PRIYANJNA SHARMA

SUMIT DEO

GAUTAM PAWAR

ADS Assignment – 2

TEAM 2

Table of Contents

[2 Docker 2](#_Toc479366706)

[3 Running Pipeline on Cloud 3](#_Toc479366707)

[1: Create an EC2 Instance: We should create an Amazon EC2 Instance to run on cloud where the application can be hosted. 3](#_Toc479366708)

[4 Pipelining – Luigi 5](#_Toc479366709)

[4.1 Loan Data Flow 6](#_Toc479366710)

[4.2 Decline Data Flow 6](#_Toc479366711)

[5 Data Download 7](#_Toc479366712)

[5.1 Downloading Loan Data 7](#_Toc479366713)

[5.2 Downloading Decline Data 7](#_Toc479366714)

[6 Data Preprocessing And Feature Engineering 8](#_Toc479366715)

[6.1 Loan Data 8](#_Toc479366716)

[6.1.1 Preprocessing 8](#_Toc479366717)

[6.1.2 Derived Columns 8](#_Toc479366718)

[6.1.3 Storing on S3 8](#_Toc479366719)

[6.2 Declined Loans Data 9](#_Toc479366720)

[6.2.1 Storing on S3 9](#_Toc479366721)

[7 Exploratory Data Analysis 9](#_Toc479366722)

[7.1 Using Python 10](#_Toc479366723)

[7.1.1 LOAN DATA 10](#_Toc479366724)

[1. Exploring the relationship of variables to late payment 10](#_Toc479366725)

[7.1.2 DECLINED LOAN DATA 15](#_Toc479366726)

[7.2 Power BI 19](#_Toc479366727)

[7.2.1 Lending Loan Club Analysis for Approved and Declined Loans 19](#_Toc479366728)

[7.2.2 Lending Loan Club Analysis for Approved Loans 20](#_Toc479366729)

[7.2.3 Lending Loan Club Analysis for Declined Loans 21](#_Toc479366730)

[8 Contribution 23](#_Toc479366731)

[23](#_Toc479366732)

[9 REFERENCES 24](#_Toc479366733)

# Docker

1.Pull the docker image from the docker hub:

   docker pull sumit91188/assignment2part1

2.Create an image on the local docker terminal:

   docker run -d sumit91188/assignment2part1 tail -f /dev/null

3.Check the container created from the above query

   docker ps -a

4.Run the pipeline for loan data  on the docker container

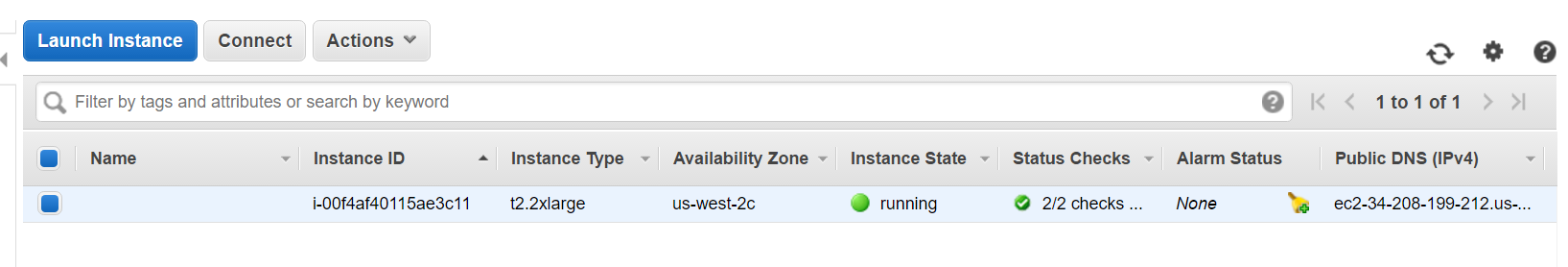
   docker exec -it <container\_id> python pipeline\_loan\_data.py start\_task --local-scheduler

5.Run the pipeline for declined loan data on the docker container

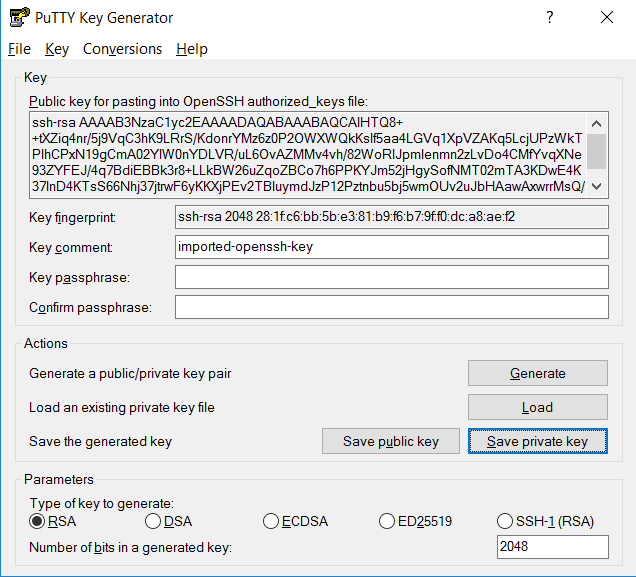
   docker exec -it <container\_id> python pipeline\_declined\_loan\_data.py start\_task --local-scheduler

# Running Pipeline on Cloud

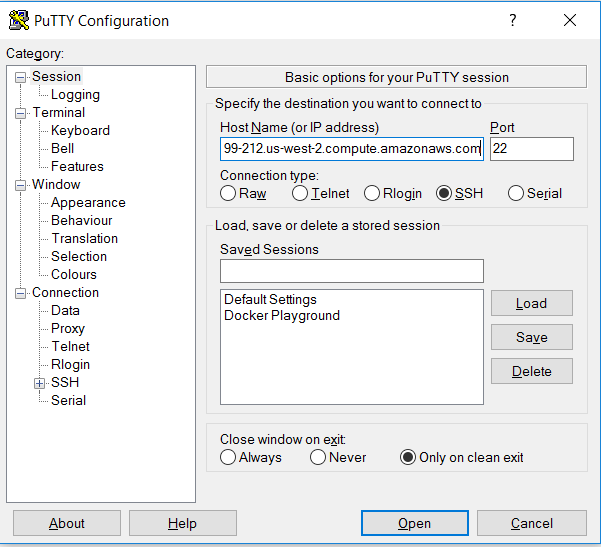
## 1: Create an EC2 Instance: We should create an Amazon EC2 Instance to run on cloud where the application can be hosted.



2**. Use Putty to connect to the instance:** Use puttygen to convert the .pem file generated from the instance and convert it to .ppk (private key) file.

