data, policy code must not be 1, Check if any data is present which is not a declined loan.
3. Check for date syntax.
4. **Look of the data after validation:**



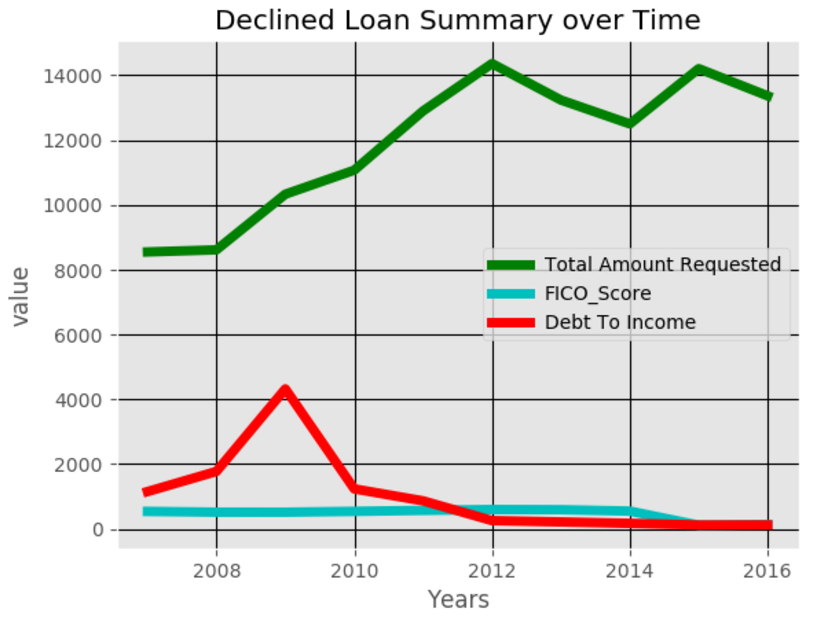
**DATA EXPLORATION:**

1. **DECLINED LOAN FILE:**
2. **Analysis of declined loan file over time (for each year):**

Columns taken: Risk Score, Amount Requested, Debt to income ratio.



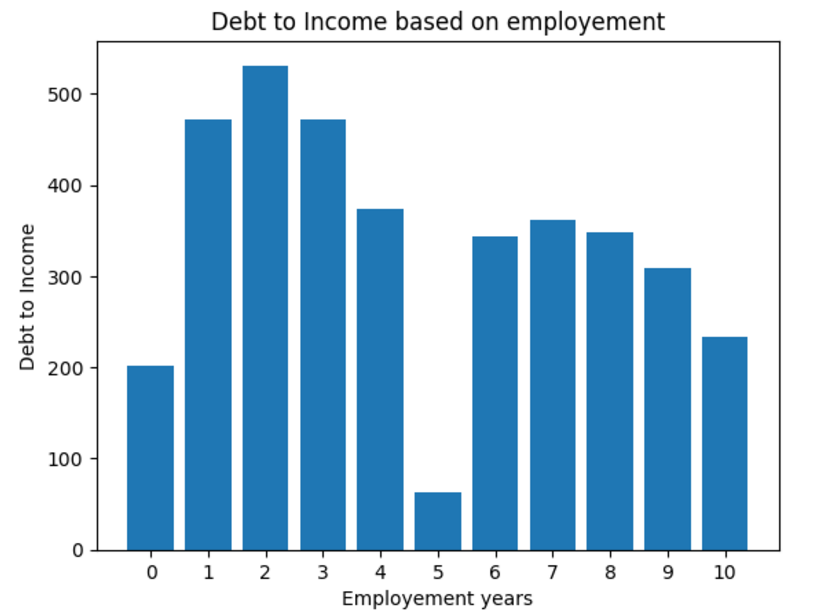


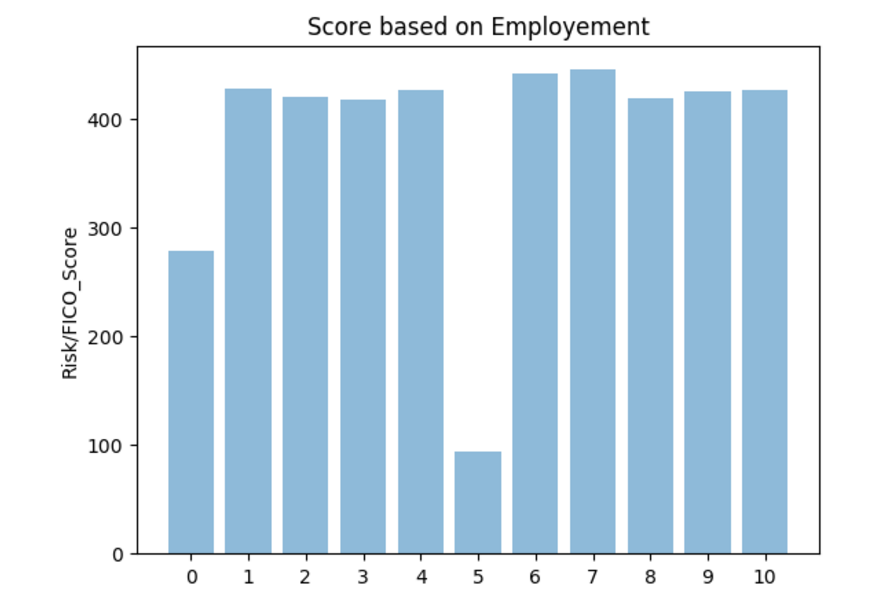


**Insights:**

1)The Debt to income ratio is highest for years around 2009 (economic slowdown year)  
2) Amount requested for loans was the lowest for years around 2007-2008 (slowdown yrs)   
3) There is a gradual increase In the number of declined loans with highest for 2016.

1. **Analysis of declined loan data for applicant’s number of employment years:**



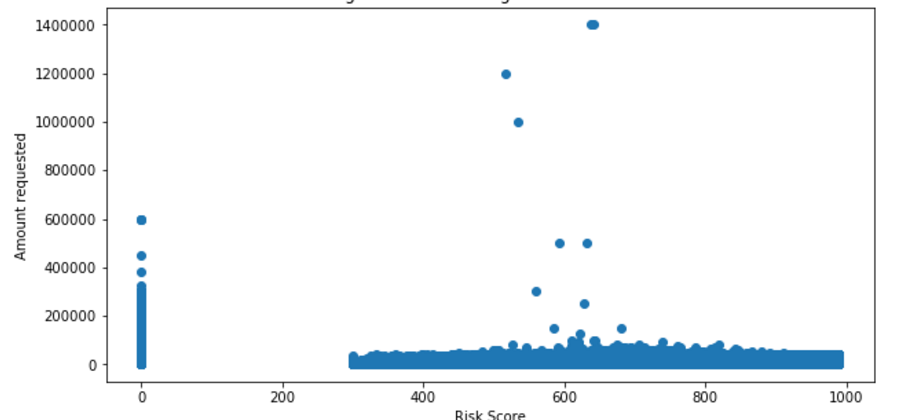


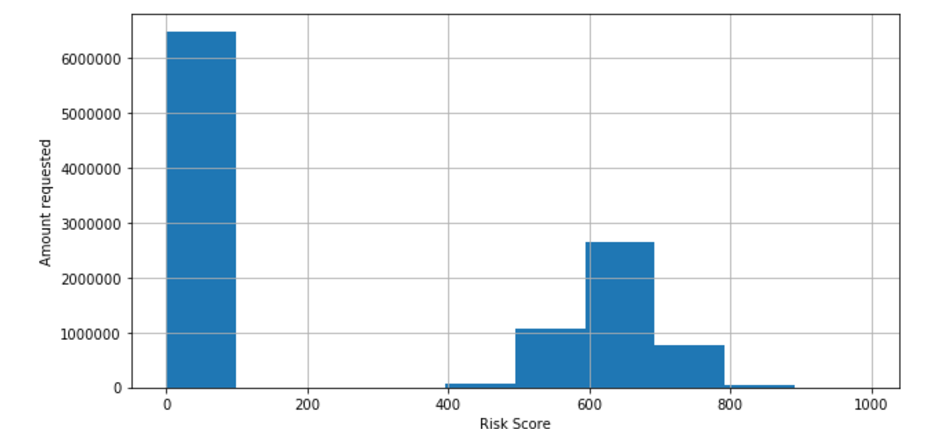
**Insights:**

The debt to income ratio is low for applicants with no employment years (0)  
The debt to income ratio is highest for applicants with employment years (1-3)  
 Very high loans were provided to applicants whose employment years were low , hence the   
 high debt to income ratio.

