HMDA Data Analysis

Manoj Kumar Billa

Executive Summary

The House Mortgage Disclosure Act (HMDA) was enacted by Congress in 1975[1]. This regulation provides the public Housing Loan data, reported by various institutions every year. Change Financial wishes to understand the regional home lending market by analyzing this data.

This project will combine and Analyze 2 datasets – Loans and Institutions. Both contain House Loan data from the years 2012 – 2014 for five states. The data is merged based on the key values, it is transformed and then some descriptive and outlier analytics is performed on the data. Rules to maintain the integrity of the data are defined and some visualizations are created so that the leaders at Change Financial can better understand their regional home loan market. Ultimately, a region where the market is better compared to other regions is identified. A shiny app is also created so that the VP can see market share for competitors in each geography.

The data cleaning, descriptive analytics, visualizations and app have been made using open source tools – R and RStudio.

Software: R Studio 1.0.143, R 3.4.0

Data Source: HMDA (Home Mortgage Disclosure Act) [2]

Initial Preparation

The packages required for Data Cleaning, Plotting and Creating App are installed and loaded. The Loans and Institutions' datasets are downloaded and copied to the working directory.

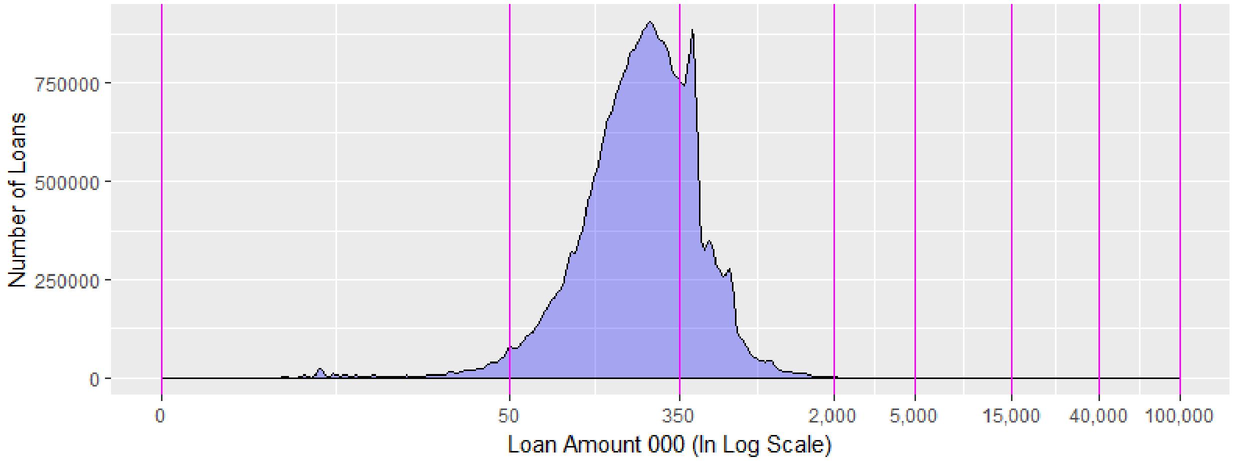
Data Munging

Raw data from both the datasets can be joined using the unique combination of Year, Agency Code and Respondent ID as the key. The loans data has 1.3 million rows, which, when loaded using read.csv( ) takes some time. To quicken this, fread( ) function from data.table package is used.

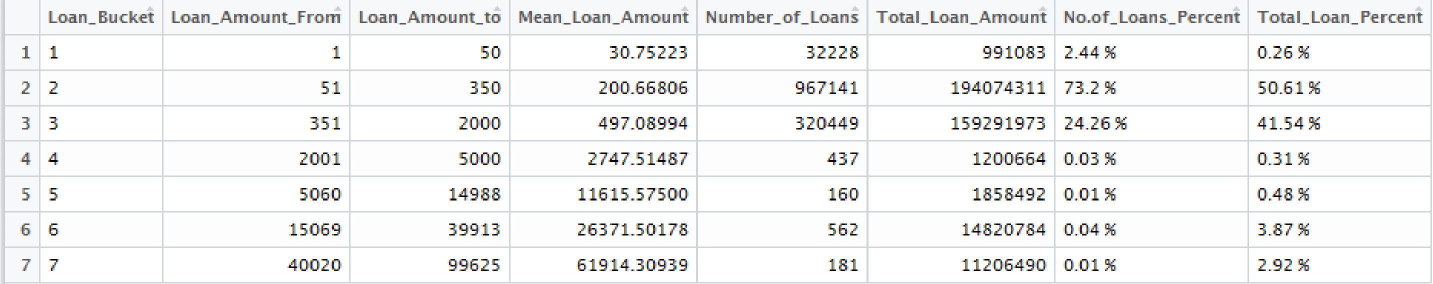
After loading, both the datasets are merged using the merge( ) function from dplyr package. The data is merged using the combination of As_of_year + Agency_Code + Respondent_ID values. The final dataset had ~1.3 million rows (Same number of rows as in the Loans dataset) and 27 columns.

R identified some of the numeric columns correctly but several other columns are not. The class of these columns is to be changed. The mutate( ) function from dplyr package is used to do this. In this step, the Applicant Income, Loan Amount 000, Number of Owner-Occupied Units, FFIEC Median Family Income, Tract to MSA MD Income Pct columns are converted into numeric form and the Conventional Conforming Flag, As of Year, Loan Purpose Description, Agency Code Description, Lien Status Description, Loan Type Description, Conventional Status and Conforming Status are converted into factors. The Loan_Amount_000 along with some other variables are present in the multiples of thousands. So, they are represented with a _000 appended to their column names. The median family income is still in its original form, so this is divided by 1000 in this step.

Market segmentation helps us identify and target specific groups of customers. In this case, the Loan_Amount_000 variable a key metric, which can show the market size, etc. Hence, a new attribute called as "Loan_Bucket" is created based on this. This column has 7 buckets into which the dataset is divided based on the Loan_Amount_000 and cutoff. Prior to this, the distribution of Loan_Amount_000 is visualized. The loan buckets are chosen such that they capture most of the variations in the distribution of Loan_Amount_000. In this case, the loan amount is divided based on the peaks observed in the density plot.



The above density plot shows the distribution of the Loan Amount, separated by purple lines representing bucket cutoffs. It is observed that there are several values concentrated in the 50 ($50,000) to 2000 ($2 Million) range and there are very high loan amounts ranging to $ 100 Million. Based on the data distribution observed, the loan amount is segregated into 7 buckets in the Loan_Bucket column, which has a class of factor.



	Loan_Bucket	Loan_Amount_From	Loan_Amount_to	Mean_Loan_Amount	Number_of_Loans	Total_Loan_Amount	No.of_Loans_Percent	Total_Loan_Percent
1	1	1	50	30.75223	32228	991083	2.44 %	0.26 %
2	2	51	350	200.66806	967141	194074311	73.2 %	50.61 %
3	3	351	2000	497.08994	320449	159291973	24.26 %	41.54 %
4	4	2001	5000	2747.51487	437	1200664	0.03 %	0.31 %
5	5	5060	14988	11615.57500	160	1858492	0.01 %	0.48 %
6	6	15069	39913	26371.50178	562	14820784	0.04 %	3.87 %
7	7	40020	99625	61914.30939	181	11206490	0.01 %	2.92 %

The above table shows the total loan amount and a total number of loans present in each of these buckets. 73% of the loans are present in the bucket 2 while 24% are in bucket 3. Rest of the buckets have a negligible number of loans. Also, the buckets 2 and 3 account for 50% and 41% of the total loan amount respectively. Buckets 6 and 7 though have very less number of loans, they have around 3% of total loan amount each. From this table, it is apparent that capturing the customers in buckets 2 and buckets 3 is important if Change Financial decides to enter these markets. However, we will analyze this subsequently.

When this function hmda_init( ) is called with the correct file paths, the function creates a global variable called as “combined”, which is stored as a tbl. This dataframe is used everywhere in our subsequent analysis. So, it is important that the combined data frame is present before performing other actions.

A function to export the data to json format is also created. This function can be passed arguments for State and Conventional_Conforming_Flag and the data is filtered based on that. If there are no filter values specified, then the function exports the complete dataframe as json. For filtering, the pipe operator (%>%) from dplyr package is used. For converting the dataframe to Json, toJSON( ) function from jsonlite package is used. This function saves the json file in the current working directory.