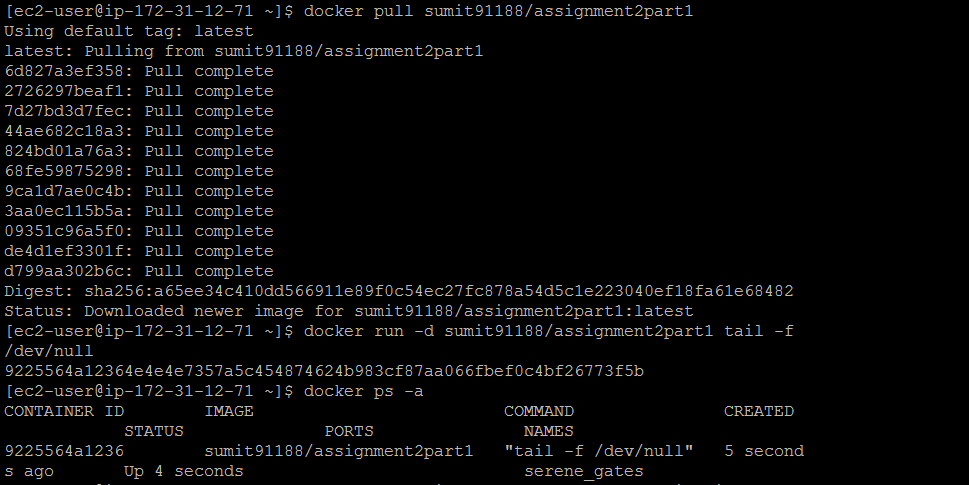


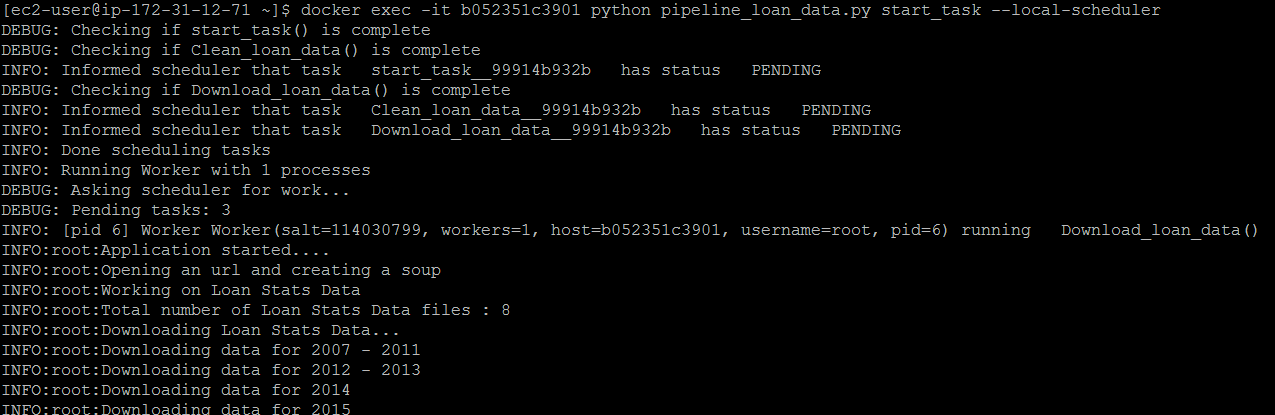
**3. Connect to EC2 instance:** Enter the public domain DNS for that particular instance and use the private key generated to connect to the EC2 instance.

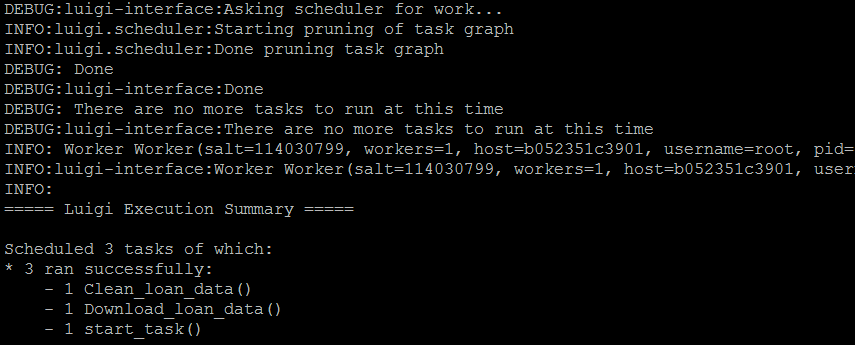


**4. Download Docker on the EC2 Instance:**

**5. Pull and Run the Docker Image:** Below screenshots shows the status of the completed job and we can check the S3 bucket for the data uploaded.





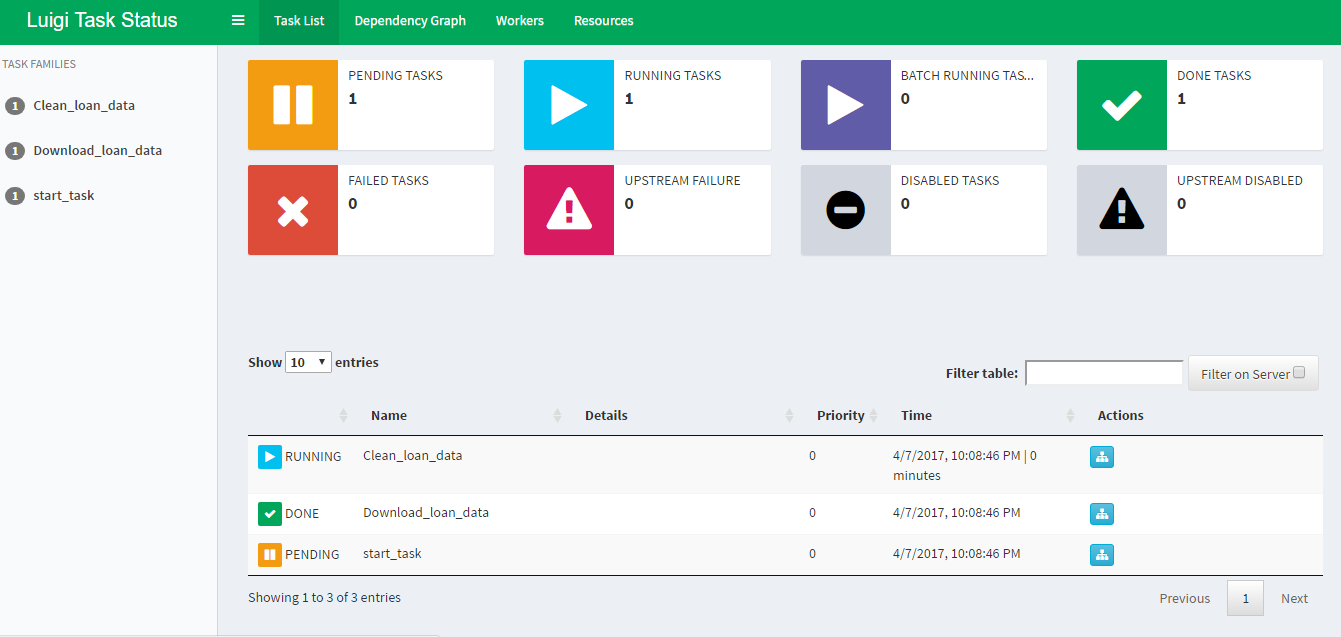


# Pipelining – Luigi

We have used Luigi to pipeline our tasks of Data Download, Data Preprocessing and Saving data to AWS S3. It helped us manage the workflow of our application.

