CIS5200 Term Project Tutorial

Authors: Philip Wong ; Pratiksha Yadav; Pooja Madhup ; Shailja Pandit

Instructor: Jongwook Woo

Date: 05/22/2022

Lab Tutorial

Philip Wong (pwong4@calstatela.edu)

Pratiksha Yadav(pyadav@calstatela.edu)

Pooja Madhup (pmadhup@calstatela.edu)

Shailja Pandit (spandit3@calstatela.edu)

05/22/2022

Yelp Data Analysis using Hive

Objectives

In this hands-on lab, you will learn how to:

- Download data from the yelp website and using SCP upload the data to the Hadoop cluster
- Create Hive tables in HDFS using HiveQL.
- Create HiveQL queries to manipulate and analyze the data
- Visualize the result in Excel and tableau

Introduction

Yelp is an internet-based company that focuses on crowd-sourced reviews for millions of businesses and establishments across the US and globally. Yelp users can submit feedback in terms of reviews, ratings, and list information pertaining to the establishment. The yelp data set and will include three Json files (Business, User, and Reviews) which we will join with the dictionary dataset required for sentiment analysis.

- Provide a high-level, general, and aggregated approach to sentiment analysis. Visualize the results to find insights.

- Overall sentiment of yelp users (yelpers) and behaviors
- Healthcare and Food/Beverage category
- What insights can we obtain for the US healthcare system?
- What does the data show towards food establishments across the country?

Platform Spec

- Cluster Version: 3.2.1-amzn-3.1
- CPU Speed: 2.40 GHz
- Number of CPU cores: 4
- Number of nodes: 3
- Total Memory Size: 481 GB

Dataset Details

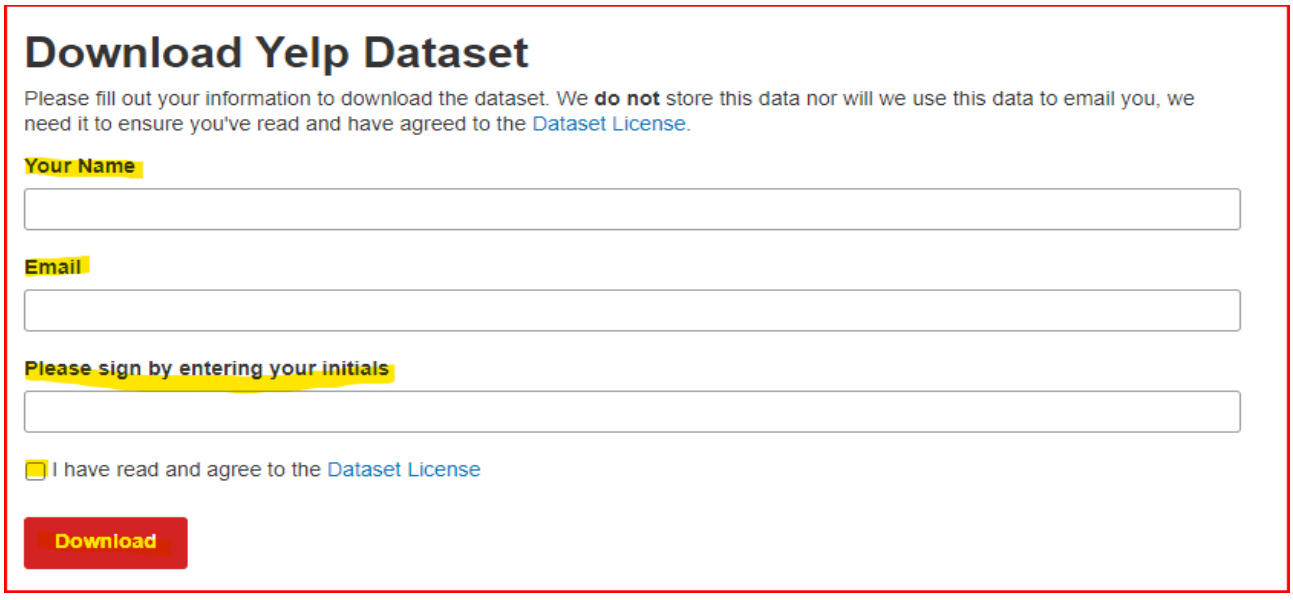
- DATASET NAME: Yelp Dataset
- DATASET URL: https://www.yelp.com/dataset/download
- TOTAL SIZE: 8.21 GB
- COUNTRIES CONSIDERED: USA and World
- NUMBER OF FILES: 3
- FILE FORMAT: JSON, CSV

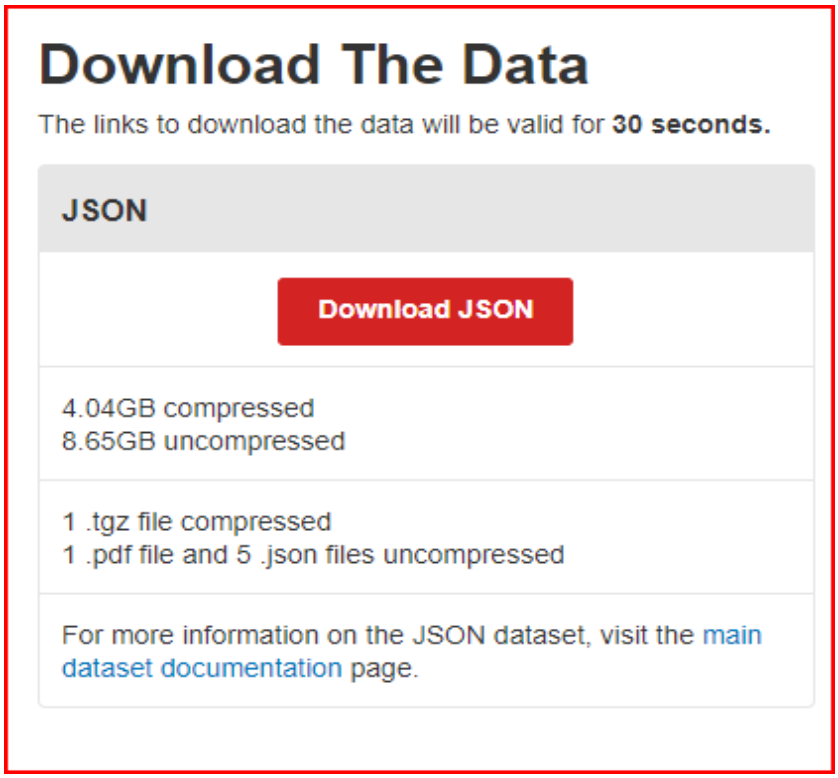
Step 1: Download the Dataset

This step is to get data manually. You need to remotely access your Oracle Cloud Big Data Compute Editions that you executed in your Oracle Cloud account using ssh using the information - ip address and connect command in beeline CLI-

Yelp Dataset – Download Dataset to local machine. Fill out the highlighted fields and click on 'Download JSON'. You should have the yelp_dataset.tar on your local machine.

Note: This file contents can change depending on when it was downloaded.





Extract the Yelp_dataset.tar file using a tool such as 7-zip

Extract the yelp_dataset file using a tool such as 7-zip

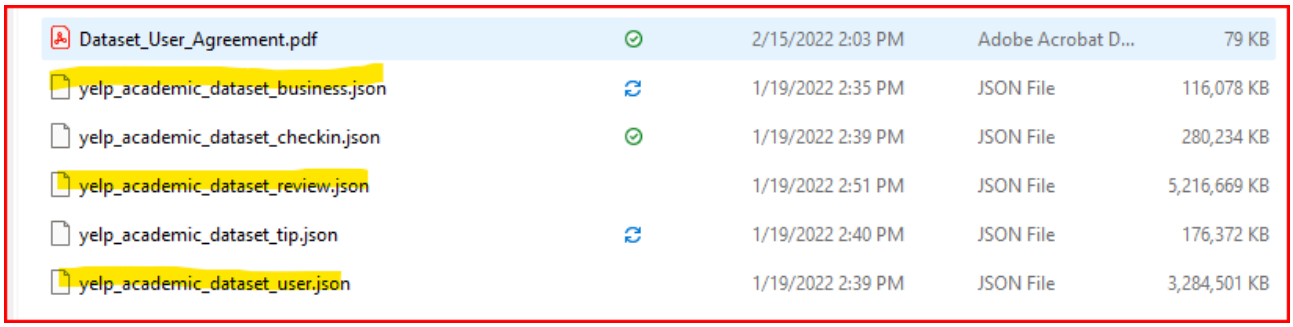
Output 5 JSON formatted files.

We then need to transfer the JSON 3 files from the local machine to the Hadoop cluster

yelp_academic_dataset_business.json

yelp_academic_dataset_review

yelp_academic_dataset_user



Step 2: Upload Files to Hadoop File System (HDFS)

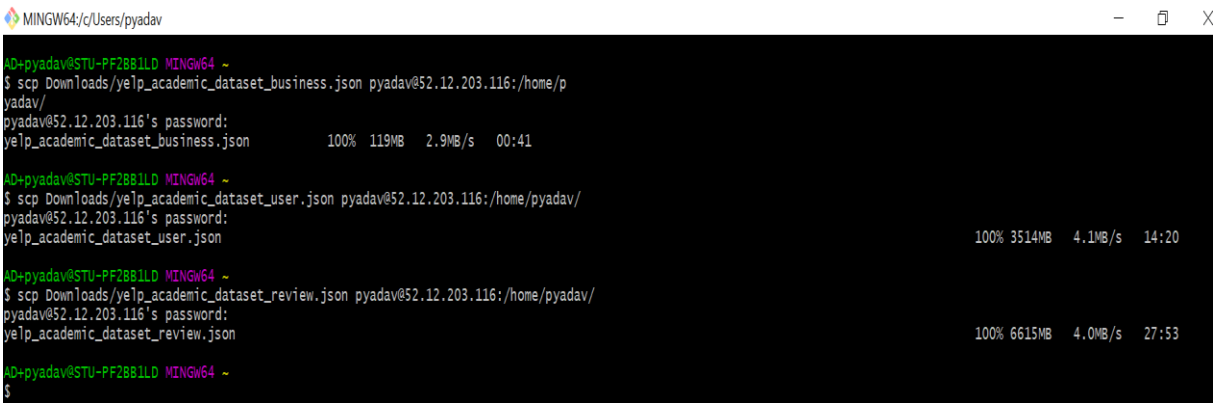
Using SCP:

Open a command prompt session and from the directory of the extracted files in the previous step and perform the following commands:

```
 scp Downloads/yelp_academic_dataset_business.json
pyadav@35.87.184.21:/home/pyadav/
 scp Downloads/yelp_academic_dataset_review.json
pyadav@52.12.203.116:/home/pyadav/
 scp Downloads/yelp_academic_dataset_user.json
```

Note: Use your own userid and server ip address.

Note: Alternatively, you can use a free tool such as WINSCP to upload the files.



