CIS5200 Term Project Tutorial

Authors: **Fereshteh Mamaghani, Adrian Marroquin, Aleksander Sekowski, Jinhui Liu & Siying Chen**

Instructor: Jongwook Woo

Date: 12/08/2020

Lab Tutorial

Fereshteh Mamaghani(fmamagh@calstatela.edu)

Adrian Marroquin(amarro15@calstatela.edu)

Siying Chen(schen112@calstatela.edu)

Jinhui Liu (jliu2@calstatela.edu)

Aleksander Sekowski(asekows@calstatela.edu)

12/08/2020

Pain Pills Data Analysis using Hadoop HDFS and MapReduce on Oracle cloud

---

Objectives

- **Objective 1: Top 15 product market share and Top 10 product market share by every year**
- **Objective 2 : Sales volume by state over time**
- **Objective 3: 10 top distributors by state and by city**
- **Objective 4 : ETL for ARCOS buyers of pain pills' Components and their transactions in USA**

Platform Specifications

- Oracle BDCE
- Cluster Version:20.3.3-20
- CPU Speed: 2195.287 MHz
- # of CPU cores: 12
- # of nodes: 3
- Total Memory Size: 180GB
- Storage: 960 GB

Objective 1: Top 15 product market share and Top 10 product market share by every year

Objectives 1

In this part of the lab, you will learn how to:

- Create tables in Hive/Beeline
- Rename and move files in HDFS, get the files to Oracle Cloud and download the data to local PC
- Visualize the data in Excel and Power BI

Step 1: Create Tables in Hive/Beeline

1. Connect to Oracle Cloud:

```
ssh jliu2@129.150.69.91
```

2. Create a folder called Painpill:

```
hdfs dfs -mkdir Painpill
```

Note: The data has already been upload to the HDFS, therefore, in this Objective, additional copies of the dataset in HDFS can be generated by following code:

```
hdfs dfs -cp /user/fmamagh2/Group5/arcos_all_washpost1.tsv
/user/jliu2/Painpill
```

For more information about how to get the data and upload it to the cloud, please refer to the Objectives 4.

3. Create folders in Painpill:

```
hdfs dfs -mkdir /user/jliu2/Painpill/top15ms

hdfs dfs -mkdir /user/jliu2/Painpill/productms

hdfs dfs -mkdir /user/jliu2/Painpill/top10date
```

Note: In case for having error message (Error: Error while compiling statement: FAILED: RuntimeException…) in hive/beeline:

```
hdfs dfs -mkdir tmp
```

4. Give the permission to edit files/data:

```
hdfs dfs -chmod -R o+w .
```

5. Connect to beeline/hive and using your database in beeline:

```
beeline

!connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-
2:2181,bigdai-nov-bdcsce-
3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?
tez.queue.name=interactive  bdcsce_admin

create database jliu2;

use jliu2;
```

6. Create main table which will include everything from the data:

```
create external table if not exists painpilldata (

reporter_dea_no string, reporter_bus_act string, reporter_name string,
reporter_addl_co_info string, reporter_address1 string,
reporter_address2 string, reporter_city string, reporter_state string,
reporter_zip string, reporter_county string, buyer_dea_no string,
buyer_bus_act string, buyer_name string, buyer_addl_co_info string,
buyer_address1 string, buyer_address2 string, buyer_city string,
buyer_state string, buyer_zip string, buyer_county string,
transaction_code string, drug_code string, ndc_no string, drug_name
string, quantity int, unit int, action_indicator string, order_form_no
string, correction_no string, strength string, transaction_date
string, calc_base_wt_in_gm float, dosage_unit int, transaction_id
string, product_name string, ingredient_name string, measure string,
```

```
mme_conversion_factor string, combined_labeler_name string,
revised_company_name string, reporter_family string, dos_str int )

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

LOCATION '/user/jliu2/Painpill'

TBLPROPERTIES ("skip.header.line.count"="1");
```

7. Using describe and select from function to check the table is created correct:

```
describe painpilldata;

select * from painpilldata limit 10;
```

8. Create a Market Share by Product table:

```
drop table if exists ms_product;


create table if not exists ms_product

row format delimited fields terminated by '\t'

stored as textfile location '/user/jliu2/Painpill/productms'

as

select product_name, sum(quantity) as quantity

from painpilldata group by product_name order by quantity desc;
```

Note: May also use describe and select from function to check the table.

9. Create a table by Top 15 Products' Market Share:

```
create table if not exists top15msp

row format delimited fields terminated by '\t'

stored as textfile location '/user/jliu2/Painpill/top15ms'

as
```

```
with top15 as (

select product_name, quantity from ms_product order by quantity desc
limit 15)

select * from top15

union all

select "all other" as product_name, sum(quantity) as quantity

from ms_product

where product_name not in (select product_name from top15);
```

10. Create a table include all Products' Market Share with date:

```
create table if not exists omspwd

row format delimited fields terminated by '\t'

stored as textfile location '/user/jliu2/Painpill/productms'

as

select product_name, reverse(substr(reverse(transaction_date),0,4)) as
tdate, quantity

from painpilldata;
```

11. Create a table include all Products' Market Share with date and have a desc order by date and quantity:

```
create table if not exists mspwd

row format delimited fields terminated by '\t'

stored as textfile location '/user/jliu2/Painpill/productms'

as

select product_name, tdate, sum(quantity) as quantity,

from omspwd group by product_name, tdate order by tdate desc, quantity
desc;
```

12. Create a table with Top 10 Products' Market Share Every Year:

```
create table if not exists top10ms_by_year

row format delimited fields terminated by '\t'

stored as textfile location '/user/jliu2/Painpill/top10date'

as

select product_name, tdate, quantity

from(

select product_name, tdate, quantity, row_number() over (partition by
tdate order by quantity desc) as row_num

from mspwd ) t

where row_num < 11;
```

Step 2: Rename and move files in HDFS, get the files to Oracle Cloud and download the data to local PC

1. Open another terminal connect to Oracle Cloud

Check the files and folders in top15ms and top10date folders:

```
hdfs dfs -ls ./Painpill/top15ms

hdfs dfs -ls ./Painpill/top10date
```

2. Rename/Move the files:

```
hdfs dfs -mv ./Painpill/top15ms/1/000000_0
./Painpill/top15ms/1/000000_1

hdfs dfs -mv ./Painpill/top15ms/1/000000_1 ./Painpill/top15ms/000000_1

hdfs dfs -mv ./Painpill/top15ms/2/000000_0 ./Painpill/top15ms/000000_2
```

3. Combine the files as named "top15productmarketshare" and get it to Oracle Cloud. Also get top10date file into Oracle Cloud:

```
hdfs dfs -cat /user/jliu2/Painpill/top15ms/000000_* | hdfs dfs -put -
/user/jliu2/Painpill/top15productmarketshare

hdfs dfs -get /user/jliu2/Painpill/top15productmarketshare
```

```
hdfs dfs -get /user/jliu2/Painpill/top10date/000000_0
```