## Loan Data Flow

## Decline Data Flow



# Data Download

## Downloading Loan Data

* Traversed to the link <https://www.lendingclub.com/info/download-data.action>
* Selected the loan data files to be downloaded by finding the div that stores it
* Extracted the files and stored them in the home folder dynamically if not already present at the below mentioned location:

Downloads/LoanData/

* Logged each of the steps in log files

## Downloading Decline Data

* Traversed to the link <https://www.lendingclub.com/info/download-data.action>
* Selected the loan data files to be downloaded by finding the div that stores it
* Extracted the files and stored them in the home folder dynamically if not already present at this location: Downloads/DeclinedLoans/
* Logged each of the steps in log files

# Data Preprocessing And Feature Engineering

## Loan Data

We created the below mentioned derived columns that we thought would directly impact the interest rate

### Preprocessing

* We assumed columns which have >= 90% data as null will not aid our analysis
* So, we dropped columns which have >=90% data as null which gave us 86 columns out of 113 columns
* Missing Values :
* Replace missing values in **annual\_inc** by mean
* For other columns replaced missing values by their respective modes
* Dropped column zipCode as it was encrypted

### Derived Columns

* **cat\_loan\_amnt :** Bins of loan\_amount based on quantiles
* **cat\_annual\_inc:** created bins of annual\_inc based on quantiles
* **Loan\_Status\_Binary**: This column indicates if the borrower is a defaulter or not

**Defaulter :** if the loan\_status has any of the following values the value for Loan\_Status\_Binary will be 1 else 0

* Charged Off
* Default
* Does not meet the credit policy. Status:Charged Off
* In Grace Period
* Default Receiver
* Late (16-30 days)
* Late (31-120 days)
* **cr\_line\_history :** this column gives the number of years of the borrower’s credit history

We calculated this column by subtracting issue year and earliest\_cr\_line value of the borrower

**Calculation :** ['issue\_d'].dt.year - ['earliest\_cr\_line'].dt.year

* **Verification Status** : if the verification\_status is not verified then 0 else 1

For this we assumed that 'Verified' and 'Verified Source' means the same

### Storing on S3

* Finally stored the clean and preprocessed data on S3 using Luigi Pipeline

## Declined Loans Data

* Created bins for Risk score using the below code according to FICO score ranges:

groupNames = ['Invalid','Very High','High','Moderate', 'Low', 'Very Low']

bins = [110, 299, 400, 600, 700, 800, 991]

df['RiskCategories'] = pd.cut(pd.to\_numeric(df['Risk\_Score'], errors='coerce'), bins, labels=groupNames)

* Replaced missing values in Risk score by an invalid value ‘111’

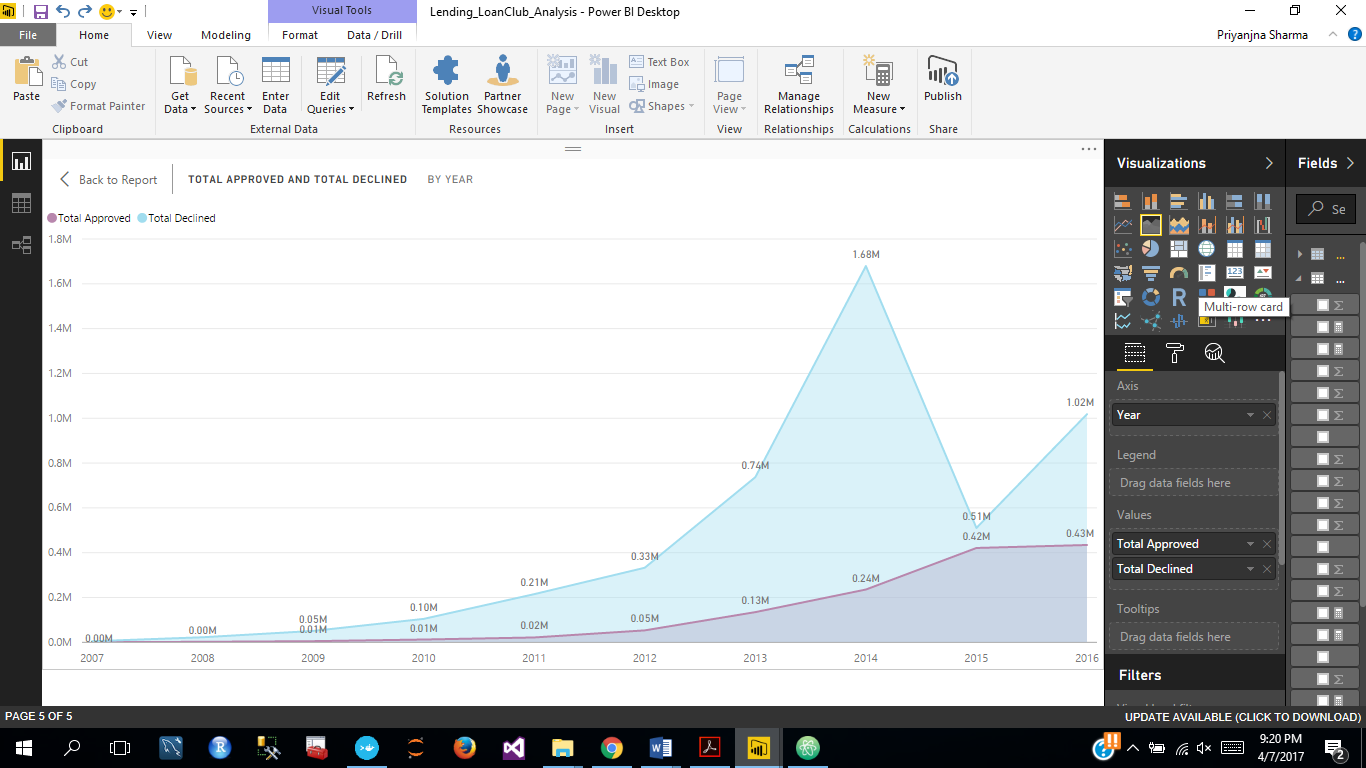
### Storing on S3

* Finally stored the clean and preprocessed data on S3 using Luigi Pipeline

# Exploratory Data Analysis

We performed exploratory data analysis by using our derived columns and the provided columns. Our analysis for Loan Data and Decline Loan data is as follows:

**POWER BI LINK:** https://app.powerbi.com/groups/me/dashboards/c6745b00-47ab-4d32-8de6-a570d13902d7



Observations :

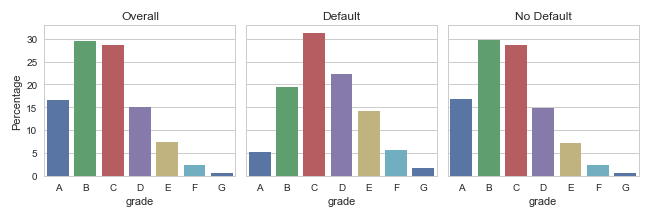
* From the graph we can see a hike in the number of declined loans as well as the number of approved loans. Which implies an increase in the number of applications in the year 2014.
* This supports the fact that in Year 2014, Lending Club went public and started providing loans to small businesses. It also partnered with Union Bank and in the end of 2014, Lending club raised $900 million which we can see with the spike in the number of applications.