Confirm files transferred using ls command.

```
$ ls -ltr
// confirm files transfered
hdfs dfs -ls yelp/business
hdfs dfs -ls yelp/review
hdfs dfs -ls yelp/user
```

```
-bash-4.2$ ls -ltr
total 8617560
-rw-rw-r-- 1 pwong4 pwong4     308921 May 12 02:25 dictionary.tsv
-rw-r--r-- 1 pwong4 pwong4  118863795 May 15 02:08 yelp_academic_dataset_busines
s.json
-rw-r--r-- 1 pwong4 pwong4 5341868833 May 15 02:10 yelp_academic_dataset_review.
json
-rw-r--r-- 1 pwong4 pwong4 3363329011 May 15 02:13 yelp_academic_dataset_user.js
on
```

Create the directory on the hdfs and copy from linux server to hdfs:

```
hdfs dfs -mkdir yelp
hdfs dfs -mkdir yelp/business
hdfs dfs -mkdir yelp/review
hdfs dfs -mkdir yelp/user
hdfs dfs -put yelp_academic_dataset_business.json yelp/business
hdfs dfs -put yelp_academic_dataset_review.json yelp/review
hdfs dfs -put yelp_academic_dataset_user.json yelp/user
```

Confirm file transfer to hdfs.

```
-bash-4.2$ hdfs dfs -ls yelp/business
Found 1 items
-rw-r--r--   3 pwong4 hdfs  118863795 2022-05-12 02:23 yelp/business/yelp_academic_dataset_business.json
-bash-4.2$ hdfs dfs -ls yelp/review
Found 1 items
-rw-r--r--   3 pwong4 hdfs 5341868833 2022-05-12 02:23 yelp/review/yelp_academic_dataset_review.json
-bash-4.2$ hdfs dfs -ls yelp/user
Found 1 items
-rw-r--r--   3 pwong4 hdfs 3363329011 2022-05-12 02:23 yelp/user/yelp_academic_dataset_user.json
```

```
hdfs dfs -ls yelp/business
hdfs dfs -ls yelp/review
hdfs dfs -ls yelp/user
```

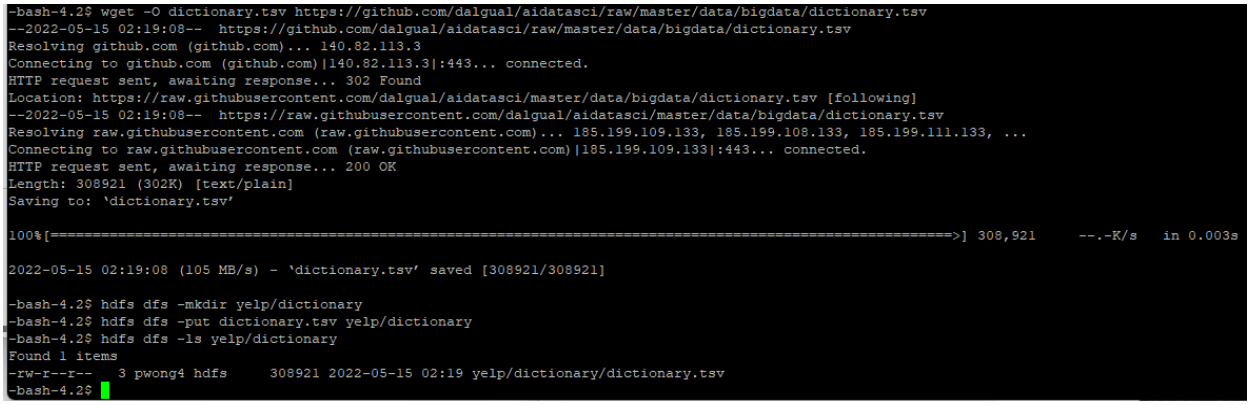
Now perform similar process for the dictionary file, but instead use wget from the linux server bash prompt:

```
-bash-4.2$
```

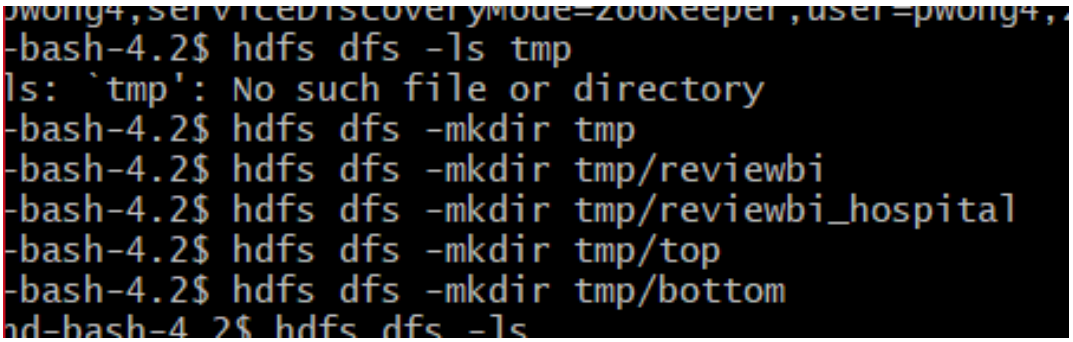
Upload file directly from file server using wget.
https://github.com/dalgual/aidatasci/raw/master/data/bigdata/dictionary.tsv

Create hdfs directory → Copy to HDFS using put command → Confirm if the file exists

```
wget -
O dictionary.tsv https://github.com/dalgual/aidatasci/raw/master/data/
bigdata/dictionary.tsv
hdfs dfs -mkdir yelp/dictionary
hdfs dfs -put dictionary.tsv yelp/dictionary
hdfs dfs -ls yelp/dictionary
```




```
hdfs dfs -mkdir yelp
hdfs dfs -mkdir yelp/business
hdfs dfs -mkdir yelp/review
hdfs dfs -mkdir yelp/user
hdfs dfs -mkdir tmp
hdfs dfs -mkdir tmp/reviewbi
hdfs dfs -mkdir tmp/reviewbi_hospital
hdfs dfs -mkdir tmp/reviewbi_restaurant
hdfs dfs -mkdir tmp/reviewbi_hospital_pa
hdfs dfs -mkdir tmp/reviewbi_restaurant_pa
```



- bash-4.2$ hdfs dfs -ls tmp

```
-bash-4.2$ hdfs dfs -ls tmp
Found 4 items
drwxr-xr-x   - pwong4 hdfs          0 2022-05-16 04:22 tmp/bottom
drwxr-xr-x   - pwong4 hdfs          0 2022-05-16 04:20 tmp/reviewbi
drwxr-xr-x   - pwong4 hdfs          0 2022-05-16 04:21 tmp/reviewbi_hospital
drwxr-xr-x   - pwong4 hdfs          0 2022-05-16 04:22 tmp/top
-bash-4.2$
```

Step 3: Dictionary Table Creation

Execute the following command to create the table.

```
CREATE EXTERNAL TABLE if not exists dictionary (
type string,
length int,
word string,
pos string,
stemmed string,
polarity string )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION 'yelp/dictionary';
```

Confirm contents in table with the SELECT statement.

```
SELECT * from dictionary limit 5;
```

```
INFO  : OK
+-----------------+-------------------+-----------------+----------------+----------------
-----+----------------------+
| dictionary.type  | dictionary.length  | dictionary.word  | dictionary.pos  | dictionary.stem
med  | dictionary.polarity  |
+-----------------+-------------------+-----------------+----------------+----------------
-----+----------------------+
| weaksubj        | 1                  | abandoned        | adj            | n
     | negative             |
| weaksubj        | 1                  | abandonment      | noun           | n
     | negative             |
| weaksubj        | 1                  | abandon          | verb           | y
     | negative             |
| strongsubj      | 1                  | abase            | verb           | y
     | negative             |
| strongsubj      | 1                  | abasement        | anypos         | y
     | negative             |
+-----------------+-------------------+-----------------+----------------+----------------
-----+----------------------+
5 rows selected (0.229 seconds)
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai>
```

Step 4: JSON Conversion to Structure Data

We will create the schema of the 3 review_table, business_table and user_table. Subsequently, we will insert the values from the 3 json files.

```
CREATE TABLE IF NOT EXISTS review_table (
review_id STRING,
user_id STRING,
business_id STRING,
stars INT,
useful INT,
funny INT,
cool INT,
text STRING,
`date` STRING);
```

```
CREATE TABLE IF NOT EXISTS business_table(
business_id STRING,
name STRING,
address STRING,
city STRING,
state STRING,
postal STRING,
lattitude float,
longitude float,
stars INT,
review_count INT,
is_open INT,
attributes STRING,
categories STRING,
hours INT);
```

```
CREATE TABLE IF NOT EXISTS user_table (
user_id STRING,
name STRING,
review_count INT,
yelping_since INT,
useful INT,
funny INT,
cool INT,
fans INT,
elite STRING,
average_stars FLOAT,
compliment_hot INT,
compliment_more INT,
compliment_profile INT,
compliment_cute INT,
compliment_list INT,
compliment_note INT,
compliment_plain INT,
compliment_cool INT,
compliment_funny INT,
compliment_writer INT,
compliment_photos INT
);
```

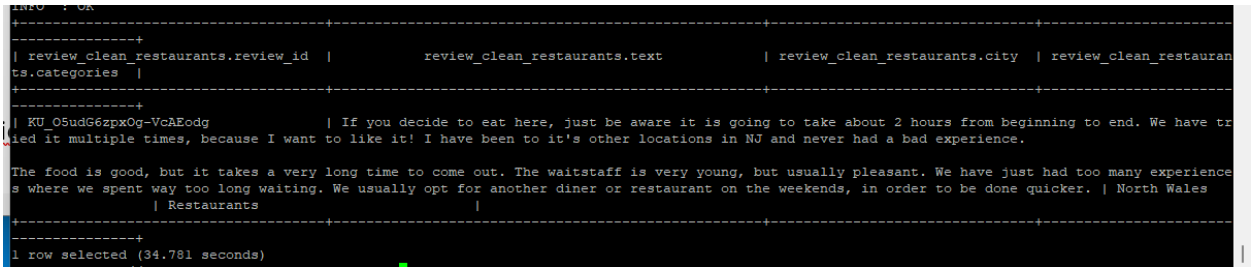
Step 5: Create Table in Hive /Mention all the Queries

The following view will help filter out the review table with the specific fields for our visualization. We will join the business table so we can determine which reviews map to what specific businesses. In this case we want to focus on restaurants and hospitals. We will use the REGEX function to extract both keywords in review_clean_restaurants and review_clean_hospital respectively.

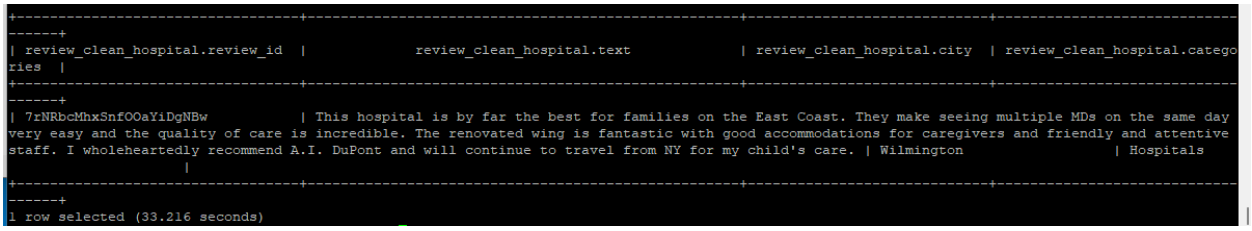
```
CREATE VIEW IF NOT EXISTS review_clean_restaurants AS
SELECT
rt.review_id,
rt.text,
b.city,
REGEXP_EXTRACT(b.categories, 'Restaurants', 0)  AS categories
FROM review_table rt LEFT OUTER JOIN
business_table b
ON rt.business_id = b.business_id where b.categories LIKE
'%Restaurants%';
```

```
select * from review_clean_restaurants LIMIT 1;
```

```
CREATE VIEW IF NOT EXISTS review_clean_hospital AS
SELECT
rt.review_id,
rt.text,
b.city,
REGEXP_EXTRACT(b.categories, 'Hospitals', 0) AS categories
FROM review_table rt LEFT OUTER JOIN
business_table b
ON rt.business_id = b.business_id where b.categories LIKE
'%Hospitals%';
```




```
select * from review_clean_hospital LIMIT 1;
```



Step 5A Part1: Hive Queries to Compute Sentiment

In this step we want to utilize the dictionary imported from STEP 3 to determine the polarity broken down by word and grouped by review_id and user_id for objectives. We will perform simple visualization of the 2 tables.

