**HMDA Data Challenge**

1. **Data Munging**
2. **Merge Data**

**Code:**

# set a working directory

setwd('C:\\Users\\dalalshobhit\\Desktop\\CapitalOneData')

# Provide paths for loan data and institution data on your disk

loanDataPath <- "C:\\Users\\dalalshobhit\\Desktop\\CapitalOneData\\2012\_to\_2014\_loans\_data.csv"

institutionDataPath <- "C:\\Users\\dalalshobhit\\Desktop\\CapitalOneData\\2012\_to\_2014\_institutions\_data.csv"

mergedData <- merge(loanData, institutionData, by=c("Agency\_Code","Respondent\_ID", "As\_of\_Year"), all.x = TRUE)

1. **Provide a simple API of two functions:**

* **hmda\_init()**

**Code:**

# hmda\_init() implements a method to merge and get expanded data

hmda\_init <- function() {

loanData <- read.csv(loanDataPath, header = TRUE)

institutionData <- read.csv(institutionDataPath, header = TRUE)

mergedData <- merge(loanData, institutionData, by=c("Agency\_Code","Respondent\_ID", "As\_of\_Year"), all.x = TRUE)

return(mergedData)

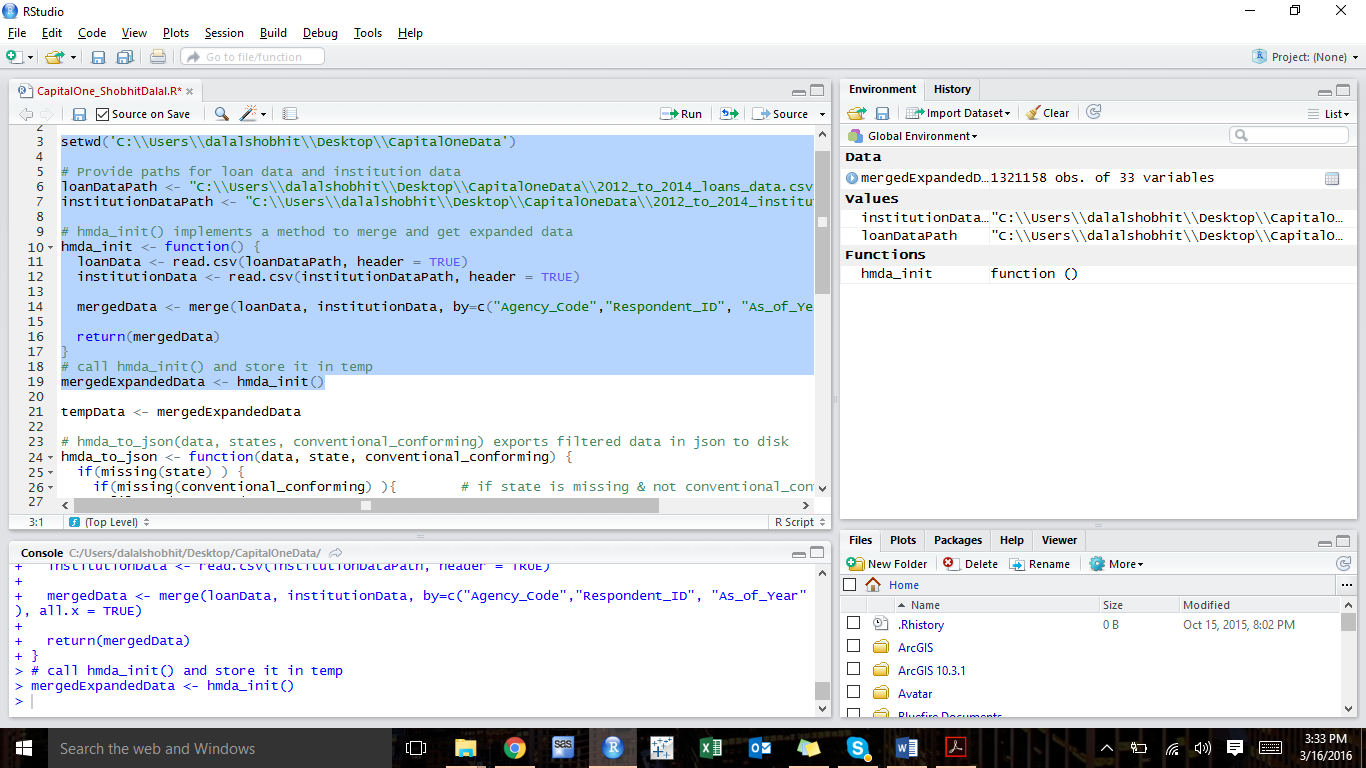
}

# call hmda\_init() and create a data frame mergedExpandedData

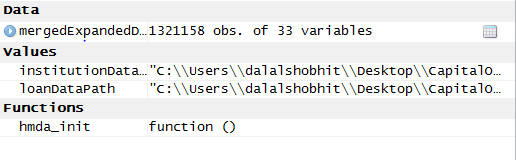
mergedExpandedData <- hmda\_init()