## Using Python

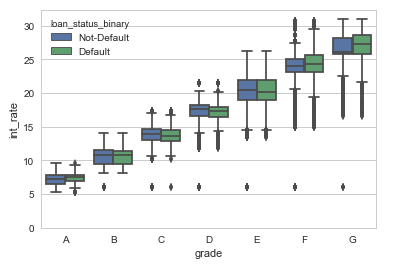
### LOAN DATA

1. ***Exploring the relationship of variables to late payment***

The grade of the loan is the companies estimate of the likelihood of default for the loan. As should probably be expected the best graded loans (A and B) have a higher percentage of loans with no default than with a default. C is approximately the same percentage across no default and default and the worst graded loans (D, E, F and G) have a higher percentage of loans with default than with no default.

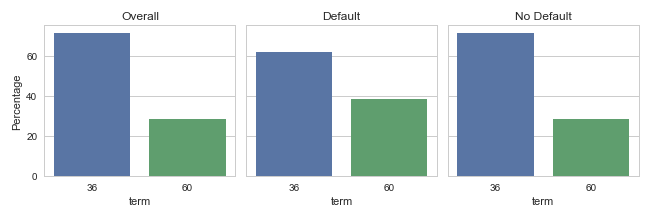


Even controlling for the grade of the loan (as this will be used to calculate the interest rate) the defaulting loans still have a higher interest rate than non-defaulting loans for most of the grades.



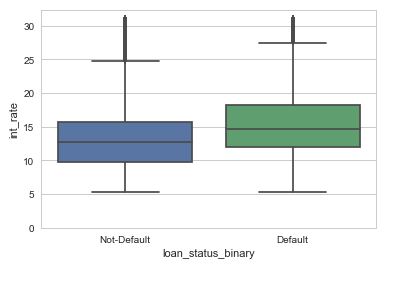
1. ***Loan Term:***

The longer-term loans (60 months) make up a higher percentage of the defaults than the non-defaulting loans.



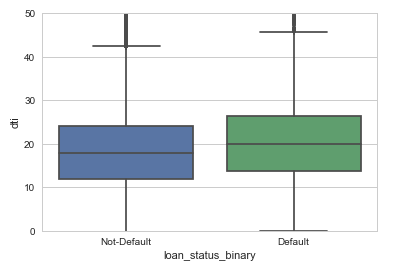
1. ***Interest Rate:***

The defaulting loans have a higher interest rate than non-defaulting loans.



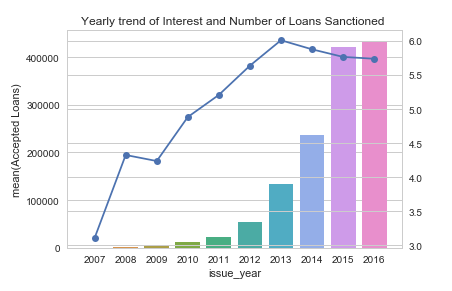
1. ***Debt to Income Ratio:***

Defaulting loans have a higher DTI.