1. **Analysis of Amount requested for loan based on risk score (Fico/Vantage):**

We performed a scatter plot to see the concentration of amount requested for loan with respect to the risk/fico score

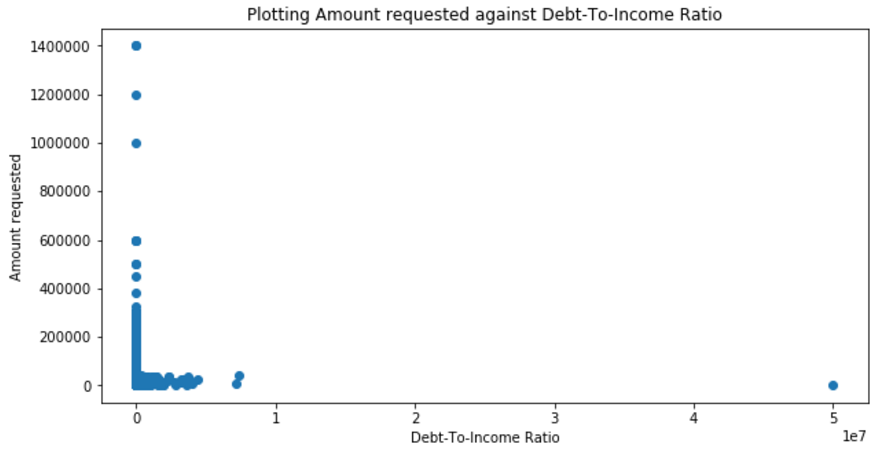




**Insights:**

1)We see some outliers for the amount requested that have a very high value to the average 2) We also inferred that most of the loans requested had fico/vantage score between 600-700  
Only a few loans were declined for score over 800

1. **Analysis of Amount requested with respect to Debt to income ratio:**

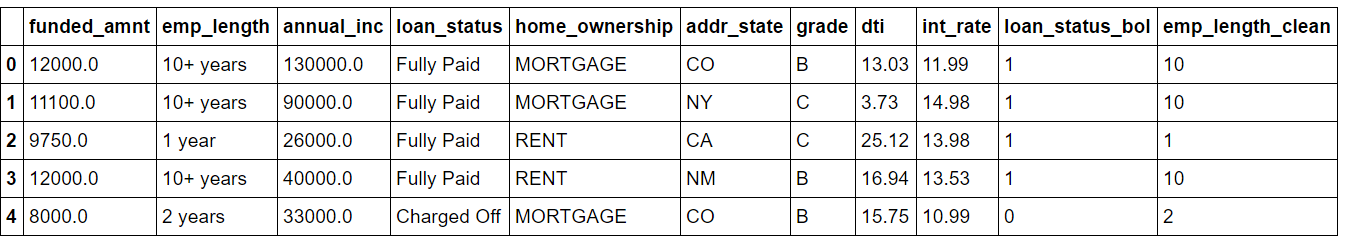


**Insights:**

1. Few outliers are visible which refer to incorrect values. Most of the deb to income ratio is scattered between 0 to 5000
2. **DATA EXPLORATION ON LOAN DATA FILE:**

**Note: As the loan data file, had considerably high number of columns, we only took the columns that we considered necessary or important for our data exploration**

**Columns chosen for analysis:**

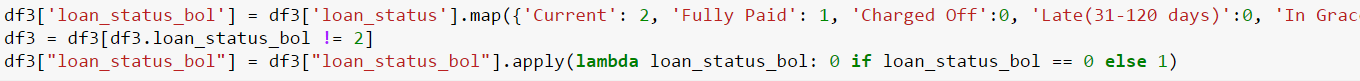


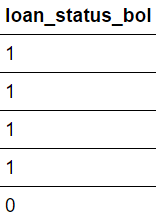
1. Funded Loan Amount
2. Employee employment length
3. Annual income of the requestor
4. Loan status
5. Home Ownership of the requestor
6. Address State
7. Grade
8. Debt to income ratio
9. Interest Rate
10. Status of Loan (Derived Attribute)
11. Numerical clean employment length

**Derived Column:**

We created a derived column from Loan status which was a text column and created a Boolean column wherein we put Fully Paid loans as 1 and those not paid as 0.

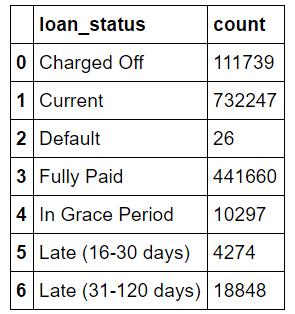
We assigned a number to the different loan statuses with Fully Paid as 1 and others set to 0.  
We did not consider the current loans.

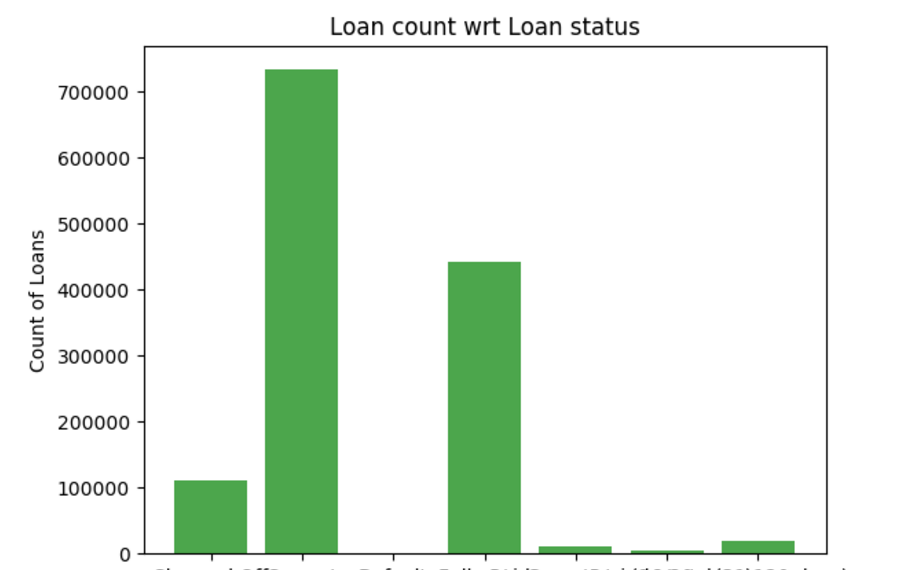




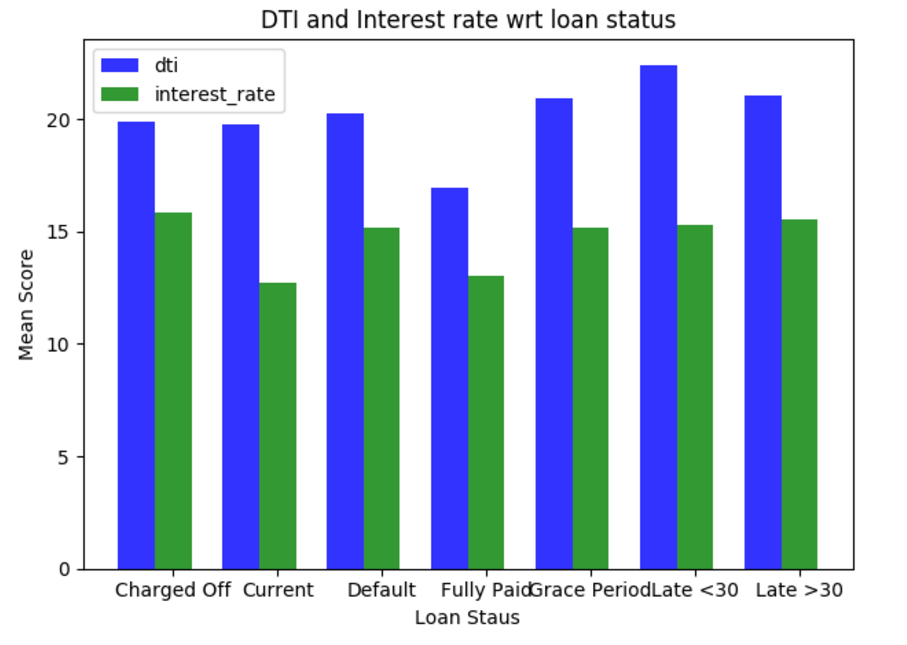
1. **ANALYSIS OF LOAN DATA SET WITH RESPECT TO LOAN STATUS:**