Create L1, L2, L3 views to compute sentiment and provide a numerical representation of polarity for each word referenced by its review_id. Reference L3_OUTPUT

```
--Create view l1 to compute sentiment
create view IF NOT EXISTS l1 as
select review_id, words
from review_table
lateral view explode(sentences(lower(text))) dummy as words;

-- Create view l2 from l1 to compute sentiment
create view IF NOT EXISTS l2 as
select review_id, word
from l1
lateral view explode( words ) dummy as word;

-- Create view l3 from l2 to compute sentiment
create view IF NOT EXISTS l3 as select
review_id,
l2.word,
case d.polarity
when 'negative' then -1
when 'positive' then 1
else 0 end as polarity
from l2 left outer join dictionary d on l2.word = d.word;
```

```
SELECT * FROM l3 LIMIT 3;
```

l3 output -

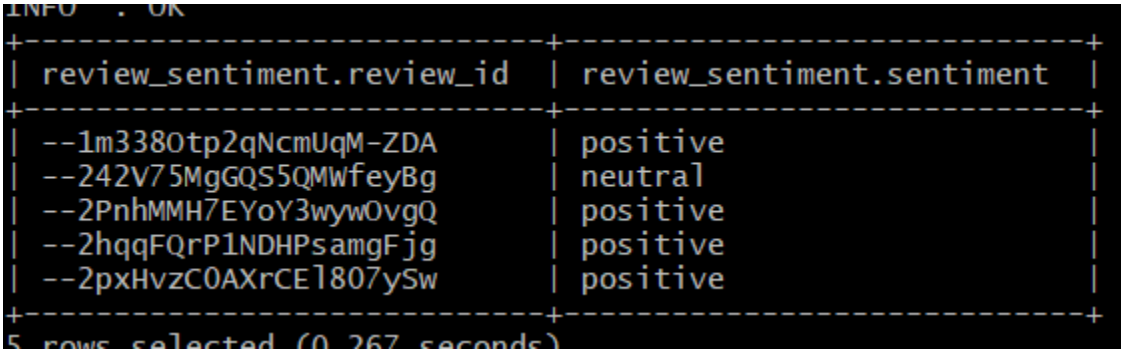
```
+-------------------------+----------+--------------+
|      l3.review_id       | l3.word  | l3.polarity  |
+-------------------------+----------+--------------+
| KU_O5udG6zpxOg-VcAEodg  | if       | 0            |
| KU_O5udG6zpxOg-VcAEodg  | you      | 0            |
| KU_O5udG6zpxOg-VcAEodg  | decide   | 0            |
+-------------------------+----------+--------------+
3 rows selected (37.454 seconds)
```

The following table will store as orc datatype, sum the polarity, and assign the word positive, negative, and neutral the numerical value. It will be grouped by review_id.

```
create table IF NOT EXISTS review_sentiment
stored as orc as select
review_id,
case
when sum( polarity ) > 0 then 'positive'
when sum( polarity ) < 0 then 'negative'
else 'neutral' end as sentiment
from l3 group by review_id;
```

Let's see what the values of the table review_sentiment :

```
SELECT * FROM review_sentiment LIMIT 5;
```



```
INFO  : OK
+-----------------------------+-----------------------------+
| review_sentiment.review_id  | review_sentiment.sentiment  |
+-----------------------------+-----------------------------+
| --1m338Otp2qNcmUqM-ZDA      | positive                    |
| --242V75MgGQS5QMWfeyBg      | neutral                     |
| --2PnhMMH7EYoY3wywOvgQ      | positive                    |
| --2hqqFQrP1NDHPsamgFjg      | positive                    |
| --2pxHvzC0AXrCEl807ySw      | positive                    |
+-----------------------------+-----------------------------+
5 rows selected (0.267 seconds)
```

Analysis 1: Analysis for Top 10 and Bottom 10 Users.

In this section new will consolidate and sum the polarity of all reviews by userid to obtain the "NET" sentiment for each user. We will leverage the l3 polarity table we have completed in the previous step.

Sum polarity by review_id

```
create table IF NOT EXISTS review_sentiment_TOP
stored as orc as select
review_id, sum( polarity ) as sentiment
from l3 group by review_id;
```

Compute the sum of the sentiment by userid

```
CREATE TABLE IF NOT EXISTS user_id_TOP
AS SELECT
rt.user_id, sum( rs.sentiment ) as sentiment
FROM review_sentiment_TOP rs LEFT OUTER JOIN review_table rt
on rs.review_id = rt.review_id group by user_id;
create table IF NOT EXISTS review_sentiment_TOP
stored as orc as select
review_id, sum( polarity ) as sentiment
from l3 group by review_id;
```

Confirm if output of step 1 or 2 is the following.

```
+------------------------+------------------------+
|   user_id_top.user_id  | user_id_top.sentiment  |
+------------------------+------------------------+
| ---2PmXbF47D870stHljqA  | 222                    |
| --0S2HVJui8bEa2iVgUisg  | 5                      |
| --0kuuLmuYBe3Rmu0Iycww  | 17                     |
| --2F5G5LKt3h2cAXJbZptg  | 1                      |
| --2bpE5vyR-2hAP7sZZ41A  | 82                     |
| --4ZhTMV2fIlGhcUk8S5rQ  | 31                     |
| --5jl6efLh3S3NXvn0AsTA  | 4                      |
| --6RLpoufvX9f5gQs_LOuw  | 8                      |
| --78aksX3obHJ667lpjJqA  | 70                     |
| --8IGwUi6ta3OcJjhAhXgg  | 6                      |
+------------------------+------------------------+
10 rows selected (0.206 seconds)
```

Compute overall average of all users in the dataset

```
SELECT AVG(sentiment) as overall_avg from user_id_top;
```

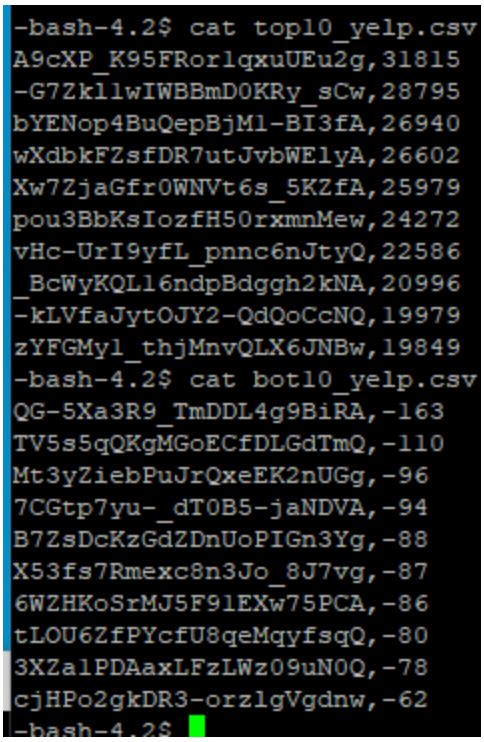
```
+---------------------+
|     overall_avg     |
+---------------------+
| 26.215337167474292  |
+---------------------+
```

Create CSV file of top 10 users by sentiment

```
INSERT OVERWRITE DIRECTORY '/user/pwong4/tmp/top'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT * from user_id_TOP
ORDER BY sentiment DESC LIMIT 10;
```

Create CSV file of top 10 users by sentiment

```
INSERT OVERWRITE DIRECTORY '/user/pwong4/tmp/bottom'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT * from user_id_TOP
ORDER BY sentiment ASC LIMIT 10;
INSERT OVERWRITE DIRECTORY '/user/pwong4/tmp/top'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT * from user_id_TOP
ORDER BY sentiment DESC LIMIT 10;
```



Analysis 2: Sentiment analysis of Restaurant by city.

We need to create the schema and join the review table filtered by 1) restaurant and 2) hospital and join it to the newly created review sentiment on previous step to match the review with the sentiment.

```
CREATE TABLE IF NOT EXISTS reviewbi
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "tmp/reviewbi"
AS SELECT
rc.review_id,rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_restaurants rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id
WHERE 1=0 -- Not to copy value but only schema
```

Insert data into reviewbi table

```
INSERT into table reviewbi
SELECT
rc.review_id, rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_restaurants rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id;
```

Execute the following to confirm:

```
SELECT COUNT(*) FROM reviewbi;
```

```
INFO  : OK
+----------+
|   _c0    |
+----------+
| 4724471  |
+----------+
```

```
select * from reviewbi LIMIT 3;
```

```
+------------------------+----------------+----------------------+----------------------+
|   reviewbi.review_id   | reviewbi.city  | reviewbi.categories  | reviewbi.sentiment   |
+------------------------+----------------+----------------------+----------------------+
| JNe0d2JLZhgaBEo8pBvBrQ | New Orleans    | Restaurants          | 2                    |
| VcMHiUc8qQcsjKa5pWvrpQ | Franklin       | Restaurants          | 2                    |
| 5SsAJAOTBoAudXhckKgdLw | Saint Louis    | Restaurants          | 2                    |
+------------------------+----------------+----------------------+----------------------+
3 rows selected (0.231 seconds)
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai>
```

Analyze for a state " Pennsylvania (PA)" of cities for restaurant category of sentiment for yelp user.

For this analysis, we created a schema and insert data into it:

```
CREATE TABLE IF NOT EXISTS reviewbi_restaurants_pa
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "tmp/reviewbi_restaurants_pa"
AS SELECT
rc.review_id,rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
```

```
when 'negative' then 0
end as sentiment
FROM review_clean_restaurants rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id
WHERE 1=0
LIMIT 1;
```

```
Insert into table reviewbi_restaurants_pa
SELECT
rc.review_id, rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_restaurants_pa rc LEFT OUTER JOIN review_sentiment
rs
on rc.review_id = rs.review_id;
```

Analysis 3: Sentiment analysis of Hospitals by city.

Analyses of Sentiment of hospital by different cities in USA. Also, Analysis of Sentiment Breakdown using the same query for both visualization with 3D power map and tableau

NOTE: Don't forget to replace pwong4 to your account name in the following Hive QL code.

For reviewbi_hospital, we are creating table and inserting the data into it.

```
Drop table if exists reviewbi_hospital;

CREATE TABLE IF NOT EXISTS reviewbi_hospital
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "tmp/reviewbi_hospital"
AS SELECT
rc.review_id, rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_hospital rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id
WHERE 1=0
LIMIT 1;
```

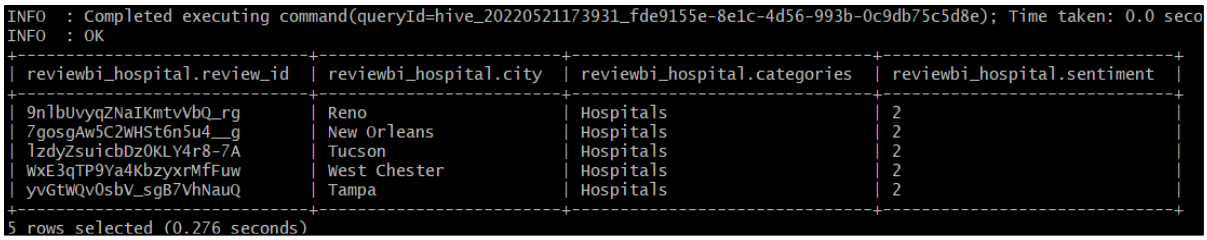
```
Insert into table reviewbi_hospital
SELECT
```

```
rc.review_id, rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_hospital rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id;
```

Now you can query the content of the table:

```
SELECT * FROM reviewbi_hospital LIMIT 5;
```

It will display the result as follows:



```
INFO  : Completed executing command(queryId=hive_20220521173931_fde9155e-8e1c-4d56-993b-0c9db75c5d8e); Time taken: 0.0 seco
INFO  : OK
+-----------------------------+-------------------------+-------------------------------+-------------------------------+
| reviewbi_hospital.review_id | reviewbi_hospital.city  | reviewbi_hospital.categories  | reviewbi_hospital.sentiment   |
+-----------------------------+-------------------------+-------------------------------+-------------------------------+
| 9nlbUvyqZNaIKmtvVbQ_rg      | Reno                    | Hospitals                     | 2                             |
| 7gosgAw5C2WHSt6n5u4__g      | New Orleans             | Hospitals                     | 2                             |
| lzdyZsuicbDzOKLY4r8-7A      | Tucson                  | Hospitals                     | 2                             |
| WxE3qTP9Ya4KbzyxrMfFuw      | West Chester            | Hospitals                     | 2                             |
| yvGtWQvOsbV_sgB7VhNauQ      | Tampa                   | Hospitals                     | 2                             |
+-----------------------------+-------------------------+-------------------------------+-------------------------------+
5 rows selected (0.276 seconds)
```

Analyze for a state" Pennsylvania (PA)" of cities for hospital category of sentiment for yelp user.

