**Tutorial Data analysis with Hive and Pig**

**Of Health Insurance Marketplace data**

**Using Powerview and Tableau**

By Kulwinder Kaur ([kkaur7@calstatela.edu](mailto:kkaur7@calstatela.edu)), Jalashri Raval ([jraval @calstatela.edu](mailto:rdonda@calstatela.edu)) and Moath Aldarrab ([maldarr@calstatela.edu](mailto:maldarr@calstatela.edu))

Prof. Jongwook Woo ([jwoo5@calstatela.edu](mailto:jwoo5@calstatela.edu))

12/10/2016

**Objectives**

This tutorial will explain how to run analysis and data manipulation in the Hive and pig Hadoop environment by running a step by step analysis on Health Insurance Marketplace in the US. Then the results will be visualized by Powerview, Excel and Tableau in 2D graphs. Throughout this tutorial, we will concentrate on the average rates for the individuals in different states and in different years.

In this tutorial, we will use different Hive and Pig Operations including:

* How to download data to the local systems in Bluemix BigInsights.
* Then, how to upload it to HDFS.
* We will figure out how to manipulate and analyze data in HDFS using HiveQL. Create tables, Showing contents of the tables and use of JOIN and GROUP BY with filters.
* We will also show how to visualize the result in Powerview.
* Define a relation with a specific schema
* Define a new relation from an existing relation
* Select specific columns from within a relation
* Group the data by using ‘GROUP BY’ and ‘FLATTEN’

**Introduction**

Health Insurance Industry is one of the biggest and all time booming industry across the word. But, there are many ambiguities and dilemmas for common people for understanding which plan is most suitable and how much average rate should be paid, which criteria should be considered for which age-group. A large scale data set facilitates the transparency of analysis and visualization. In this term paper we delve into the Health Insurance Marketplace data using Hive and Hive Ql codes to better understand the average rates of Health plans, average rates of dental plans only in particular state, comparing the rates and comparing the tobacco rates for different age groups for different years. In this tutorial, you’ll learn how HDInsight can be used to process data available in health insurance marketplace in order to comparisons of the rates.

**Prerequisites**

* IBM Bluemix account with an already created hadoop cluster.
* You must have Microsoft Excel 2010, 2013 or 2016 installed.
* You must have Winscp installed for file transfer.

**OUTLINE**

1. Download the data
2. Upload the data files to Hadoop file system (HDFS)
3. Make directories and subdirectories to arrange the files in HDFS
4. Write the HIVEQL code for analysis
   1. Create tables with data in .csv files
   2. Calculate average rates using AVG() function and GROUP BY age and State.
   3. Comparison of average rates of health plans provided by different providers using JOIN on tables.
   4. Calculate average rates of Dental only plans in a particular state.
5. Import output to excel and visualization in Powerview.
6. Write the pig script for analysis
   1. Read from .csv file
   2. Create a relation
   3. Group the data
   4. Store the output
7. Import the output in Tableau
8. Generate the graphs

**Part 1**

**STEP 1: Download the data**

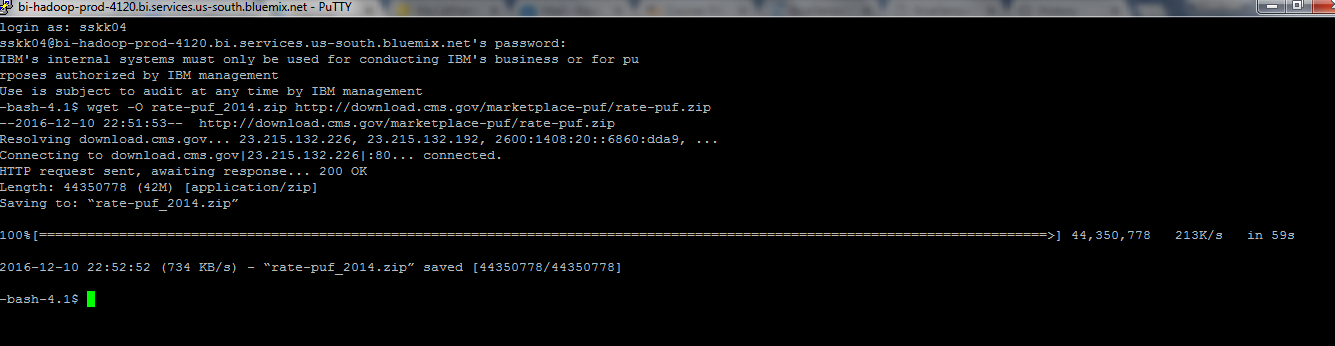
Download the Health Insurance Market place data from The Centers for Medicare & Medicaid Services (CMS) to your local file system in the cluster. Use this command at the shell command after creating the ssh session.

wget -O rate-puf\_2014.zip <http://download.cms.gov/marketplace-puf/rate-puf.zip>

wget -O rate-puf\_2015.zip <http://download.cms.gov/marketplace-puf/rate-puf.zip>

wget -O network-2014.zip <http://download.cms.gov/marketplace-puf/2014/network-puf.zip>

wget -O rate-puf\_2016.zip <http://download.cms.gov/marketplace-puf/2016/rate-puf.zip>



unzip the content:

unzip rate-puf\_2014.zip

unzip rate-puf\_2015.zip

unzip rate-puf\_2016.zip

unzip network-2014.zip



For a better file organization, it is better to rename the file. So, to rename run this Linux command:

mv Rate\_PUF\_2016.csv Rate\_PUF\_2016.csv

**STEP 2: Upload the data files to Hadoop file system (HDFS)**

You need to run the following HDFS commands to upload the file that we just unzipped, but first we need to make a directory for our project:

hdfs dfs -mkdir projectfile

hdfs dfs -mkdir projectfile/2014

hdfs dfs -mkdir projectfile/2014/rate/Rate\_puf2014

hdfs dfs -mkdir projectfile/2014/net

hdfs dfs -mkdir projectfile/2015

hdfs dfs -mkdir projectfile/2015/rate

hdfs dfs -mkdir projectfile/2016

You can check if the directory has been made by running the following command:

hdfs dfs -ls

Now we can upload the .csv file to HDFS system:

hdfs dfs -put Rate\_PUF2014.csv projectfile/2014/rate/Rate\_puf2014

hdfs dfs -put Network\_PUF.csv projectfile/2014/net

hdfs dfs -put Rate\_PUF2015.csv projectfile/2015/rate

hdfs dfs -put Rate\_PUF\_2016.csv projectfile/2016

we can make a quick check to make sure we uploaded the file successfully:

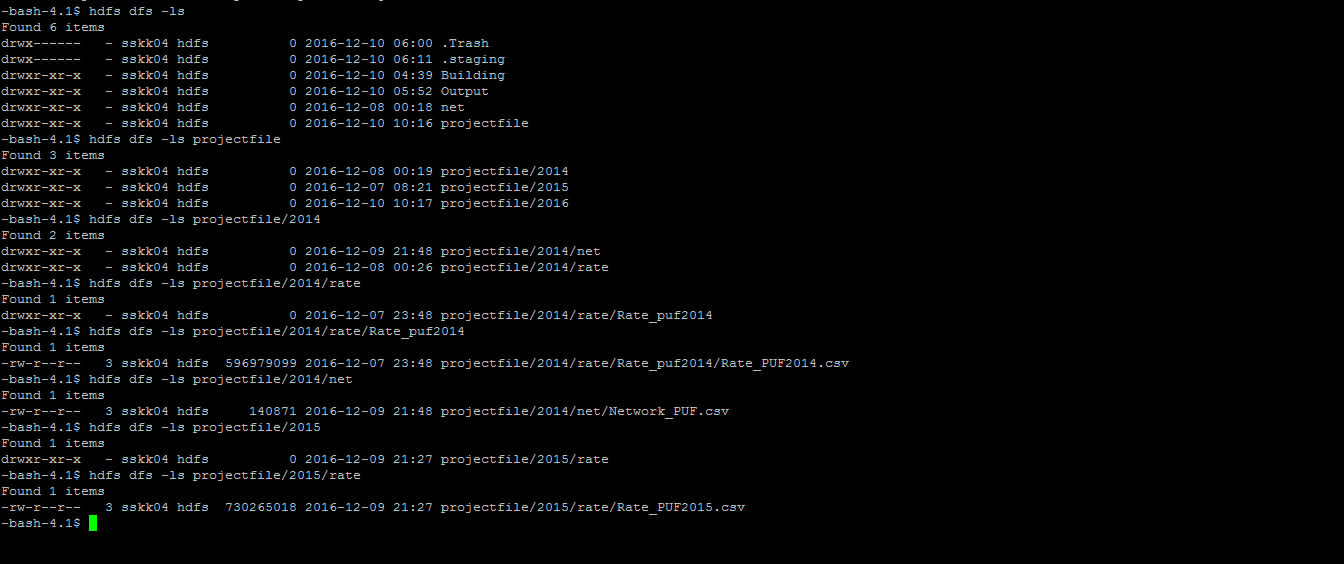
hdfs dfs -ls projectfile

hdfs dfs -ls projectfile/2014/rate/Rate\_puf2014

hdfs dfs -ls projectfile/2014/net

hdfs dfs -ls projectfile/2015/rate

hdfs dfs -ls projectfile/2016



**STEP 3: Writing HIVEQL code:**

The following Hive statement creates an external table that allows Hive to query data stored in HDFS. External tables preserve the data in the original file format, while allowing Hive to perform queries against the data within the file.

1. **Create tables:**

Make tables for rates 2014 and 2015 and calculate average rates of different age groups.For that, We need to go in hive:

hive

Create Table for 2014 from Rate\_PUF2014 in user/sskk04/projectfile/2014/rate/Rate\_puf2014:

CREATE EXTERNAL TABLE IF NOT EXISTS rate2014(

BusinessYear string,

StateCode string,

IssuerId string,

SourceName string,

VersionNum string,

ImportDate string,

IssuerId2 string,

FederalTIN string,

RateEffectiveDate string,

RateExpirationDate string,

PlanId string,

RatingAreaId string,

Tobacco string,

Age string,

IndividualRate int,

IndividualTobaccoRate int,

Couple int,

PrimarySubscriberAndOneDependent int,

PrimarySubscriberAndTwoDependents int,

PrimarySubscriberAndThreeOrMoreDependents int,

coupleAndOneDependent int,

CoupleAndTwoDependents int,

CoupleAndThreeOrMoreDependents int,

RowNumber string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sskk04/projectfile/2014/rate/Rate\_puf2014'

TBLPROPERTIES ('skip.header.line.count'='2');

Create table for rate2015 from Rate\_PUF2015.csv in user/sskk04/projectfile/2015/rate:

CREATE EXTERNAL TABLE IF NOT EXISTS rate2015(

BusinessYear string,

StateCode string,

IssuerId string,

SourceName string,

VersionNum string,

ImportDate string,

IssuerId2 string,

FederalTIN string,

RateEffectiveDate string,

RateExpirationDate string,

PlanId string,

RatingAreaId string,

Tobacco string,

Age string,

IndividualRate int,

IndividualTobaccoRate int,

Couple int,

PrimarySubscriberAndOneDependent int,

PrimarySubscriberAndTwoDependents int,

PrimarySubscriberAndThreeOrMoreDependents int,

coupleAndOneDependent int,

CoupleAndTwoDependents int,

CoupleAndThreeOrMoreDependents int,

RowNumber string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sskk04/projectfile/2015/rate'

TBLPROPERTIES ('skip.header.line.count'='2');

Create Network table from Network\_PUF2014.csv in user/sskk04/projectfile/2014/net:

CREATE TABLE IF NOT EXISTS net2014(

BusinessYear string,

StateCode string,

IssuerId string,

SourceName string,

VersionNum string,

ImportDate string,

IssuerId2 string,

StateCode2 string,

NetworkName string,

NetworkId string,

NetworkURL string,

RowNumber string,

MarketCoverage string,

DentalOnly string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/sskk04/projectfile/2014/net'

TBLPROPERTIES ('skip.header.line.count'='2');

1. **Create tables with columns needed from original tables by filtering unknown values using where:**

CREATE TABLE r2014 As

select BusinessYear, StateCode, IssuerId, PlanId, Tobacco, Age, IndividualRate, IndividualTobaccoRate

from rate2014

where IndividualRate != 999999 and IndividualRate != 99 and IndividualRate != 0;

CREATE TABLE r2015 As

select BusinessYear, StateCode,IssuerId, PlanId, Tobacco, Age, IndividualRate, IndividualTobaccoRate

from rate2015;