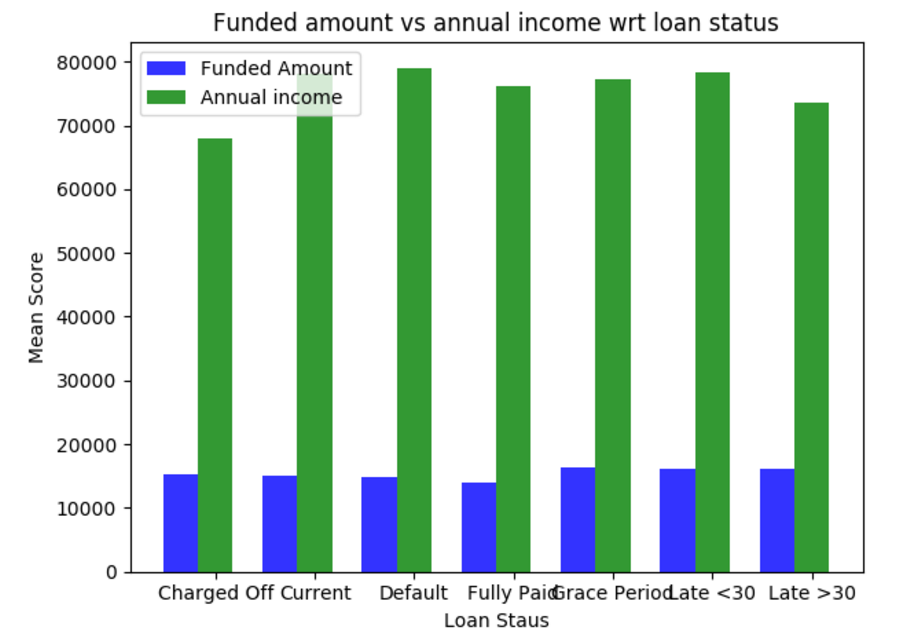
**Count of number of loans with the current loan status**





**Analyzing DTI and interest rate with respect to each loan status:**



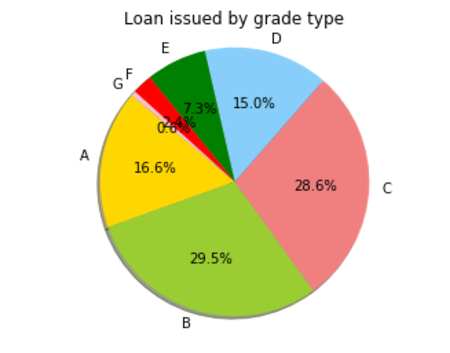


**Insights:**

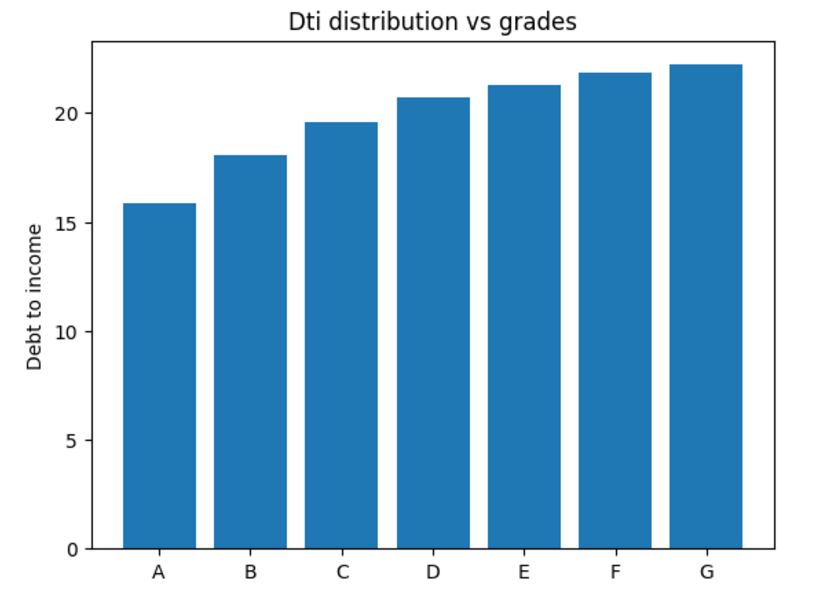
The debt to Income ratio was lowest for loans where the loans were fully paid   
It was highest for loans where the loan status was late  
 Interest rate was considerably higher for defaulted and charged off loans.  
 Interest rate seems lowest for current loans

1. **ANALYSIS OF LOAN DATA WITH RESPECT TO GRADES:**

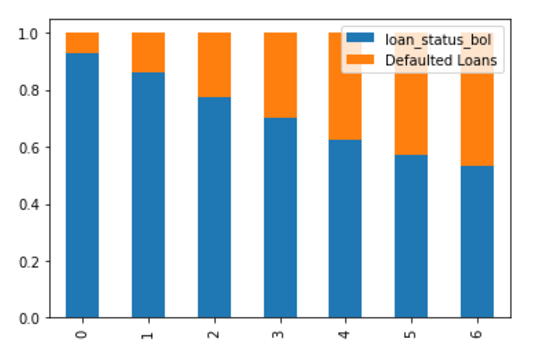
**Percentage of Loan issued in respect to grade types**



**Debt to income ratio pattern with respect to grade type**



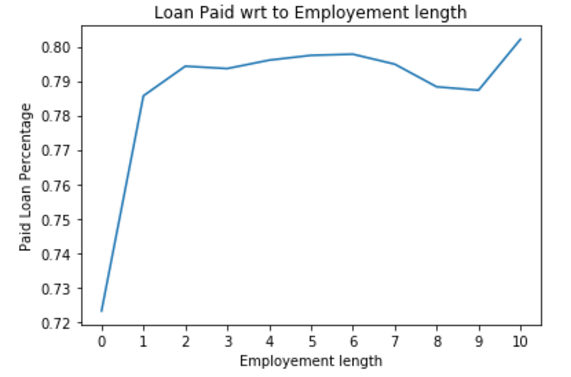
**Percentage of Paid Loans vs defaulted Loans for each grade type**



**Insights:**

The paid loans percentage and debt to income ratio were lowest for Grade A loan type.  
The quality of loans decreases as we go from Grade A to G  
Most of the loans are clustered between Grade A to C (approx. 75%)

1. **ANALYSIS OF LOAN DATA BASED ON REQUESTER’S EMPLOYEMENT LENGTH**



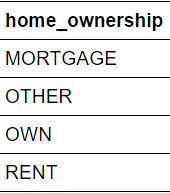


**INSIGHTS:**

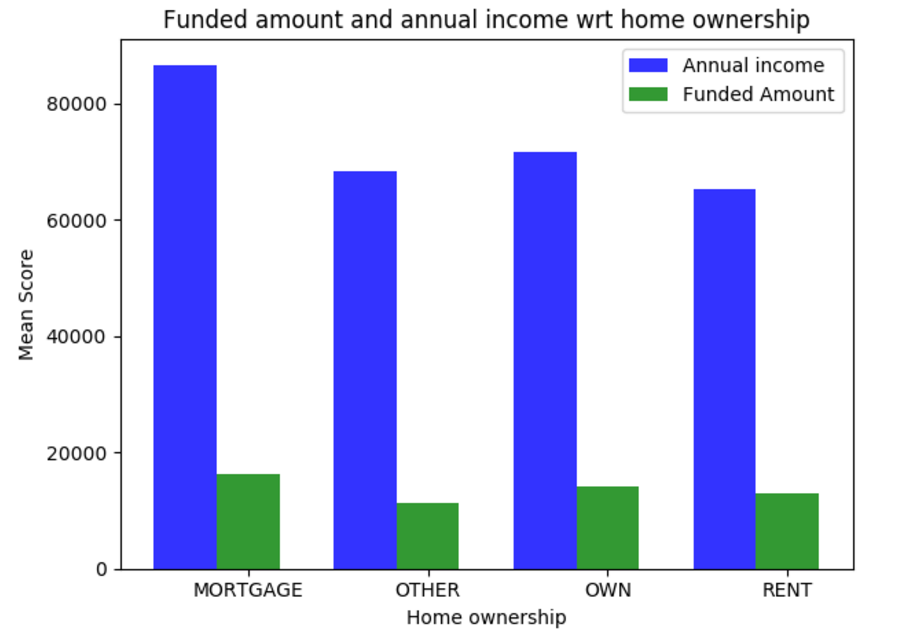
The paid Loan percentage is lowest for requestor’s who are unemployed  
 The pain load percentage gradually increases except for years 8-9  
 As expected unemployed loan has highest debt to income ratio.