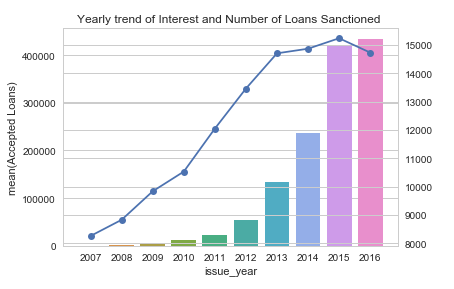


1. ***Summarizing the data by vintage:***

*Summary Chart for Year Wise Accepted Data and Approved Loans:*

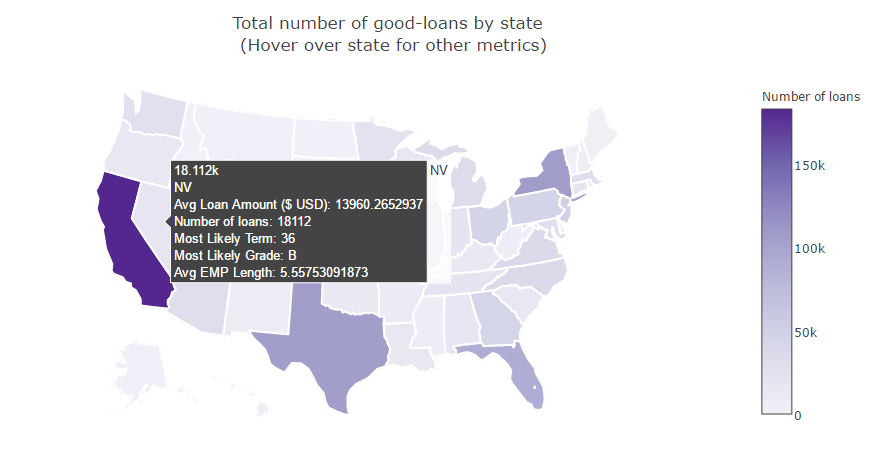


1. ***Summarizing the data by vintage:***

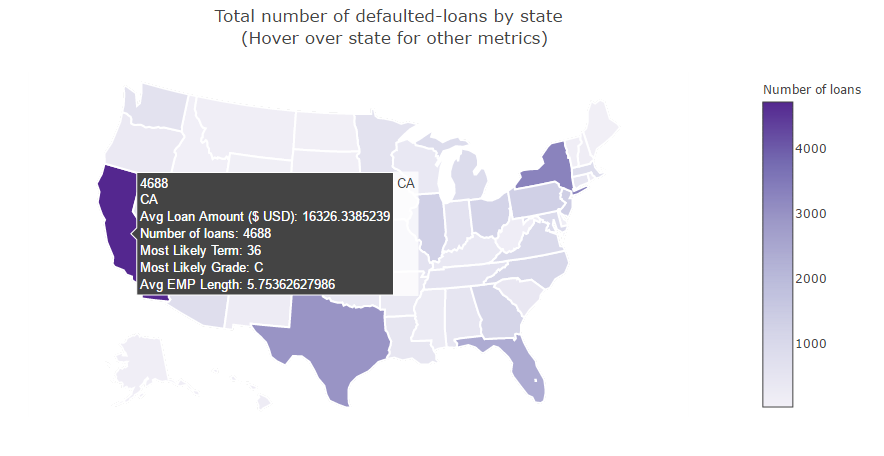


1. ***Summarizing good & default loans by state:***

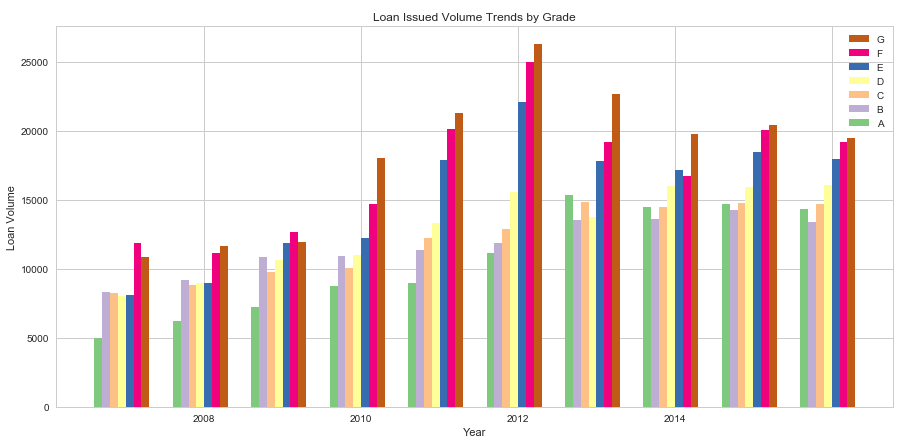
*Summary by state for Good loans:*

****

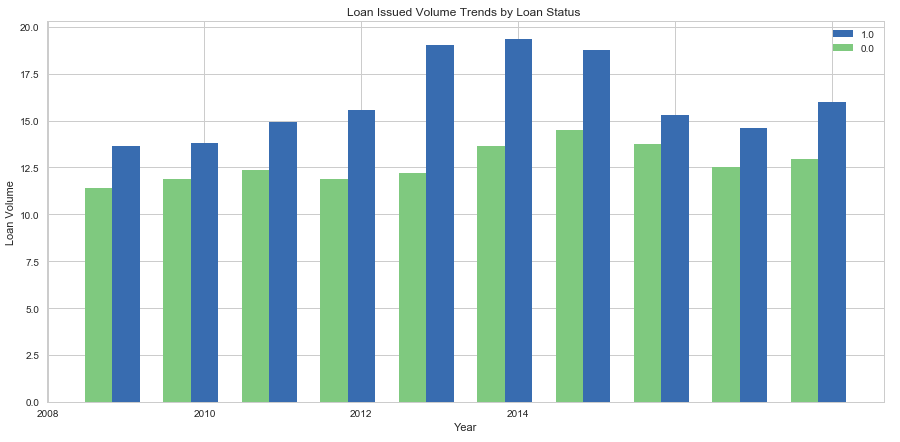
1. ***Summary by state for default loans:***

****

1. ***Summarizing loans by Grade:***



1. ***Summarizing loans by Year & Loan-Status:***

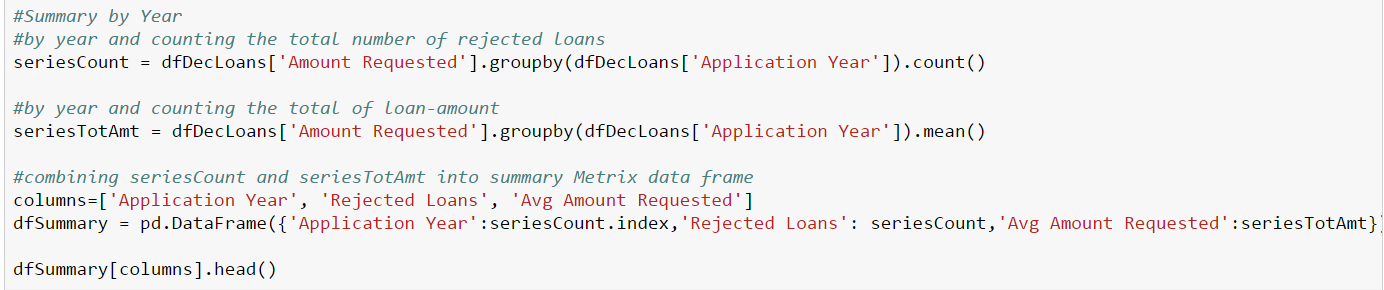


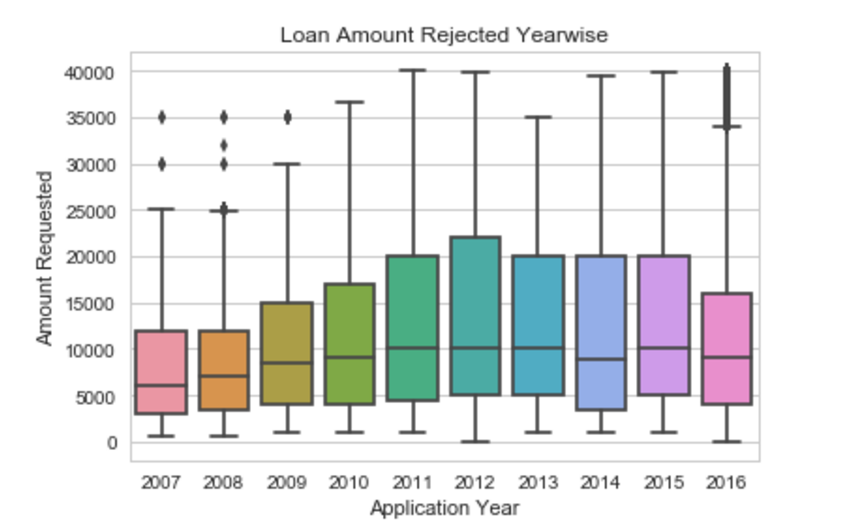
### DECLINED LOAN DATA

Used seaborn, matplotlib, plotly libraries for plotting the graph.

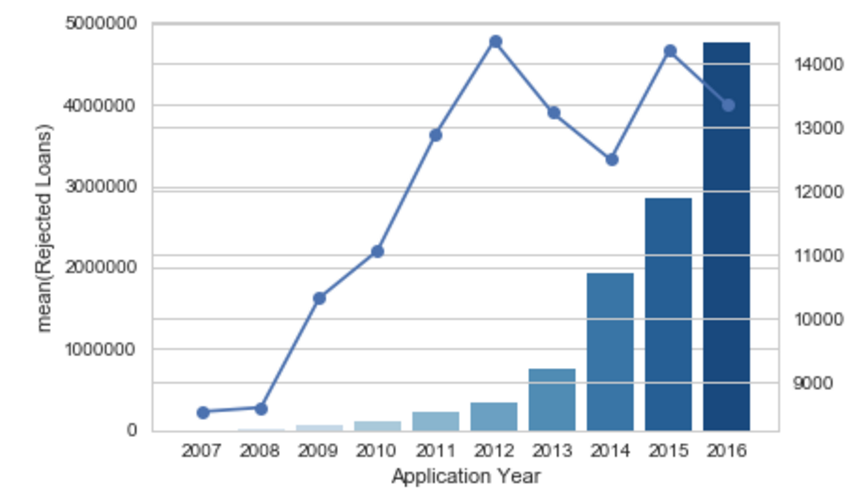
Year wise Analysis:

Created a summary data frame dfSummary to analyze the yearly trend for the loan volume and the total loan amount.