Quality Check

Data Cleaning is the most important steps of Data Analysis as the subsequent analysis results are dependent on this. The “Garbage in, Garbage Out” criteria needs to be diminished as much as possible, because the results of our analysis are used to make key strategic decisions. For this, the quality of data is assessed using descriptive analysis and rules are created so that data quality can be monitored. Apart from this, specific restrictions and guidelines are also established based on general understanding of the data and functional knowledge. Knowing the stakeholder perceptions of data is also important to get a contextual understanding of data and frame better rules.

A dataset can have Validity, Accuracy, Completeness, Consistency and Uniformity related issues [3].

Validity: The degree to which data measures conform to business rules. For good validity of data, these constraints are to be enforced while data recording itself. However, that is not something which is in our control. But, we can create our own constraints to validate Year ranges, State and County Fields, ZIP codes etc. We can also validate the referential integrity constraints, where we make sure that there are no null values in either of the As_of_year, Agency_Code and Respondent_ID fields. The uniqueness of the values in Sequence_Number field, for a given combination of As_of_Year, Agency_Code and Respondent_ID can also be checked.

Accuracy: HMDA data is the data reported by different institutions. So, there might be Accuracy issues. However, without integrating external datasets, Accuracy issues cannot be identified easily. We can do descriptive analysis and outlier Analysis to identify anomalies. Again, we cannot remove the outlier rows as we cannot be sure that these are anomalies. So, this type of Analysis only serves the purpose of gauging the data quality. With the help of a subject matter expert, we can get better results as then we will be able to know which is erroneous data.

Completeness: The completeness of data is another issue that cannot be resolved completely. In our dataset, we can impute the missing values, if there are any in the State or County_Name based on the State_Code. However, in the other fields, the missing values cannot be identified easily. For example, identifying the missing values in FFIEC Median Family Income may need an external dataset. Also, based on the State and County details, we can identify the MSA. However, even for this, we need external MSA data.

Consistency: There can be inconsistent data entries in the Respondent_Name, Parent_Name and MSA_MD_Description fields. The same name may be entered differently in any of these fields. It is necessary to identify which is the correct name and populate the other fields with this name. We can also modify some of the names using regular expressions in R or Python.

Uniformity: The data measures need to have same units. In our case, the monetary values are present in the multiples of 1000 in all the fields except FFIEC_Median_Family_Income. So, this column is divided by 1000 to make all the measures uniform.

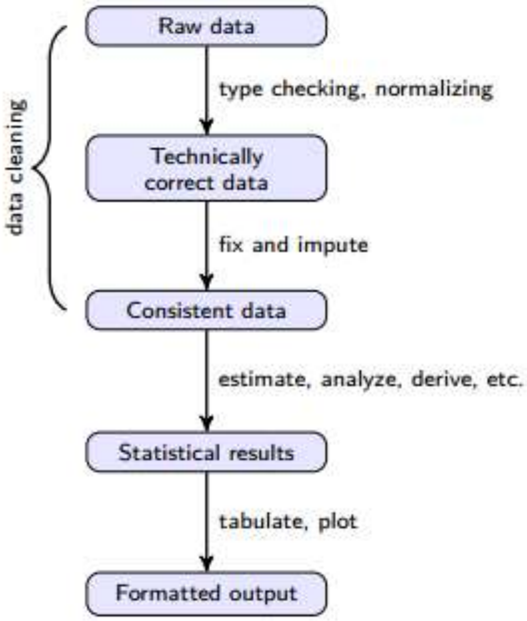


Figure 1: Statistical Analysis Value Chain [4]

As seen in Figure 1, there are various stages of data cleaning and various levels of data cleanliness. As part of this project, at least one step of each of these processes is done.

Detailed attribute wise Analysis:

a) **Loan_Amount_000:** This is the most important variable after Year, Agency_Code and Respondent ID. Not only because this plays an important role in further analysis, but also because this can also be used to validate other columns like Conforming_Status. The Loan_Amount_000 column has values in the multiples of 1000 and data type of this should be numeric. Highest quality data has no missing values in this column. For the current dataset, it is observed that there are no missing values.

In the future, if a different dataset is used, then this column can be read as a character using stringsAsFactors = FALSE while reading the data and then using as.numeric() to convert this to a numeric form. When we do this, if there are any characters in the Loan_Amount_000 column, then the non-numeric fields are changed to NA's.

The density plot shows the distribution of the data in the Loan_Amount_000 column.

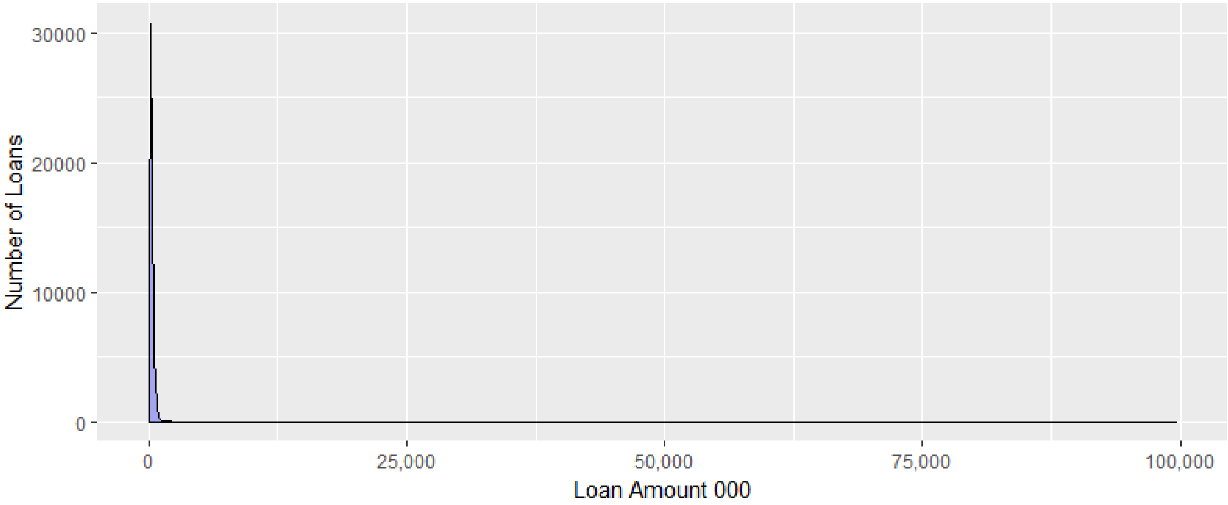


Figure 2: Density plot of Loan Amount

This is the density plot without any log transformation on the x axis. The distribution is heavily right skewed. When we check for the outliers, based on Agency and State, we get these results.

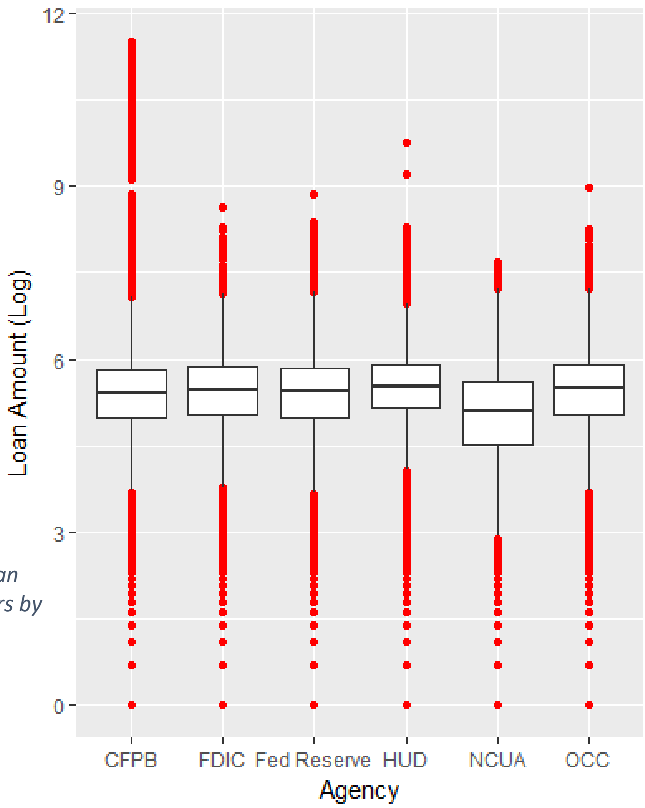


Figure 4: Loan Amount outliers by Agency

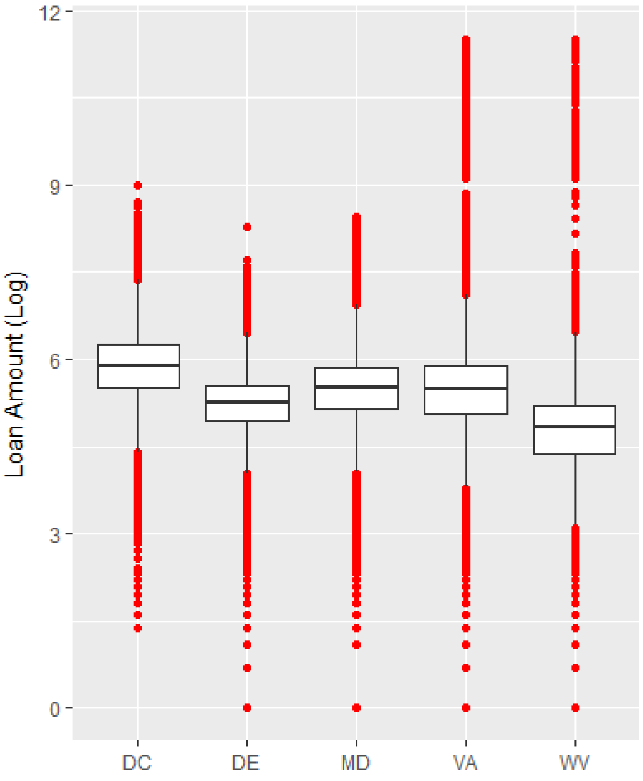


Figure 3: Loan Amount outliers by State

There are many outliers for the CFPB agency. With respect to state, VA and WV have the most outliers.