For this analysis, we created a schema and insert data into it:

```
CREATE TABLE IF NOT EXISTS reviewbi_hospital_pa
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ","
STORED AS TEXTFILE
LOCATION "tmp/reviewbi_hospital_pa"
AS SELECT
rc.review_id, rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_hospital_pa rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id
WHERE 1=0
LIMIT 1;
```

```
INSERT into table reviewbi_hospital_pa
SELECT
rc.review_id, rc.city, rc.categories,
case rs.sentiment
when 'positive' then 2
```

```
when 'neutral' then 1
when 'negative' then 0
end as sentiment
FROM review_clean_hospital_pa rc LEFT OUTER JOIN review_sentiment rs
on rc.review_id = rs.review_id;
```

Analysis 4: Total count of yelp users for last 10 years.

NOTE: Don't forget to replace pwong4 to your username name in the following Hive QL code.

```
Drop table if exists review_count_last10_years;

CREATE TABLE IF NOT EXISTS total_user_count_last10_years(user_id
string, yelping_since int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
STORED AS TEXTFILE LOCATION
'/user/pwong4/yelp/user/total_user_count_last10_years';

INSERT OVERWRITE TABLE total_user_count_last10_years
Select count(user_id) as user_id, yelping_since from user_table
group by yelping_since order BY yelping_since DESC limit 10;
```

Now you can query the content of the table:

```
select * from total_user_count_last10_years;
```

It will display the result as follows:

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> select * from total_user_count_last10_years;
INFO  : Compiling command(queryId=hive_20220519175350_def9523b-8bd3-4608-a794-7cee7a8c3013): select * from total_user_count_last10_years
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:total_user_count_last10_years.user_id, type:string, comment:null), F
st10_years.yelping_since, type:int, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20220519175350_def9523b-8bd3-4608-a794-7cee7a8c3013); Time taken: 0.296 seconds
INFO  : Executing command(queryId=hive_20220519175350_def9523b-8bd3-4608-a794-7cee7a8c3013): select * from total_user_count_last10_years
INFO  : Completed executing command(queryId=hive_20220519175350_def9523b-8bd3-4608-a794-7cee7a8c3013); Time taken: 0.0 seconds
INFO  : OK
+----------------------------------------+----------------------------------------------+
| total_user_count_last10_years.user_id  | total_user_count_last10_years.yelping_since  |
+----------------------------------------+----------------------------------------------+
| 2782                                   | 2022                                         |
| 40485                                  | 2021                                         |
| 47444                                  | 2020                                         |
| 104655                                 | 2019                                         |
| 133568                                 | 2018                                         |
| 151024                                 | 2017                                         |
| 217620                                 | 2016                                         |
| 247850                                 | 2015                                         |
| 233465                                 | 2014                                         |
| 209762                                 | 2013                                         |
+----------------------------------------+----------------------------------------------+
10 rows selected (0.372 seconds)
```

Analysis 5 : Top states with the highest review count

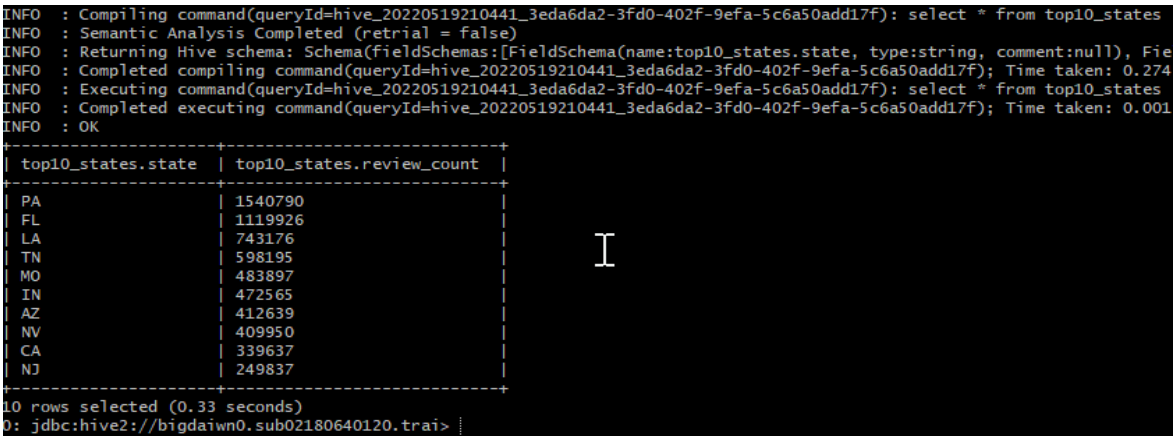
NOTE: Don't forget to replace pwong4 to your username name in the following Hive QL code.

```
DROP TABLE IF EXISTS top10_states;

CREATE TABLE IF NOT EXISTS top10_states(state string, review_count
int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
STORED AS TEXTFILE LOCATION
'/user/pwong4/yelp/business/top10_states';

INSERT OVERWRITE TABLE top10_states
SELECT state, sum(review_count) as review_count from business_table
```

Now you can query the content of the table:

```
select * from top10_states;
```

It will display the result as follows:



Step 6: Downloading data into PC for Sentiment Analysis & Yelp User Analysis

1. Switch on to the first terminal connected to the Oracle cloud to download the output

file at the HDFS path

After the Hive tables are created, you can download the file to your lab (or personal PC/Laptop) as follows:

Note: use your username and IP address below.

$ ssh pwong4@ipaddress

pwong4@ipaddress's password:

Run the following command to check if files are present:

For Top 10 User sentiment and Bottom 10 user sentiment, restaurant and hospital:

```
hdfs dfs -ls tmp/top
hdfs dfs -ls tmp/bottom
hdfs dfs -ls tmp/reviewbi
hdfs dfs -ls tmp/reviewbi_restaurant_pa
hdfs dfs -ls tmp/reviewbi_hospital
hdfs dfs -ls tmp/reviewbi_hospital_pa
```

```
-bash-4.2$ hdfs dfs -ls tmp/top


Found 1 items
-rw-r--r--   3 pwong4 hdfs        290 2022-05-22 02:51 tmp/top/000000_0

-bash-4.2$
-bash-4.2$ hdfs dfs -ls tmp/bottom

Found 1 items
-rw-r--r--   3 pwong4 hdfs        272 2022-05-22 03:06 tmp/bottom/000000_0
```

Total count of yelp users for last 10 years & Top states with the highest review count

```
-bash-4.1$ hdfs dfs -ls yelp/user/total_user_count_last10_years/*

-bash-4.1$ hdfs dfs -ls yelp/business /top10_states/*
```

You will see only one file named 000000_0 is present in all the following folders:

```
-bash-4.2$ hdfs dfs -ls yelp/user/*
Found 1 items
drwxr-xr-x   - pwong4 hdfs          0 2022-05-19 17:53 yelp/user/total_user_count_last10_years/base_0000001
-rw-r--r--   3 pwong4 hdfs 3363329011 2022-05-12 02:23 yelp/user/yelp_academic_dataset_user.json
-bash-4.2$ hdfs dfs -ls yelp/user/total_user_count_last10_years/base_0000001/*
-rw-r--r--   3 pwong4 hdfs        116 2022-05-19 17:53 yelp/user/total_user_count_last10_years/base_0000001/000000_0
```

Similarly, do it for the other files.

```
yelp/user/total_user_count_last10_years/000000_0
yelp/business /top10_states /000000_0
```

2. Download the output files to the local file systems and rename it

For these steps, we need to concatenate the csv files into the local Linux file system for downloading in step.

```
hdfs dfs -get tmp/top/000000_0 top10_yelp.csv
hdfs dfs -get tmp/bottom/000000_0 bot10_yelp.csv
cat top10_yelp.csv
cat bot10_yelp.csv
```

```
-bash-4.2$ cat top10_yelp.csv
A9cXP_K95FRorlqxuUEu2g,31815
-G7Zk1lwIWBBmD0KRy_sCw,28795
bYENop4BuQepBjM1-BI3fA,26940
wXdbkFZsfDR7utJvbWElyA,26602
Xw7ZjaGfr0WNVt6s_5KZfA,25979
pou3BbKsIozfH50rxmnMew,24272
vHc-UrI9yfL_pnnc6nJtyQ,22586
_BcWyKQL16ndpBdggh2kNA,20996
-kLVfaJytOJY2-QdQoCcNQ,19979
zYFGMyl_thjMnvQLX6JNBw,19849
-bash-4.2$ cat bot10_yelp.csv
QG-5Xa3R9_TmDDL4g9BiRA,-163
TV5s5qQKgMGoECfDLGdTmQ,-110
Mt3yZiebPuJrQxeEK2nUGg,-96
7CGtp7yu-_dT0B5-jaNDVA,-94
B7ZsDcKzGdZDnUoPIGn3Yg,-88
X53fs7Rmexc8n3Jo_8J7vg,-87
6WZHKoSrMJ5F91EXw75PCA,-86
tLOU6ZfPYcfU8qeMqyfsqQ,-80
3XZalPDAaxLFzLWz09uN0Q,-78
cjHPo2gkDR3-orzlgVgdnw,-62
-bash-4.2$
```

```
hdfs dfs -cat tmp/reviewbi/delta_0000002_0000002_0000/0000* >
/home/pwong4/combined_restaurant.csv
cat combined_restaurant.csv | tail -n 2

hdfs dfs -cat tmp/reviewbi_hospital/delta_0000002_0000002_0000/0000* >
/home/pwong4/combined_hospital.csv
cat combined_hospital.csv | tail -n 2

hdfs dfs -cat
tmp/reviewbi_hospital_pa/delta_0000002_0000002_0000/0000* >
/home/pwong4/combined_hospital_pa.csv
cat combined_hospital_pa.csv | tail -n 2

hdfs dfs -cat
tmp/reviewbi_restaurants_pa/delta_0000002_0000002_0000/0000* >
/home/pwong4/combined_restaurant_pa.csv
cat combined_restaurant_pa.csv | tail -n 2
```

Since all the folders have the same name file 000000_0, while downloading into the local file system you will rename the file. Moreover, cat command allows to read the file.

```
-bash-4.2$ hdfs dfs -get yelp/business/top10_states/base_0000001/000000_0 top10_states.csv
-bash-4.2$ cat top10_states.csv | tail -n 2
CA,339637
NJ,249837
```

Similarly, do it for the other files.

```
-bash-4.1$ hdfs dfs -get yelp/business/top10_states/000000_0
top10_states.csv
-bash-4.1$ cat top10_states.csv | tail -n 2

-bash-4.1$ hdfs dfs -get
yelp/user/total_user_count_last10_years/000000_0
total_user_count_last10_years.csv
-bash-4.1$ cat total_user_count_last10_years.csv |tail -n 2
```

3. Open another terminal with git bash in order to import the output file using your lab

computer (or your PC/Laptop) - you have to download the file to your lab computer

(or your PC/Laptop). For example, your output file at the oracle cloud server is located

at /home/pwong4/top10_states.csv and remotely download the files.

Open a terminal session on your local PC and navigate to the directory where you'd like to save your files. In this example, I've navigated to the following directory of "Project"

```
C:\Users\sonic\Desktop\CIS5200\Project>
```

Run the following command to copy the combined files to local machine. You will be prompted for your credentials. Provide your password

Navigate to local path and use scp to transfer files as previously completed.

```
scp pwong4@129.146.154.176:/home/pwong4/bot10_yelp.csv .
scp pwong4@129.146.154.176:/home/pwong4/top10_yelp.csv .
```

```
 scp pwong4@129.146.154.176:/home/pwong4/combined* .
 Similarly,
 scp pwong4@129.146.154.176:/home/pwong4/combined_hospital.csv .
 scp pwong4@129.146.154.176:/home/pwong4/combined_hospital_pa.csv .
 scp pwong4@129.146.154.176:/home/pwong4/combined_restaurant_pa.csv .
```

```
C:\Users\sonic\Desktop\CIS5200\Project>scp pwong4@129.146.154.176:/home/pwong4/combined* .
pwong4@129.146.154.176's password:
combined_hospital.csv                                   100%  439KB   6.7MB/s   00:00
combined_hospital_pa.csv                                100%  150KB  10.4MB/s   00:00
combined_restaurant.csv                                 100%  188MB  96.4MB/s   00:01
combined_restaurant_pa.csv                              100%   52MB  90.4MB/s   00:00
```

3. Confirm the files have been transferred using the dir command:

```
Dir combined*
```

```
:\Users\sonic\Desktop\CIS5200\Project>dir combined*
 Volume in drive C is Windows
 Volume Serial Number is 585C-A593

 Directory of C:\Users\sonic\Desktop\CIS5200\Project

05/21/2022  06:06 PM           449,806 combined_hospital.csv
05/21/2022  06:06 PM           153,751 combined_hospital_pa.csv
05/21/2022  06:06 PM       197,417,749 combined_restaurant.csv
05/21/2022  06:06 PM        54,080,611 combined_restaurant_pa.csv
               4 File(s)    252,101,917 bytes
               0 Dir(s)  44,307,431,424 bytes free
```

```
$ scp pwong4@129.146.154.176:/home/pwong4/top10_states.csv
```

```
$ scp pwong4@129.146.154.176:/home/pwong4/top10_states.csv .
pwong4@129.146.154.176's password:
top10_states.csv                                        100%  102     2.2KB/s   00:00
```

Similarly, do it for other files.

