CUNY MSDA - IS607 : DATA ACQUISITION AND MANAGEMENT

Week 10 Project

A. Leopold HILLAH

Part1 – Obtaining the Data

The data used was obtained from https://data.medicare.gov/ on 09/11/2014 4pm CET. The description of the dataset from the site follows:

"Medicare Hospital Spending Per Patient – Hospital"

The "Medicare hospital spending per patient (Medicare Spending per Beneficiary)" measure shows whether Medicare spends more, less or about the same per Medicare patient treated in a specific hospital, compared to how much Medicare spends per patient nationally. This measure includes any Medicare Part A and Part B payments made for services provided to a patient during the 3 days prior to the hospital stay, during the stay, and during the 30 days after discharge from the hospital.

Dataset

The data was downloaded in **csv** format and made available in the current working directory. The file name is **Medicare_Hospital_Spending_Per_Patient_-_Hospital.csv** and contains **3320** records.

Use Case

The objective here is to find the top 10 states with the most hospitals having a spend score per patient higher than national scores.

Comparison between the three technologies

After loading and manipulating the data in each of the three technologies R, PostgreSQL and MongoDB, we can summarize findings in the table below for comparison:

CRITERIA	R	POSTGRESQL	MONGODB
Data Load	Straightforward. Method for loading data directly into R exists.	Straightforward. Utility exits to load csv data into the database.	Straightforward. Utility exists to load csv data directly into MongoDB (**mongoimport)**
Data Load Speed	Fast data load	Fast data load	Very fast data load
Data Retrieval	Very fast and simple data querying	Fast but complex Data Query (table join).	Very fast and simple data querying
Data Model	Simple model, un-normalized (data frame).	Complex model with normalization. Need to create extra tabi	Simple model, un-normalzed

Query type	Built-in functions	SQL, although the MEDIAN function was not available out of the box. Had to build it first before invocation.	NOSQL construct, although the MEDIAN function was not available out of the box. Had to use AVERAGE instead.
Ease of Use	Easy to use	Somewhat easy to use.	Easy to use
Overall Performance	Very good performance overall	Good performance	Very good performance overy

For this use case, both MongoDB and R appear to be the most effective and simple way to store and retrieve the data in the **csv** file, although MongoDB did not have the MEDIAN function implemented in the aggregation framework.

To store the data in PostgreSQl, the major drawback is the time to load the data into a temporary table, then create a normalized model and dispatching the data into three more tables.

For this use case, R appears to be the most effective and simple technology to use to load, aggregate and query the data.

Bringing the Data into R

Here is the code needed to load and manipulate the data in R

```
setwd("F:/CUNY - Masters Data Science/Data Acquisition & Management/Datasets")

library(dplyr)

mhspend <- read.csv("Medicare_Hospital_Spending_Per_Patient_-_Hospital.csv", stringsAsFactors = FALSE)

str(mhspend)

topmh_states <- mhspend %>% group_by(State) %>%
                                    summarize (MedScore = median(Score, na.rm=T)) %>%
                                    arrange(desc(MedScore)) %>%
                                    mutate (MedRank = order(MedScore, decreasing = TRUE))

str(topmh_states)

head(topmh_states, 10)
```