**Output:**

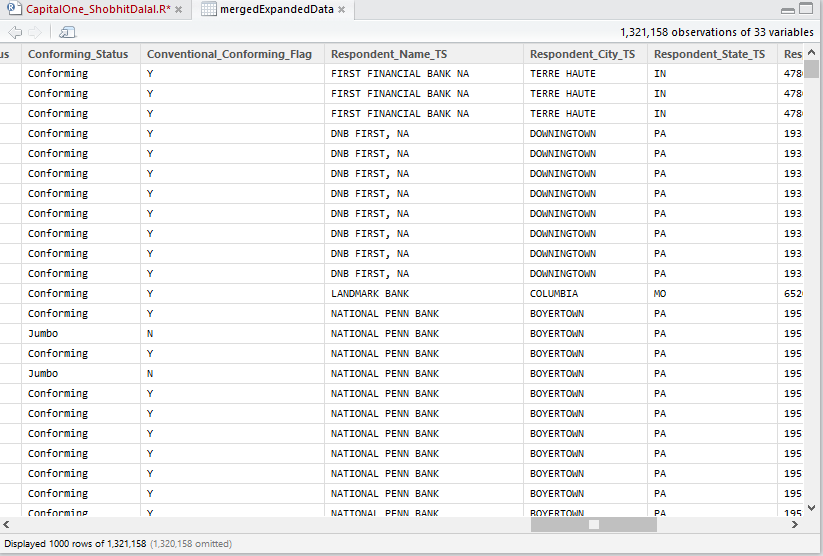
Execution of hmda\_init() function



Observation:



Merged Expanded Output:



* **hmda\_to\_json(data, state, conventional\_conforming)**

**Code:**

# install jsonlite package

install.packages(“jsonlite”, repos=<http://cran.r-project.org>)

# load jsonlite

library(jsonlite)

# hmda\_to\_json(data, states, conventional\_conforming) exports filtered data in json to disk

hmda\_to\_json <- function(data, state, conventional\_conforming) {

# if else loops for filtering data

if(missing(state) ) {

if(missing(conventional\_conforming) ){

filteredData <- data

}

else { # if state is missing & conventional\_conforming

filteredData <- subset(data, Conventional\_Conforming\_Flag == conventional\_conforming)

}

}

else {

if(missing(conventional\_conforming)) {

filteredData <- subset(data, State == state)

}

else { # if state is given & conventional\_conforming

filteredData <- subset(data, State == state & Conventional\_Conforming\_Flag == conventional\_conforming)

}

}

# Convert to JSON data

sink("jsonData.json")

cat(toJSON(filteredData))

sink()

# export the expanded dataset to disk

write.table(filteredData, "C:\\Users\\dalalshobhit\\Desktop\\CapitalOneData\\jsonData.json", sep=":")

return(filteredData)