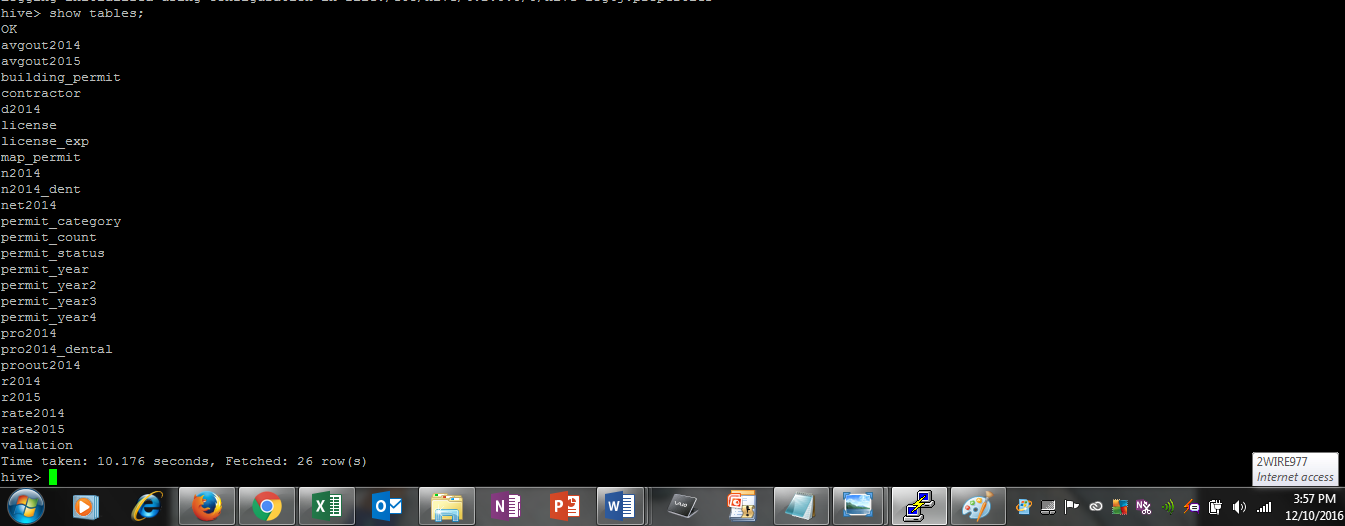
CREATE TABLE n2014 As

select BusinessYear, StateCode, IssuerId, NetworkName, DentalOnly

from net2014;

We can use the following to see the output:

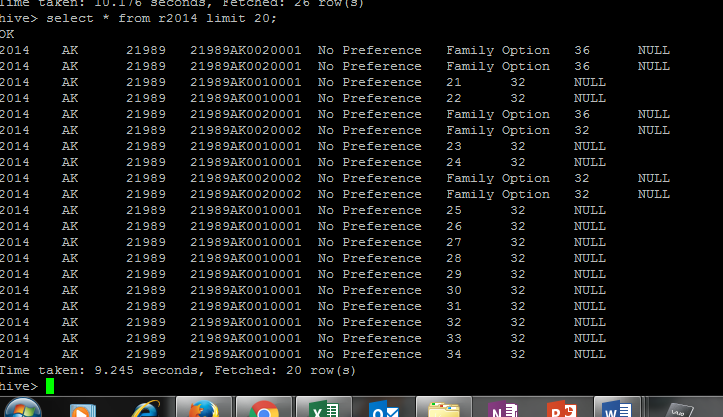
hive> show tables;



The above screenshot contains the tables we needed.

Use select statement to see contents of the tables:

Select \* from r2014 limit 20;



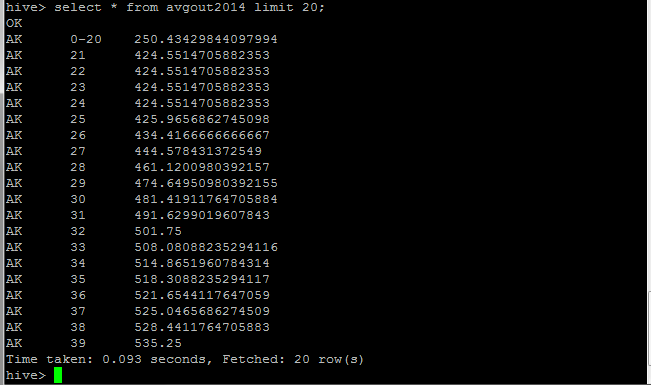
**Step 4: Analysis of average rates of different age groups and states in 2014 and 2015 with the use of above tables:**

1. **Create table for the average rates and name it as avgout2014**

Create table avgout2014 As

Select Statecode, age, avg(IndividualRate) FROM r2014 GROUP BY Statecode,age;

Select \* from avgout2014 limit 20;



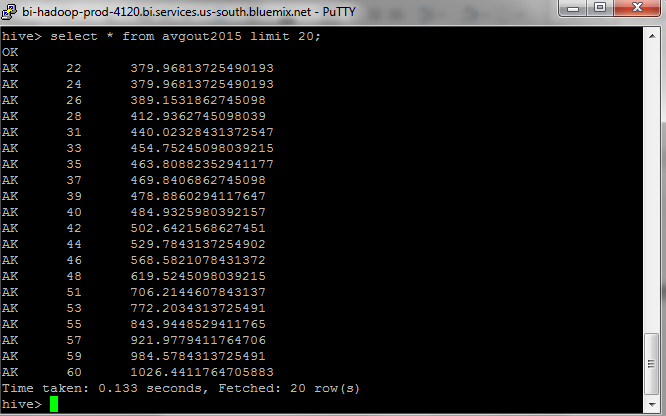
Above shows the average rates of different age groups in different states in 2014

1. **Create table for the average rates and name it as avgout2015**

Create table avgout2015 As

Select Statecode, age, avg(IndividualRate) FROM r2015 GROUP BY Statecode,age;

Select \* from avgout2015 limit 20;



Above shows the average rates of different age groups in different states in 2015. We can see difference in averagerates.

**Step 5: Analysis of Average rates of a particular age group provided by different providers using JOIN and GROUP BY:**

1. **Create new table by pro2014 joining r2014 and n2014 with a unique key “issuerId” for state Florida and age 25 in 2014**

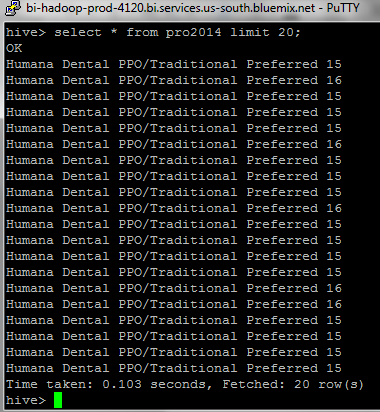
Create table pro2014 As

select NetworkName, IndividualRate from r2014

JOIN n2014 ON( r2014.issuerId = n2014.issuerId)

where r2014.statecode ="FL" and r2014.age = "25";

select \* from pro2014 limit 20;