APPENDIX 1 - R DATA STRUCTURES AND OUTPUT

```
> str(mhspend)
'data.frame':  3320 obs. of  15 variables:
 $ Provider.ID       : int  10001 10005 10006 10007 10008 10011 10012 10016
10018 10019 ...
 $ Hospital.Name     : chr  "SOUTHEAST ALABAMA MEDICAL CENTER" "MARSHALL ME
DICAL CENTER SOUTH" "ELIZA COFFEE MEMORIAL HOSPITAL" "MIZELL MEMORIAL HOSPI
TAL" ...
 $ Address           : chr  "1108 ROSS CLARK CIRCLE" "2505 U S HIGHWAY 431
NORTH" "205 MARENGO STREET" "702 N MAIN ST" ...
 $ City              : chr  "DOTHAN" "BOAZ" "FLORENCE" "OPP" ...
 $ State             : chr  "AL" "AL" "AL" "AL" ...
 $ ZIP.Code          : int  36301 35957 35631 36467 36049 35235 35968 35007
35233 35660 ...
 $ County.Name       : chr  "HOUSTON" "MARSHALL" "LAUDERDALE" "COVINGTON" .
..
 $ Phone.Number      : num  3.35e+09 2.57e+09 2.57e+09 3.34e+09 3.34e+09 ..
.
 $ Measure.Name      : chr  "Medicare hospital spending per patient (Medica
re Spending per Beneficiary)" "Medicare hospital spending per patient (Medi
care Spending per Beneficiary)" "Medicare hospital spending per patient (Me
dicare Spending per Beneficiary)" "Medicare hospital spending per patient (
Medicare Spending per Beneficiary)" ...
 $ Measure.ID        : chr  "MSPB_1" "MSPB_1" "MSPB_1" "MSPB_1" ...
 $ Score             : num  0.97 0.98 0.99 1.02 0.98 0.97 1.01 1.01 NA 1.06
...
 $ Footnote          : logi  NA NA NA NA NA NA ...
 $ Measure.Start.Date: chr  "01/01/2013" "01/01/2013" "01/01/2013" "01/01/2
013" ...
 $ Measure.End.Date  : chr  "12/31/2013" "12/31/2013" "12/31/2013" "12/31/2
013" ...
 $ Location          : chr  "1108 ROSS CLARK CIRCLE\nDOTHAN, AL 36301\n(31.
215379379000467, -85.36146587999968)" "2505 U S HIGHWAY 431 NORTH\nBOAZ, AL
35957\n(34.22133455500045, -86.15937514799964)" "205 MARENGO STREET\nFLOREN
CE, AL 35631\n(34.795039606000444, -87.68507485299966)" "702 N MAIN ST\nOPP
, AL 36467\n(31.292159523000464, -86.25539902199966)" ...
```

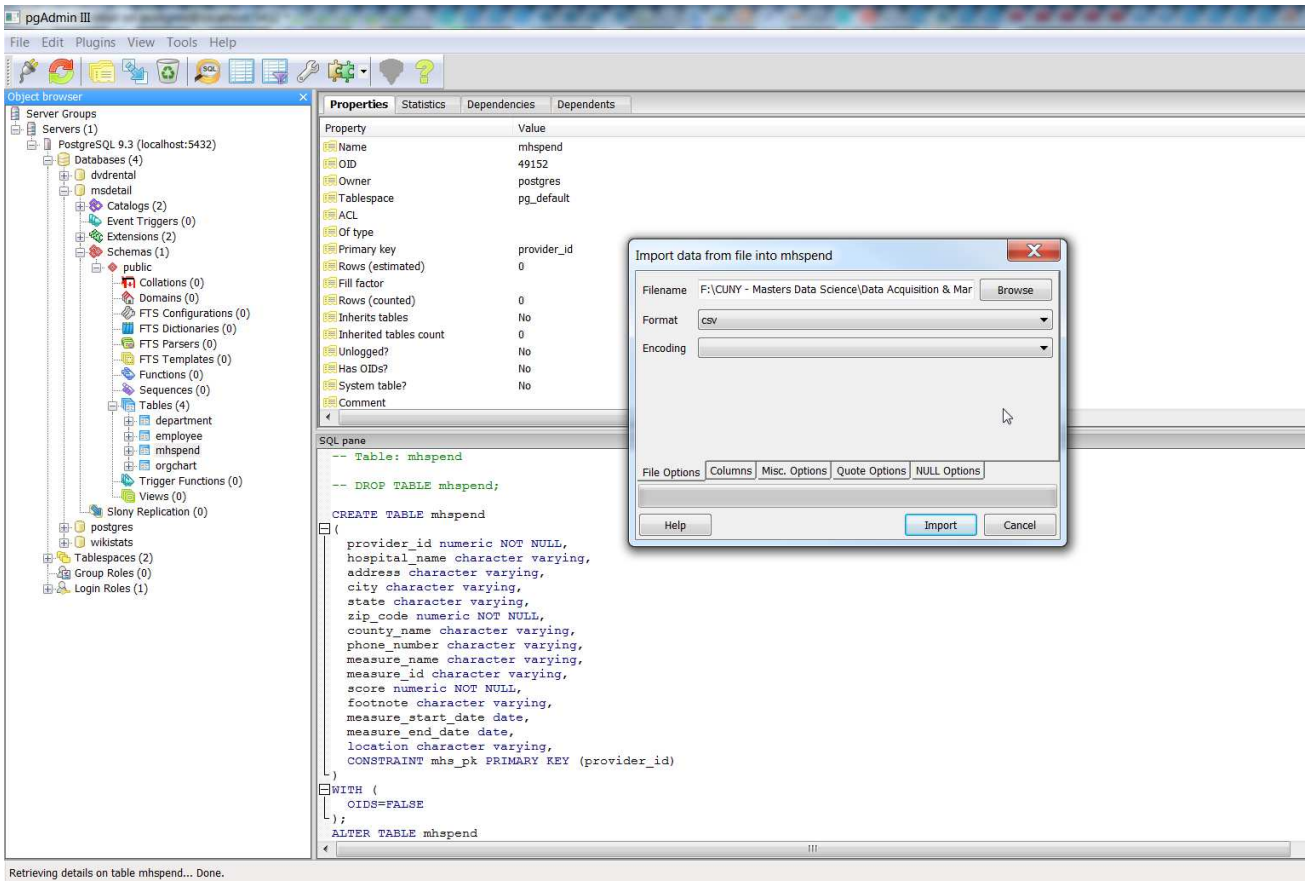
```
> str(topmh_states)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':     50 obs. of  3 variables:
 $ State   : chr  "NJ" "NV" "FL" "TX" ...
 $ MedScore: num  1.08 1.08 1.04 1.04 1.03 ...
 $ MedRank : int  1 2 3 4 5 6 7 8 9 10 ...
>
```