The standalone outliers for the entire range of Loan_Amount_000 can be known using the boxplot.stats( ) function. This function has a cutoff specified, and all the points beyond this cutoff are considered as outliers. Using the outlierKD script from datascienceplus.com [5], the statistics before and after removal of outliers are gathered as shown below. In this case, all the Loan_Amount's which lie outside 1.5 times IQR (Inter Quantile Range) are outliers.

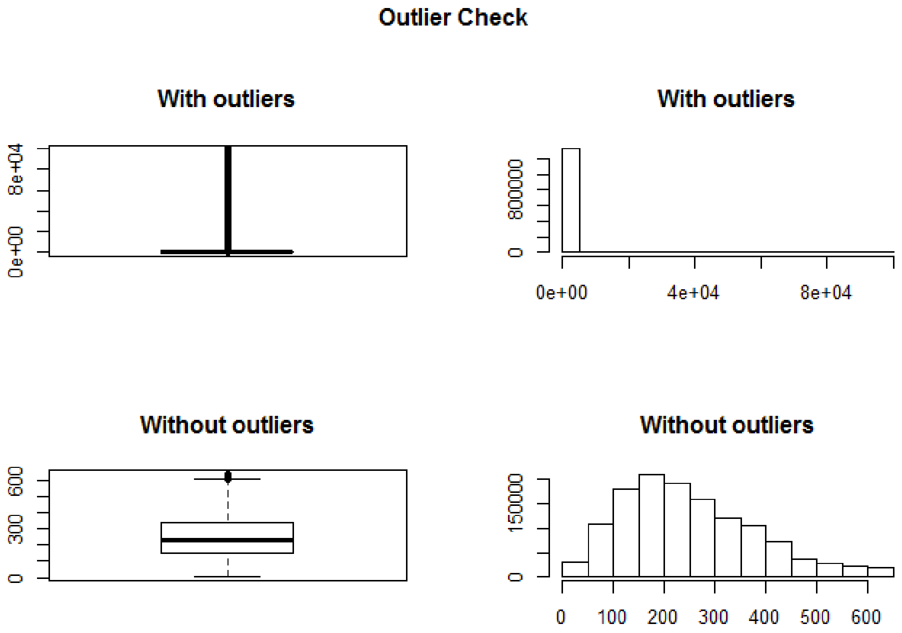


Figure 5: Outlier Analysis

```
Outliers identified: 38983 from 1321158 observations
Proportion (%) of outliers: 2.95066903428659
Mean of the outliers: 1587.55821768463
Mean without removing outliers: 290.233111406811
Mean if we remove outliers: 250.789490514165
```

These outliers may be actual loan amounts and not error's in measurements. So, we do not remove them from our Analysis

b) **Respondent_Name:** Respondent name is mapped to each loan from the Institutions Data. It is observed that this column has different names for even the same respondent ID. Some of the common issues observed are double spaces, repetitions of the same letter, abbreviations like NA, ASSN, State Names etc., in some cases expanded in some cases not. In most of the cases, the name is incomplete. Duplicate values are checked for in the Respondent_Name_TS column and 8740 duplicates have been found to be present. This is expected because the institutions report once every year, so almost all the institutions are supposed to have their

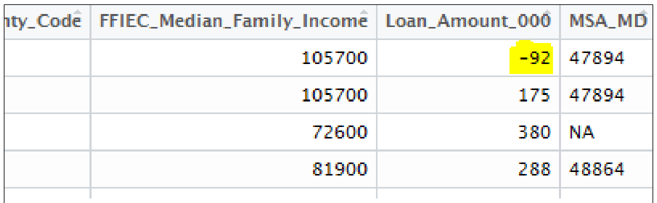
Respondent ID in the dataset 3 times. So out of the 21655 values, around 14000 values are supposed to be duplicates if no bank shut down or merged in the 3 years. However, our duplicates analysis gave us only 8740 duplicates, meaning that at most, ~6000 values have wrong names.

There are different ways of cleaning this column data based on the importance of the data within it. In this case, this column does not play a major role in our analysis. So, we can just replace the double spaces and trim spaces at the beginning and end. We can even create a new column where all the rows with same respondent_id will have same name (The longest name after removing double spaces for each respondent_ID). As the dataset is rather large, we cannot set factor labels and check for undefined factor labels. Apart from this, spell check and regular expressions can also be used to clean the data.

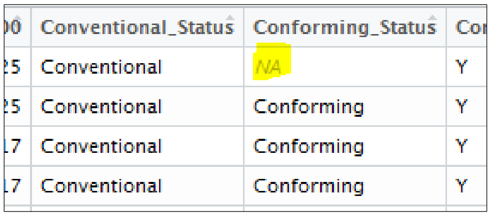
c) **Conforming_Status:** When the loan amount is greater than Conforming_Limit_000, the Conforming_Status should have the value "Conforming". If the amount is lesser, it should have the value "Jumbo". The entire dataset should have either of these 2 values only. However, in the current dataset it is observed that in all the rows where Conforming_Limit_000 is missing, the Conforming_Status column had the value "Jumbo". This is observed even in some cases where the loan amount is as low as 5 ($5000) which is obviously not a jumbo loan. Perhaps the HMDA used a wrong formula to calculate this field. To resolve this, we need to know on what basis Conforming_Limit_000 is determined. If we are unable to find that, and if we need to use this column for some analysis, then it is better to remove the rows where the conforming status is Jumbo even when there is no Conforming_Limit and Loan Amount is below a certain threshold (say $400,000). In extreme cases, the values can be imputed based on an average Conforming_Limit cutoff value.

d) HMDA website states that there may be duplicate records for a single company due to discrepancies in reporting. So, the duplicates must be removed from the dataset if there are any. The yearly also data needs to be analyzed to identify missing data. For example, if the data is available for 2012, 2014 years but not 2013, that can mean that the data is not recorded. The other 2 cases where data is missing in either 2012 or 2014, are also possible but in these cases, we cannot say for sure whether data is not recorded or whether the Bank is acquired/merged or if the Bank is just opened.

Apart from this, rules can also be defined to maintain data integrity and check for data quality issues. For this, violatedEdits( ) function from editRules package is used. This function, can take the rules as input from a text file and identify the % and number of violations. This also identifies the type of violation (numeric, character etc.). For example, in our dataset, the first row is modified and this data is tested against the rules in violatedEdits( ) function.