1. **ANALYSIS OF LOAN DATA WITH RESPECT TO HOMEOWNER TYPE:**

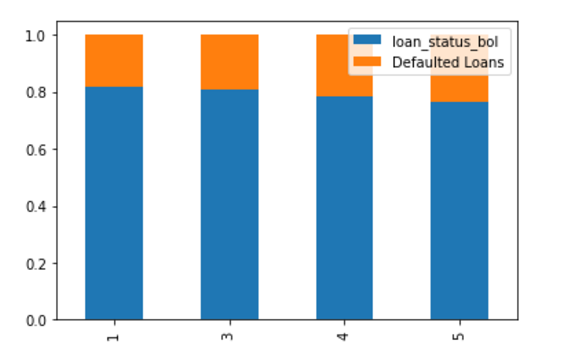
**Note: Here we are only considering 4 homeowner types**



**Funded amount vs annual income for every home ownership type.**

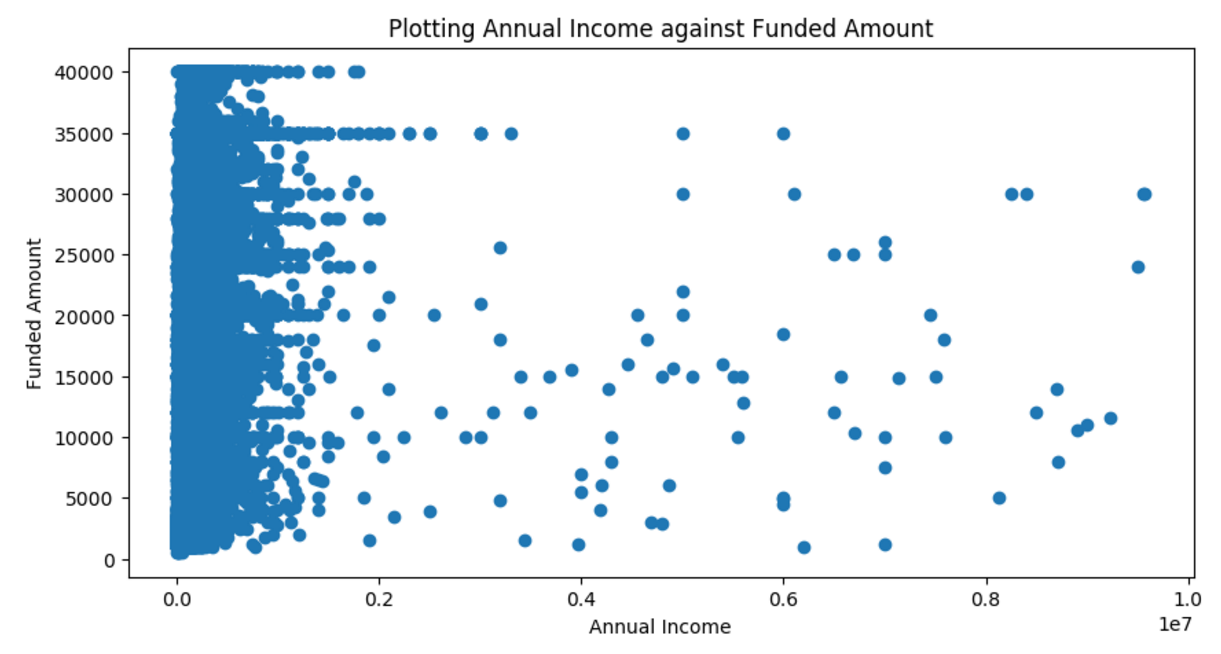


**Paid Loan percentage vs defaulted for different home owner types:**

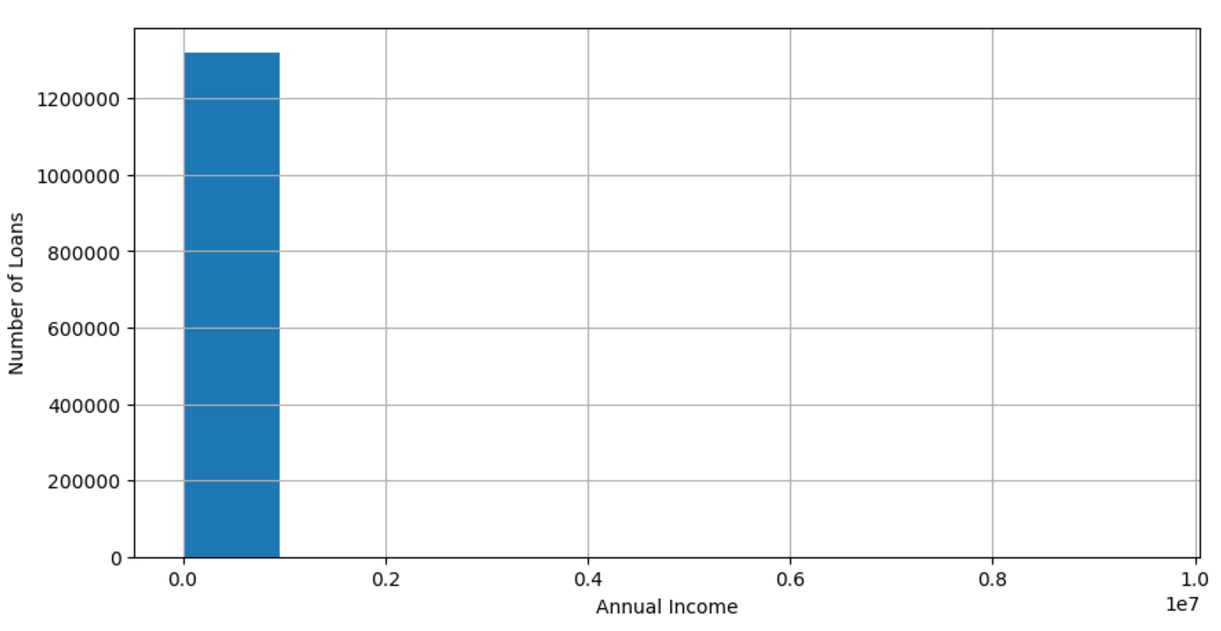


1. **ANALYSIS BASED ON ANNUAL INCOME:**

We first plotted a scatter plot between annual income and funded income to see the correlation between the two.

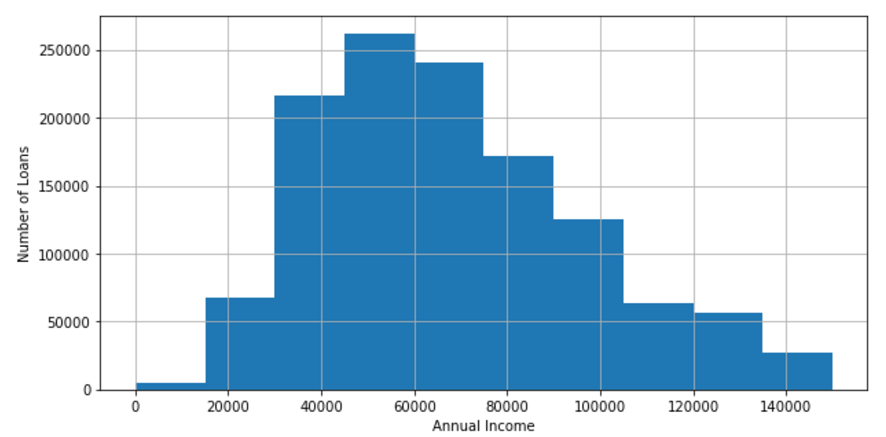


We can see that most of the annual income of people was around 150000, and we can see a lot of outliers for higher value of annual income.