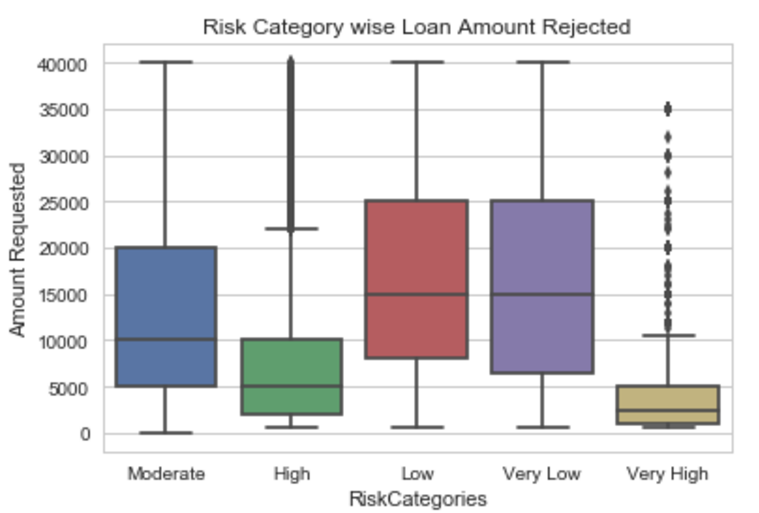
1. As we can see in the above chart, the total amount rejected has **increased** from **2008-2012** and is steady after that.
2. As we can see below the hike in average amount requested from the year **2010-12 got a huge hike** and then suddenly got declined from **2012- 2014** with the total count of rejected loans being increased constantly over years.



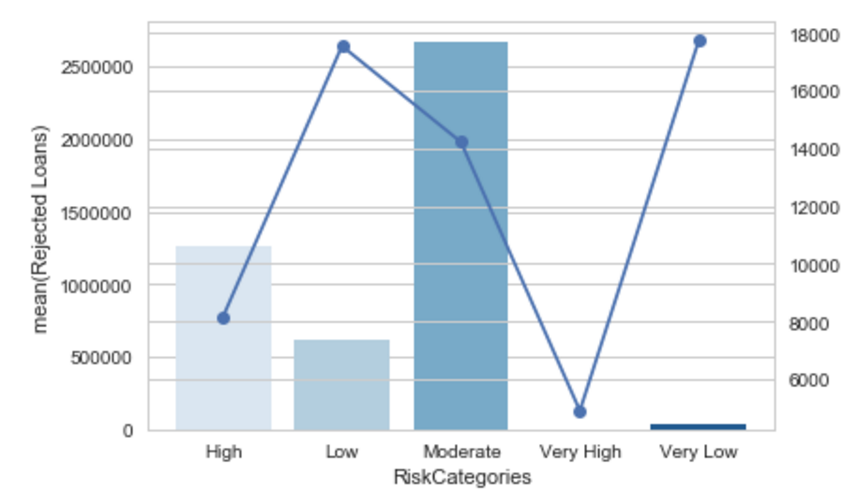
Risk Categories wise Analysis:

We have created bins on the basis of Risk Scores as: Very High, Very Low, Low and High.

1. As we can see in the below box plot, the people lying in **“Very High”** bucket have the least amount of loan rejected as they apply for less amount of loan.

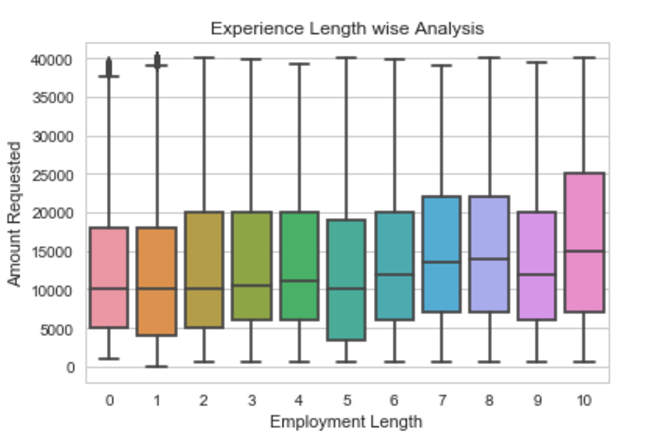


1. People falling in “**Very Low”** i.e Very Low Risk bucket have the least count of rejection and the highest in the amount requested.

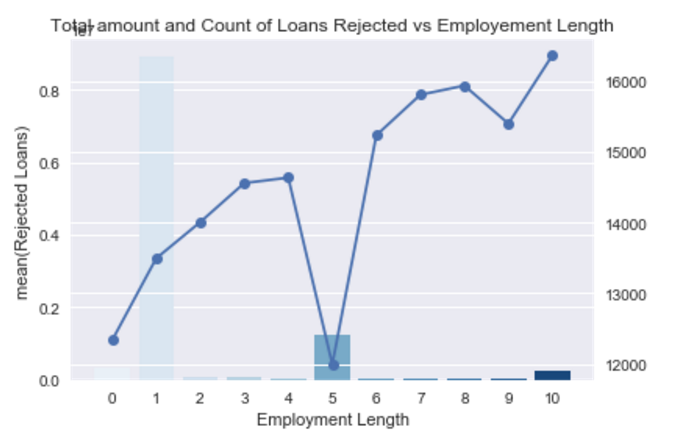


Employee Experience Length wise Analysis:

1. People with **>10 years of experience** have the most loan amount requested.



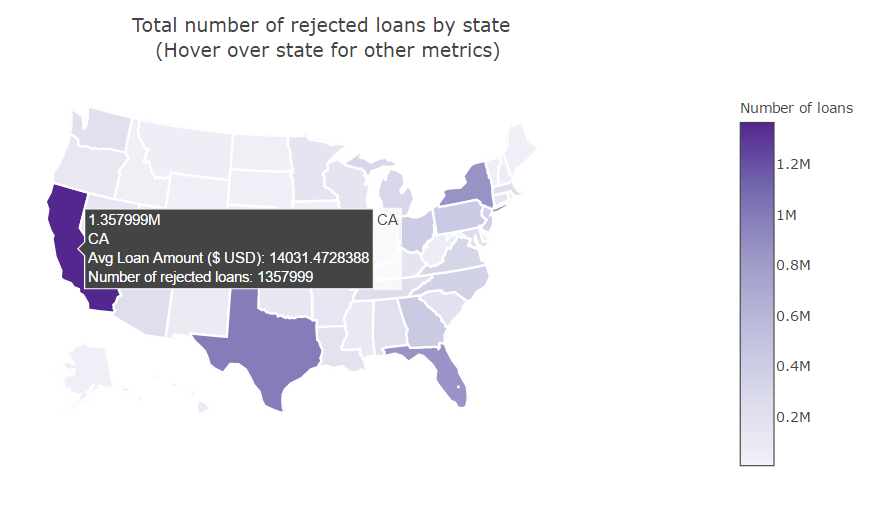
1. Population with 5 years of experience got their loan rejected most.