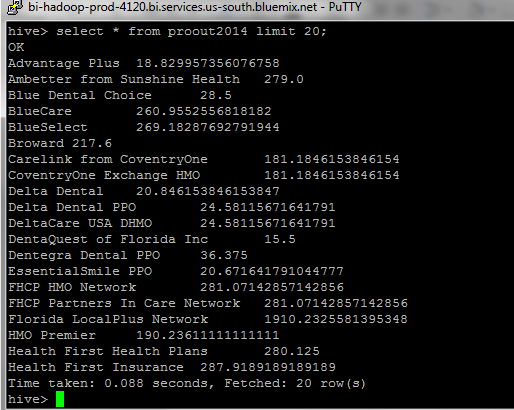


1. **create table proout2014 for the output of above created table pro2014:**

create table proout2014 As

SELECT NetworkName, avg(IndividualRate) FROM pro2014 GROUP BY NetworkName;

Select \* from proout2014 limit 20;



**Step6: Analysis of average rates of Dental Only plans for state a particular in particular agegroup:**

1. **Create new table n2014\_dent from net2014 filtering the dental plans only**

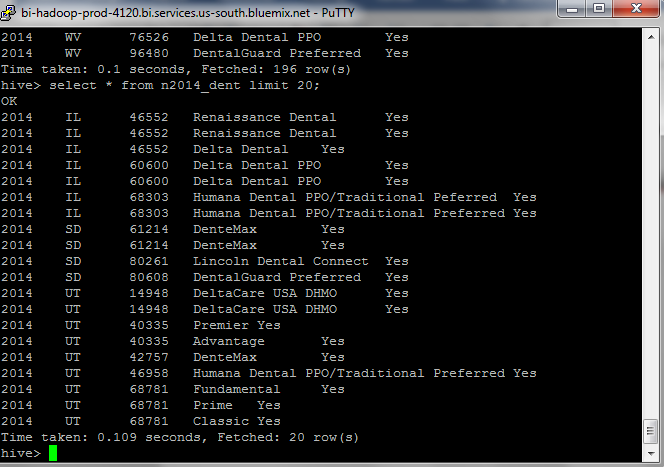
CREATE TABLE n2014\_dent As

select BusinessYear, StateCode, IssuerId, NetworkName, DentalOnly

from net2014

where DentalOnly = "Yes";

select \* from n2014\_dent limit 20;



1. **Create table pro2014\_dental to retrieve the rates of dental health plans only for Virginia and age 25 by joining n2014\_dent with r2014**

Create table pro2014\_dental As

select NetworkName, IndividualRate from r2014

JOIN n2014\_dent ON ( r2014.issuerId = n2014\_dent.issuerId)

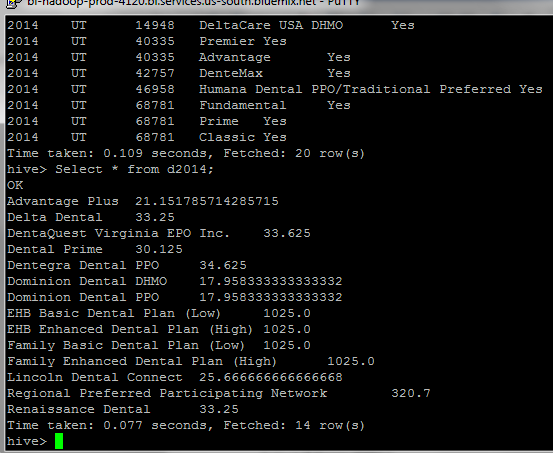
where r2014.statecode ="VA" and r2014.age = "25";

1. **Create table for the output of pro2014\_dental group by networkname**

create table d2014 As

SELECT NetworkName, avg(IndividualRate) FROM pro2014\_dental GROUP BY NetworkName;

Select \* from d2014;



Above output shows the rates of dental only plans in Virginia in 2014 for age 25 by different providers.

**Step7: Visualization in Powerview:**

1. **Put the output in a location in under the username(sskk04) for downloading file in excel:**

INSERT OVERWRITE LOCAL DIRECTORY '/home/sskk04/temp1' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT \* FROM avgout2014;

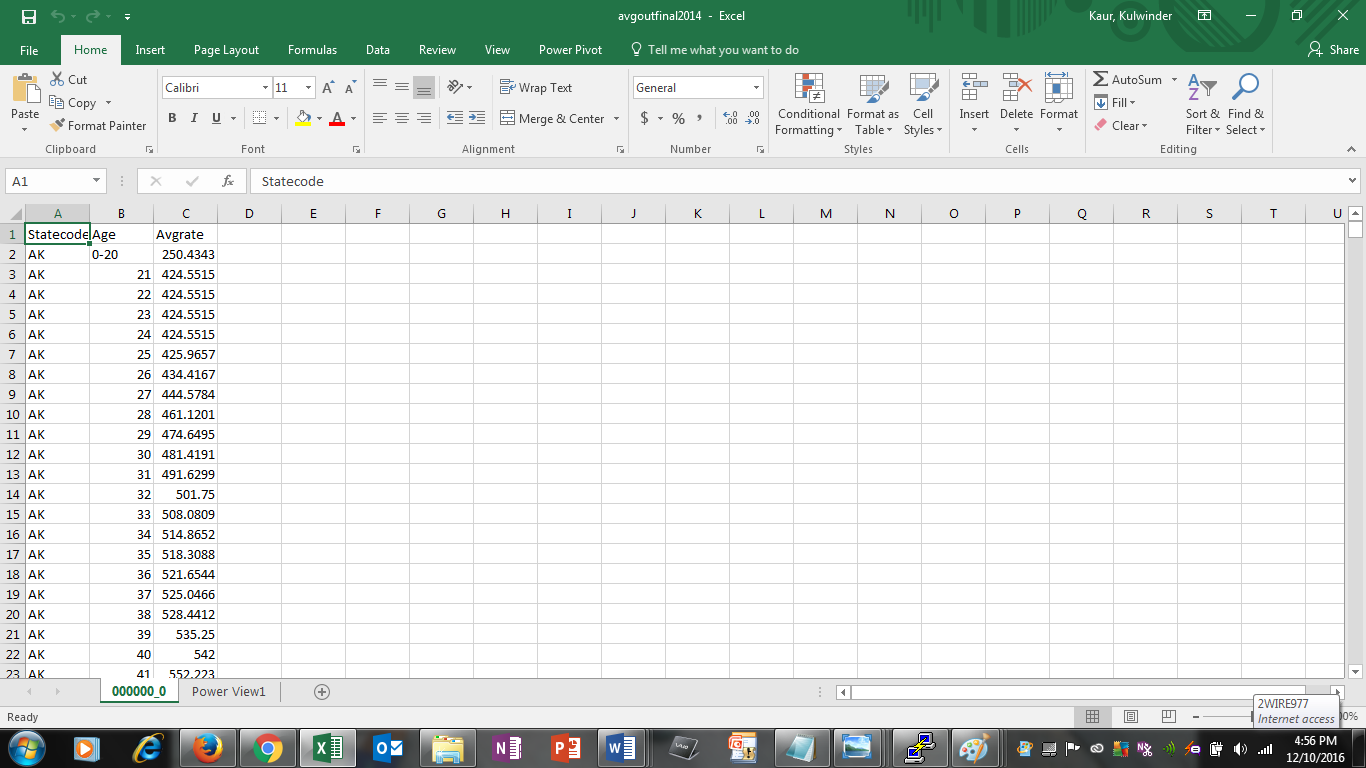
INSERT OVERWRITE LOCAL DIRECTORY '/home/sskk04/temp2' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT \* FROM avgout2015;