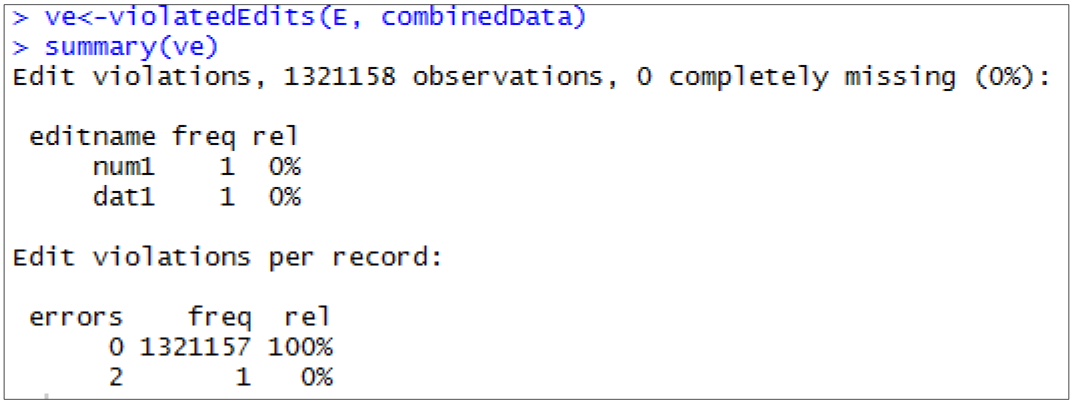


nty_Code	FFIEC_Median_Family_Income	Loan_Amount_000	MSA_MD
	105700	-92	47894
	105700	175	47894
	72600	380	NA
	81900	288	48864



00	Conventional_Status	Conforming_Status	Co
5	Conventional	NA	Y
5	Conventional	Conforming	Y
7	Conventional	Conforming	Y
7	Conventional	Conforming	Y

Figure 6: Dataset manually altered to test the rules



```
> ve<-violatedEdits(E, combinedData)
> summary(ve)
Edit violations, 1321158 observations, 0 completely missing (0%):

 editname freq rel
     num1    1  0%
     dat1    1  0%

Edit violations per record:

 errors    freq  rel
      0 1321157 100%
      2       1   0%
```

Figure 7: violatedEdits( ) showing type of errors and number of errors

This package helps us to specify rules and automatically generate the bar graphs for various types of issues

Visual Narrative

Hypothesis: Change Financial should enter a market where the total loan amount and number of loans are increasing. If there is no such area, change Financial should enter a region where the market is relatively big and where there is more space for new companies.

Market Size by State and Year:

From the plot below, it can be understood that that the loan amount has been steadily decreasing over the 3 years. The number of loans also decreased indicating that Home buying is on a decreasing trend overall in these five states. The positive aspect is the average loan amount, which slightly reduced in 2013 but increased again in 2014. This indicates increasing housing prices.

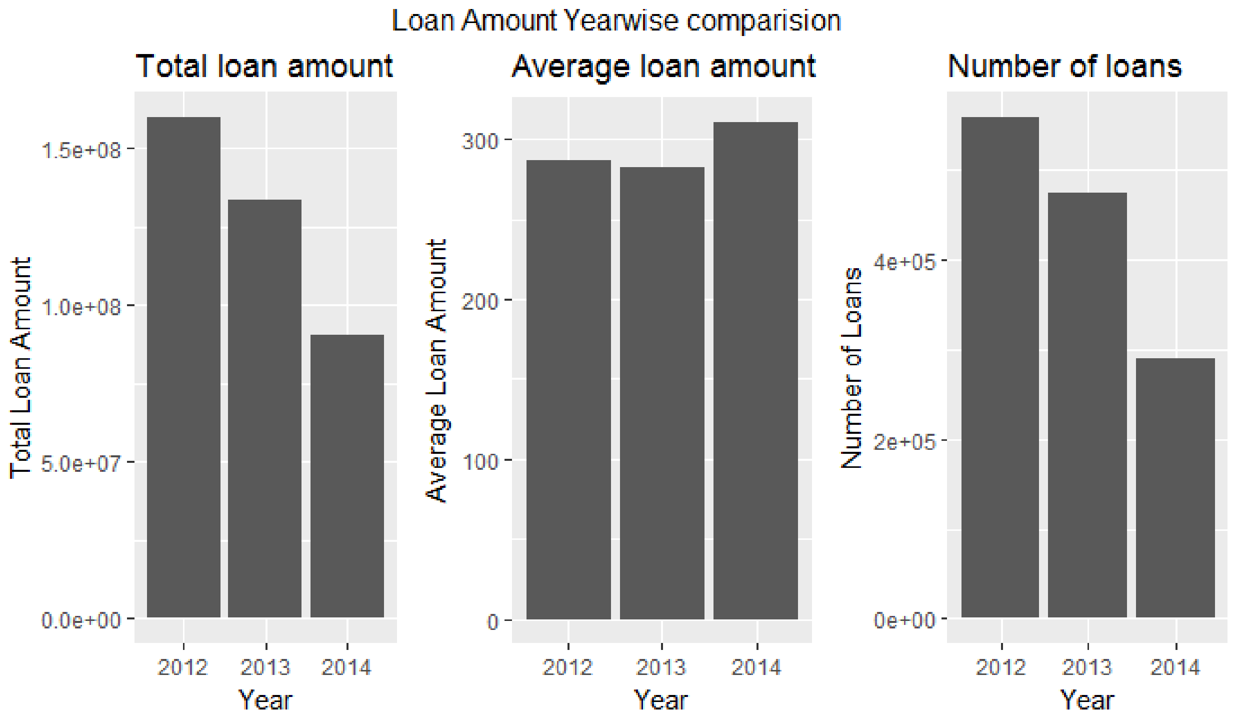


Figure 8: Loan Amount variation with years

Further drilling down, the State-wise comparison is done to have a look at the market size in all the states.

Initial state-wise comparison plot showed that the number of loans and subsequently the total loan amount are very high in both MD and VA. This might be a result of the areas of these states. The areas of DC, DE, MD, VA and WV are 68.34, 2491, 12407, 42775 and 24038 mi^2 respectively (from Google). Plotting a graph with these values to get a better idea.

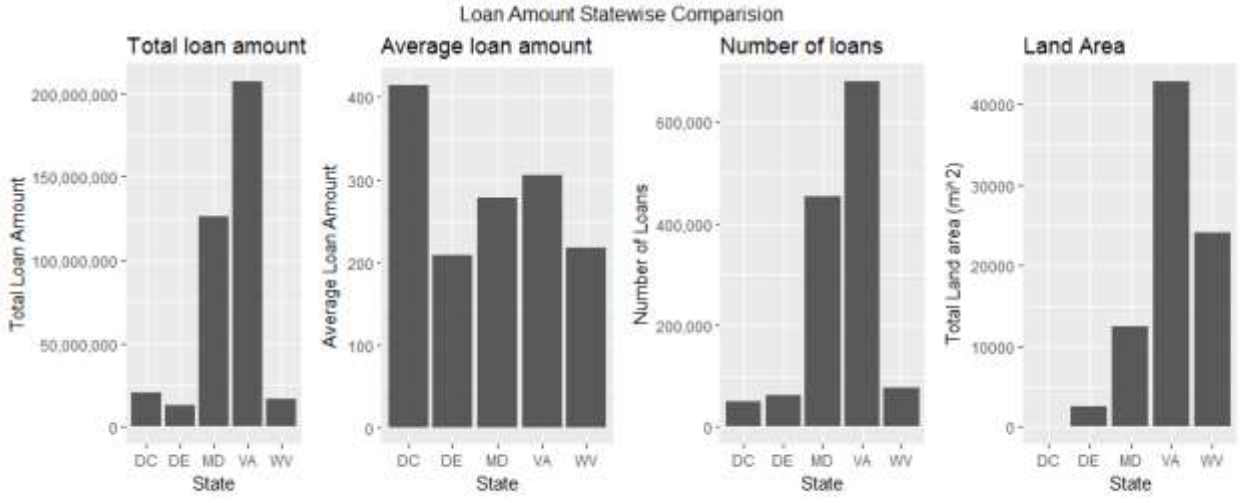


Figure 9: Loan Amount Variation with State

VA has the largest land area among all the states, followed by WV and MD. DE and DC have the least areas. VA proportionately has a high number of loans and total loan amount. WV, however, has very less number of loans and very less total loan amount. WV doesn't seem like a lucrative market for our expansion. DE has low area and low everything else. So even DE doesn't seem like a lucrative market. Now comparing DC, MD and VA, all have proportionately higher total loan amounts and the number of loans compared to their land area. Of these three, even though VA has a high market base and high total loan amount, its land area is also large. So, from this analysis, MD and DC appear to be lucrative followed by VA. The year on year growth/decline also needs to be checked for these states.

Dividing the total loan amount and the Number of loans by Area of each state and plotting their graphs might give us a better idea. So, doing that, we have this graph.