**So taking Annual income only below 150000 to reduce the irrelevant records**

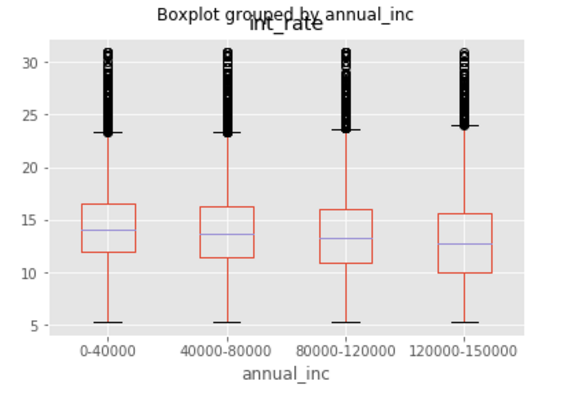
**Number of loans with respect to the annual income:**



**INSIGHT:**

The highest number of loans provided were for range 50000-75000

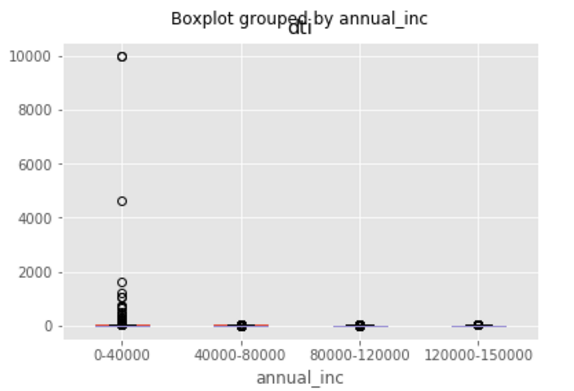
**Interest rate comparison for annual income:**



**INSIGHTS:**

Interest rate is lower for higher annual income, but that’s not the only criteria for interest rate as we see a lot of outliers.

**DTI comparison for annual income**



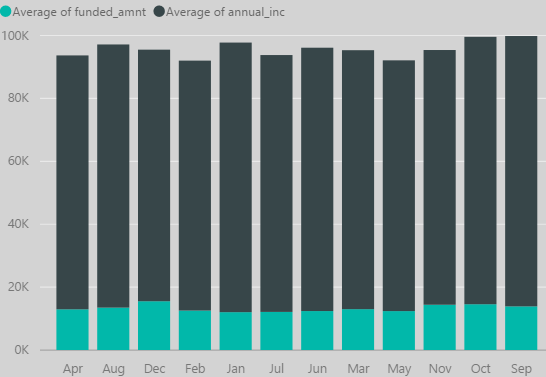
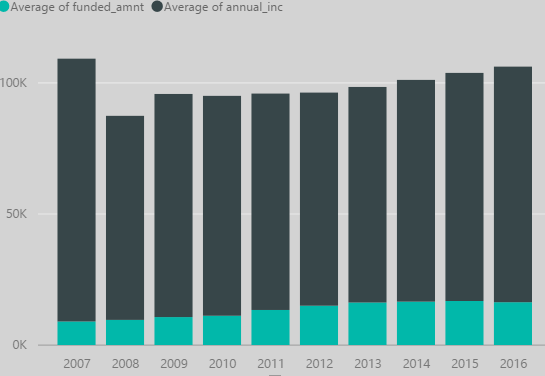
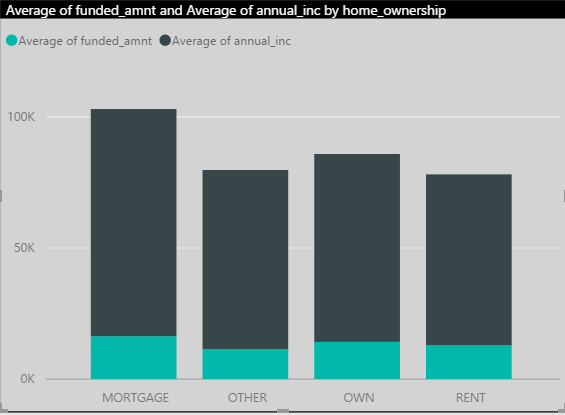
**INSIGHT:**

There are no visible outliers for higher annual income bracket but see some outliers for lower income bracket.

**POWER BI ANALYSIS:**

1. **Time Series Analysis of Loan file:**

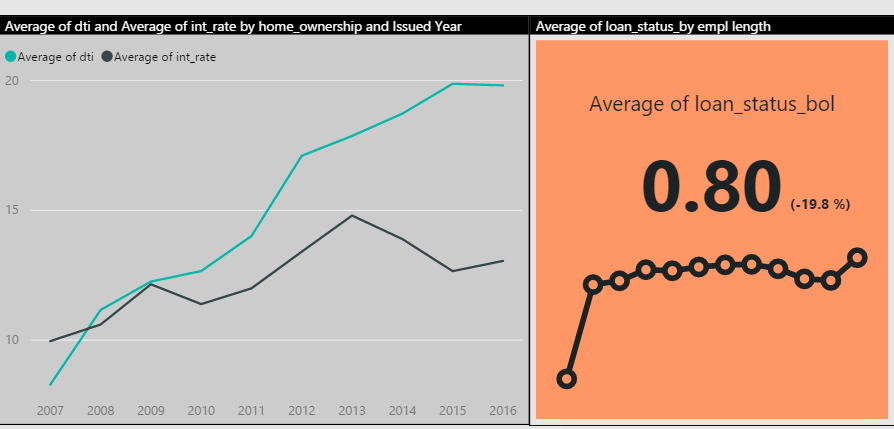
**Looking at funded amount vs annual income by home ownership by year and month:**



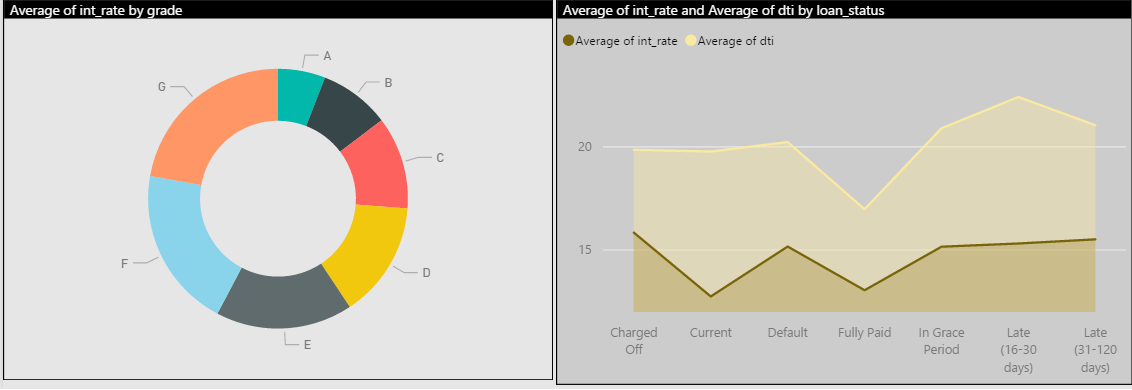
DRILLED DOWN TO YEAR AND MONTH FOR EVERY TYPE OF HOME OWNERSHIP

**Debt to income ratio and interest rate plotted for every year/month**

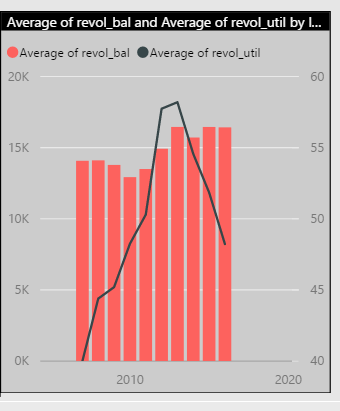
**Paid Loan vs Defaulted Loan ration by employment length**