4. Open a new terminal and Download the files to local pc and name the file as following:

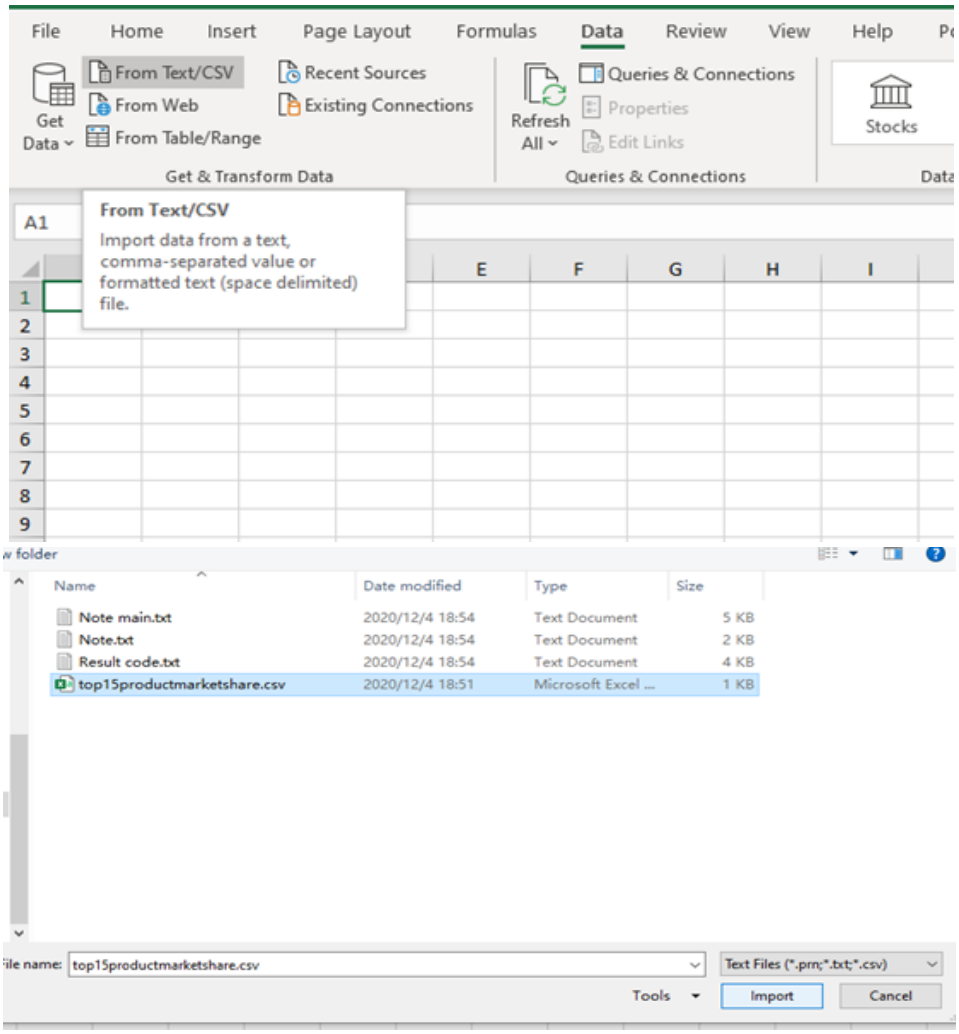
```
scp jliu2@129.150.69.91:/home/jliu2/top15productmarketshare
top15productmarketshare.csv
```

```
scp jliu2@129.150.69.91:/home/jliu2/000000_0 top10productmsbyyear.csv
```

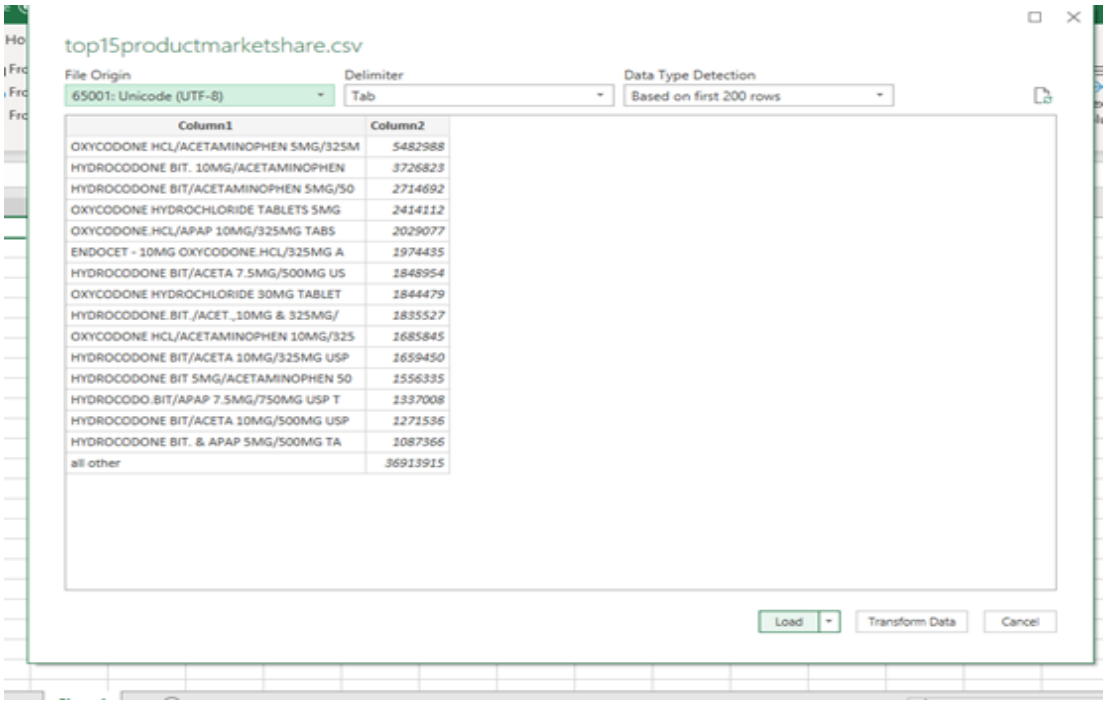
Step 3: Visualize the data in Excel and Power BI