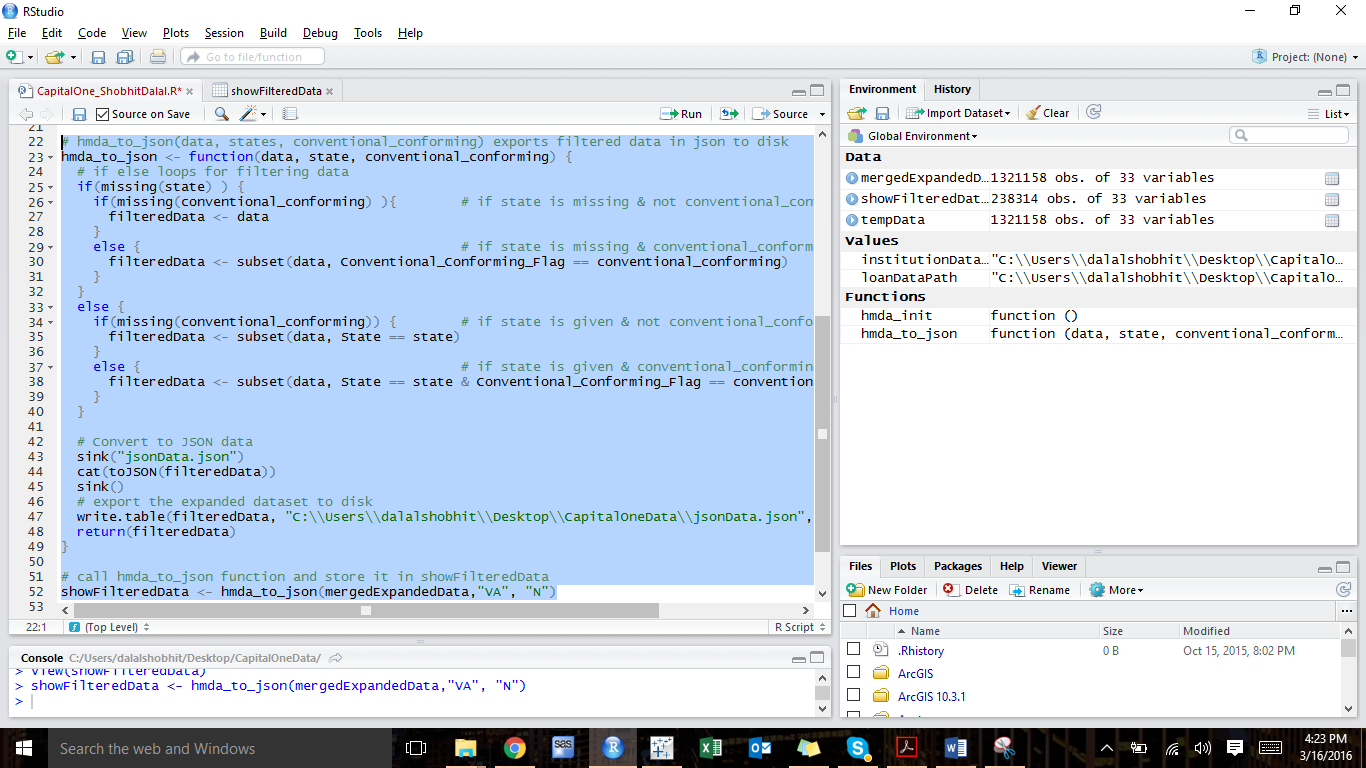
}

# call hmda\_to\_json function and store it in showFilteredData (Give a path on your local machine where you want to write this file)

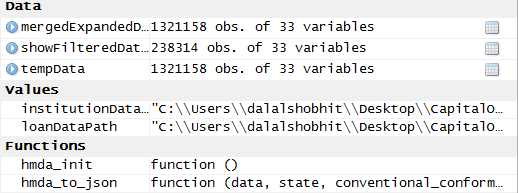
showFilteredData <- hmda\_to\_json(mergedExpandedData,"VA", “N”)

**Output:**

Execution of hmda\_to\_json() function:

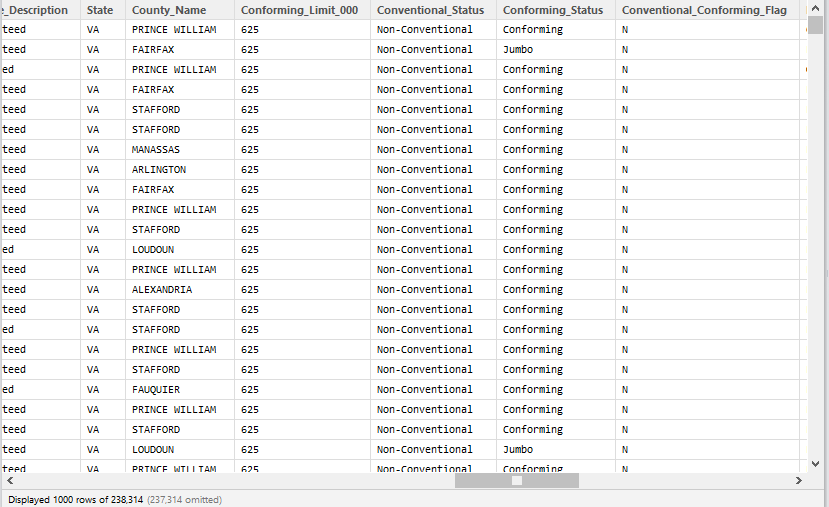


**Observation:**

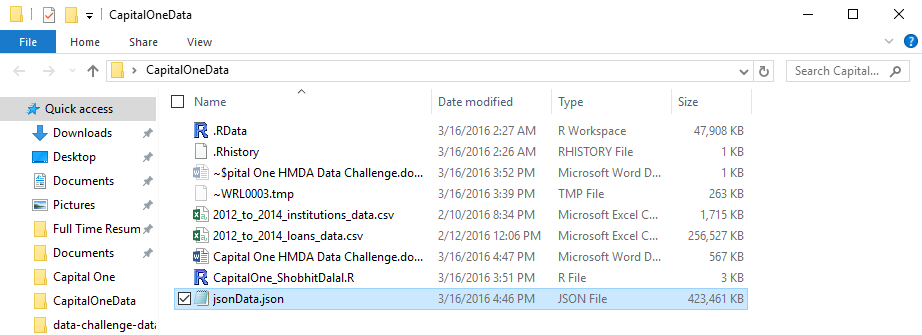


**Filtered Data Output:**

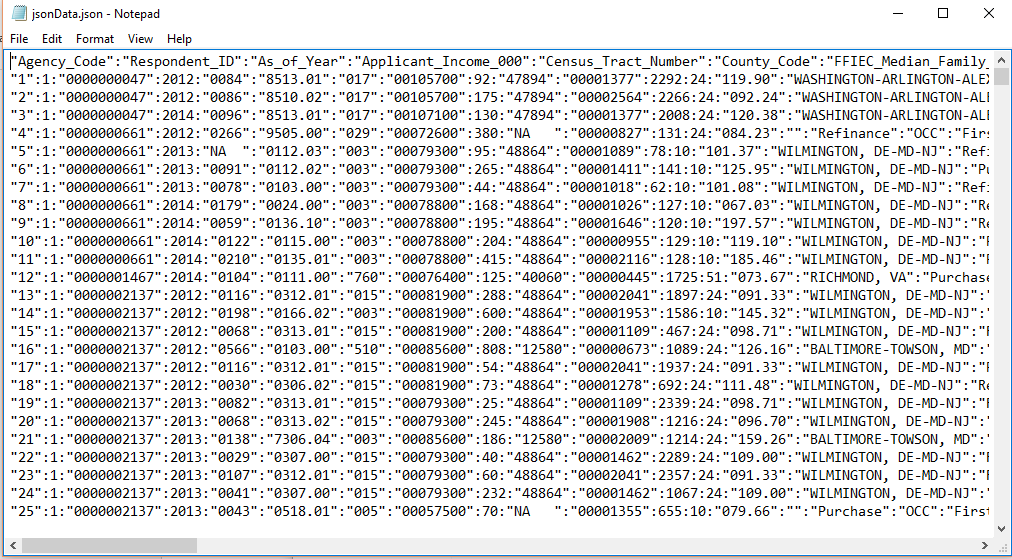
Both state(“VA”) and conventional\_conforming(“N”) filters are applied



**Output of filtered & expanded JSON file on disk:**



**JSON data file:**



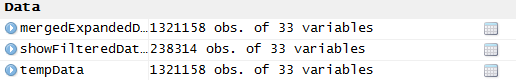
**Output for various test cases:**

* When both state & conventional\_conforming filters are applied:

showFilteredData <- hmda\_to\_json(mergedExpandedData,"VA", “N”)

**Output:**

238,314 observations with state=”VA” and conventional\_conforming=”N”

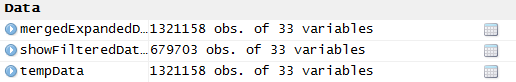


* When only state filter is applied:

showFilteredData <- hmda\_to\_json(mergedExpandedData,"VA")

**Output:**

679,703observations with state=”VA”

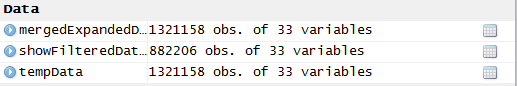


* When only conventional\_conforming filter is applied:

showFilteredData <- hmda\_to\_json(mergedExpandedData, , “Y”)

**Output:**

882,206observations with conventional\_conforming=”Y”

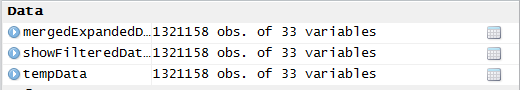


* When no filters are applied:

showFilteredData <- hmda\_to\_json(mergedExpandedData)

**Output:**

1,321,158observations with no filters applied



1. **Quality Check**
2. Apart from “Loan\_Amount\_000” and “Respondent\_Name” two columns I think are most important to monitor are:

* **MSA\_MD** – This data will be helpful to analyze the type of loans and amount of loans that are originated or purchased within a specific MSA or MD. It will be used to analyze which location will be more profitable, and where many loans are applied and originated.
* **Loan\_Type\_Description** – It states if a loan is conventional or non-conventional loan. Conventional loan indicates these are ideal applicants with good or excellent credit. Non-conventional loans can be FHA, FSA or VA loans and they are for low income borrowers or for borrowers with low credit scores. Based on the type of loan we can identify categories of people in a specific region and what kind of loans will be in more demand in that region.

1. **Quality assessment on ‘Loan\_Amount\_000’**

On analyzing ‘Loan\_Amount\_000’ data, it is noticed that there are entries with values of 5 digit values. To clean the data and make the entries uniform, a function cleaning\_LoanAmount(data) is created.

This function will replace the 5 digit value by dividing it by 1000 to correctly represent them.