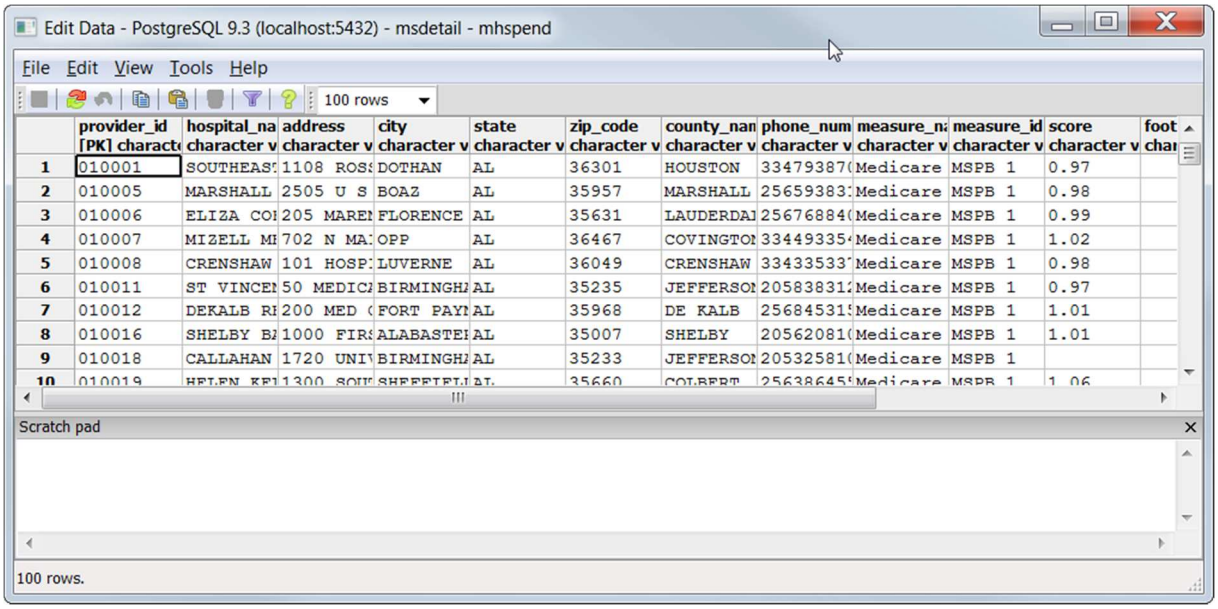
```
> head(topmh_states, 10)
Source: local data frame [10 x 3]

   State MedScore MedRank
1     NJ    1.080       1
2     NV    1.080       2
3     FL    1.045       3
4     TX    1.040       4
5     MA    1.030       5
6     NH    1.030       6
7     CT    1.020       7
8     LA    1.020       8
9     PA    1.020       9
10    RI    1.020      10
>
```