1. Open Excel and load the data/.csv files to excel file as following:

Data-> From Text/CSV -> Import



2. Leave the default setting or you may change the file origin to Unicode (UTF-8) if you want and Load the data:

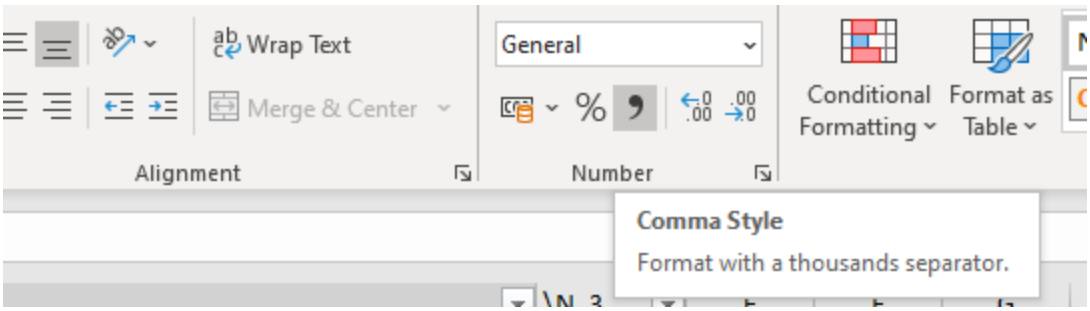


Column1	Column2
OXYCODONE HCL/ACETAMINOPHEN 5MG/325M	5482988
HYDROCODONE BIT. 10MG/ACETAMINOPHEN	3726823
HYDROCODONE BIT/ACETAMINOPHEN 5MG/50	2714692
OXYCODONE HYDROCHLORIDE TABLETS 5MG	2414112
OXYCODONE.HCL/APAP 10MG/325MG TABS	2029077
ENDOCET - 10MG OXYCODONE.HCL/325MG A	1974435
HYDROCODONE BIT/ACETA 7.5MG/500MG US	1848954
OXYCODONE HYDROCHLORIDE 30MG TABLET	1844479
HYDROCODONE.BIT./ACET.,10MG & 325MG/	1835527
OXYCODONE HCL/ACETAMINOPHEN 10MG/325	1685845
HYDROCODONE BIT/ACETA 10MG/325MG USP	1659450
HYDROCODONE BIT 5MG/ACETAMINOPHEN 50	1556335
HYDROCODO.BIT/APAP 7.5MG/750MG USP T	1337008
HYDROCODONE BIT/ACETA 10MG/500MG USP	1271536
HYDROCODONE BIT. & APAP 5MG/500MG TA	1087366
all other	36913915

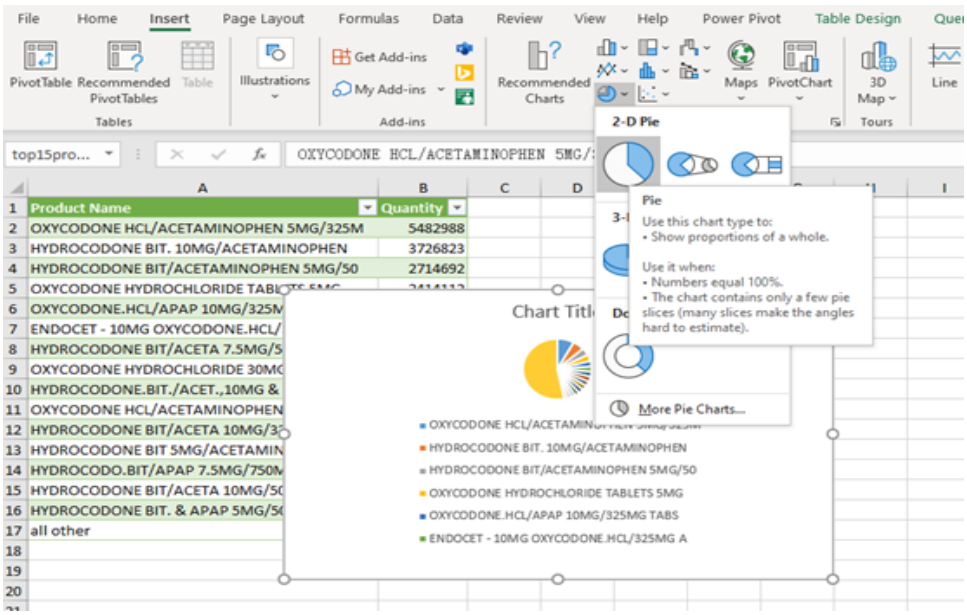
3. Changing Column1 and Column2 to “Product Name” and “Quantity”:

	A	B	C
1	Product Name	Quantity	
2	OXYCODONE HCL/ACETAMINOPHEN 5MG/325M	5482988	
3	HYDROCODONE BIT. 10MG/ACETAMINOPHEN	3726823	
4	HYDROCODONE BIT/ACETAMINOPHEN 5MG/50	2714692	
5	OXYCODONE HYDROCHLORIDE TABLETS 5MG	2414112	
6	OXYCODONE.HCL/APAP 10MG/325MG TABS	2029077	
7	ENDOCET - 10MG OXYCODONE.HCL/325MG A	1974435	
8	HYDROCODONE BIT/ACETA 7.5MG/500MG US	1848954	
9	OXYCODONE HYDROCHLORIDE 30MG TABLET	1844479	
10	HYDROCODONE.BIT./ACET.,10MG & 325MG/	1835527	
11	OXYCODONE HCL/ACETAMINOPHEN 10MG/325	1685845	
12	HYDROCODONE BIT/ACETA 10MG/325MG USP	1659450	
13	HYDROCODONE BIT 5MG/ACETAMINOPHEN 50	1556335	
14	HYDROCODO.BIT/APAP 7.5MG/750MG USP T	1337008	
15	HYDROCODONE BIT/ACETA 10MG/500MG USP	1271536	
16	HYDROCODONE BIT. & APAP 5MG/500MG TA	1087366	
17	all other	36913915	
18			
19			

4. Select B Column and click comma style in Number Section and adjust the value by clicking decrease decimal:

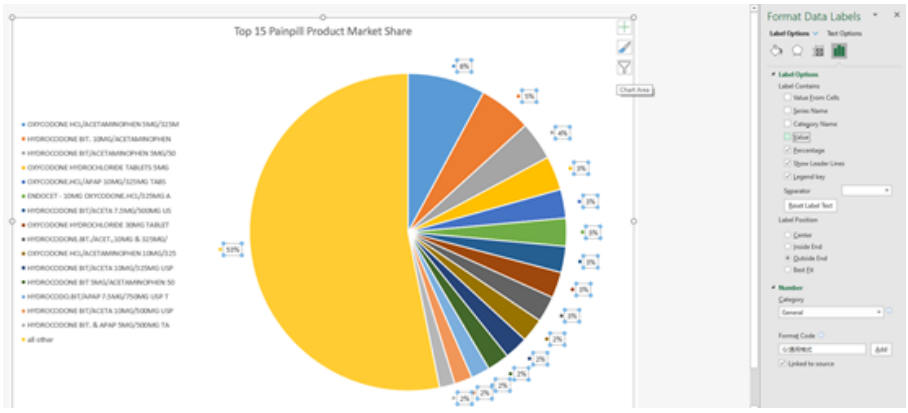


5. Select the cell from A2 to B17 and click Insert -> Charts -> 2-D Pie -> Pie:

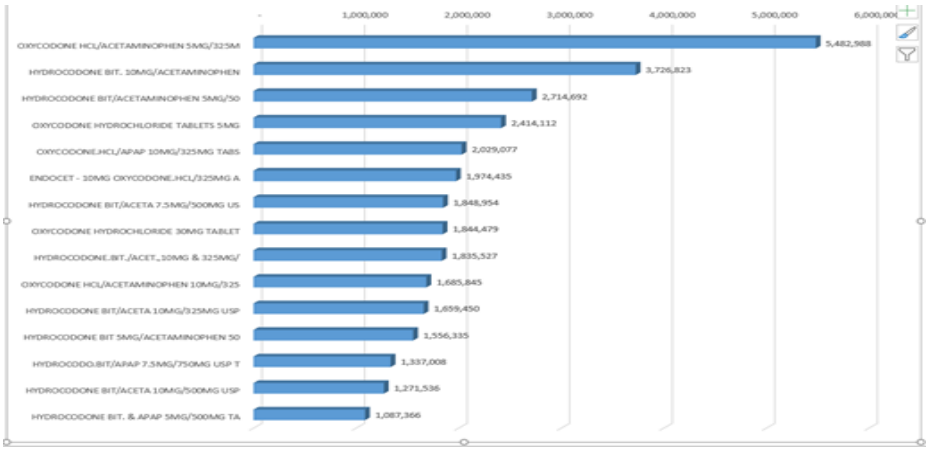


6. Right Click the Chart area and select Move Chart -> New sheet

And Now you can change with the Chart Design and Format option:



7. Repeat the Steps to create a 3-D Bar Chart(do not include the "All other" data) and the final result should be like this:



8. Repeat the Steps to import the top10 product market share csv file in to Excel and adjust the value in C(Quantity) column and rename the columns as following: Product Name, Date, Quantity

	A	B	C
1	Product Name	Date	Quantity
2	OXYCODONE HCL/ACETAMINOPHEN 5MG/325M	2006	624,786
3	HYDROCODONE BIT. 10MG/ACETAMINOPHEN	2006	458,247
4	HYDROCODONE BIT/ACETAMINOPHEN 5MG/50	2006	374,968
5	HYDROCODONE BIT/ACETA 7.5MG/500MG US	2006	230,317
6	OXYCODONE HYDROCHLORIDE TABLETS 5MG	2006	224,941
7	HYDROCODONE BIT 5MG/ACETAMINOPHEN 50	2006	198,411
8	HYDROCODO.BIT/APAP 7.5MG/750MG USP T	2006	191,940
9	HYDROCODONE BIT/ACETA 10MG/325MG USP	2006	177,985

9. Select D2 cell and type following code (it should auto fill in all the rest cells in D column) and change the column name as Full Date:

```
=DATE([@Date],12,31)
```

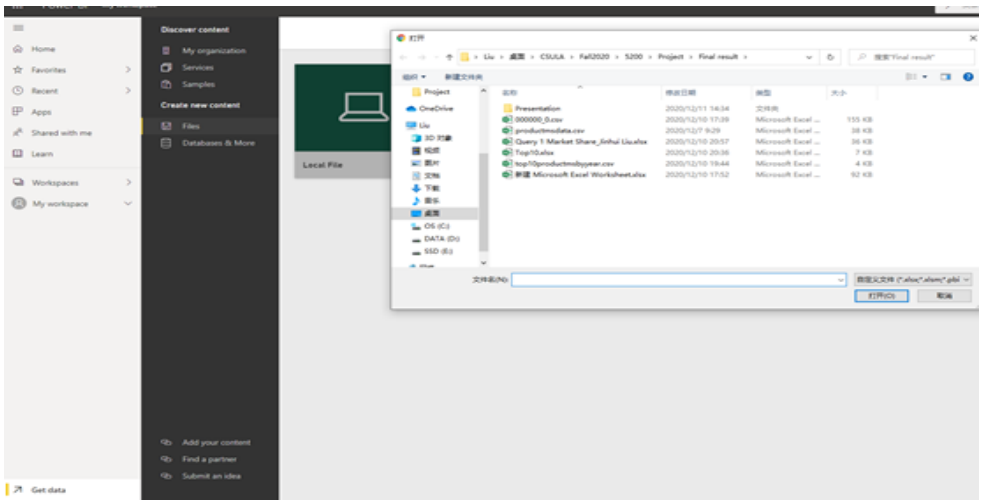
The final result should be like this:

D2 =DATE([@Date],12,31)

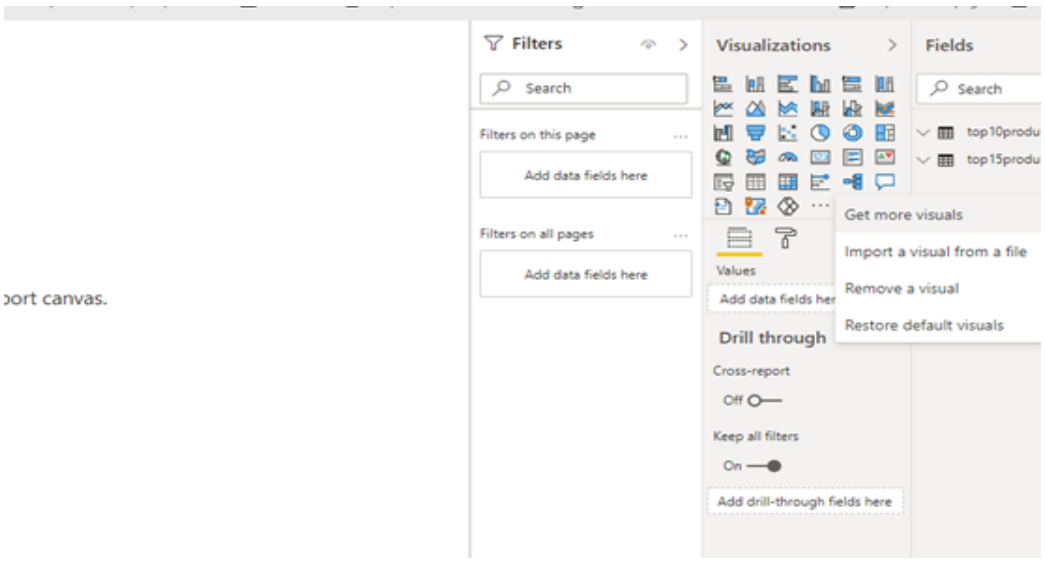
	A	B	C	D
1	Product Name	Date	Quantity	Full Date
2	OXYCODONE HCL/ACETAMINOPHEN 5MG/325M	2006	624,786	2006/12/31
3	HYDROCODONE BIT. 10MG/ACETAMINOPHEN	2006	458,247	2006/12/31
4	HYDROCODONE BIT/ACETAMINOPHEN 5MG/50	2006	374,968	2006/12/31
5	HYDROCODONE BIT/ACETA 7.5MG/500MG US	2006	230,317	2006/12/31
6	OXYCODONE HYDROCHLORIDE TABLETS 5MG	2006	224,941	2006/12/31
7	HYDROCODONE BIT 5MG/ACETAMINOPHEN 50	2006	198,411	2006/12/31
8	HYDROCODO.BIT/APAP 7.5MG/750MG USP T	2006	191,940	2006/12/31
9	HYDROCODONE BIT/ACETA 10MG/325MG USP	2006	177,985	2006/12/31
10	ENDOCET - 10MG OXYCODONE.HCL/325MG A	2006	153,323	2006/12/31
11	HYDROCODONE BIT/ACETA 10MG/500MG USP	2006	149,976	2006/12/31
12	OXYCODONE HCL/ACETAMINOPHEN 5MG/325M	2007	756,178	2007/12/31
13	HYDROCODONE BIT. 10MG/ACETAMINOPHEN	2007	500,159	2007/12/31
14	HYDROCODONE BIT/ACETAMINOPHEN 5MG/50	2007	372,968	2007/12/31