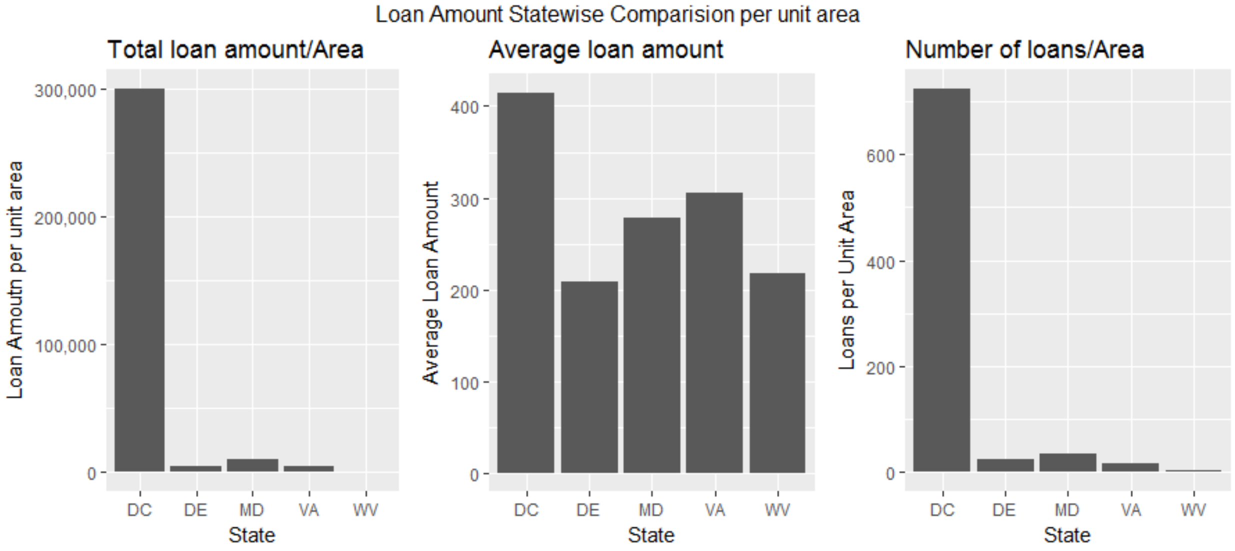


Figure 10: Loan Amount variation with state, per unit area

The DC area had the highest number of loans and loan amount per unit area and highest average loan amount overall. The other states have very less loan amount per unit area. MD is at the second highest place but it does not compare with the DC.

We know that DC has very high loan amounts. Checking whether the loan amount and number of loans are decreasing or increasing based on the following plot which is created with states as facets.

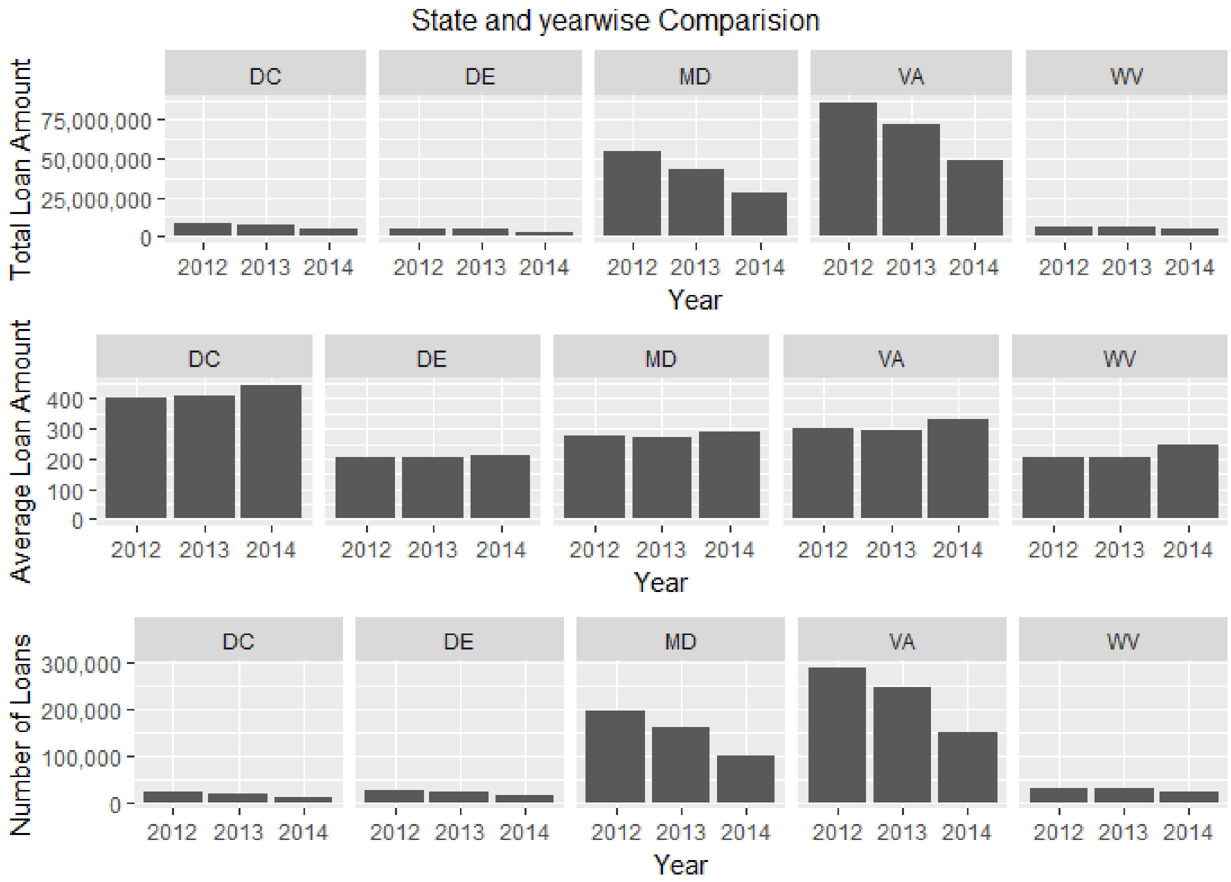


Figure 11: Loan Amount state and year wise variation

The average loan amount has been increasing in all the states, which is obvious because property prices increase every year. This increase is more prominent in DC. The number of loans, by 2014, almost became half of what they are in 2012. Compared to others, WV has shown less decline. Hence per this analysis, WV seems to be an emerging market. Even DC's total loan amounts and number of loans decreased but the decrease is not as drastic as MD or VA.

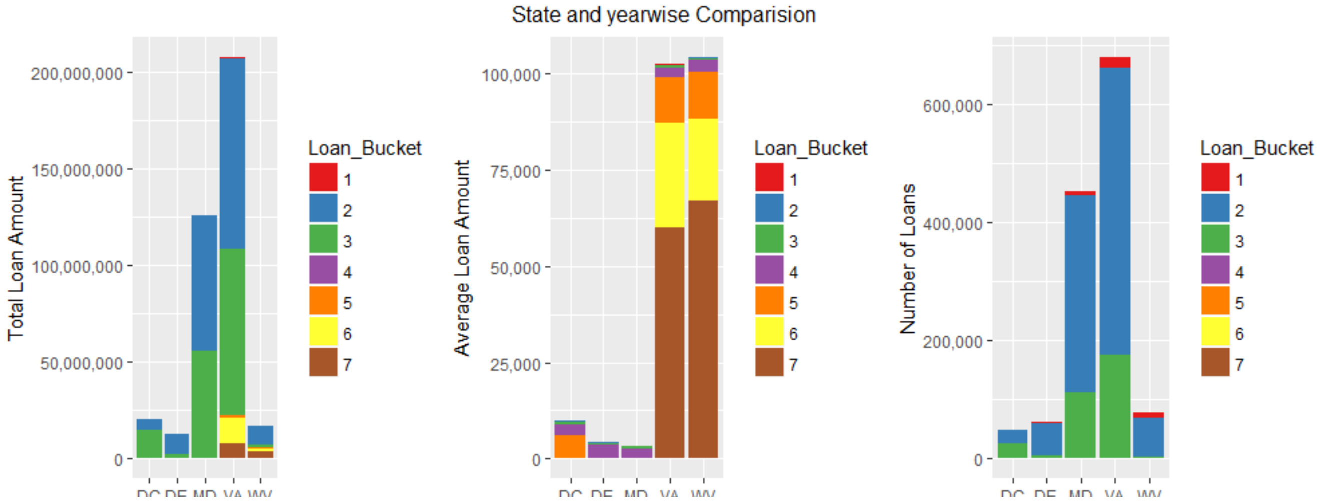
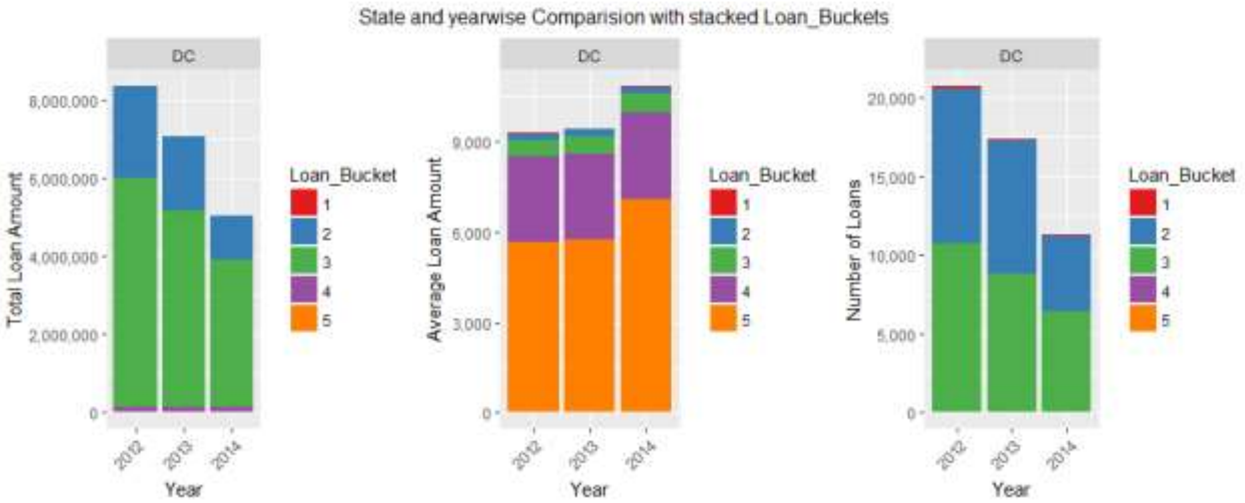