**Code:**

# Create a function cleaning\_LoanAmount() for Quality Assssment of Loan\_Amount\_000

cleaning\_LoanAmount <- function(clean\_data) {

clean\_data$Loan\_Amount\_000=as.numeric(as.character(clean\_data$Loan\_Amount\_000))

class(clean\_data$Loan\_Amount\_000)

# clean\_data$Loan\_Amount\_000[ which(clean\_data$Loan\_Amount\_000 < 999)]

clean\_data$Loan\_Amount\_000[clean\_data$Loan\_Amount\_000>999 & clean\_data$Loan\_Amount\_000<99999] =

clean\_data$Loan\_Amount\_000/1000

return(clean\_data)

}

# Call cleaning\_LoanAmount(data) function for Quality assessment of Loan\_Amount\_000

clean\_data <- cleaning\_LoanAmount(clean\_data)

**Metadata:**

Loan\_Amount\_000: Enter value in thousands. For eg. 95,000 should be entered as 95

Applicant\_Income\_000: Enter value in thousands. For eg. 300,000 should be entered as 300

1. **Craft a visual data narrative**
2. **Market size by year & state**

* **Comparison of all types of loans grouped by year**

**Code:**

library(sqldf)

# Create SQL queries for each type of loan grouped by year

conventionalLoanSQL <- "SELECT As\_of\_Year,count(\*) AS Conventional\_Loans FROM mergedExpandedData WHERE Loan\_Type\_Description LIKE 'Conventional' GROUP BY As\_of\_Year"

VALoanSQL <- "SELECT As\_of\_Year,count(\*) AS VA\_Loans FROM mergedExpandedData WHERE Loan\_Type\_Description LIKE 'VA guaranteed' GROUP BY As\_of\_Year"

FSA\_RHS\_LoanSQL <- "SELECT As\_of\_Year,count(\*) AS FSA\_RHS\_Loans FROM mergedExpandedData WHERE Loan\_Type\_Description LIKE 'FSA/RHS guaranteed' GROUP BY As\_of\_Year"

FHALoanSQL <- "SELECT As\_of\_Year,count(\*) AS FHA\_Loans FROM mergedExpandedData WHERE Loan\_Type\_Description LIKE 'FHA insured' GROUP BY As\_of\_Year"

# Create tables for each type of loans

conventionalLoans <- sqldf(conventionalLoanSQL)

VALoans <- sqldf(VALoanSQL)

FSA\_RHSLoans <- sqldf(FSA\_RHS\_LoanSQL)

FHA\_Loans <- sqldf(FHALoanSQL)

# Query to merge all tables

all\_LoansSQL <- "SELECT c.As\_of\_Year, c.Conventional\_Loans, v.VA\_Loans, f.FSA\_RHS\_Loans, fh.FHA\_Loans

FROM ((conventionalLoans c INNER JOIN VALoans v

ON c.As\_of\_Year = v.As\_of\_Year) INNER JOIN FSA\_RHSLoans f

ON c.As\_of\_Year = f.As\_of\_Year) INNER JOIN FHA\_Loans fh

ON c.As\_of\_Year = fh.As\_of\_Year"

allLoans <- sqldf(all\_LoansSQL)

library(ggplot2)

library(reshape)

dfm <- melt(allLoans[,c('As\_of\_Year','Conventional\_Loans','VA\_Loans', 'FSA\_RHS\_Loans', 'FHA\_Loans')],id.vars = 1)

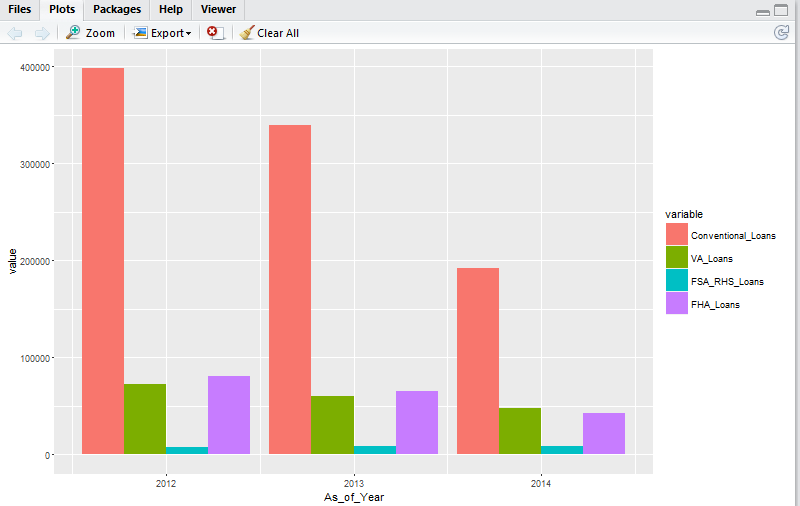
# Bar plot each type of loans grouped by year

options(scipen=10000)

ggplot(dfm, aes(x=As\_of\_Year,y=value)) +

geom\_bar(stat='identity', aes(fill=variable), position='dodge')

**Output:**



**Comparison of all types of loans grouped by year**

* **Percentage share of loans for each state**

**Code:**

# Pie chart of total loans grouped by State (no segregation of loans)

total\_loansSQL <- "SELECT State, COUNT(\*) AS Tot\_loans FROM mergedExpandedData GROUP BY State"

total\_loans <- sqldf(total\_loansSQL)

Loan\_Count <- as.vector(total\_loans['Tot\_loans'])