INSERT OVERWRITE LOCAL DIRECTORY '/home/sskk04/tempdf' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT \* FROM proout2014;

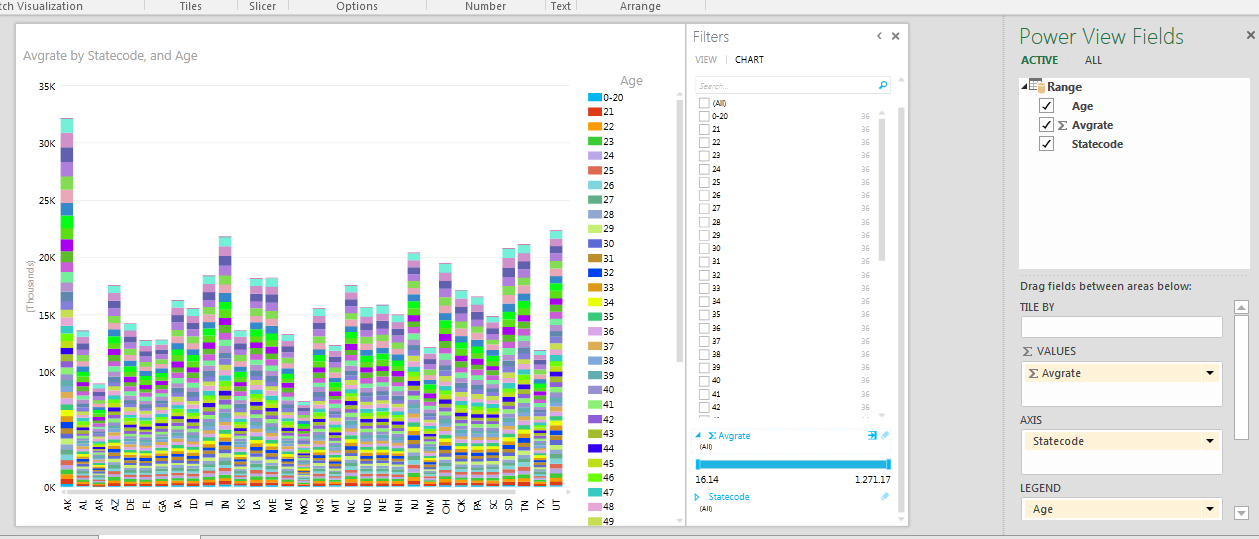
INSERT OVERWRITE LOCAL DIRECTORY '/home/sskk04/tempd2014' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT \* FROM d2014;

Launch an Excel 2016 (or 2013) and open/import the file “000000\_0”. Now that the data has been imported into Excel, we will use Power View to visually explore the data. We have to insert a header row with the following column names of the data using a text editor such as notepad, textpad:

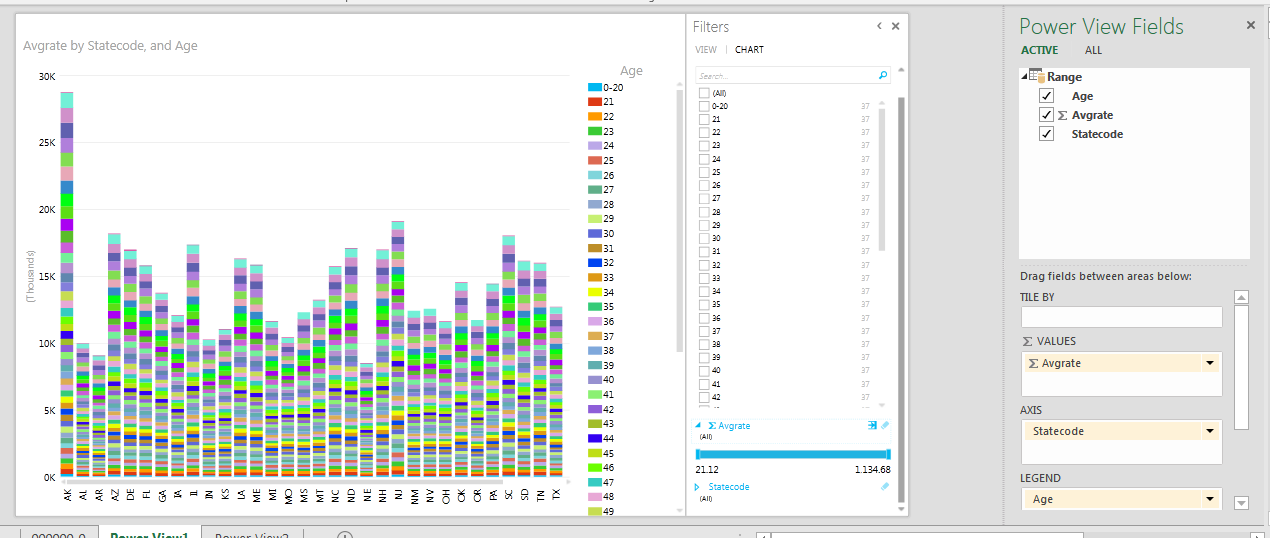
1. **Now, you need to save and close the text editor. Then, open the file in Excel. Save the file as an excel format, for example, avgout2014.xlsx**.