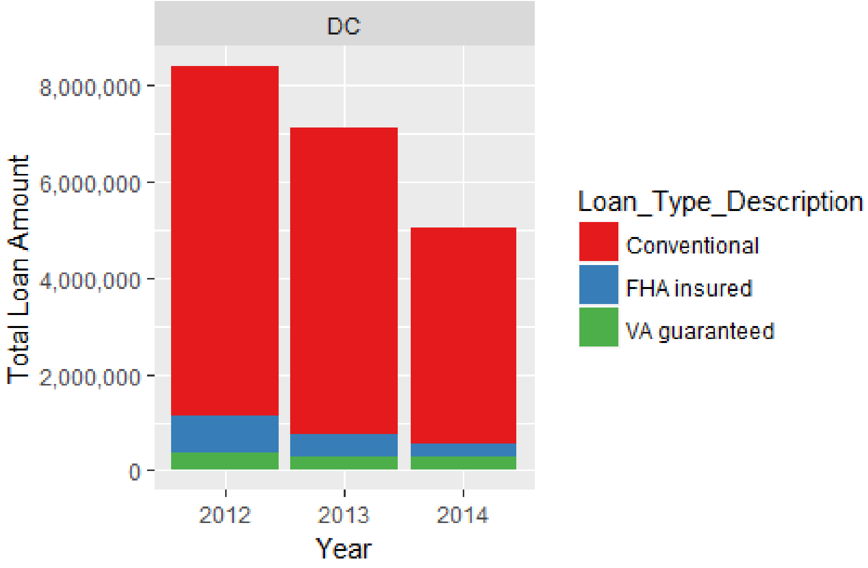
Figure 12:Loan Amount variation by State with Loan_Bucket in Stacked Bars

Using the buckets that we have created earlier, we can see that in our favorite state: DC, most loan takers belong to Bucket 3 followed by bucket 2. This might also be possible because most of the loan amounts are present in these buckets. Clearly, if Change Financial enters DC, it should target the customers in the buckets 2 & 3.

In the below plot, only our focus state: DC is visualized. From this, clearly, the total loan amount and number of loans have been decreasing because the people in the second and third bucket have been decreasing. From the first and third plots, we see that the decline in loan amount and number of loans in bucket 2 is more rapid than that of bucket 3. Bucket 3 also has the most loan amount for DC, so targeting people from bucket 3 is very important to penetrate DC market. Another observation is that DC does not have any loans from the high loan average loan buckets 6 and 7. Similarly, visualizing the



Now further analyzing DC's categorical aspects



Loan Type:

Most the loans are of conventional loan type in DC. VA guaranteed loans have remained stable while other loan types saw a reduction in total loan amount. Hence, Conventional and VA guaranteed loan types are important for Change Financial if they decide to enter DC market.

Figure 13: Loan Amount variation by year with Loan_Bucket in Stacked bars, for DC

Figure 14: DC Total Loan Amount variation with time and Loan Type in Stacked Bars

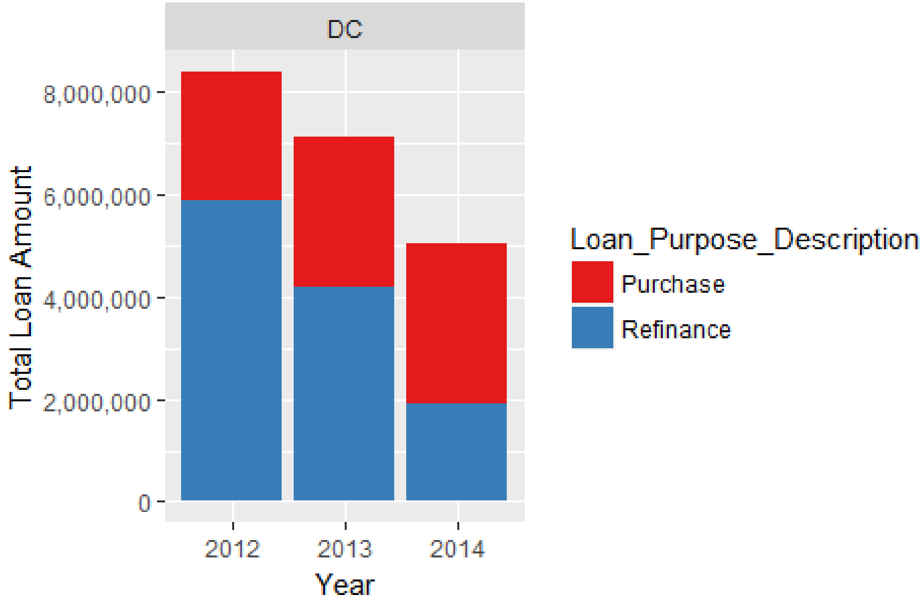


Figure 15: DC Loan Amount variation with year with Loan Purpose in Stacked bars

Loan Purpose:

The total loan amount for refinancing purpose has diminished drastically, reducing to almost one-third of its original value in 2012. However, the total loan amount for Purchase purpose has remained same or even slightly increased over the three years, Hence Change Financial must target the Purchase Loan purpose category.

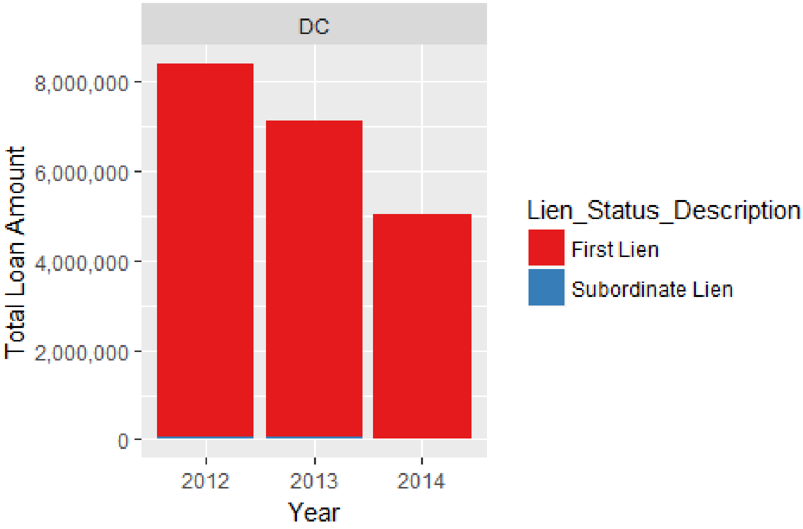


Figure 16: Loan Amount variation with year with Lien Status in stacked bars

Lien Status:

Almost all of the loans are of First Lien in DC. This observation is not as significant because the same criteria has been observed in all the states. Banks have the propensity to give loans when there is low risk involved. As First Lien involves higher chances of getting loan amount back when the borrower defaults on the loan, this factor is observed to comprise of almost 99%of the total loan amount.