State wise Analysis:

We have performed different analysis based on state. As we can see below:

1. California has the highest number of loan amount requested and rejected both.



## Power BI

We analyzed the trends in Approved Loans and Declined Loans on the Lending Club data using Power BI

### Lending Loan Club Analysis for Approved and Declined Loans

* Summarized the loans and declined loans data based on Year and State Using group by in MDX(Power BI)
* Joined the two summaries into one table using DAX

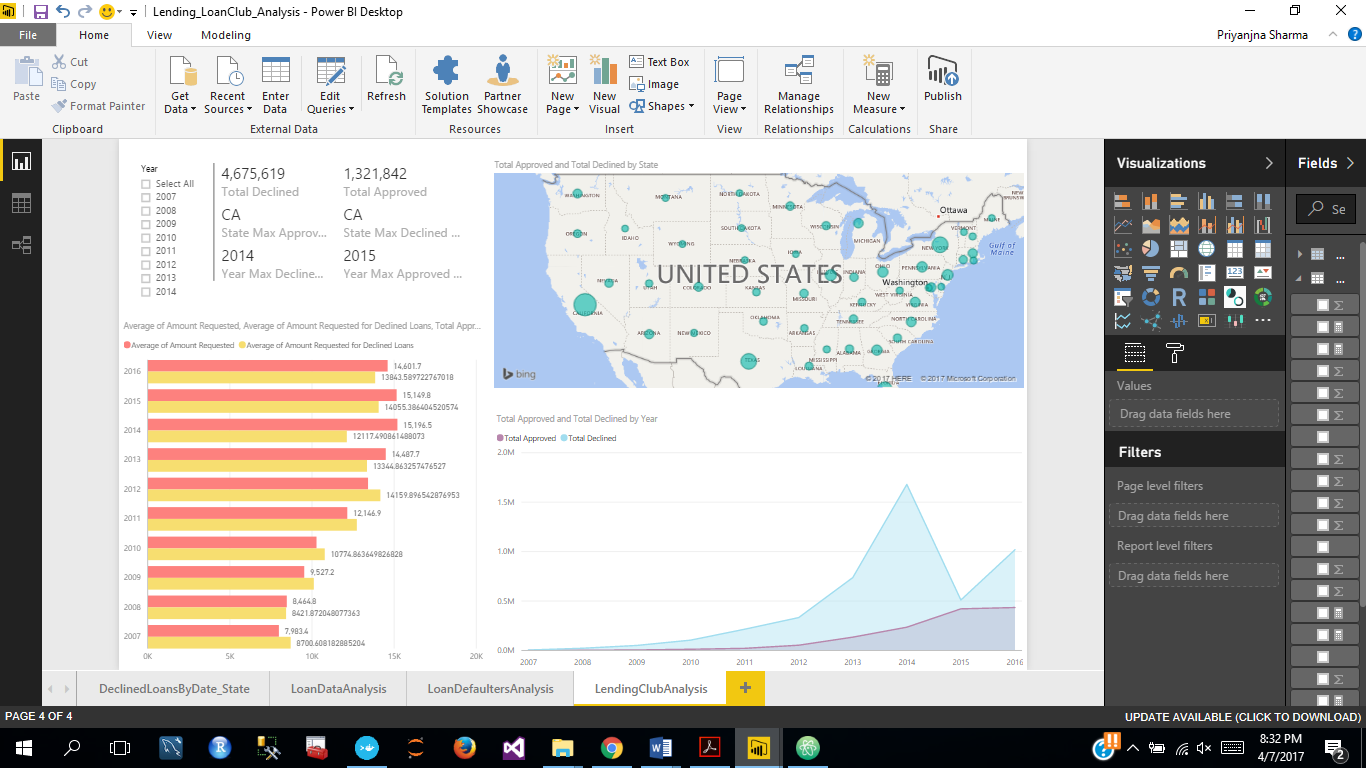
LendingClubAnalysis = FILTER (

CROSSJOIN ( LendingClub, 'Table' ),

LendingClub[Year] = 'Table'[ApplicationYear] && LendingClub[State] = 'Table'[State1]

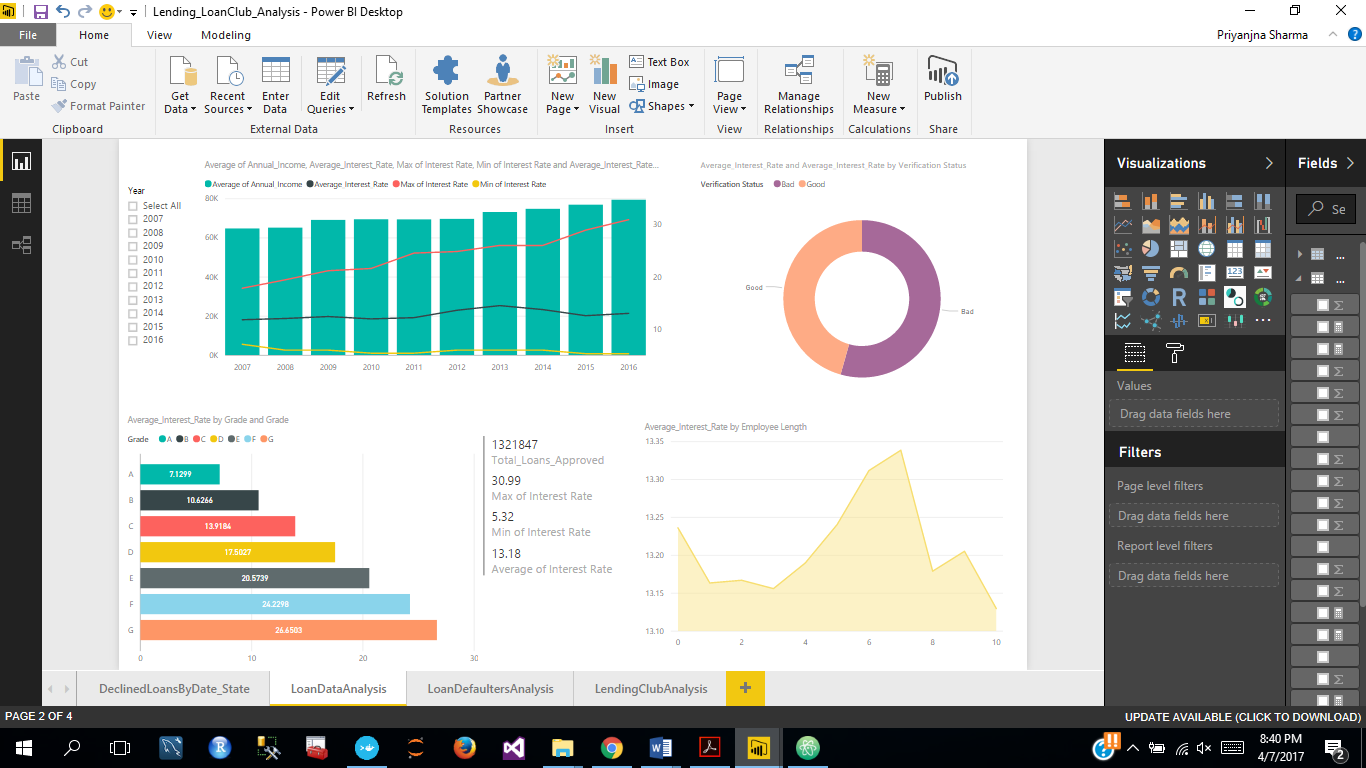
)

* Analyzed the relation between total loans approved and declined over the years by
* Lending Loan club