10. Now save the Excel file(you may also want to rename the sheets at the bottom) and Open PowerBI (https://powerbi.microsoft.com/en-us/) and sign in PowerBI by using calstate email address

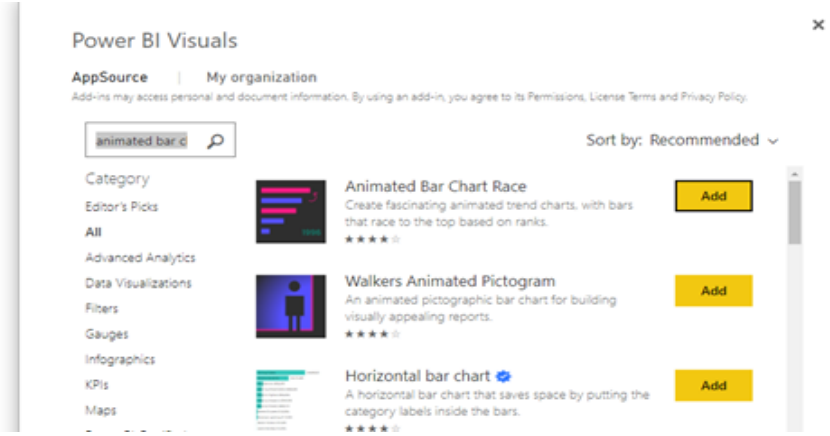
Click “Get Data” at the left bottom corner to import the excel file we just created:



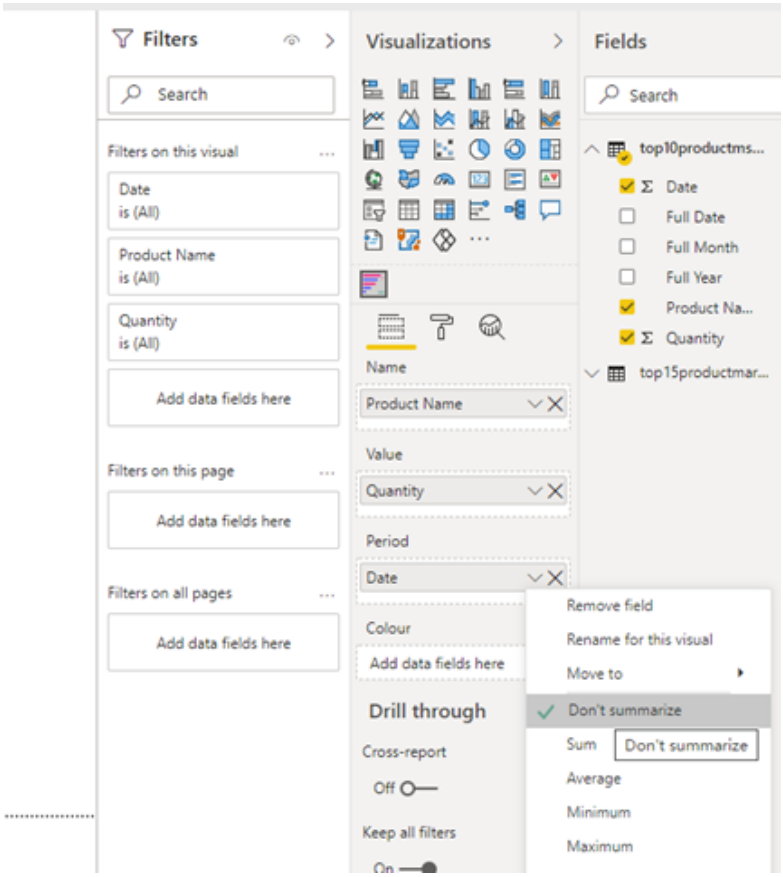
11. Click Query 1 Market Share.xlsx or the name you give to the file in “My workspace” with type-Dashboard and click “Get more visuals” in Visualizations section:



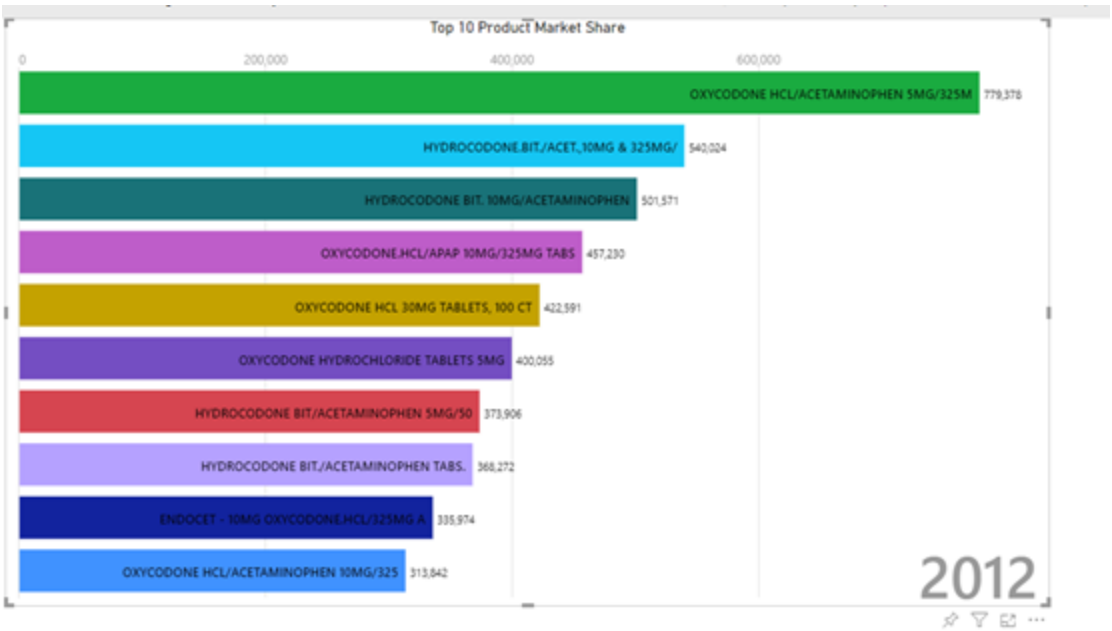
12. Search “animated bar chart race” and click Add:



13. Click "Animated Bar Chart Race" and on the Fields section select Date, Product Name and Quantity under top10productmsbyyear (the data sheet with top 10 product data in excel). Put/Drug "Product Name" into Name section, "Quantity" in Value and "Date" in Period in Visualizations area. And change all of them to "Don't summarize":



Now you will get the final result which is an animated Bar Chart (Note: you can play the animation by double click the chart and you may also change the title or else in the format section.)