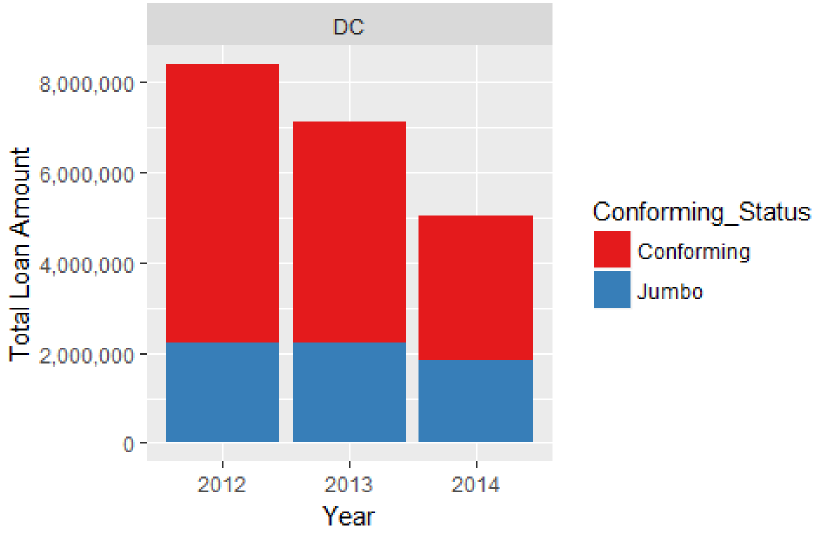


Figure 17: Loan Amount variation with year with Conforming status in stacked bars

Conforming Status:

The results of this graph might not be accurate as most of the Conforming Statuses were wrongly entered as Jumbo instead of Conforming. However, it appears as if Conforming is diminishing year by year. This can again be analyzed by removing the rows with NA's in Conforming Limit column.

Ultimately, though the market is going down in all states, DC seems to be the best state if Change Financial wants to enter a new market, as the average loan amount here is rising rapidly and the loan amount per unit area and number of loans are drastically high compared to other states. If Change Financial decides to enter these markets, it should focus on retaining the customers from bucket 3 in DC. Also, it should focus on Conventional, VA guaranteed loan types and Purchase loan purpose.

Some other interesting insights:

There are very few loans with subordinate Lien Status. Most of the loans are of First Lien Status. This is understandable because banks give loan whenever there is less risk involved. First Lien has the lowest risk, so the number of loans is also greater. The distribution of Number of loans by Lien Status can be seen via a histogram.

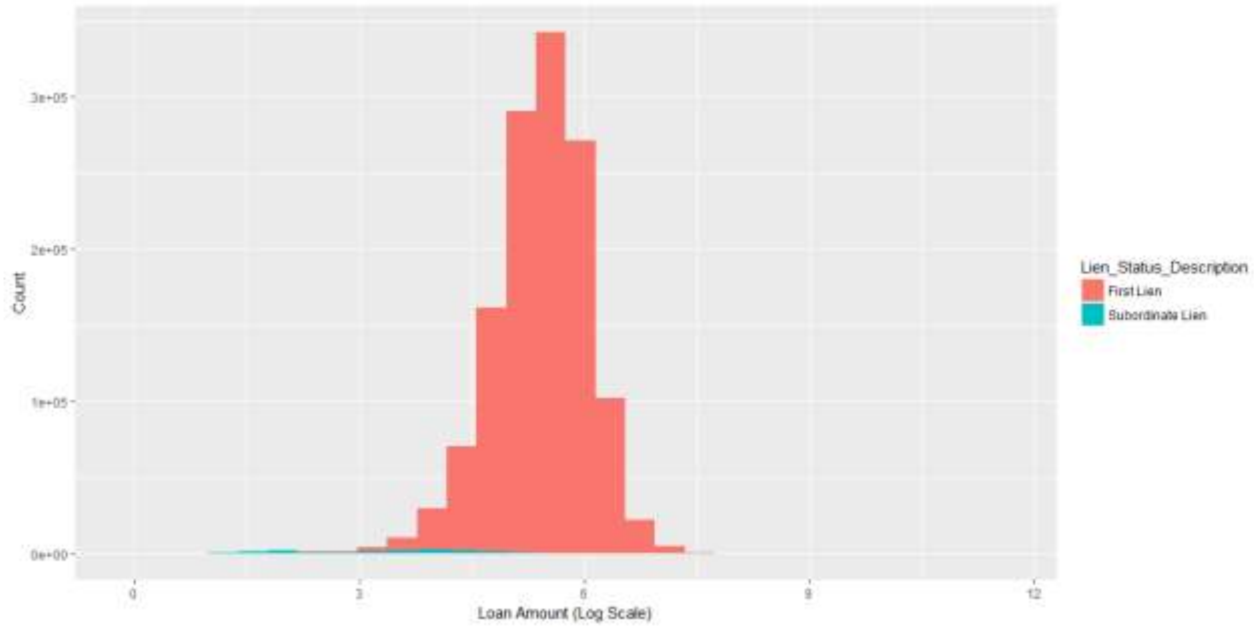


Figure 18: Distribution of Loan Amount divided by Lien Status

We can further analyze this by checking when people are taking subordinate Lien loans (These loans involve high risk, but the banks give this loan because they also have higher interest rate. A scatterplot between the Loan_Amount_000 and number of owner-occupied units gives an interesting insight. It shows that the first lien loans do not influence the number of owner-occupied units, however the subordinate lien loans show a sharp increase in the number of owner-occupied units with increase in loan

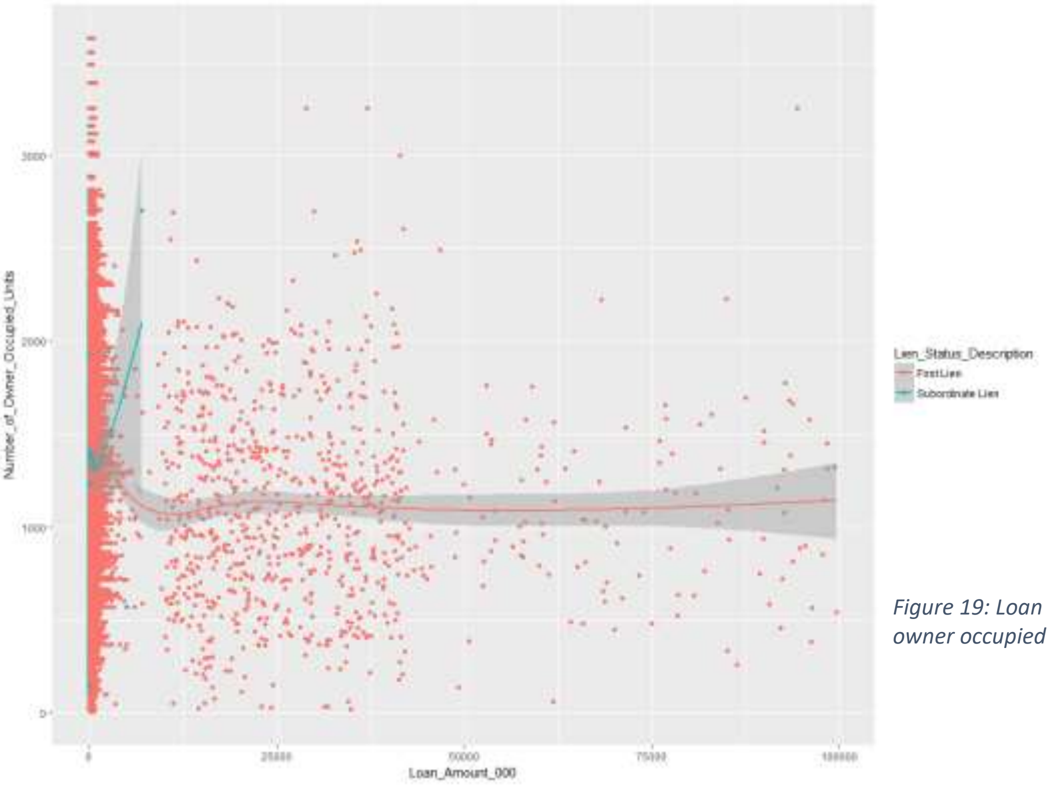


Figure 19: Loan Amount vs number of owner occupied units scatter plot

amount. This suggests that the subordinate Lien loans (at a higher interest rates) are taken by people who want to own a house.

We can also compare the conventional and Non-conventional loans to see if we can find some pattern. As seen in the adjacent image, the total loan amount of conventional loans is higher than the Non-conventional loan amounts. However, conventional loan amount is reducing more rapidly year by year, compared to the Non-conventional loan amount. We can further drill down to see which Loan types are increasing in popularity as shown in the figure below.