**Interest rate analysis for every grade and every loan status**



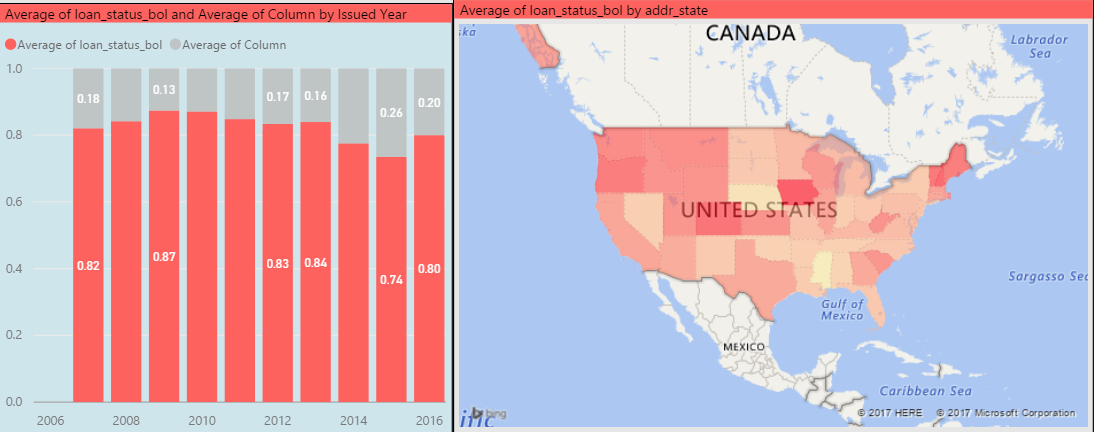
**Total credit balance vs percentage of credit used by customers**



**INSIGHTS:**

1. The credit percentage used was maximum was for years 2012. Shows heavy purchases and spending as maximum credit is used.
2. The credit provided to customer has increased since slowdown
3. Interest rate is lowest for the loan which have been paid and higher for loans not paid.
4. Debt to income ratio is lowest for paid loans and higher for loans not paid .
5. Paid Loan percentage is highest for customers employed for more years, highest for 10 years
6. **State wise analysis of loan data:**

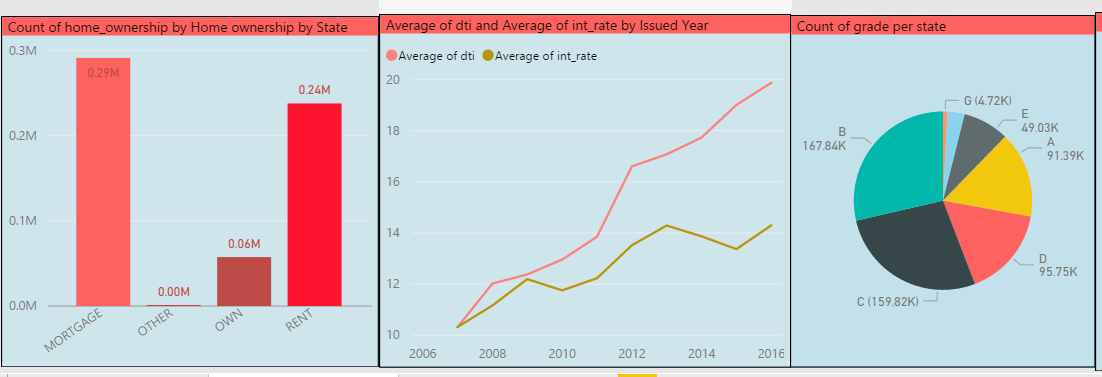
**State wise analysis of paid loans vs defaulted loans analyzed over time**



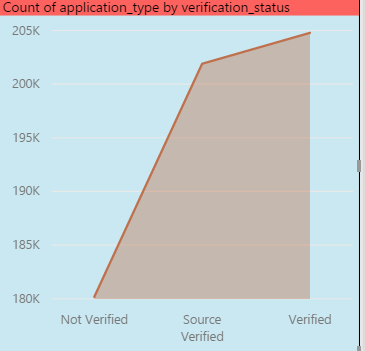
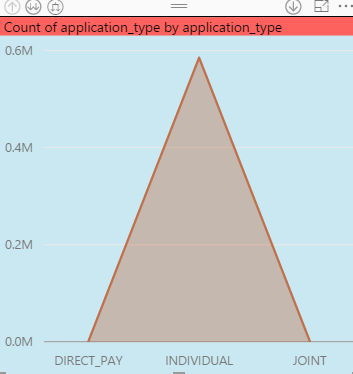
**Count of home ownership for every state analyzed by time**

**Interest rate and debt to income ratio over time**

**Count of grade loans for every state**



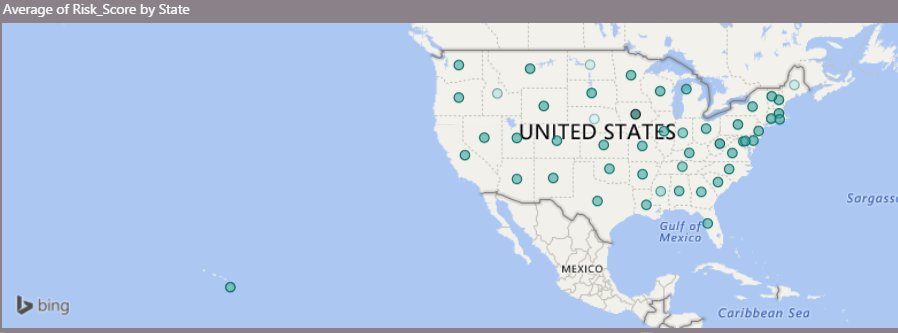
**Count of application type and checking whether verified for every application type**



**INSIGHTS:**

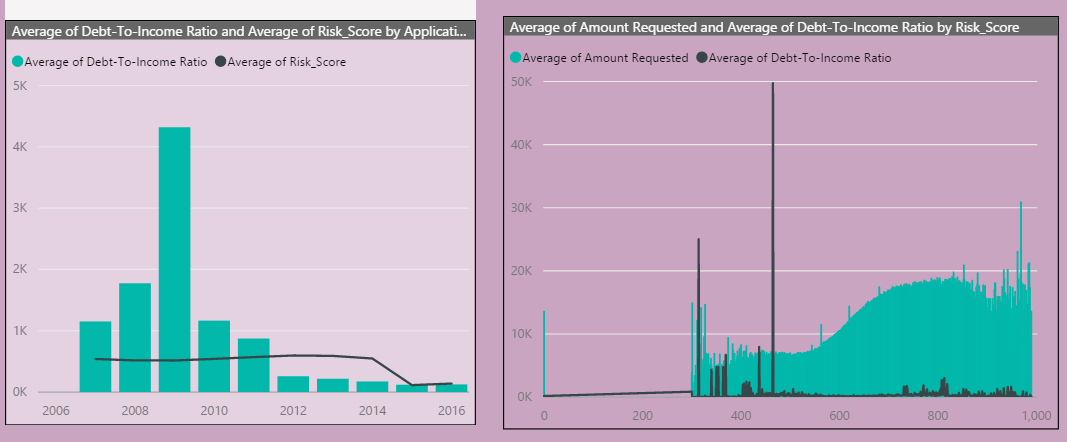
1. Washington DC has the best paid percentage of loans, whereas Mississippi has the lowest paid percentage of loans.
2. Iowa also has the lowest interest rate which might explain the highest paid loan percentage.
3. North Dakota has the highest interest rate but not the lowest paid percentage.
4. New Mexico has the highest debt to income ratio for all the states whereas Washington DC has the lowest debt to income ratio.
5. **Statewise analysis of Declined loan data:**

**State wise description based on risk score (fico/vantage score)**



**Plotting debt to income ratio against the risk score over time to check dependency between debt to income and risk score for declined files.**

**Plotting of debt to income ratio and loan amount requested over time to see dependency of debt to income and loan amount on risk score.**



**INSIGHTS:**

1. There is no visible dependency between risk score and debt to income ratio for declined loans
2. The highest number of failed loans are for the risk score around 500-700.
3. Debt to income ratio is considerably high for higher risk score
4. People with lower risk score have higher debt to income ratio.
5. Iowa state has the highest fico/vantage score and least default values.