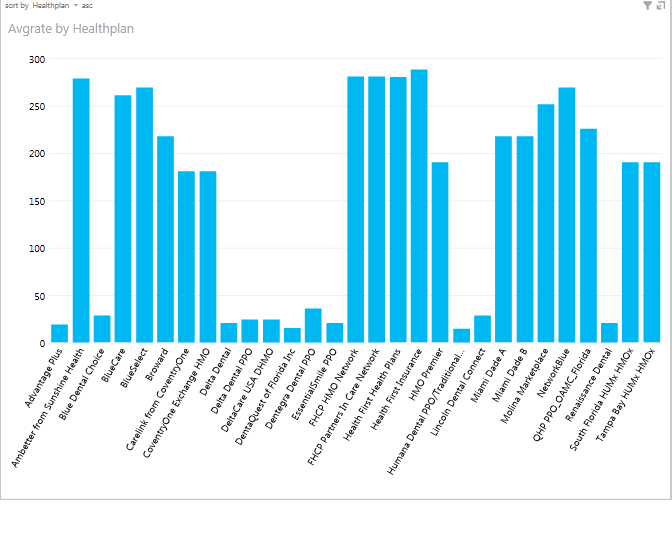
1. **In the Excel worksheet, select the Insert tab, and then select Power View to open a new Power View report. From design tab select column chart and then stacked column.**

****

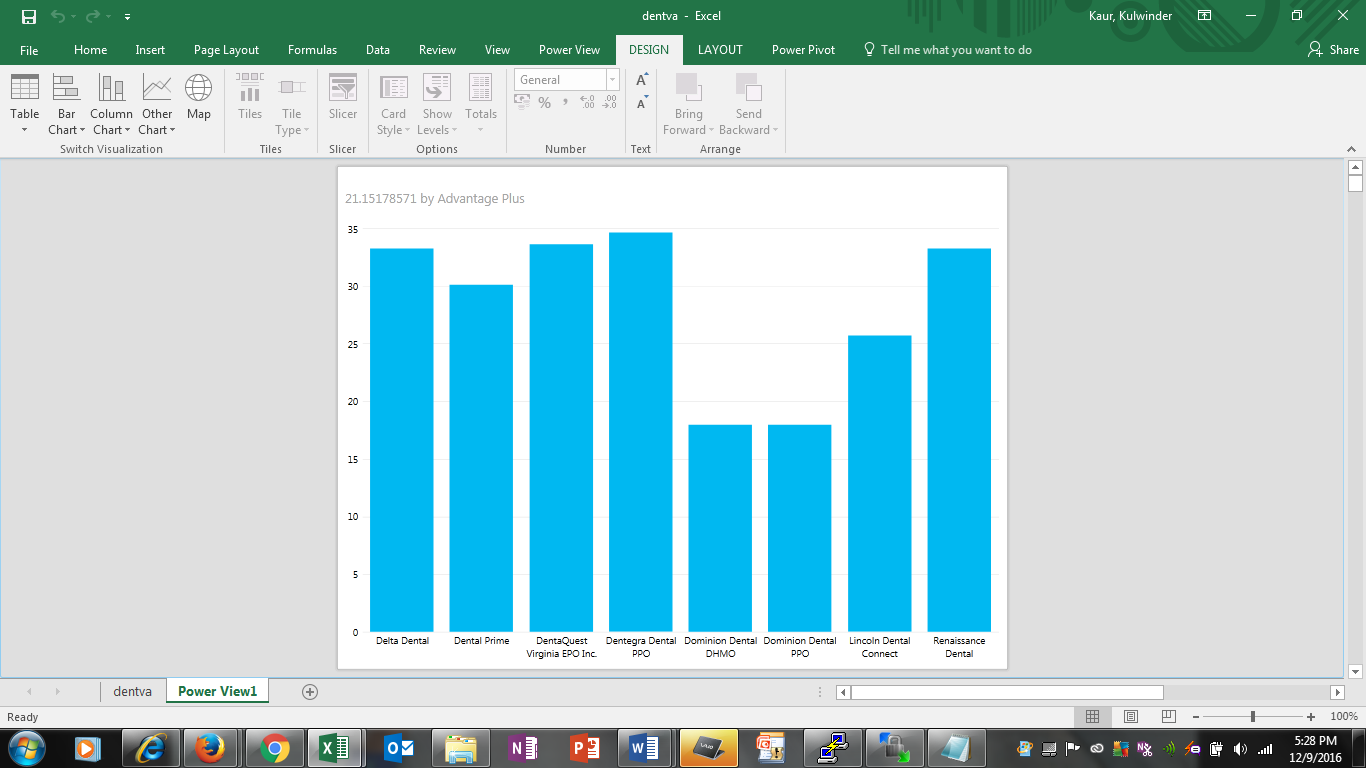
1. **Steps b and c can be repeated for visualization of average rates of 2015, average rates provided by different providers in Florida for age 25 and rates of Dental plans only in Virginia for age 25. The folder for the file to be downloaded should selected carefully such as temp2, tempdf and tempd2014.**

****

Above is the visualization of average rates of 2015 of different age groups in different states



Above is the visualization average rates of Health plans provided by health providers considering different age groups in different states.



Above is the visualization of rates of dental health plans in 2014 for state Virginia and age 25

**Part 2**

**Step 1 : Writing the pig script for analysis**

First, we need to run Pig, so we can enter the Pig interface (Grunt):

Pig

1. **Read from .csv file and creating a relation**

At this point we need to load the .csv data from Rate\_PUF\_2016.csv by using a special java class called CSVExcelStorage to handle the .csv format (for [more info](https://pig.apache.org/docs/r0.12.0/api/org/apache/pig/piggybank/storage/CSVExcelStorage.html)). And at the same step, we will create a relation called **Rate\_PUF\_2016** with a schema based on the [official documentation](https://www.cms.gov/CCIIO/Resources/Data-Resources/Downloads/Rate_DataDictionary_2016.pdf):

Rate\_PUF\_2016 = LOAD 'projectfile/2016/Rate\_PUF\_2016.csv'

USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO\_MULTILINE', 'NOCHANGE', 'SKIP\_INPUT\_HEADER')

AS (BusinessYear: int, StateCode: chararray, IssuerId: chararray, SourceName: chararray, VersionNum: chararray, ImportDate: chararray, IssuerId2: chararray, FederalTIN: chararray, RateEffectiveDate: chararray, RateExpirationDate: chararray, PlanId: chararray, RatingAreaId: chararray, Tobacco: chararray, Age: chararray, IndividualRate: int, IndividualTobaccoRate: int, Couple: int, PrimarySubscriberAndOneDependent: int, PrimarySubscriberAndTwoDependents: int, PrimarySubscriberAndThreeOrMoreDependents: int, CoupleAndOneDependent: int, CoupleAndTwoDependents: int, CoupleAndThreeOrMoreDependents: int, RowNumber: chararray);

We can run the following command to see the output of the relation:

DUMP Rate\_PUF\_2016;

After creating the big relation that was all the fields from the original file, we need to create a sub relation that only contains the columns that we need for our analysis; To do so, we need to run the following command:

R2016 = FOREACH Rate\_PUF\_2016 GENERATE BusinessYear, StateCode, PlanId, Tobacco, Age, IndividualRate, IndividualTobaccoRate;

And dump to check the output;

DUMP R2016;