Objective 2: Sales volume by state over time

open git bash 1.

ssh asekows@129.150.69.91

hdfs dfs -mkdir tmp/drugs

hdfs dfs -chmod -R o+w tmp/

open git bash 2.

ssh asekows@129.150.69.91

beeline

!connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce_admin

use asekows;

CREATE TABLE drugs_year AS SELECT reporter_state, drug_name, quantity, REVERSE(SUBSTR(REVERSE(transaction_date), 0, 4)) as tdate FROM arcos_all_washpost1;

CREATE TABLE IF NOT EXISTS drugs_state_download ROW FORMAT DELIMITED FIELDS TERMINATED BY "," STORED AS TEXTFILE LOCATION "/user/asekows/tmp/drugs" AS SELECT reporter_state, drug_name, tdate, SUM (quantity) FROM drugs_year GROUP BY reporter_state, drug_name, tdate;

in git bash 1:

hdfs dfs -get /user/asekows/tmp/drugs/00000*_0

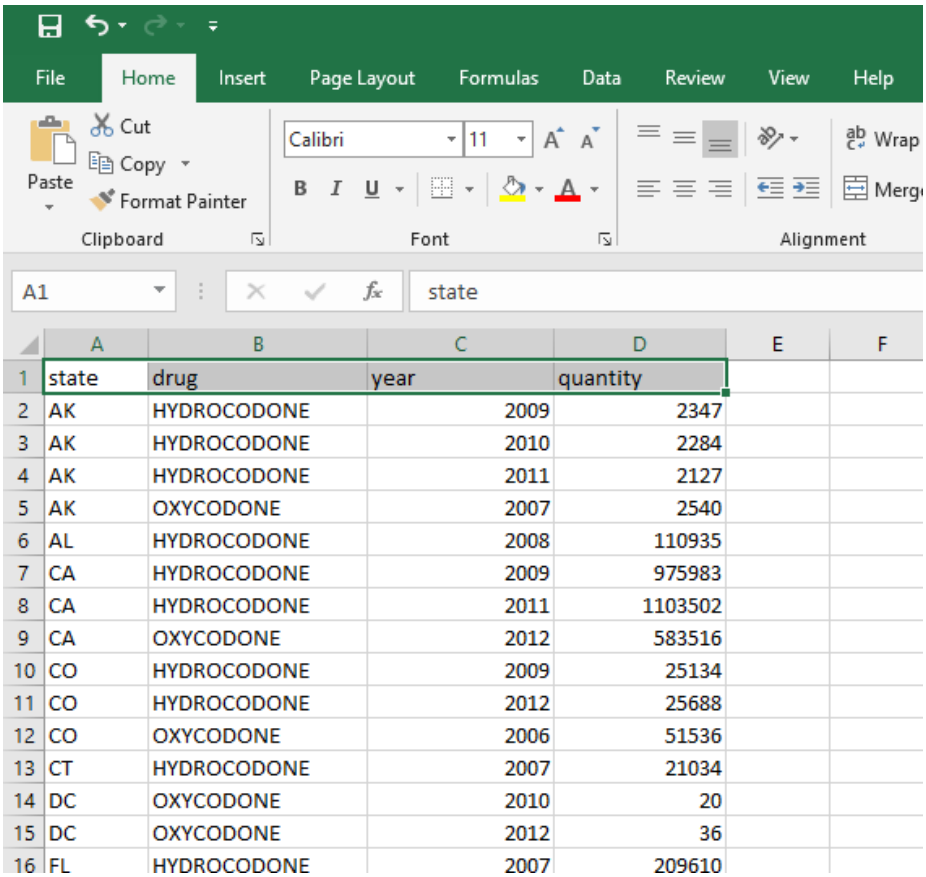
cat 000000_0 000001_0 000002_0 000003_0 000004_0 000005_0 000006_0 000007_0 > drugs_out.csv

open git bash 3:

scp asekows@129.150.69.91:/home/asekows/drugs_out.csv drugs_out.csv

Open downloaded file in Excel

Add column headings

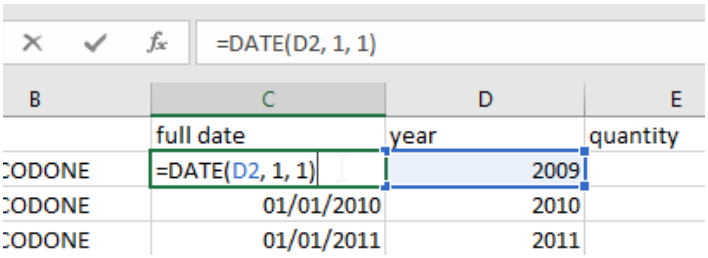


	A	B	C	D	E	F
1	state	drug	year	quantity		
2	AK	HYDROCODONE	2009	2347		
3	AK	HYDROCODONE	2010	2284		
4	AK	HYDROCODONE	2011	2127		
5	AK	OXYCODONE	2007	2540		
6	AL	HYDROCODONE	2008	110935		
7	CA	HYDROCODONE	2009	975983		
8	CA	HYDROCODONE	2011	1103502		
9	CA	OXYCODONE	2012	583516		
10	CO	HYDROCODONE	2009	25134		
11	CO	HYDROCODONE	2012	25688		
12	CO	OXYCODONE	2006	51536		
13	CT	HYDROCODONE	2007	21034		
14	DC	OXYCODONE	2010	20		
15	DC	OXYCODONE	2012	36		
16	FL	HYDROCODONE	2007	209610		

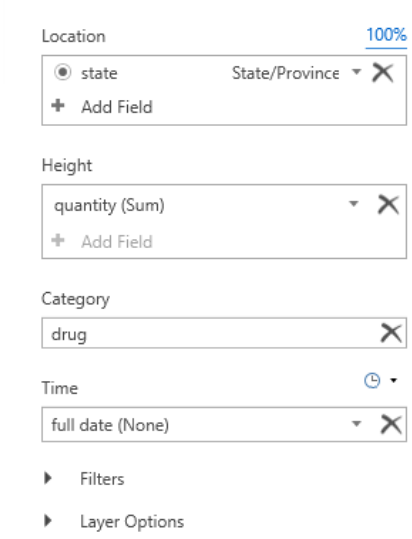
Save as xlsx

Add a new column “full date”. We need to create a formatted date column that we can use for timeline.