APPENDIX 2 - POSTGRESQL DATA STRUCTURES AND OUTPUT

1. Loading the data into a temporary table **mhspend**

```
CREATE TABLE MHSPEND
(
  PROVIDER_ID CHARACTER VARYING NOT NULL,
  HOSPITAL_NAME CHARACTER VARYING,
  ADDRESS CHARACTER VARYING,
  CITY CHARACTER VARYING,
  STATE CHARACTER VARYING,
  ZIP_CODE CHARACTER VARYING,
  COUNTY_NAME CHARACTER VARYING,
  PHONE_NUMBER CHARACTER VARYING,
  MEASURE_NAME CHARACTER VARYING,
  MEASURE_ID CHARACTER VARYING,
  SCORE CHARACTER VARYING,
  FOOTNOTE CHARACTER VARYING,
  MEASURE_START_DATE CHARACTER VARYING,
  MEASURE_END_DATE CHARACTER VARYING,
  LOCATION CHARACTER VARYING,
  CONSTRAINT MHS_PK PRIMARY KEY (PROVIDER_ID)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE MHSPEND
OWNER TO POSTGRES;
```





2. Creating three empty tables **hospital**, **hexpend** and **measure**:

```
CREATE TABLE HOSPITAL
(
  PROVIDER_ID CHARACTER VARYING,
  HOSPITAL_NAME CHARACTER VARYING,
  ADDRESS CHARACTER VARYING,
  CITY CHARACTER VARYING,
  STATE CHARACTER VARYING,
  ZIP_CODE CHARACTER VARYING,
  COUNTY_NAME CHARACTER VARYING,
  PHONE_NUMBER CHARACTER VARYING,
  LOCATION CHARACTER VARYING,
  CONSTRAINT HOSPITAL_PK PRIMARY KEY (PROVIDER_ID)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE HOSPITAL
OWNER TO POSTGRES;


CREATE TABLE HEXPEND
(
  PROVIDER_ID CHARACTER VARYING,
  MEASURE_ID CHARACTER VARYING,
  SCORE CHARACTER VARYING,
  FOOTNOTE CHARACTER VARYING,
  MEASURE_START_DATE CHARACTER VARYING,
  MEASURE_END_DATE CHARACTER VARYING,
  CONSTRAINT HEXPEND_PK PRIMARY KEY (PROVIDER_ID)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE HEXPEND
  OWNER TO POSTGRES;


CREATE TABLE MEASURE
(
  MEASURE_ID CHARACTER VARYING,
  MEASURE_NAME CHARACTER VARYING,
  CONSTRAINT MEASURE_PK PRIMARY KEY (MEASURE_ID)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE MHSPEND
```

```
  OWNER TO POSTGRES;
```

3. Inseting data into each table:

```
INSERT INTO HOSPITAL
SELECT  PROVIDER_ID,
  HOSPITAL_NAME,
  ADDRESS,
  CITY,
  STATE,
  ZIP_CODE,
  COUNTY_NAME,
  PHONE_NUMBER,
  LOCATION
FROM MHSPEND;

INSERT INTO HEXPEND
SELECT PROVIDER_ID,
  MEASURE_ID,
  SCORE,
  FOOTNOTE,
  MEASURE_START_DATE,
  MEASURE_END_DATE
FROM MHSPEND;

INSERT INTO MEASURE
SELECT DISTINCT MEASURE_ID,
  MEASURE_NAME
FROM MHSPEND;

COMMIT;
```

4. Querying the data to get the list of top states having 50% hospitals or more with highest scores:

```
WITH MH AS (SELECT A.PROVIDER_ID, A.STATE, B.PROVIDER_ID, B.SCORE
           FROM HOSPITAL A
           JOIN HEXPEND B ON A.PROVIDER_ID = B.PROVIDER_ID)
SELECT STATE, MEDIAN(CAST (SCORE AS NUMERIC)) MEDSCORE FROM MH
GROUP BY STATE
ORDER BY 2 DESC, 1
LIMIT 10
```