```
$ scp pwong4@129.146.154.176:/home/pwong4/total_user_count_last10_years.csv .
```

Step 7: Data Visualization Using Tableau

Visualization 1. Showing and visualizing the Top 10 and bottom 10 user sentiment

TOP 10 user sentiment

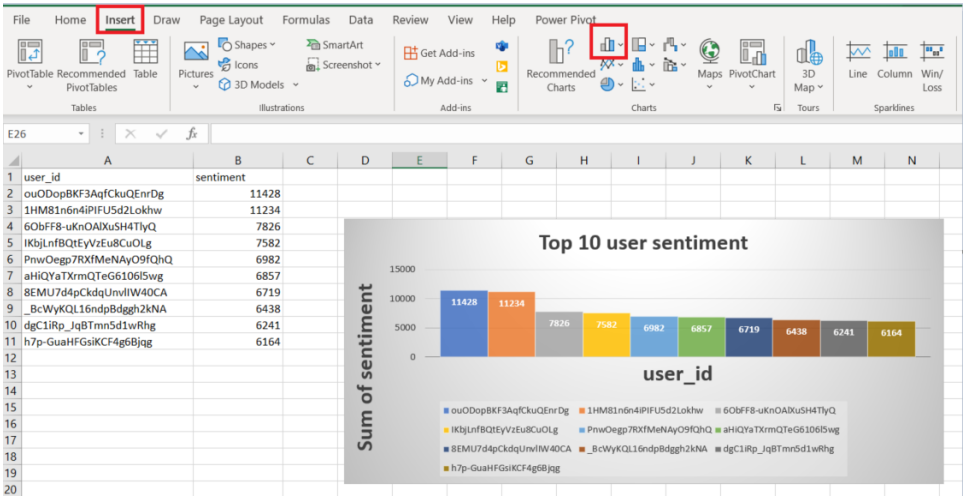
For this sentiment, we opened the top10_yelp.csv file.

1. Modified first row.

2.Go to Insert tab -> click on highlighted bar chart -> Default bar chart will be display with the user_id and sentiment column.

3.On the displayed chart, click on the right click, pop-up will be displayed.

4. Select Change chart type option and select the chart below for the top 10 user sentiment visualizations.



Bottom 10 user sentiment

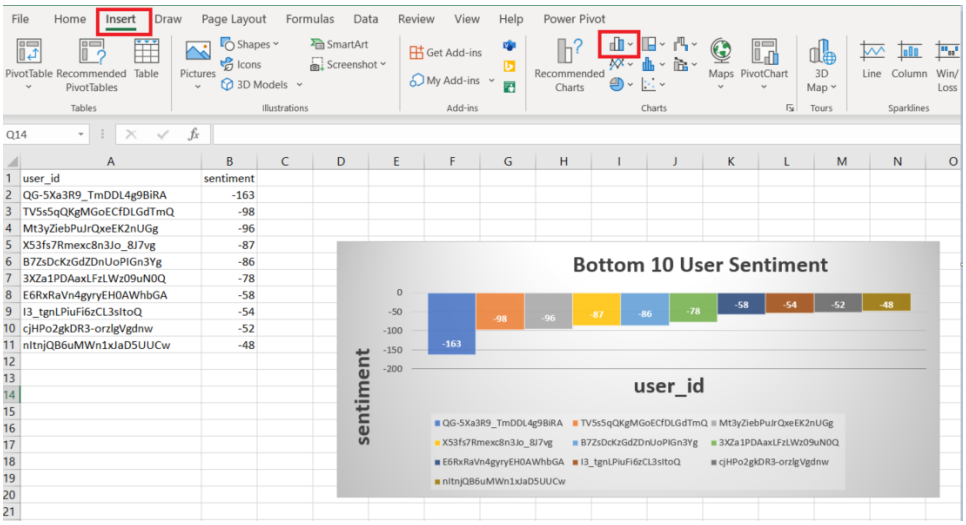
For this sentiment, we opened the bot10_yelp.csv file.

1. Modified first row.

2.Go to Insert tab -> click on highlighted bar chart -> Default bar chart will be display with the user_id and sentiment column.

3.On the displayed chart, click on the right click, pop up will be displayed.

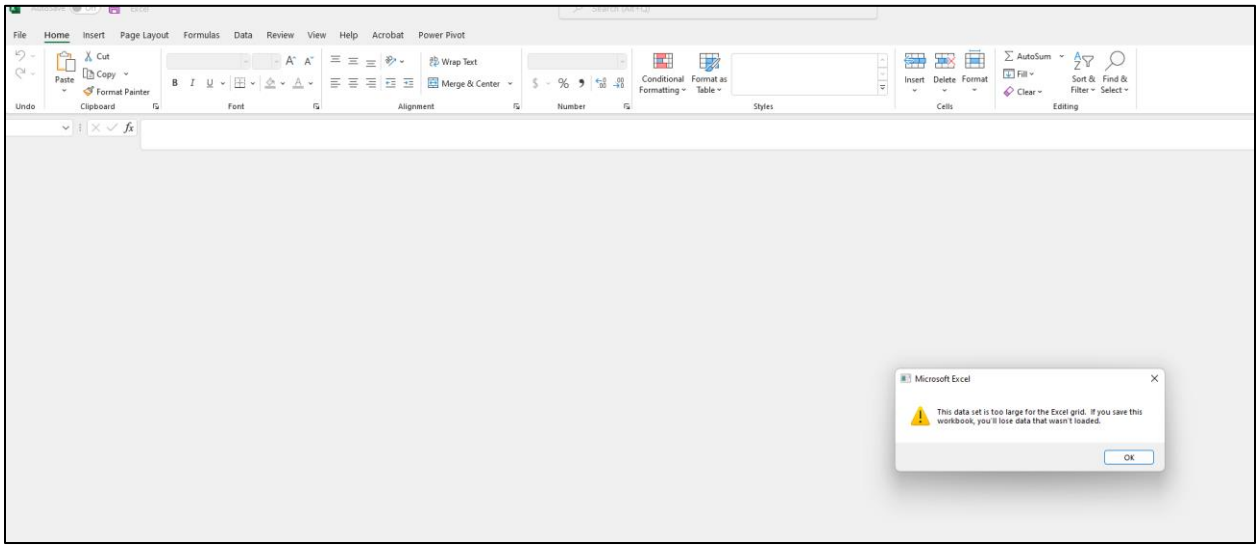
4.Select Change chart type option and select the chart below for the bottom 10 user sentiment visualization.



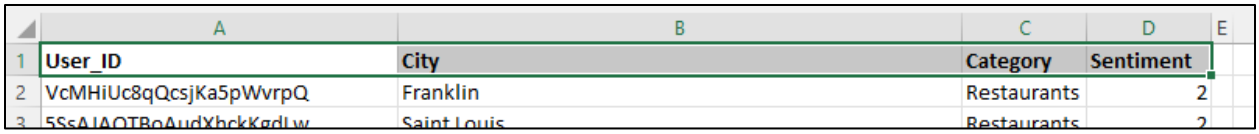
Visualization 2 - Data Visualization using Excel PowerMap for 1) Restaurants, 2)Restaurants in Pennsylvania.

Section 1 - Restaurants

a. Open the combined_restaurant.csv in excel and ignore the error below.

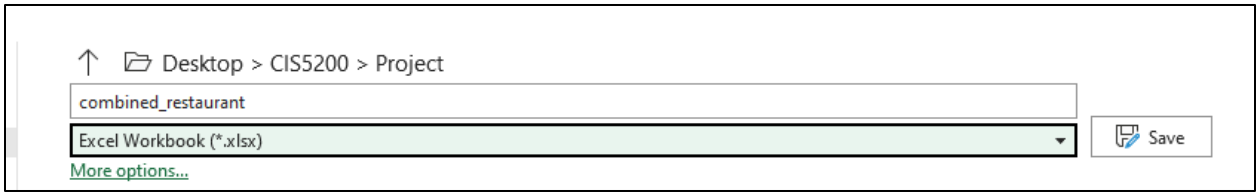


b. Modify the first row to the following shown:

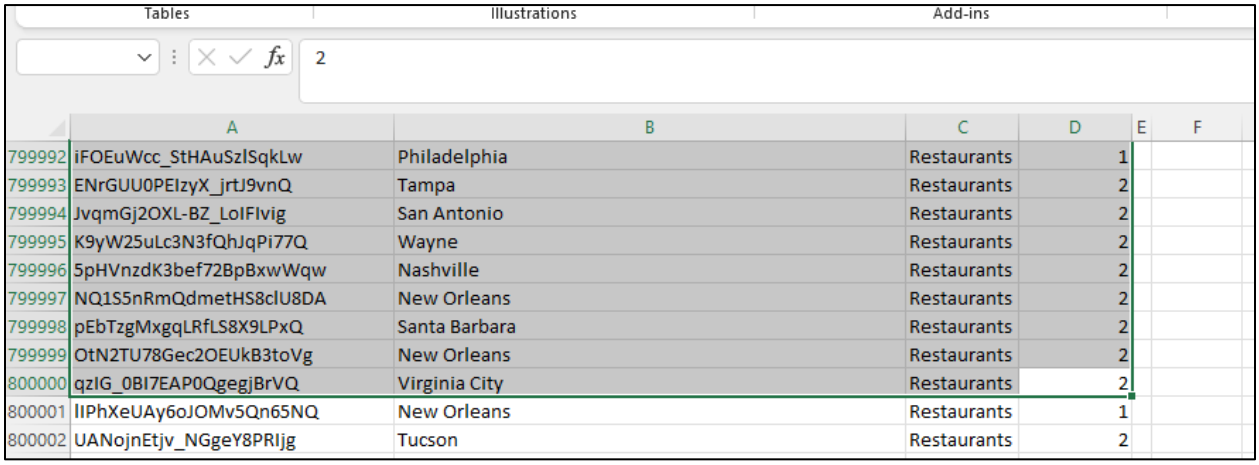


	A	B	C	D	E
1	User_ID	City	Category	Sentiment	
2	VcMHiUc8qQcsjKa5pWvrpQ	Franklin	Restaurants	2	
3	5SsAJAOTBoAudXbckKgdLw	Saint Louis	Restaurants	2	

c. Save as .xlxs as shown:

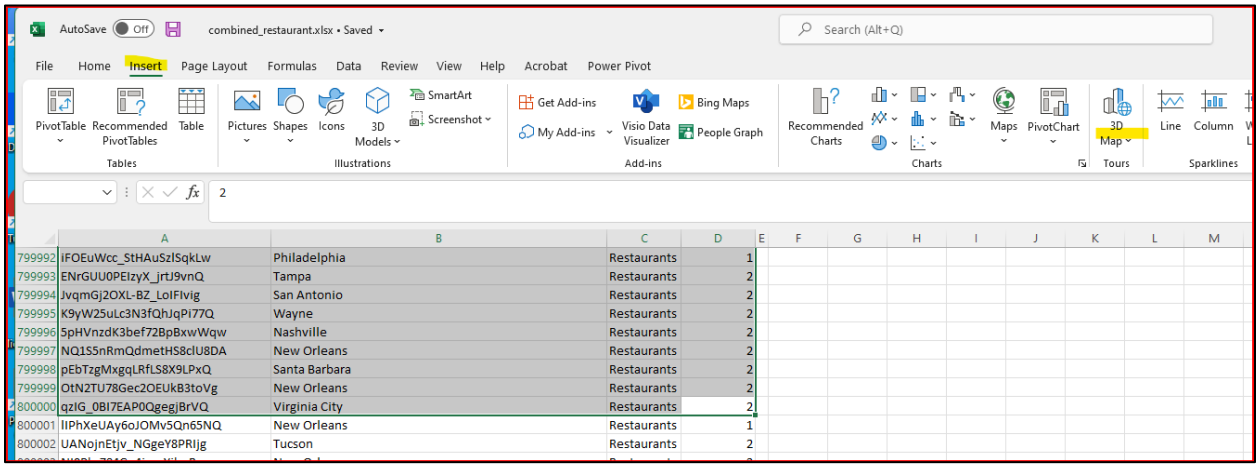


d. Select all columns to row ‘800000’ (note we will be unable to utilize the entire data-set due to the excel 3dmap limitations.)



	A	B	C	D	E	F
799992	iFOEuWcc_StHAuSzlSqkLw	Philadelphia	Restaurants	1		
799993	ENrGUU0PEIzyX_jrtJ9vnQ	Tampa	Restaurants	2		
799994	JvqmGj2OXL-BZ_LoIFIvig	San Antonio	Restaurants	2		
799995	K9yW25uLc3N3fQhJqPi77Q	Wayne	Restaurants	2		
799996	5pHVnzdK3bef72BpBxwWqw	Nashville	Restaurants	2		
799997	NQ1S5nRmQdmetHS8clU8DA	New Orleans	Restaurants	2		
799998	pEbTzgMxgqLRfLS8X9LPxQ	Santa Barbara	Restaurants	2		
799999	OtN2TU78Gec2OEUkB3toVg	New Orleans	Restaurants	2		
800000	qzIG_0BI7EAP0QgegjBrVQ	Virginia City	Restaurants	2		
800001	IlPhXeUAy6oJOMv5Qn65NQ	New Orleans	Restaurants	1		
800002	UANojnEtjv_NGgeY8PRIjg	Tucson	Restaurants	2		

e. Insert -> 3d MAP

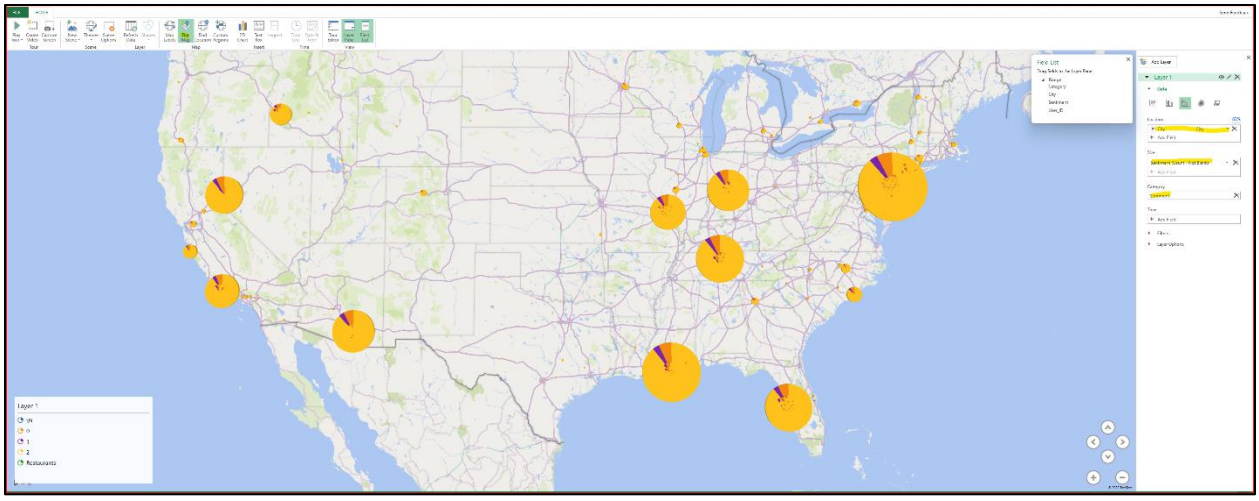


	A	B	C	D	E	F	G	H	I	J	K	L	M
799992	iFOEuWcc_StHAuSzlSqkLw	Philadelphia	Restaurants	1									
799993	ENrGUU0PEIzyX_jrtJ9vnQ	Tampa	Restaurants	2									