**FEATURE ENGINEERING ON LOAN DATA FILE:**

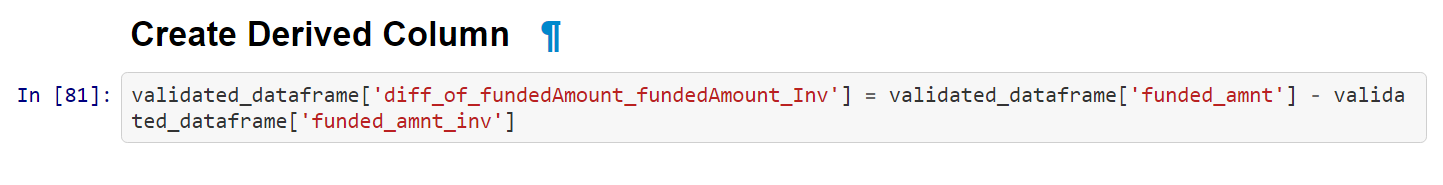
Feature engineering is the process of using domain knowledge of the data to create [features](https://en.wikipedia.org/wiki/Feature_(machine_learning)) that make [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The need for manual feature engineering can be obviated by automated [feature learning](https://en.wikipedia.org/wiki/Feature_learning).

**Steps taken for feature engineering:**

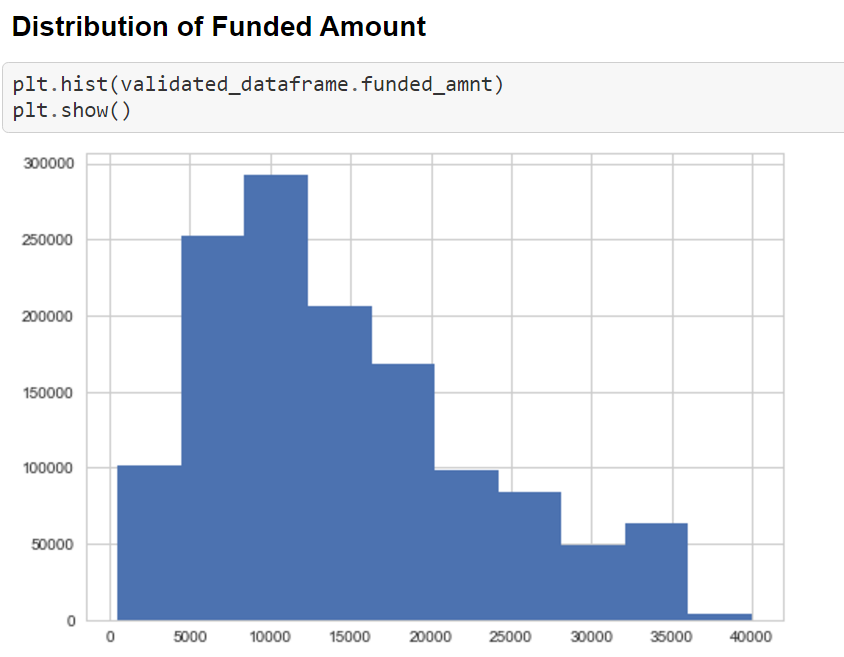
1. **Removed the columns with null value greater than 65% column**

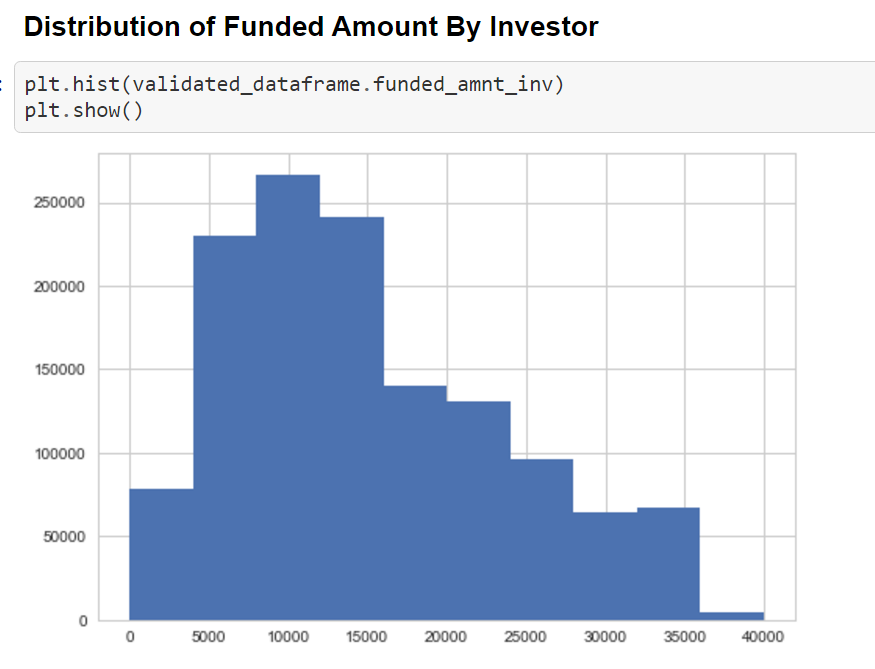


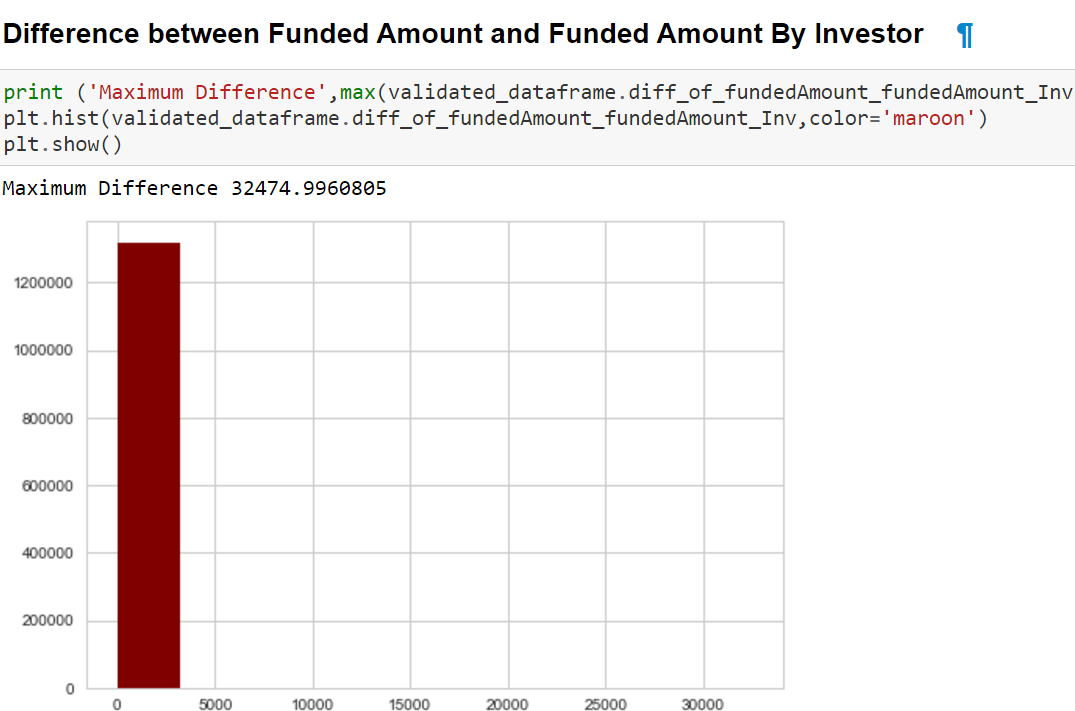
1. **Created Derived Attributes:**



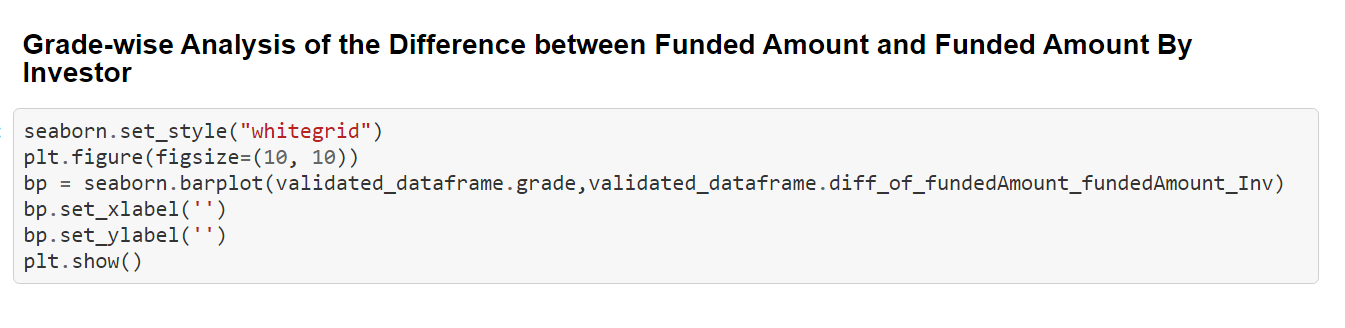
1. **Created histogram to check the range of the data:**