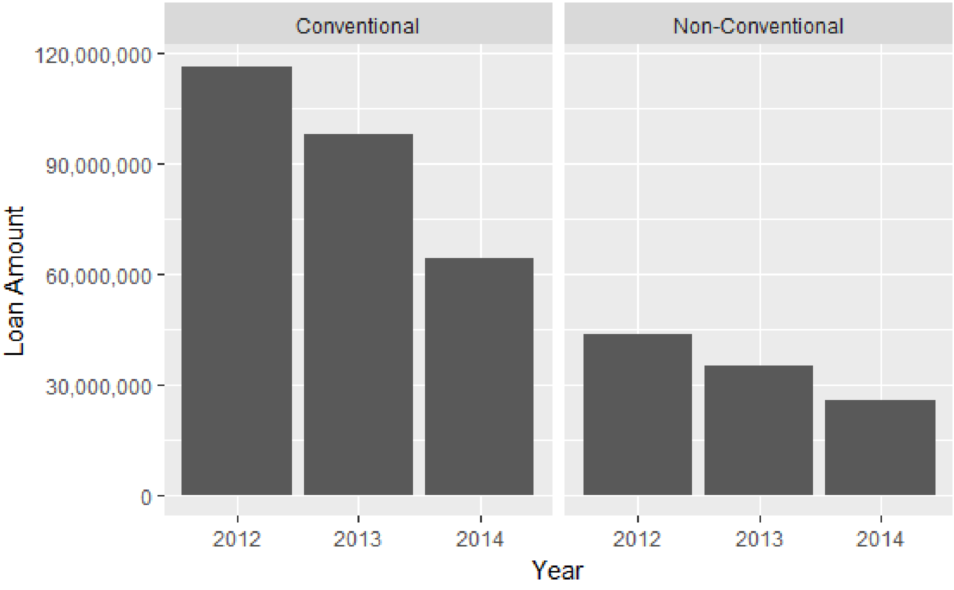


Figure 20: Loan Amount by year divided by Conventional Non Conventional loans

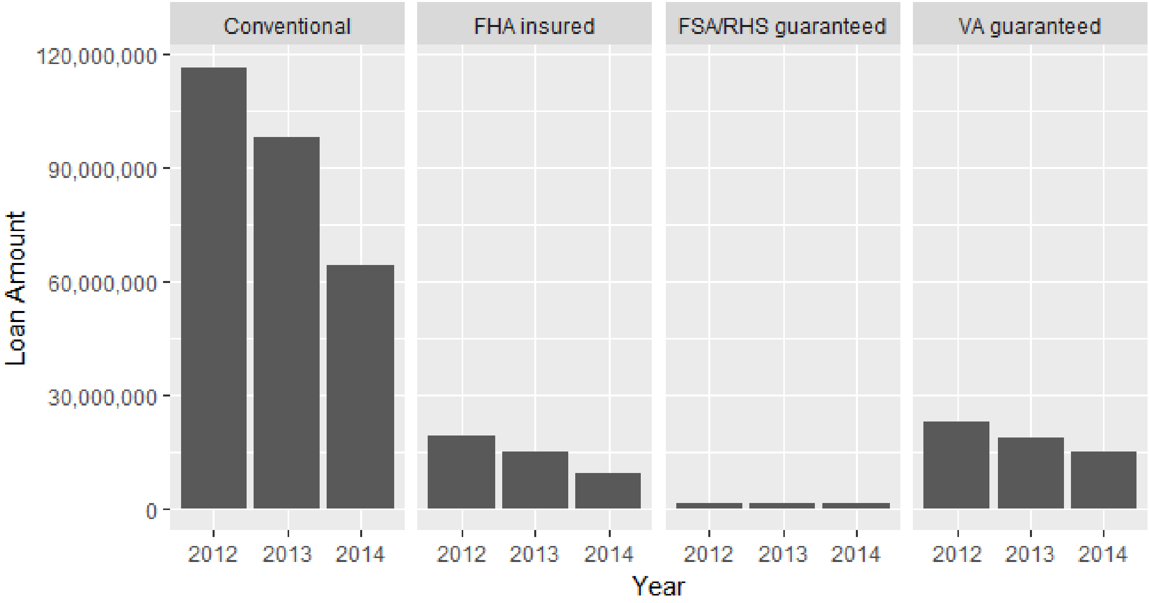
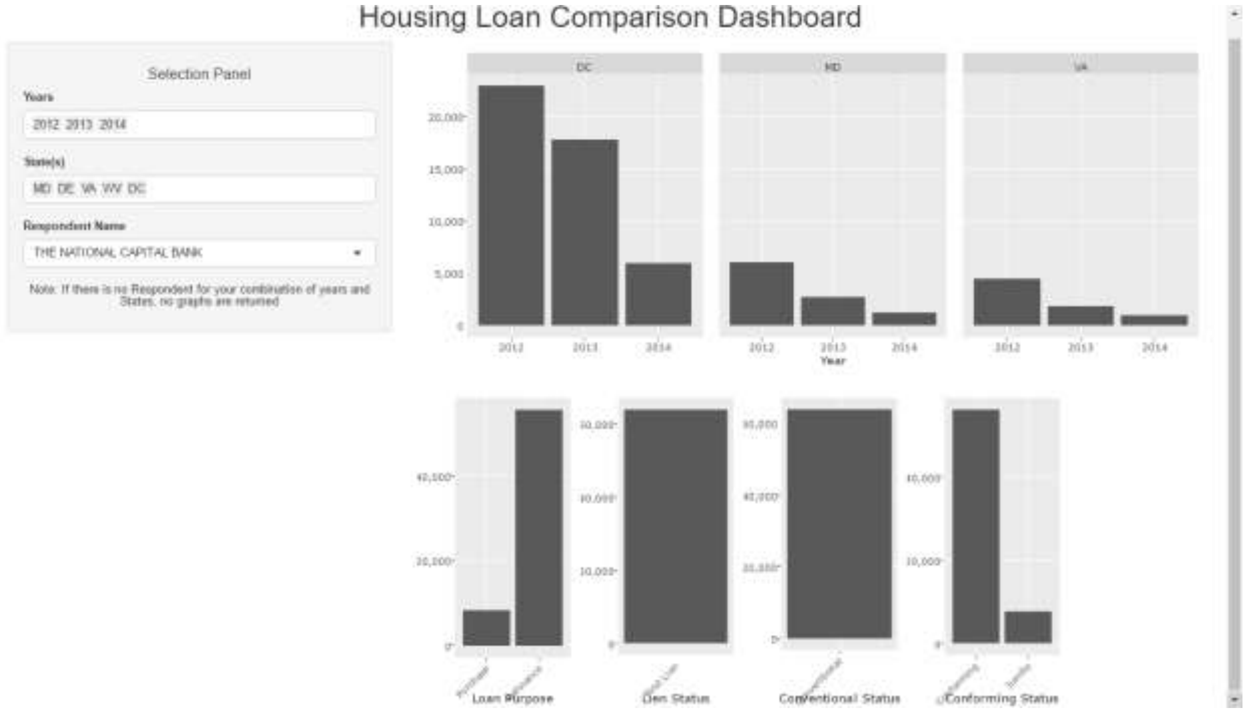


Figure 21: Loan Amount by year and Loan type

Therefore, the FSA/RHS-guaranteed loans are not falling as rapidly as others but they occupy a very little market space.

Interactive Shiny App

Shiny is a web application framework for R Studio. It can be used to create interactive web applications which can be hosted online. A shiny web application is created so that the VP of Change Financial can see the market share for a competitor in any given geography and year combination. The data is filtered based on user input and then it is aggregated. This aggregated data is used to create ggplots. The ggplots can be converted to plotly plots using ggplotly( ) function. Using this, five bar charts are created. The VP or end user can select multiple years, multiple states and a single Respondent name and all the charts get updated immediately. The user can also hover on the bar plots and it will show the value of each bar plot upon hover.



Housing Loan Comparison Dashboard

Selection Panel

Years: 2012 2013 2014

State(s): MD DE VA WV DC

Respondent Name: THE NATIONAL CAPITAL BANK

Note: If there is no Respondent for your combination of years and States, no graphs are returned

Figure 22: Screenshot of Shiny Dashboard

The first bar chart shows the variation of the total loan amount with the year, for different states. If there are no states, then only a single chart is shown. Else all the relevant states' charts are shown.

The second plot shows the Loan Purpose. The user can see what type of loans the competitors are lending.

The next plot shows Lien Status. If there are 2 liens present for the given selection criteria, then the total loan amount for both first lien and secondary lien are shown. Else, only one or none are shown (If selection criteria doesn't select any data, then nothing is shown)

The next plot shows Conventional Status. i.e. if the competitor is selling conventional or non-conventional loans. The final plot shows the conforming status, which can be conforming or jumbo.

For running this app, both the ui.r and server.r need to be opened in R studio and runApp( ) command Is to be used to start the app.