799994	JvqmGj2OXL-BZ_LoIFIvig	San Antonio	Restaurants	2									
799995	K9yW25uLc3N3fQhJqPi77Q	Wayne	Restaurants	2									
799996	5pHVnzdK3bef72BpBxwWqw	Nashville	Restaurants	2									
799997	NQ1S5nRmQdmetHS8clU8DA	New Orleans	Restaurants	2									
799998	pEbTzgMxgqLRfLS8X9LPxQ	Santa Barbara	Restaurants	2									
799999	OtN2TU78Gec2OEUkB3toVg	New Orleans	Restaurants	2									
800000	qzIG_0BI7EAP0QgegjBrVQ	Virginia City	Restaurants	2									
800001	IlPhXeUAy6oJOMv5Qn65NQ	New Orleans	Restaurants	1									
800002	UANojnEtjv_NGgeY8PRIjg	Tucson	Restaurants	2									

f. Select ‘Flat Map’ and Location -> City, Size -> Sentiment (Count – Not Blank), Category -> Sentiment.

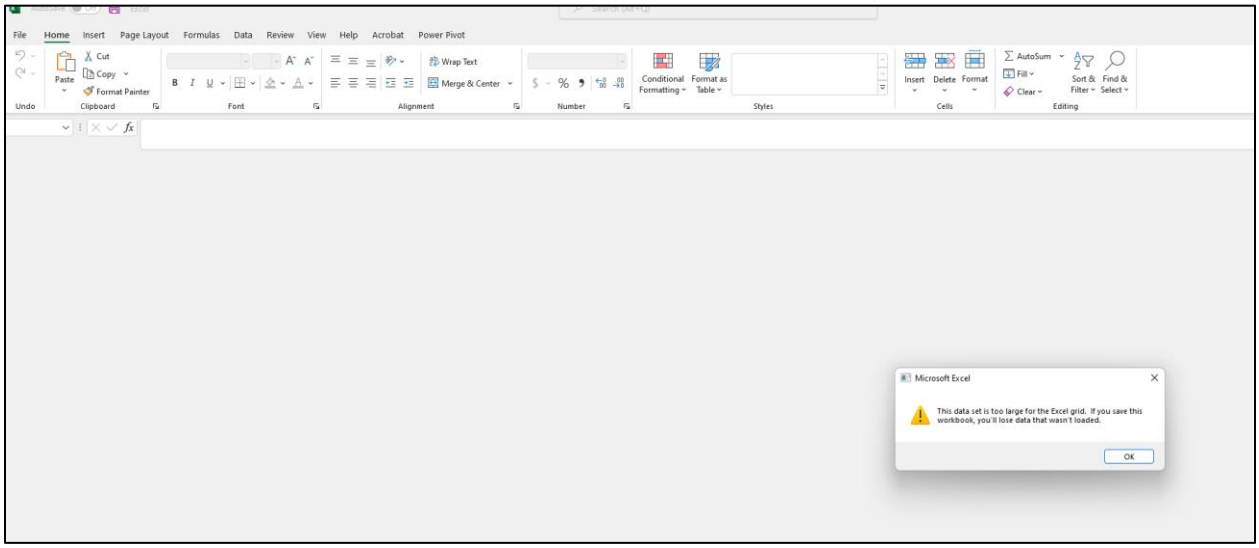
NOTE: Navigate to Insert → 3D Maps

Zoom into US country for granularity as shown.



Section 2 – Restaurants for Pennsylvania (PA)

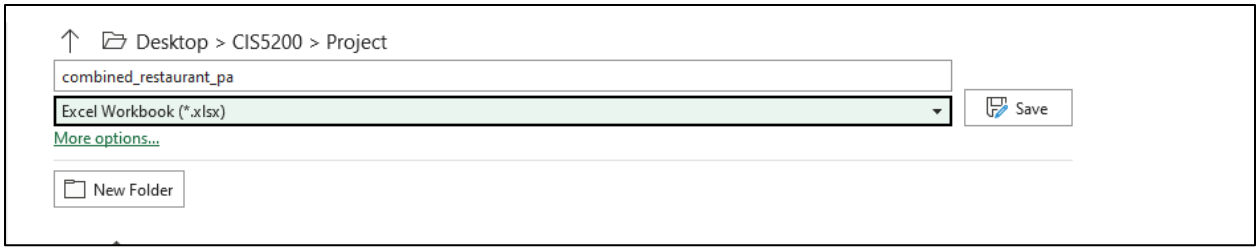
a. Open the combined_restaurant_pa.csv in excel and ignore the error below.



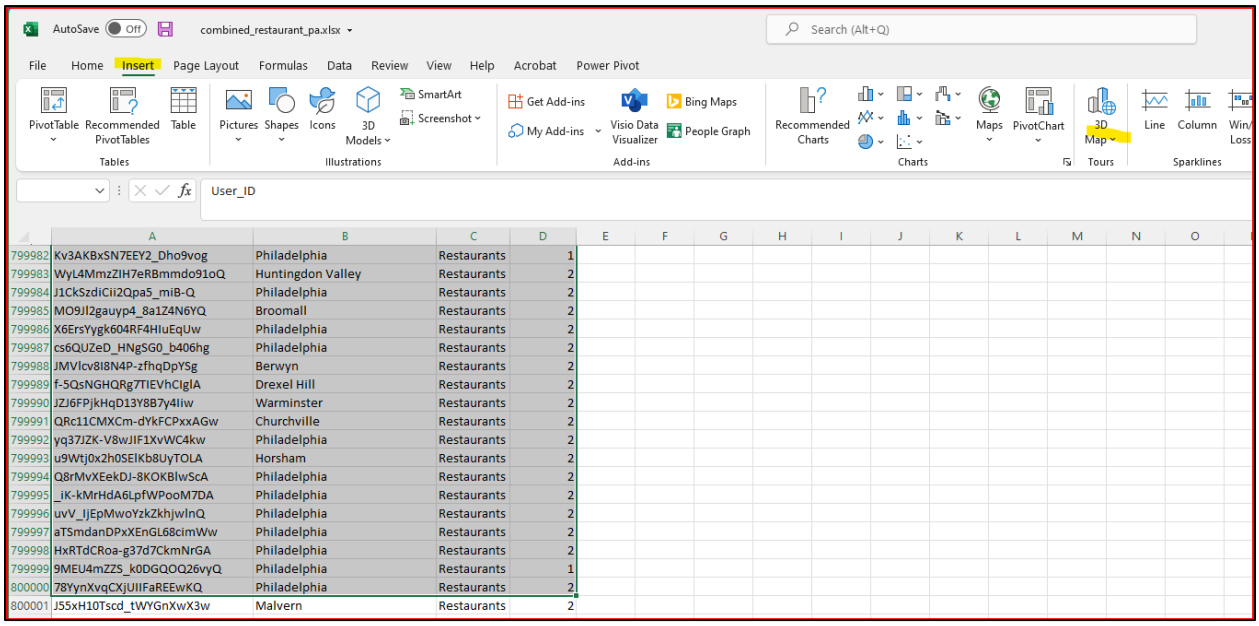
b. Modify the first row to the following shown:

	A	B	C	D	E
1	User_ID	City	Category	Sentiment	
2	VcMHiUc8qQcsjKa5pWvrpQ	Franklin	Restaurants	2	
3	5SsAJAQTBoAudXbckKgdLw	Saint Louis	Restaurants	2	

c. Save as .xlxs

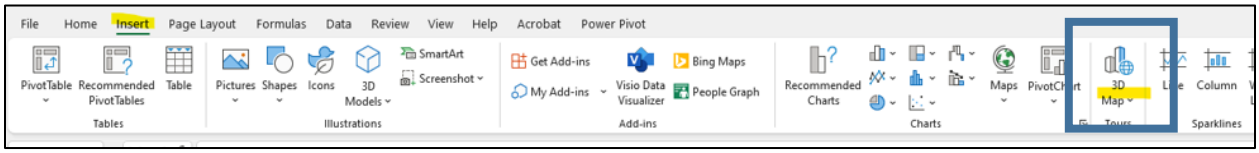


d. Select all columns to row ‘800000’ (note we will be unable to utilize the entire data-set due to the excel 3dmap limitations.)



	A	B	C	D
799982	Kv3AKBxSN7EEY2_Dho9vog	Philadelphia	Restaurants	1
799983	WyL4MmzZIH7eRBmmdo91oQ	Huntingdon Valley	Restaurants	2
799984	J1CkSzdiCii2Qpa5_miB-Q	Philadelphia	Restaurants	2
799985	MO9JI2gauyp4_8a1Z4N6YQ	Broomall	Restaurants	2
799986	X6ErsYygk604RF4HIuEqUw	Philadelphia	Restaurants	2
799987	cs6QUZeD_HNgSG0_b406hg	Philadelphia	Restaurants	2
799988	JMVIcv8I8N4P-zfhqDpYSg	Berwyn	Restaurants	2
799989	f-5QsNGHQRg7TIEVhCIgIA	Drexel Hill	Restaurants	2
799990	JZJ6FPjkHqD13Y8B7y4Iiw	Warminster	Restaurants	2
799991	QRc11CMXCm-dYkFCPxxAGw	Churchville	Restaurants	2
799992	yq37JZK-V8wJIF1XvWC4kw	Philadelphia	Restaurants	2
799993	u9Wtj0x2h0SEIKb8UyTOLA	Horsham	Restaurants	2
799994	Q8rMvXEekDJ-8KOKBlwScA	Philadelphia	Restaurants	2
799995	_iK-kMrHdA6LpfWPooM7DA	Philadelphia	Restaurants	2
799996	uvV_IjEpMwoYzkZkhjwInQ	Philadelphia	Restaurants	2
799997	aTSmdanDPxXEnGL68cimWw	Philadelphia	Restaurants	2
799998	HxRTdCRoa-g37d7CkmNrGA	Philadelphia	Restaurants	2
799999	9MEU4mZZS_k0DGQOQ26vyQ	Philadelphia	Restaurants	1
800000	78YynXvqCXjUIIFaREEwKQ	Philadelphia	Restaurants	2
800001	J55xH10Tscd_tWYGnXwX3w	Malvern	Restaurants	2

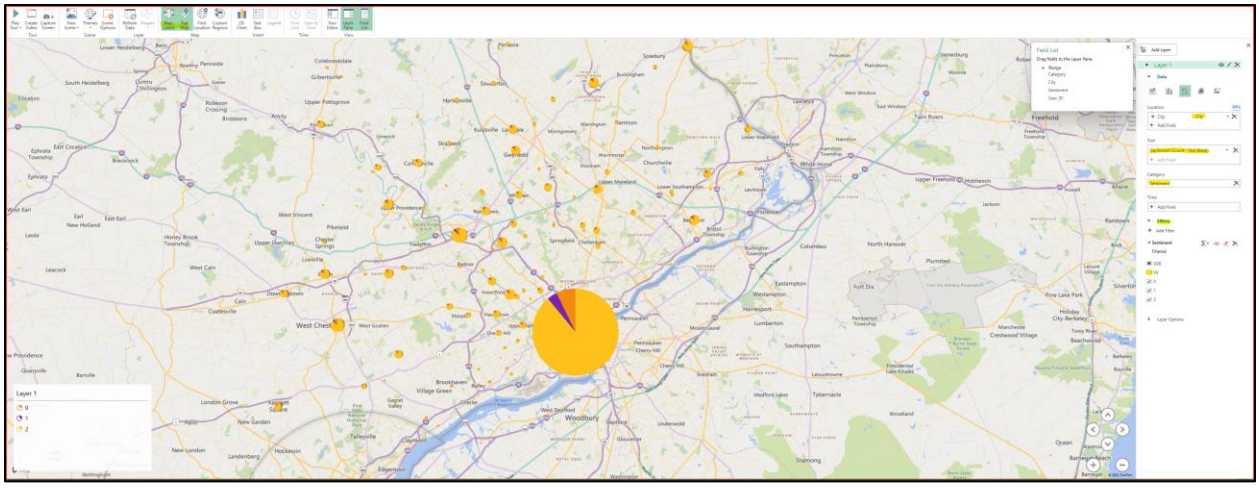
e. Insert -> 3d MAP



f. Select ‘Flat Map’ & ‘Map Labels’ and Location -> City, Size -> Sentiment (Count – Not Blank), Category -> Sentiment.

Apply Filter to remove ‘\N’ Values.