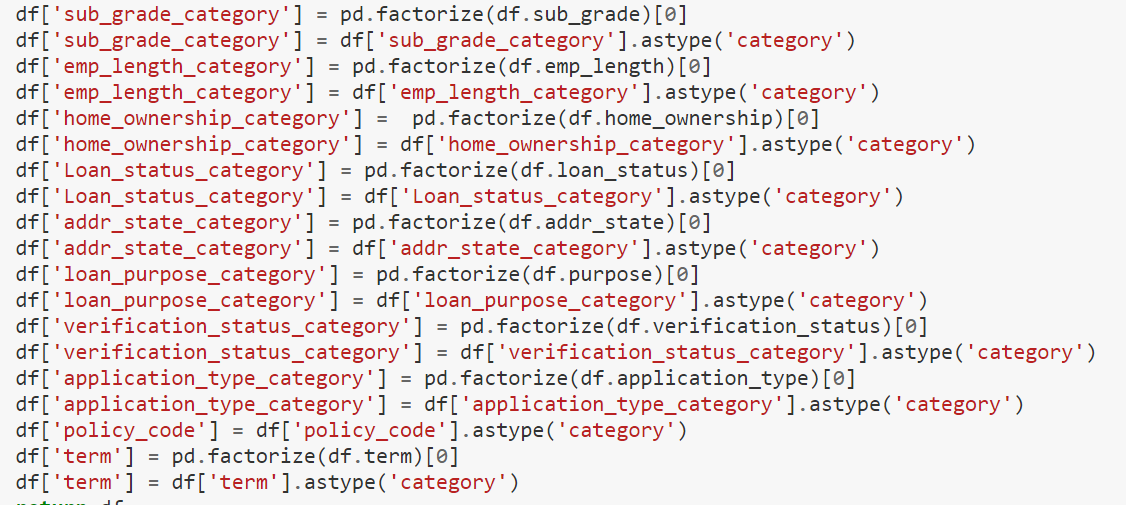


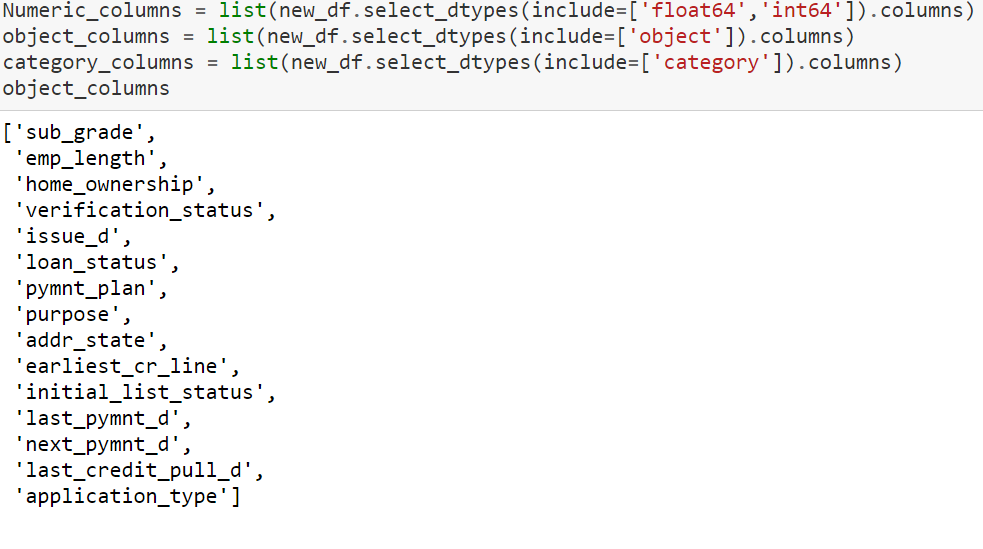




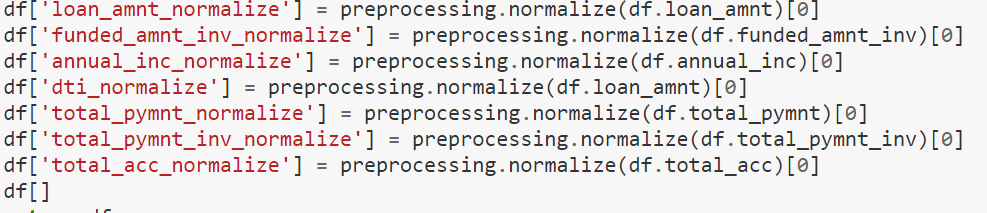
1. **Change the datatype to category for categorical columns**



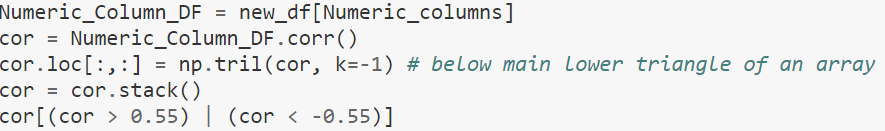


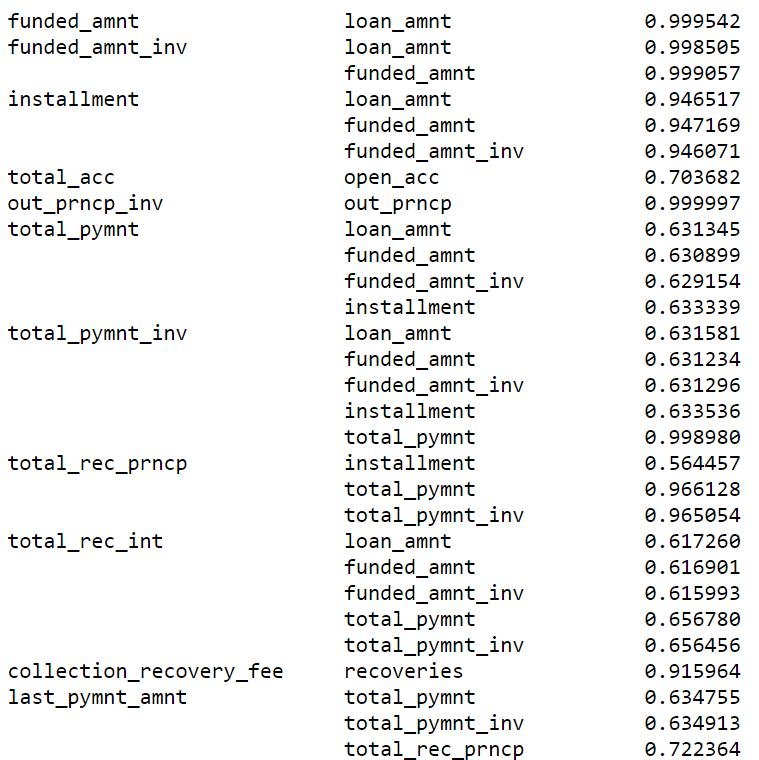


1. **Normalize the high scale columns:**



1. **Check the co -variance for the numeric columns to see the correlated columns for removal**





**Note: Fields with high co variance will be deleted when we create data for modelling.**