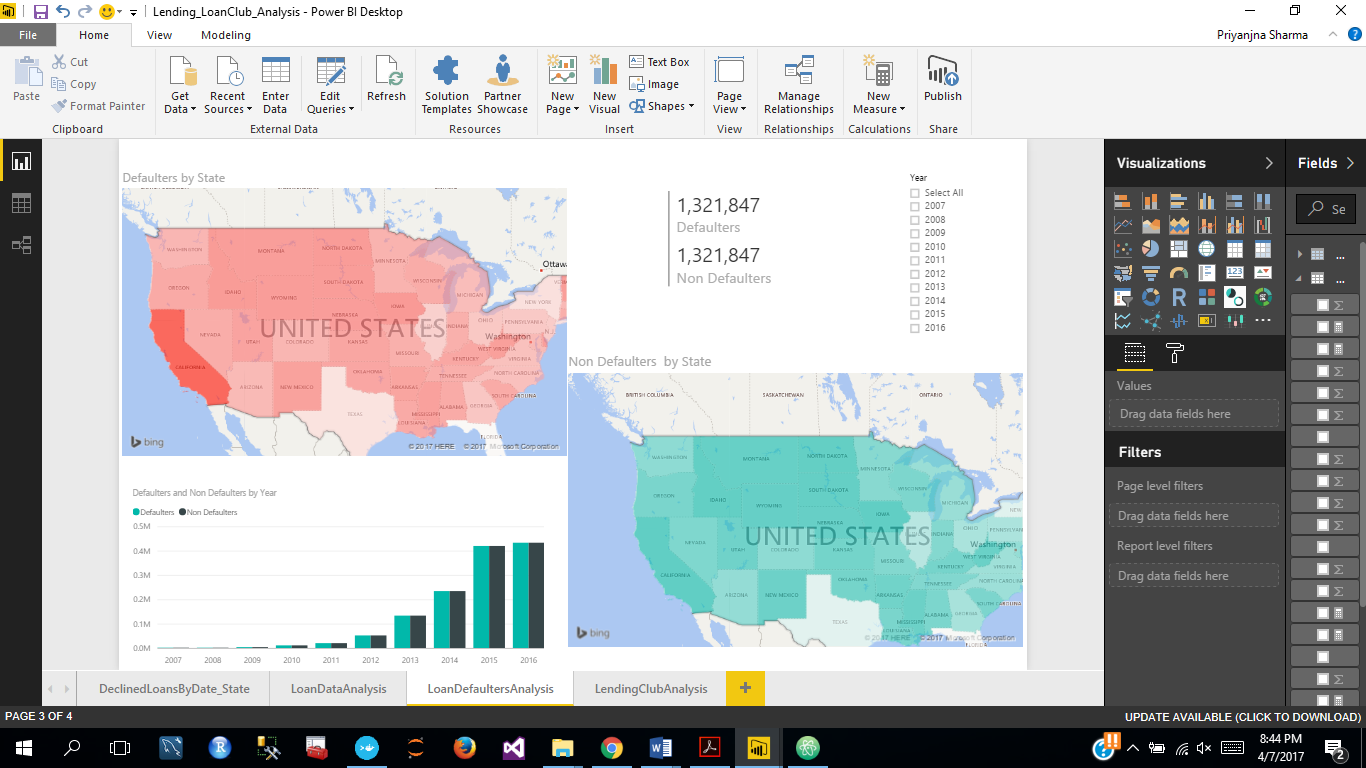


### Lending Loan Club Analysis for Approved Loans

* Analyzed approved loans based on state, year , grade , verification status , interest rate
* Grade G has the highest interest rates

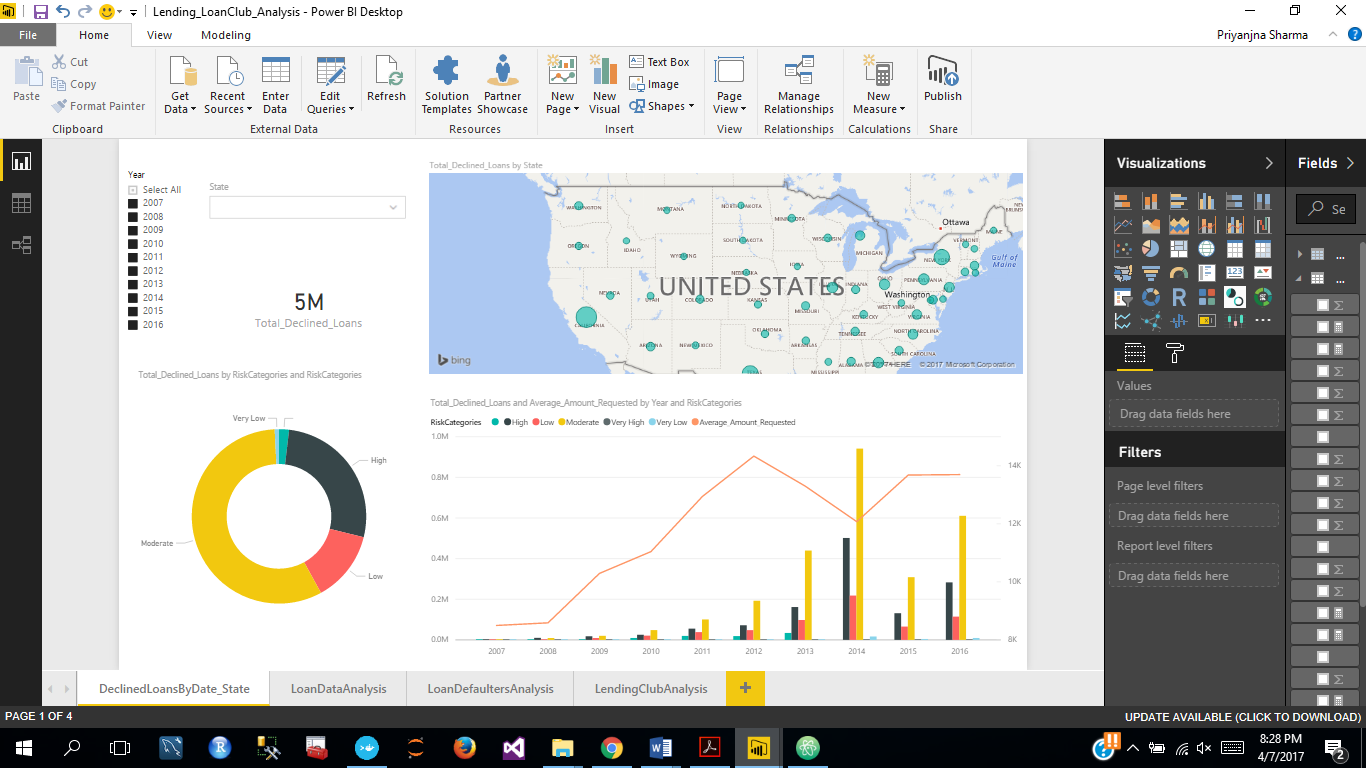


* Defaulter Analyses : analyzed the total number of defaulters by using a calculated column : loan\_status\_binary for different states



### Lending Loan Club Analysis for Declined Loans

* Analyzed declined loans based on state, year and different risk score buckets we created.
* Total declined loans were the most for moderate range of risk score
* Total declined loans were less for high risk score ranges



# Contribution

# 

# REFERENCES

1. <https://en.wikipedia.org/wiki/Lending_Club>
2. <https://www.lendingclub.com/info/download-data.action>
3. <https://www.credco.com/assets/pdfs/datasheets/FICO-booklet.pdf>
4. <http://www.lendingmemo.com/average-investor-return-lending-club-dropping/>