Zoom into Philadelphia for granularity.



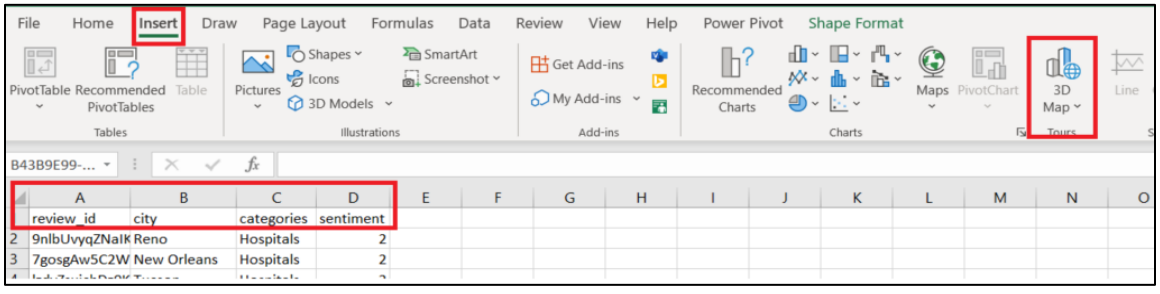
Visualization 3.Total yelpers using hospital as category in yelp for different cities in USA.

a) open the combined_hospital.csv file in Microsoft Excel. For the first row of the file, you need to insert the header to each column as follows:

review_id city categories sentiment

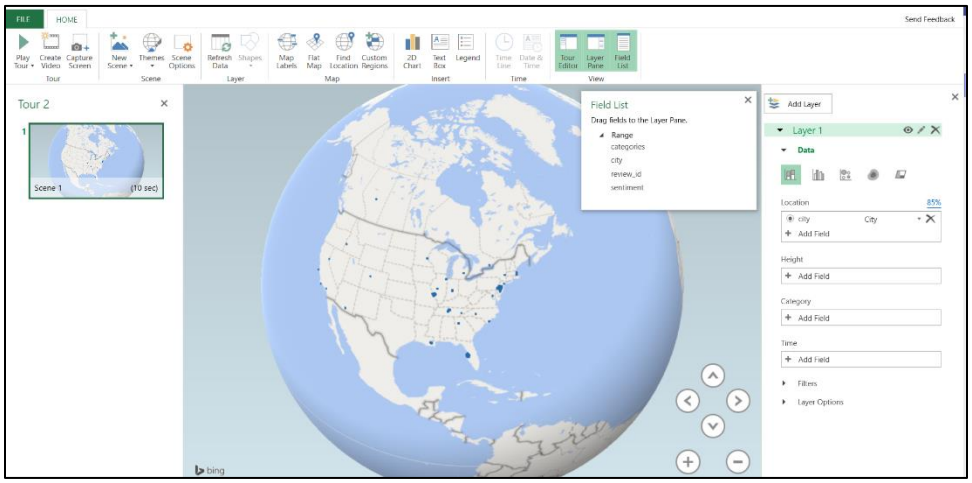
review_id	city	categories	sentiment
9nlbUvyqZNaIK	Reno	Hospitals	2
7gosgAw5C2W	New Orleans	Hospitals	2
lzdyZsuicbDz0K	Tucson	Hospitals	2
WxE3qTP9Ya4I	West Chester	Hospitals	2
yvGtWQv0sbV_	Tampa	Hospitals	2
t_xLp4eYaTuiQ	Philadelphia	Hospitals	2
YMUuqwved74	New Orleans	Hospitals	2

b) You have to save the file in excel format, that is, as combined_hospital.xlsx .Go to the Insert tab and click on “3D power-map" enabled – only excel file in XLSX should be enable “3D Map”. maps. If it complains that 3D Map cannot be open, then you need to make sure if we insert headers into the first row:



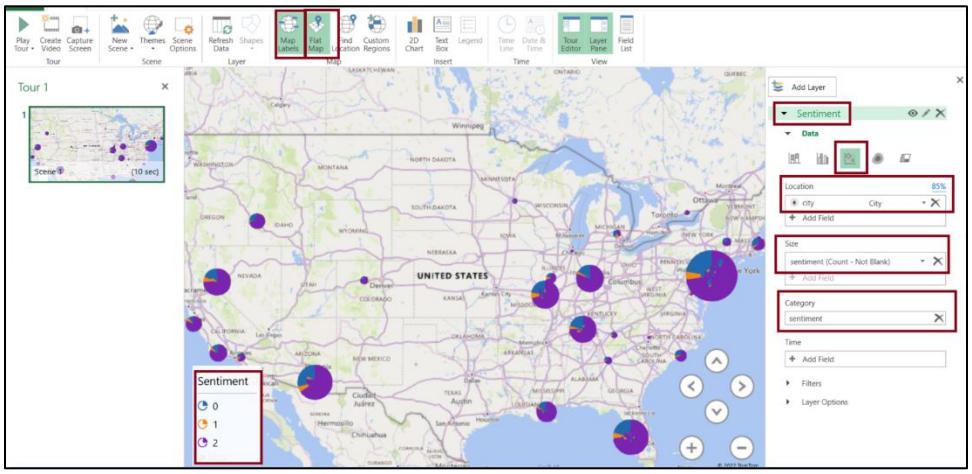
c) You will see the following 3D map.

NOTE: If you don't see the layer frame on the right side, you may select all data manually before opening 3D map:

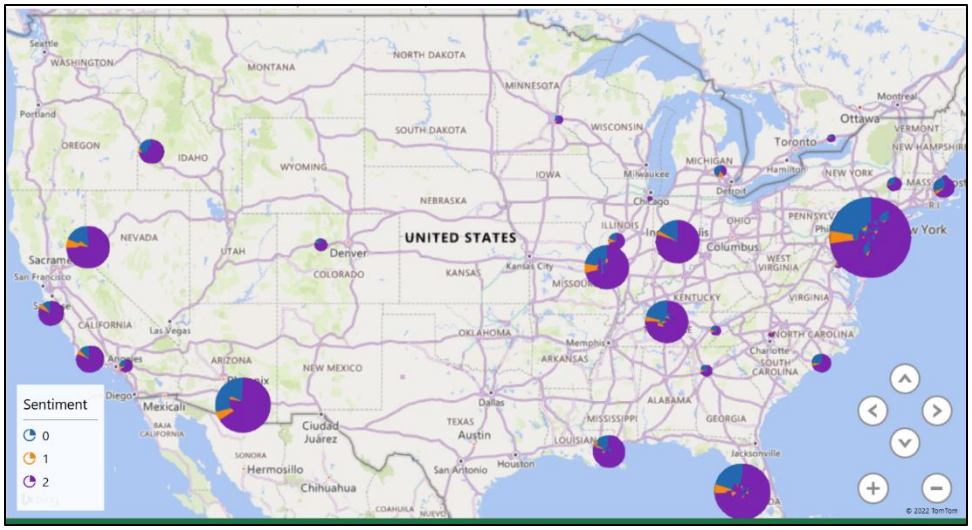


d) You need to select the properties and values in the layer as follows. Then, you can drag the earth and rotate it to observe the sentiment of the world (Only USA country).

Then rename layer 1 as Sentiment, then select 3rd graphs (pie chart), Select location, size, and category. Select Map Labels --> Flat Map on the header section on the top.

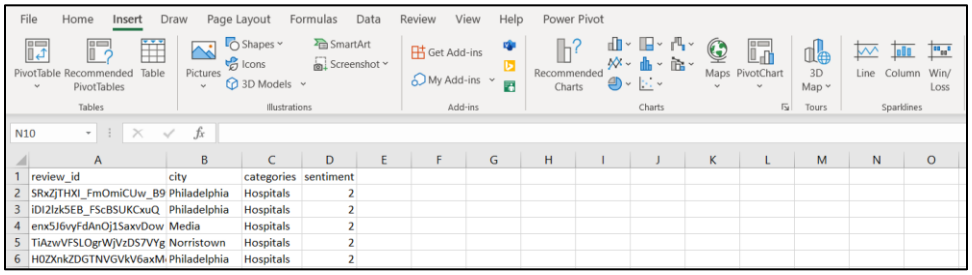


e) Now you can kill "tour 1" frame in the left. Also, you may move or resize the Layer (Sentiment) menu. You can also zoom in and zoom out by using '+' and '-' icon on the down corner side of the map. Then you can see the analysis of sentiment for the hospital category for the yelp user for the United States.



Section 2 – Hospitals for Pennsylvania (PA)

a. Open the combined_hospital_pa.csv in excel. Insert Headers into first row and convert csv file into xlsx file. Click on Insert --> 3D map enabled.

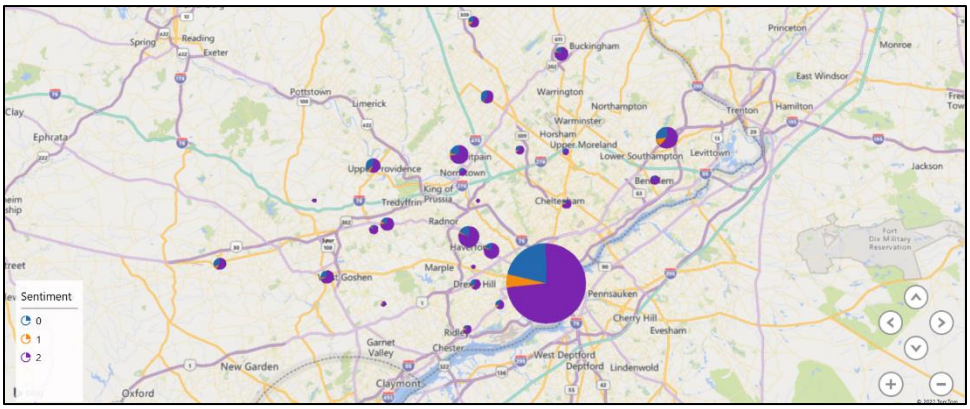


Click on 3D map, combined_hospital_pa.xlsx file in geographic map will be open

Select ‘Flat Map’ & ‘Map Labels’ and Location -> City

Size -> Sentiment (Count – Not Blank), Category -> Sentiment. See highlighted for details.

Zoom into Philadelphia for granularity.



Section 3 – Sentiment Breakdown

Sentiment Breakdown

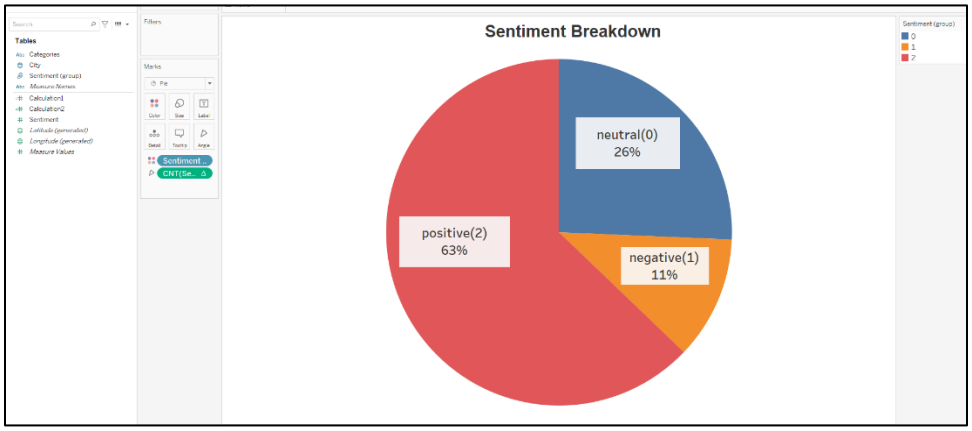
a) Open the combined_hospital.csv file in Microsoft excel. For the first row of the file, you need to insert the header to each column as follows:

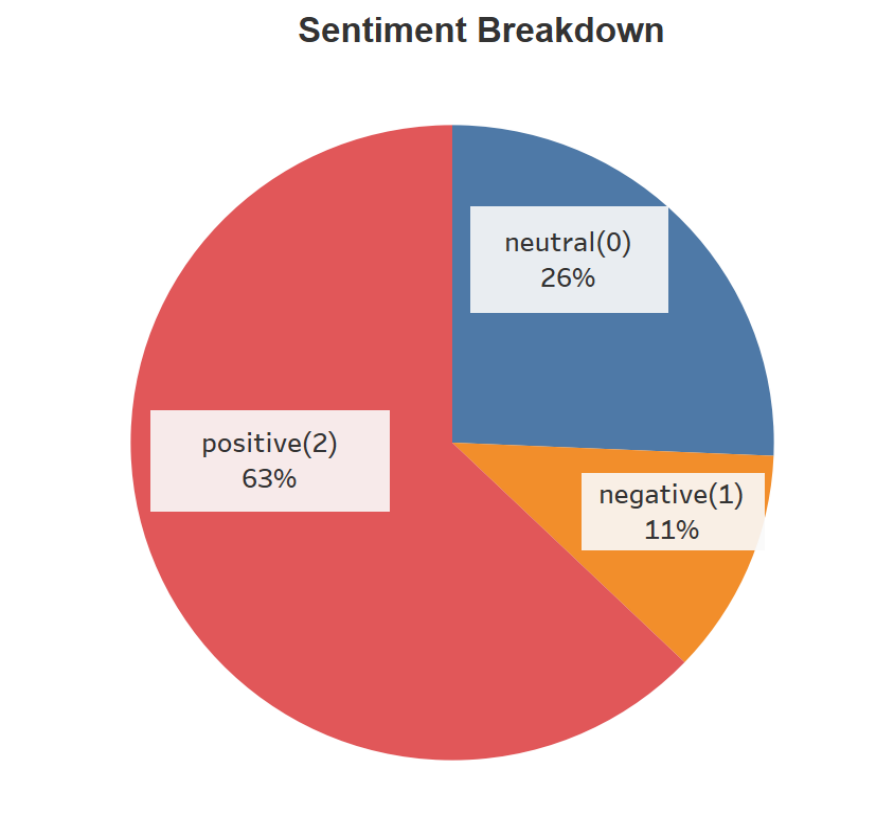
review_id city categories sentiment