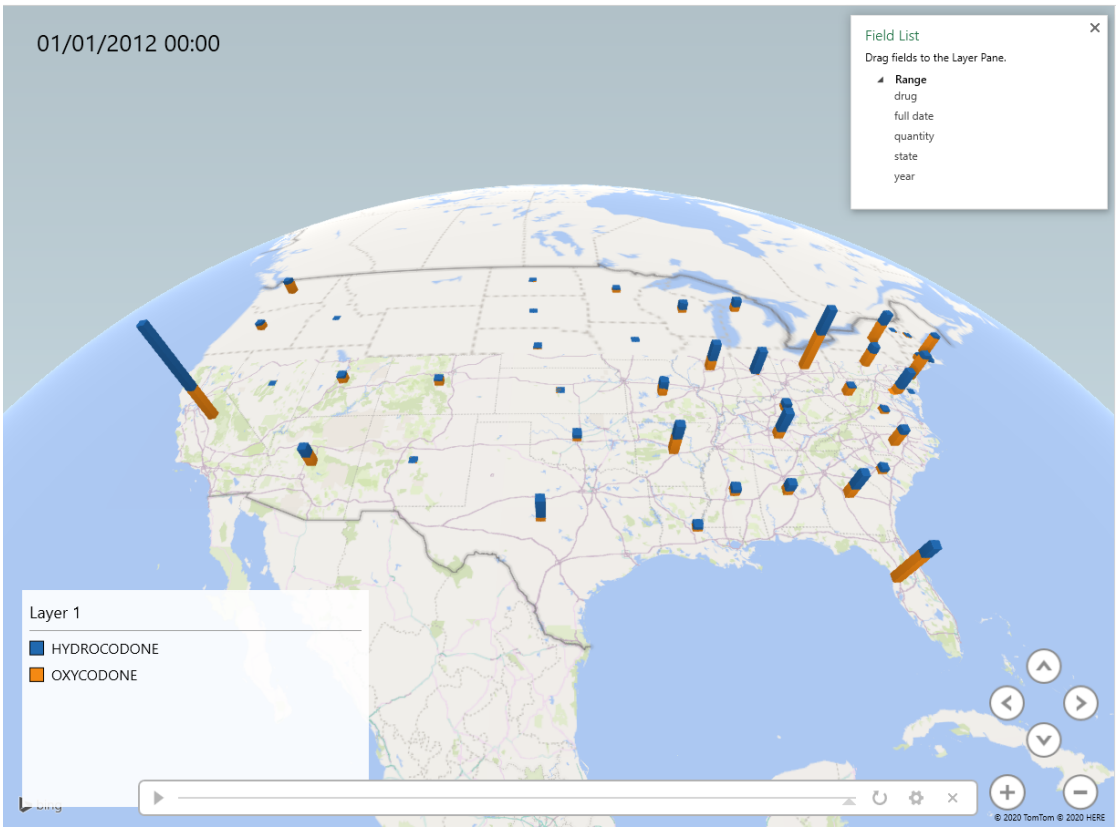
Use following formula for the “full date” column:



Create 3d Map and use:


Location
100%
state
State/Province
Add Field
Height
quantity (Sum)
Add Field
Category
drug
Time
full date (None)
Filters
Layer Options

Result


01/01/2012 00:00
Field List
Drag fields to the Layer Pane.
Range
drug
full date
quantity
state
year
Layer 1
HYDROCODONE
OXYCODONE
© 2020 TomTom © 2020 HERE

Objective 3: 10 top Cities with highest distribution by state

The goal is to query the top 10 cities that had the highest distribution by state and by drug. For the purposes of this lab, only the state of California and drug Oxycodone will be selected for visualization.

Step 1: Connect to BDCE

The data being used is a 11.43 GB sample file from a larger file. The first step is to connect to Oracle BDCE.

ssh amarro15@129.150.69.91

Step 2: Get Dataset

The data being used is already uploaded to teammate directory. The next step is to copy the file to personal user directory. Additional steps are given to check if code was successful.

hdfs dfs -mkdir /practice

hdfs dfs -cp /user/fmamagh2/Group5/arcos_all_washpost1.tsv /user/amarro15/practice/

hdfs dfs -ls /user/amarro15/practice

hdfs dfs -cat /user/amarro15/practice/arcos_all_washpost1.tsv | head -n 2

Step 3: Loading Data With PIG

Pig will be used to make our large file into smaller files to make state-specific visualizations. This step is to enter the Pig grunt shell environment and create a new relation and schema.

```
$ pig
```

```
data = LOAD '/user/amarro15/practice/arcos_all_washpost1.tsv' AS
(reporter_dea_no:chararray, reporter_bus_act:chararray,
reporter_name:chararray, reporter_addl_co_info:chararray,
reporter_address1:chararray, reporter_address2:chararray,
reporter_city:chararray, reporter_state:chararray,
reporter_zip:chararray, reporter_county:chararray,
buyer_dea_no:chararray, buyer_bus_act:chararray,
buyer_name:chararray, buyer_addl_co_info:chararray,
buyer_address1:chararray, buyer_address2:chararray,
buyer_city:chararray, buyer_state:chararray, buyer_zip:chararray,
buyer_county:chararray, transaction_code:chararray,
drug_code:chararray, ndc_no:chararray, drug_name:chararray,
```

```
quantity:int, unit:int, action_indicator:chararray,
order_form_no:chararray, correction_no:chararray, strength:int,
transaction_date:chararray, calc_base_wt_in_gm:double,
dosage_unit:int, transaction_id:chararray,
product_name:chararray, ingredient_name:chararray,
measure:chararray, mme_conversion_factor:int,
combined_labeler_name:chararray, revised_company_name:chararray,
reporter_family:chararray, dos_str:float);
```

Step 4: Filtering Data With PIG

The file is now ready to be filtered to specific data. The following code will create top distributors by city/state. There is the option to choose which state or drug name that the table highlighted in red.

```
drug_data = FILTER data BY drug_name == 'OXYCODONE';

california_subset = FILTER drug_data BY buyer_state == 'CA' AND
reporter_bus_act == 'DISTRIBUTOR';

grouped = GROUP California_subset BY reporter_city;

totals  = FOREACH grouped GENERATE group,
SUM(California_subset.quantity) AS city_count;

sorted = ORDER totals BY city_count DESC;

top_ten = LIMIT sorted 10;
```

Step 5: Storing Data With PIG

Last step in Pig is to store the output. DUMP the file to check if it is working, and store the output in a .csv file. Quit pig.

```
DUMP top_ten;

STORE top_ten INTO 'output/top_ten' USING PigStorage(',');

Quit
```

Step 6: Downloading Output File

The pig output will now be moved into local filesystem and downloaded into personal laptop/desktop. Once exiting pig, move and confirm the file is in correct order. Then download file. In order to download, last command must be executed in a new terminal window.