State <- as.vector(total\_loans['State'])

pct <- round(Loan\_Count/sum(Loan\_Count)\*100)

pct <- as.integer(unlist(pct))

State <- unlist(State)

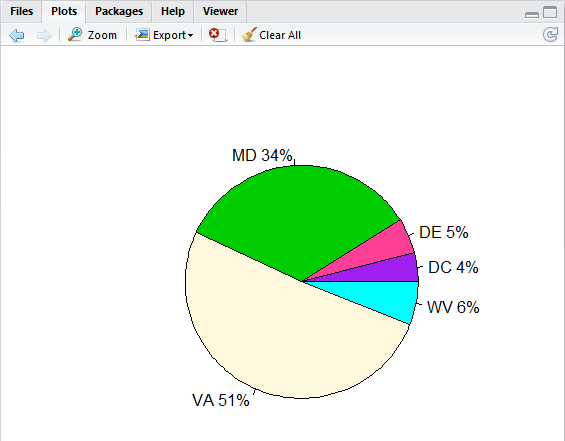
lbls <- paste(State, pct) # add percents to labels

lbls <- paste(lbls,"%",sep="") # ad % to labels

pie(pct,labels = lbls, col=c("purple", "violetred1", "green3",

"cornsilk", "cyan", "white"))

**Output:**



**Percentage share of loan application in 5 states**

* **Conventional conforming vs. non conventional conforming loans grouped by state**

**Code:**

# Create SQL queries for conventional\_conforming loans vs. not conventional\_conforming loans grouped by state

ConventionalConformingLoanSQL <- "SELECT State,count(\*) AS ConventionalConforming\_Loans FROM mergedExpandedData WHERE Conventional\_Conforming\_Flag LIKE 'Y' GROUP BY State"

ConventionalConformingLoans <- sqldf(ConventionalConformingLoanSQL)

NonConventionalConformingLoanSQL <- "SELECT State,count(\*) AS NonConventionalConforming\_Loans FROM mergedExpandedData WHERE Conventional\_Conforming\_Flag LIKE 'N' GROUP BY State"

NonConventionalConformingLoans <- sqldf(NonConventionalConformingLoanSQL)

totalLoansSQL <- "SELECT c.State, c.ConventionalConforming\_Loans, n.NonConventionalConforming\_Loans

FROM ConventionalConformingLoans c INNER JOIN NonConventionalConformingLoans n

ON c.State = n.State

ORDER BY 1"

totalLoans <- sqldf(totalLoansSQL)

totLoans <- melt(totalLoans[,c('State','ConventionalConforming\_Loans','NonConventionalConformin g\_Loans')],id.vars = 1)

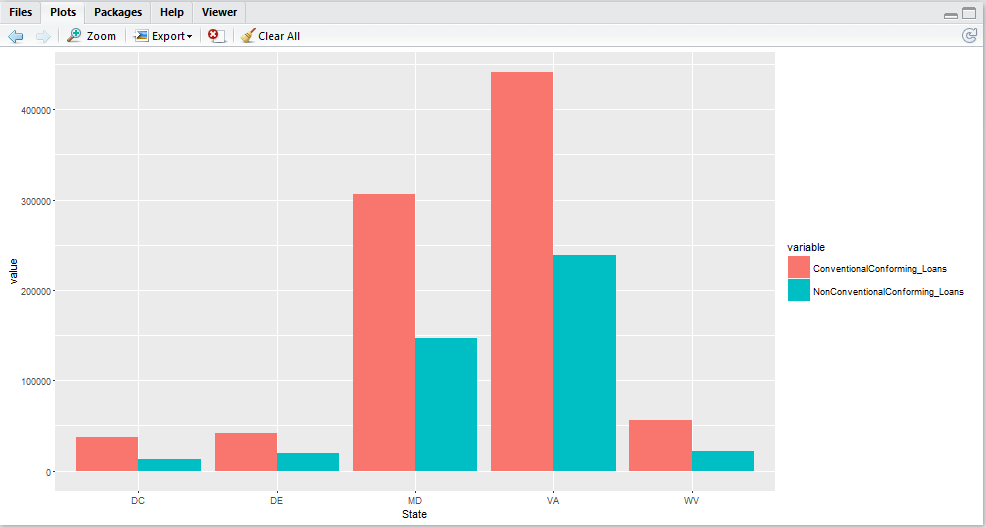
# Bar plot each type of loans grouped by year

options(scipen=10000)

ggplot(totLoans, aes(x=State,y=value)) +

geom\_bar(stat='identity', aes(fill=variable), position='dodge')