review_id	city	categories	sentiment
9nlbUvyqZNaIK	Reno	Hospitals	2
7gosgAw5C2W	New Orleans	Hospitals	2
lzdyZsuicbDz0K	Tucson	Hospitals	2
WxE3qTP9Ya4H	West Chester	Hospitals	2
yvGtWQv0sbV_	Tampa	Hospitals	2
t_xLp4eYaTuiQ	Philadelphia	Hospitals	2
YMUuqwved74	New Orleans	Hospitals	2

b) Open Tableau, Import the excel sheet.

c) Click on sheet 1, Columns name will display under dimensions and measures on the left-hand side. Click on group sentiment by selecting create group option on the right click pop up and select the count for sentiment column under measures with percent of Total in the right click of quick table calculation. Then, select Pie under drop down option in the Marks section. After that drag the sentiment(group) on the color tab and count sentiment on the Angle tab on the Marks section. We also down annotation area to get more clearly view about the analyzed the sentiment breakdown result.





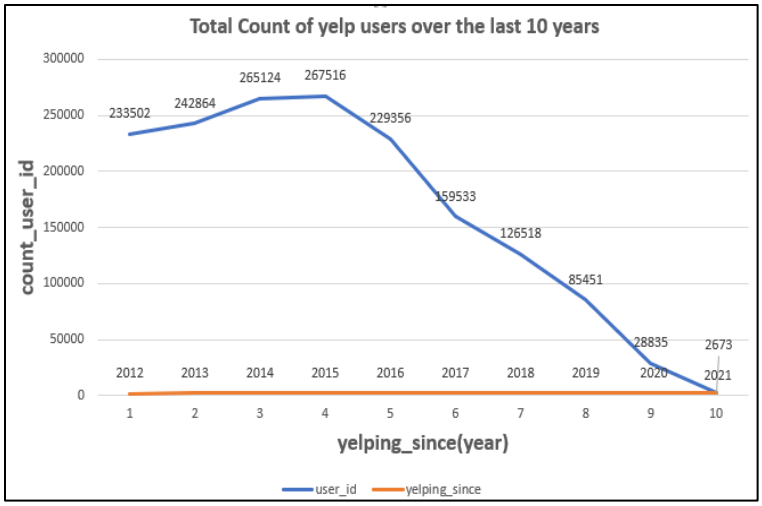
Visualization 4. Total count of yelp users for last 10 years

a) open the total_user_count_last10_years.csv file in Microsoft Excel. For the first row of the file, you need to insert the header to each column as follows:

user_id yelping_since

user_id	yelping_since
233502	2012
242864	2013
265124	2014
267516	2015
229356	2016
159533	2017
126518	2018
85451	2019
28835	2020
2673	2021

b) Go to the Insert tab and visualize the output in scatter line graph to represent the maximum number of users over the period of 10 years.



Visualization 5.Top 10 states with the highest review count

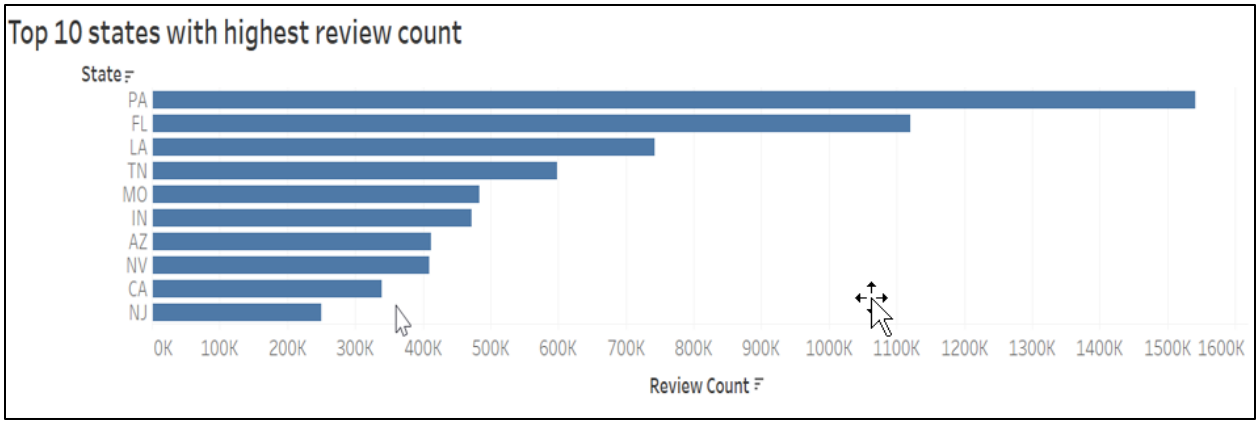
a) Open the top10_states.csv file in Microsoft Excel. For the first row of the file, you need to insert the header to each column as follows:

States Review_Count

States	Review_Count
PA	1540790
FL	1119926
LA	743176
TN	598195
MO	483897
IN	472565
AZ	412639
NV	409950
CA	339637
NJ	249837

b) Save the file in xlsx format. Open Tableau. Import the excel Sheet.

c) Select graph horizontal bars. Drag sum(review_count) to columns and states to rows.



Conclusion

In this tutorial, you learned how Hadoop Cluster can be used to analysis sentiment of yelp data using Apache Hive. You went through a flow to understand how the raw data is first upload to HDFS, and then loaded to Hive tables for performing queries. Finally, you learned how to import the results of Hive queries and to create visualizations using tableau, and 3D Map chart, bar chart in Microsoft excels.

References

1. URL of Data Source https://www.yelp.com/dataset/download

2. GitHub https://github.com/pooja9050/Yelp-Data-Analysis-Using-Hive

3. References A Very Extensive Data Analysis of Yelp | Kaggle, http://hortonworks.com/hadoop-tutorial/how-to-refine-and-visualize-sentiment-data/