APPENDIX 3 - MONGODB DATA STRUCTURES AND OUTPUT

1. Importing the **csv** file using **mongoimport** and check first 3 records loaded

```
F:\CUNY - Masters Data Science\Data Acquisition & Management\Datasets>mongoimport -d test -c mhspend --type csv --file Medicare_Hospital_Spending_Per_Patient_-_Hospital.csv --headerline

connected to: 127.0.0.1

2014-11-09T23:04:36.645+0100 check 9 3321

2014-11-09T23:04:36.666+0100 imported 3320 objects


F:\CUNY - Masters Data Science\Data Acquisition & Management\Datasets>mongo

MongoDB shell version: 2.6.5

connecting to: test

> db.mhspend.find().limit(3)

{ "_id" : ObjectId("545fe4f4dd4b347cd3d20330"), "Provider ID" : 10001, "Hospital Name" : "SOUTHEAST ALABAMA MEDICAL CENTER", "Address" : "1108 ROSS CLARK CIRCLE", "City" : "DOTHAN", "State" : "AL", "ZIP Code" : 36301, "County Name" : "HOUSTON", "Phone Number" : NumberLong("3347938701"), "Measure Name" : "Medicare hospital spending per patient (Medicare Spending per Beneficiary)", "Measure ID" : "MSPB_1", "Score" : 0.97, "Footnote" : "", "Measure Start Date" : "01/01/2013", "Measure End Date" : "12/31/2013", "Location" : "1108 ROSS CLARK CIRCLE\nDOTHAN, AL 36301\n(31.215379379000467, -85.36146587999968)" }

{ "_id" : ObjectId("545fe4f4dd4b347cd3d20331"), "Provider ID" : 10005, "Hospital Name" : "MARSHALL MEDICAL CENTER SOUTH", "Address" : "2505 U S HIGHWAY 431 NORTH", "City" : "BOAZ", "State" : "AL", "ZIP Code" : 35957, "County Name" : "MARSHALL", "Phone Number" : NumberLong("2565938310"), "Measure Name" : "Medicare hospital spending per patient (Medicare Spending per Beneficiary)", "Measure ID" : "MSPB_1", "Score" : 0.98, "Footnote" : "", "Measure Start Date" : "01/01/2013", "Measure End Date" : "12/31/2013", "Location" : "2505 U S HIGHWAY 431 NORTH\nBOAZ, AL 35957\n(34.22133455500045, -86.15937514799964)" }

{ "_id" : ObjectId("545fe4f4dd4b347cd3d20332"), "Provider ID" : 10006, "Hospital Name" : "ELIZA COFFEE MEMORIAL HOSPITAL", "Address" : "205 MARENGO STREET", "City" : "FLORENCE", "State" : "AL", "ZIP Code" : 35631, "County Name" : "LAUDERDALE", "Phone Number" : NumberLong("2567688400"), "Measure Name" : "Medicare hospital spending per patient (Medicare Spending per Beneficiary)", "Measure ID" : "MSPB_1", "Score" : 0.99, "Footnote" : "", "Measure Start Date" : "01/01/2013", "Measure End Date" : "12/31/2013", "Location" : "205 MARENGO STREET\nFLORENCE, AL 35631\n(34.795039606000444, -87.68507485299966)" }

>
```

2. Querying the top states having 50% hospitals or more with highest scores (on average):

```
> db.mhspend.aggregate( [
... { $group : { _id :  {State : "$State"} ,
... MedScore : { $avg : "$Score" } }} ,
... { $sort : { MedScore : -1 }},
... { $limit : 10 }
... ] )
{ "_id" : { "State" : "NJ" }, "MedScore" : 1.0763492063492068 }
{ "_id" : { "State" : "NV" }, "MedScore" : 1.0450000000000002 }
{ "_id" : { "State" : "TX" }, "MedScore" : 1.0430794701986752 }
{ "_id" : { "State" : "FL" }, "MedScore" : 1.039096385542169 }
{ "_id" : { "State" : "LA" }, "MedScore" : 1.036091954022988 }
{ "_id" : { "State" : "NH" }, "MedScore" : 1.0276923076923077 }
{ "_id" : { "State" : "IN" }, "MedScore" : 1.0253409090909094 }
{ "_id" : { "State" : "MA" }, "MedScore" : 1.0250819672131148 }
{ "_id" : { "State" : "CT" }, "MedScore" : 1.0176666666666667 }
{ "_id" : { "State" : "OH" }, "MedScore" : 1.015384615384616 }
>
```