1. **Grouping the data**

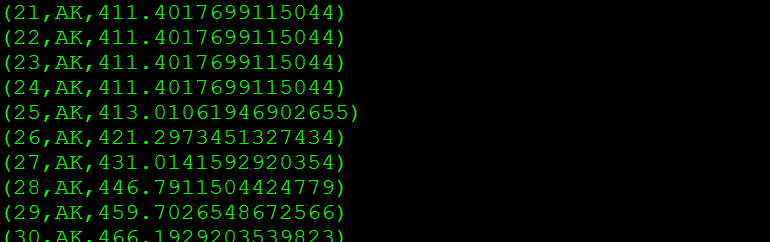
Now, we need to group the data in the way that we want to run our analysis, in this case we want to see the relation between age and price in different states. Thus, we need to group our relation by age and state code the calculate the average for each age group. This script will do what we want:

grp\_as = GROUP R2016 BY (StateCode, Age);

grp\_as\_rate = FOREACH grp\_as GENERATE FLATTEN(group) as (StateCode, Age), R2016.IndividualRate as IndividualRates;

avg\_as\_rate = FOREACH grp\_as\_rate GENERATE Age, StateCode, AVG(IndividualRates) as avg;

DUMP avg\_as\_rate;

The output should be something like this

1. **Store the output**

At this step, we need to export/save our relation by using PigStorage. Running the next command will guarantee this for us:

STORE avg\_as\_rate INTO 'avg\_as\_rate' using PigStorage(',');

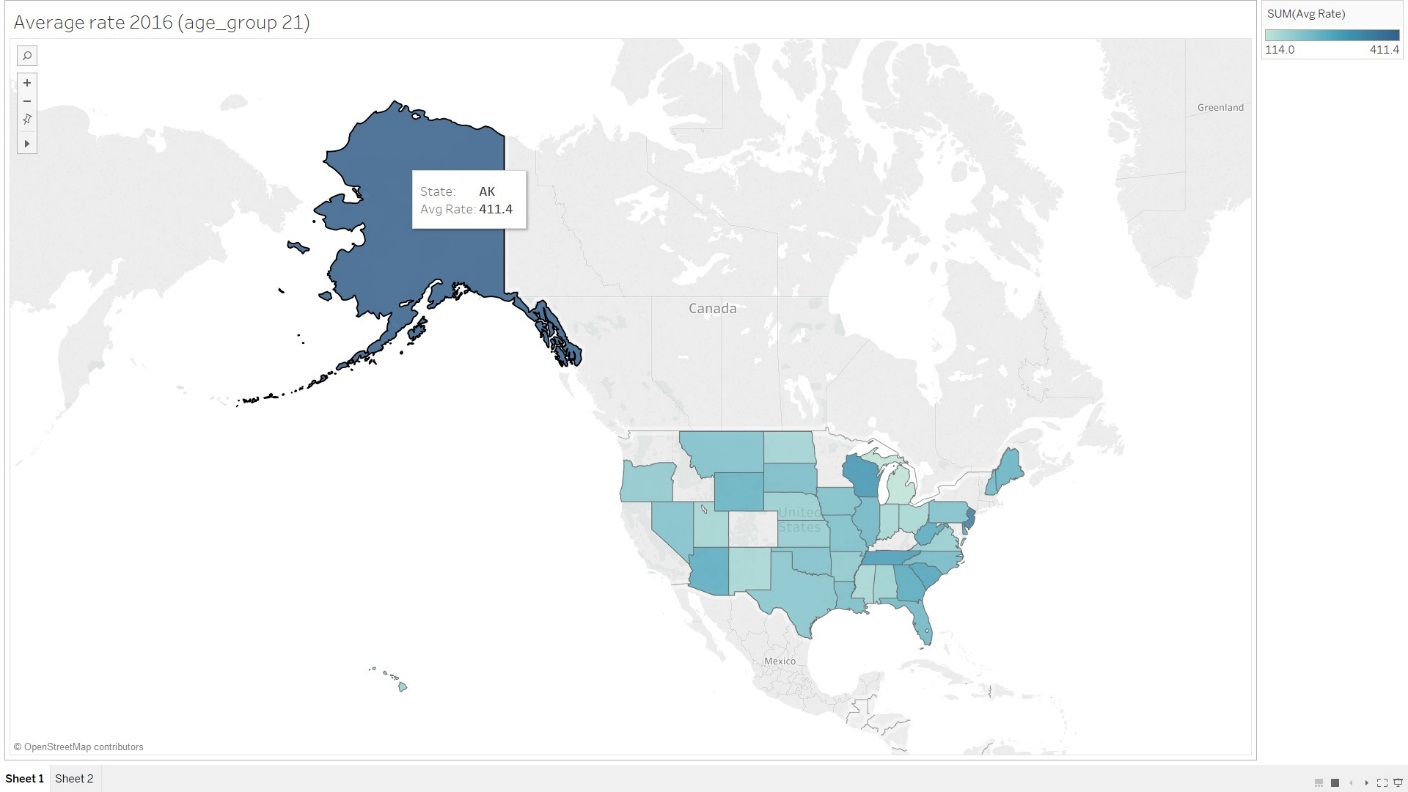
Important note: in the previous steps, we created a relation that contains only the individual rate, and if we need to calculate the individual rate with tobacco use, we need to go through the same steps from a to b but with picking a different field. So, instead of IndividualRates field, we take IndividualTobaccoRates.

