Data Wrangling for Capstone 1 Project

The data I am using for Capstone 1 project is found from Kaggle datasets. The data provider did use a SQLite database to store all 8 tables, the size of the database file is about 4GB.

The two tables I need are Rate and PlanAttributes. I tried to retrieve all data from Rate table in Jupyter Notebook, but I got memory error at the end. It looks like Rate table contains data for business year 2014, 2015 and 2016. Each year, there are about 5M rows of data. I decided only look at business year 2015.

When I got the rate and plan data from the SQLite, I use infor and describe function to get a better understanding of the data. And I also use value\_counts function to see the uniq values for each columns. Here are some steps I do for data wrangling:

1. Age column in Rate table has some entry as “Family Option”. In this case, we do not know the exact age of the individual, I explicitly have to remove the records with “Family Option” in Age. After that I examined the values for Age, they can be “0-20”, “above 65” and some specific age like 34. Per Ram’s suggestion, I created four age groups and put 0-20 as Group A, 21-44 as Group B, 45-64 as Group C and above 65 as Group D.
2. StateCode column does not have any missing values. I use value\_counts function to see how many unique states are in the Rate table, I got total 37 states.
3. IndividualRate column in Rate table is very important to my analysis. I use describe function to view the stats on this column. It does not have any missing values, but some values are extremely high (9999), and does not make any common sense. It looks like the data provider uses this seudo number for the ones that do not have individual rates. I excluded any individual rate > 9000 in the dataframe.
4. PlanAttributes table has many columns (176). But I only need PlanID and MetalLevel columns. I tried to slice on these two columns and created a new dataframe called Plan\_Data. I also found out that PlanID in PlanAttributes table is in the format of Standard Component – CSR Variant, while PlanID in Rate table only has Standard Component. In order to join these two tables on PlanID, I have to remove the CSR Variant part for PlanID in PlanAttributes table. I further reduced the number of records in Plan\_Data by removing the duplicates.
5. I was able to merge the two dataframe on PlanID to create a ready usable dataframe called df\_final. Next, I am going to work on df\_final to find out how individualrate with same level of coverage varies among different states.