**Output:**



**Conventional\_conforming vs. non\_conventional\_conforming loans grouped by state**

1. **Take it a step further!**

* **Clean Applicant\_Income\_000 for NA values**

**Code:**

# Create a function cleaningData() for Quality Assssment of Applicant\_Income\_000

cleaningData <- function(mergedExpandedData) {

mergedExpandedData$Applicant\_Income\_000[mergedExpandedData$Applicant\_Incom e\_000=='NA '] = '0'

mergedExpandedData$Applicant\_Income\_000=as.numeric(as.character(mergedExpand edData$Applicant\_Income\_000))

class(mergedExpandedData$Applicant\_Income\_000)

# Cleaning the data

meanIncome <- mean(mergedExpandedData$Applicant\_Income\_000)

mergedExpandedData$Applicant\_Income\_000[mergedExpandedData$Applicant\_Incom e\_000==0] = meanIncome

return(mergedExpandedData)

}

clean\_data <- cleaningData(mergedExpandedData)

**Explanation:**

There are many rows with applicant income = “NA “.

‘cleaningData(data)’ function is created to treat this NA values. NA values are replaced with mean value of applicant income.

* **Average income of applicant grouped by State**

**Code:**

# Mean applicant income grouped by state

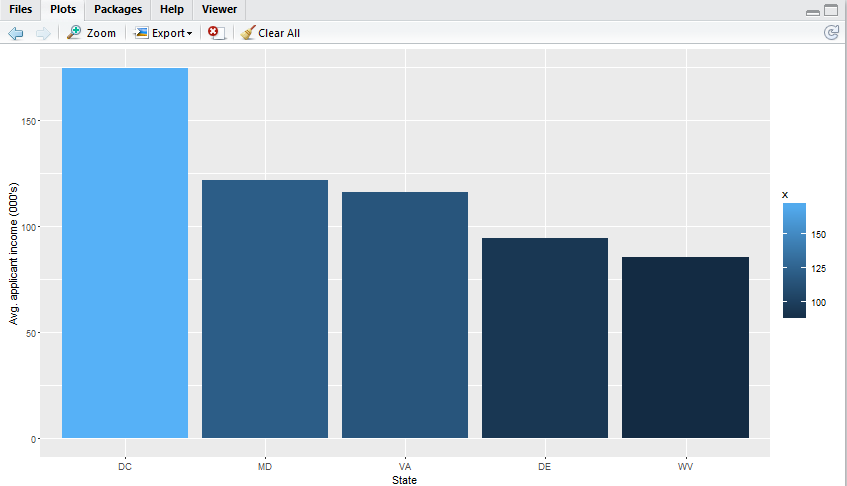
mean\_income\_state <- aggregate(clean\_data$Applicant\_Income\_000,by=list(State=clean\_data$State),mean,na. rm=TRUE)

ggplot(mean\_income\_state, aes(x=reorder(State,-x),y=x)) +

geom\_bar(stat='identity', aes(fill=x)) +

labs(x="State", y="Avg. applicant income")

**Output:**



**Average applicant income by state**

**We can see that the average income of an applicant in DC is the highest while percentage share of loan applications is lowest in DC. But, Maryland and Virginia have 2nd and 3rd highest average income of an applicant respectively.**

**They also consists of a major share of loan applications of 51% & 34% respectively. Hence, we can say that from the results of visualizations of loan data of these states, we need to further analyze data of Virginia and Maryland to see if they can be a good place to expand with promising market.**

* **Relationship between no. of loan applications and Conforming Loan limit for a given County for Maryland**

**Code:**

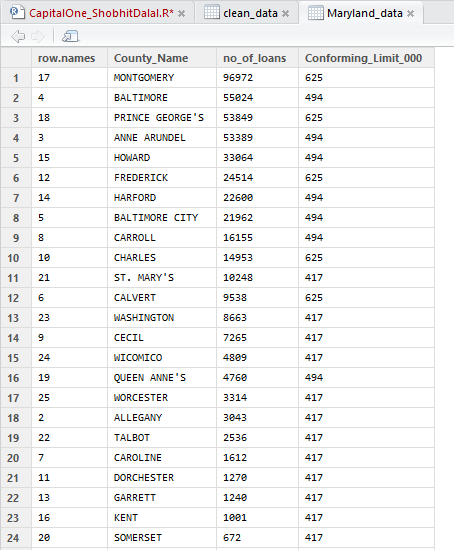
# Create SQL queries for analyzing data for Maryland

Maryland\_dataSQL <- "SELECT County\_Name,count(\*) AS no\_of\_loans,Conforming\_Limit\_000 FROM clean\_data WHERE State LIKE 'MD' GROUP BY County\_Name"

Maryland\_data <- sqldf(Maryland\_dataSQL)

Maryland\_data <- Maryland\_data[order(-Maryland\_data$no\_of\_loans),]

**Output:**



* **Relationship between no. of loan applications and Conforming Loan limit for a given County for Maryland**