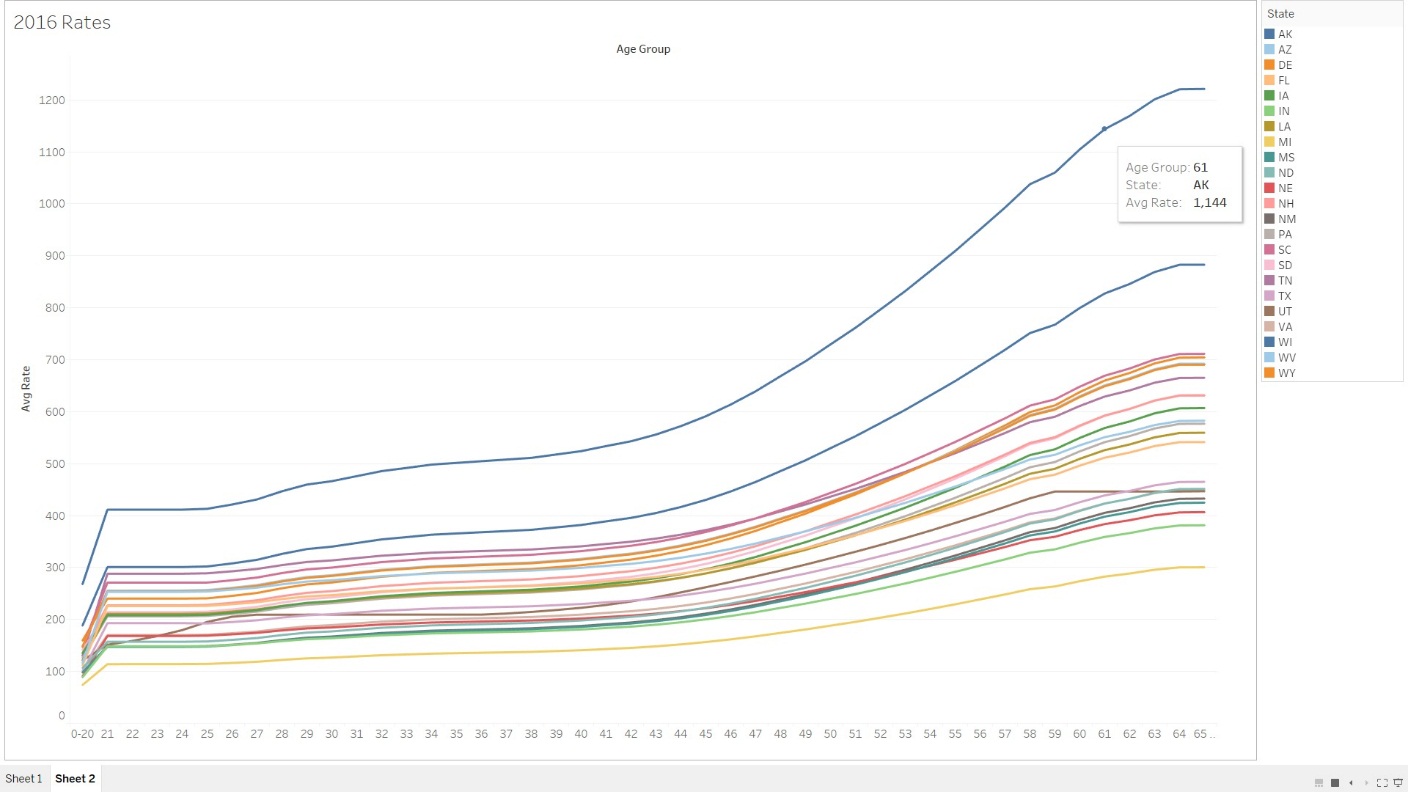
**Step 2 & 3: Import the output in Tableau and generate graphs**

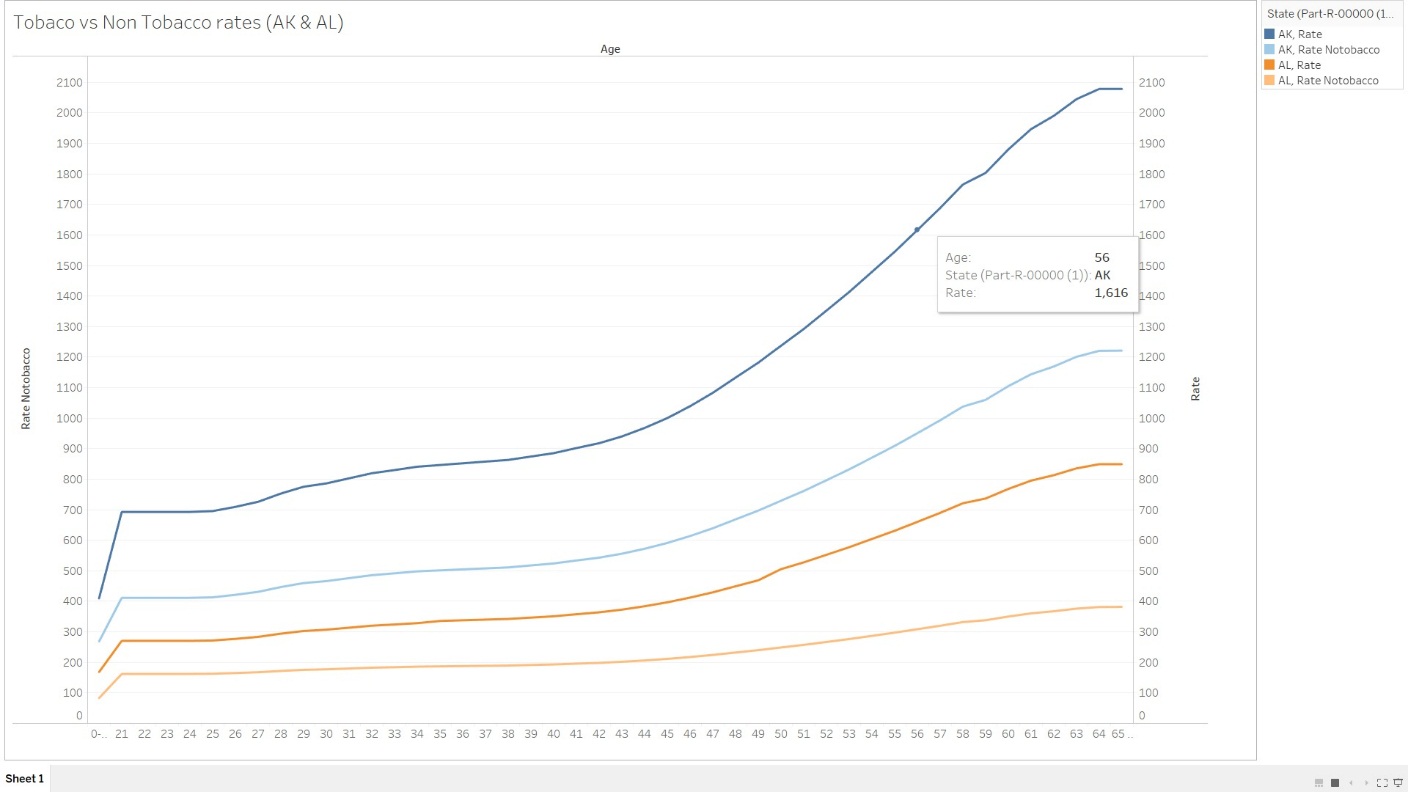
Before we jump to Tableau, we need to download the file output that we made earlier by using Ambari. After we download the file we need to open it in Excel and save it in .xlsx to import it easily in Tableau.

Now we need to lunch Tableau; and from the connect option we chose connect to Excel.

After we created the data source, now we can present our results in different ways. First, we need to click on sheet tab where we can start our visualization. The following screenshots are examples of what we can come up with using such powerful tool.







**Summary**

In this tutorial we learned how IBM BigInsight is very beneficial to analyze data using Apache Hive and Pig. We went through a flow to understand how the raw data is first uploaded to HDFS, and then loaded to Hive tables for performing queries and run pig scripts. Finally, we learned how to import the results of Hive queries and pig into Microsoft Excel and tableau respectievely. This analysis of health insurance data led us to the conclusion that large availibility of data and interactive visualization of analysis is significant and helpful for the people who are seeking for the quality and affordable healthplans.