**Code:**

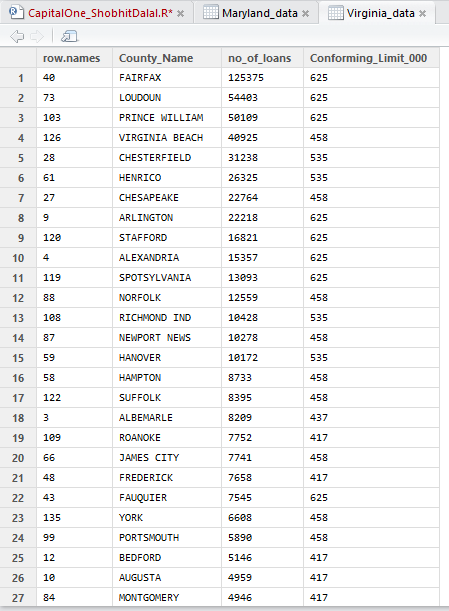
# Create SQL queries for analyzing data for Virginia

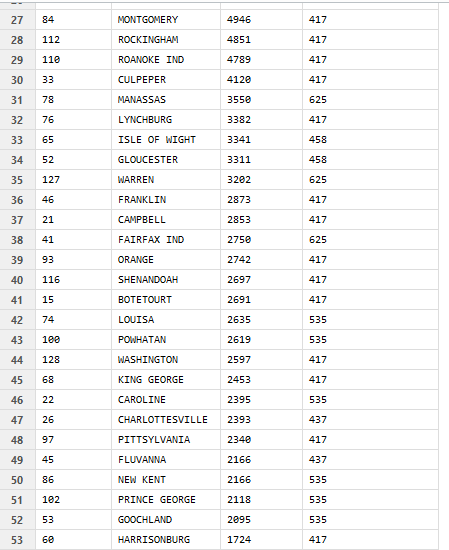
Virginia\_dataSQL <- "SELECT County\_Name,count(\*) AS no\_of\_loans,Conforming\_Limit\_000 FROM clean\_data WHERE State LIKE 'VA' GROUP BY County\_Name"

Virginia\_data <- sqldf(Virginia\_dataSQL)

Virginia\_data <- Virginia\_data[order(-Virginia\_data$no\_of\_loans),]

**Output:**





**Insights:**

**For both states Maryland and Virginia, we see that, as the single limit conforming loan limit for a given county decreases, number of loan applications also decreases.**

* **Average loan amount in various state (all loans)**

**Code:**

# Average loan amount grouped by state

mean\_loanAmt\_state <- aggregate(clean\_data$Loan\_Amount\_000,by=list(State=clean\_data$State),mean,

na.rm=TRUE)

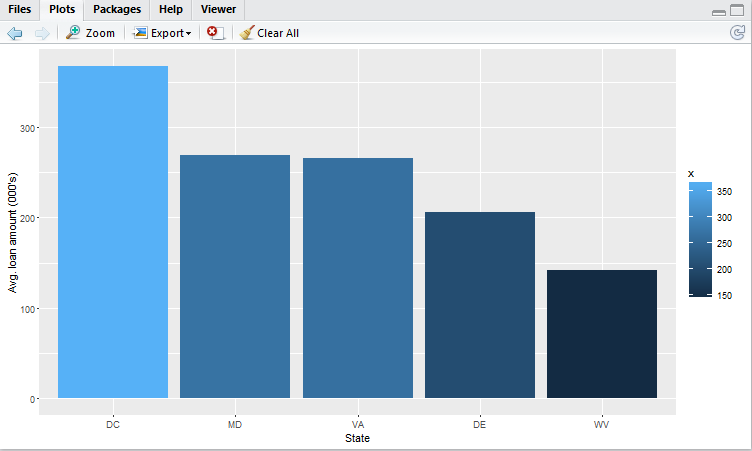
mean\_loanAmt\_state <- mean\_loanAmt\_state[order(-mean\_loanAmt\_state$x),]

ggplot(mean\_loanAmt\_state, aes(x=reorder(State,-x),y=x)) +

geom\_bar(stat='identity', aes(fill=x)) +

labs(x="State", y="Avg. loan amount (000's)")

**Output:**



**Insights:**

**Maryland and Virginia have high average loan amount of 269.43 thousand and 265.31 thousand respectively. But, District of Columbia has the highest average loan amount of 368.16 thousand.**

**Recommendation:**

**Maryland and Virginia together consists of 85% of the total loan applications among five states. One of the reasons for such a large number of loan applications may be high average income of applicants. These two states have 2nd and 3rd highest average applicant income among five states. Thus, due to large number of loan applications and high income of applicants, Maryland and Virginia looks promising regions for home loans market. Change Financial should initially consider entering home loans market in these regions. As number of applications are more in counties with higher single unit conforming loan limit, they should first target counties with higher conforming limit for entering into a new market.**

**Also, Change Financial should further analyze the data of District of Columbia. DC have the highest average loan amount and highest average income of an applicant. Still, it has the lowest percentage share of loan applications among the five states.**