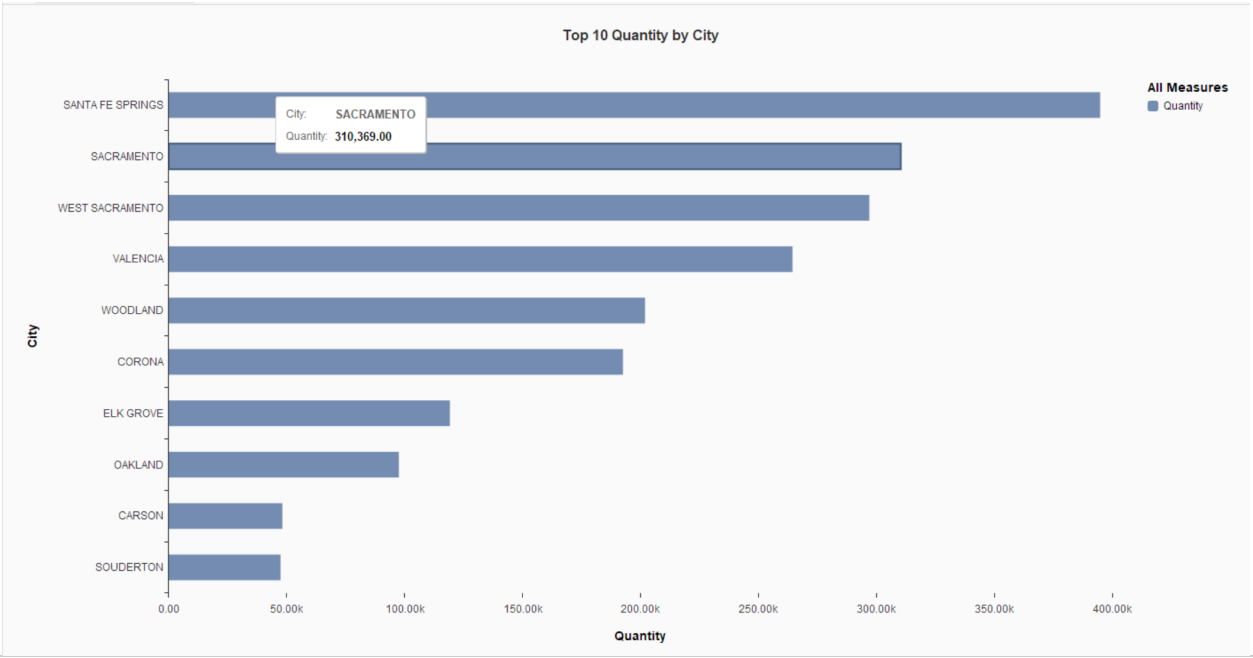
```
hdfs dfs -get output/top_ten/part-r-00000 top_ten.csv
```

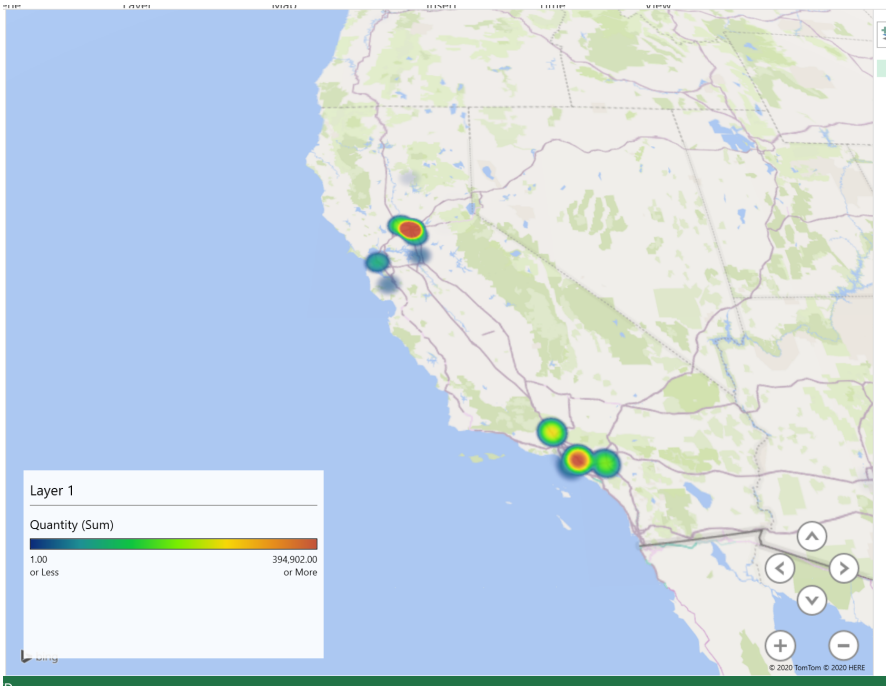
```
cat top_ten.csv | tail -n 2

scp amarro15@129.150.69.91:/home/amarro15/top_ten.csv .
```

Step 7: Visualization

The last step is to upload into visualization software. The first visualization was loaded into SAP Predictive Analytics. The second graph was created by Microsoft Excel Power Maps.





Objective 4: ETL of ARCOS buyers of pain pills' Components and their transactions in the USA

The goal is to demonstrate the details of top 100 buyers of one of the main pain pills components and subsequently, the amounts and transaction dates of top 2 purchases per each top 100 buyers

Data set source URL:

https://www.kaggle.com/paultimothymooney/pain-pills-in-the-usa

Data size : 78GB

Extracting the first 12 GB of the data set to upload it to Oracle cloud:

head -n 25500000 arcos_all_washpost.tsv > arcos_all_washpost1.tsv

scp arcos_all_washpost1.tsv fmamagh2@129.150.69.91:/dev/shm/

Downloading Pain pills dataset from Oracle cloud to Hadoop file system

Connect to Oracle Cloud:

ssh fmamagh2@129.150.69.91

Create a folder in hdfs to store the dataset:

hdfs dfs –mkdir Group5

hdfs dfs –get /dev/shm/arcos_all_washpost1.tsv Group5/

hdfs dfs -ls Group5

Found 1 items
-rw-r--rw- 2 fmamagh2 hdfs 11432128252 2020-12-01 04:25 Group5/arcos_all_washpost1.tsv

Remove pain pills dataset from oracle cloud storage:

rm /dev/shm/arcos_all_washpost1.tsv

ls dev/shm/

Creating Hive Table to Query Pain Pills Data

Give the permission to beeline for edit files/data:

hdfs dfs -chmod -R o+w Group5/

Connect to beeline/Hive:

beeline

!connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce_admin

Change the database to your username:

use fmamagh2;

Create main table which will include everything from the data:

```
CREATE  TABLE if not exists pain_pills (
REPORTER_DEA_NO  STRING ,
REPORTER_BUS_ACT STRING ,
REPORTER_NAME STRING ,
REPORTER_ADDL_CO_INFO STRING ,
REPORTER_ADDRESS1 STRING ,
REPORTER_ADDRESS2 STRING ,
REPORTER_CITY STRING ,
REPORTER_STATE STRING ,
REPORTER_ZIP STRING ,
REPORTER_COUNTY STRING ,
BUYER_DEA_NO STRING ,
BUYER_BUS_ACT STRING ,
BUYER_NAME STRING ,
BUYER_ADDL_CO_INFO STRING ,
BUYER_ADDRESS1 STRING ,
BUYER_ADDRESS2 STRING ,
BUYER_CITY STRING ,
BUYER_STATE STRING ,
BUYER_ZIP STRING ,
BUYER_COUNTY STRING ,
TRANSACTION_CODE STRING ,
DRUG_CODE INT ,
NDC_NO BIGINT ,
DRUG_NAME STRING ,
QUANTITY INT ,
UNIT INT ,
ACTION_INDICATOR INT ,
ORDER_FORM_NO INT ,
CORRECTION_NO INT ,
STRENGTH INT ,
TRANSACTION_DATE STRING ,
```

```
CALC_BASE_WT_IN_GM FLOAT ,
DOSAGE_UNIT INT ,
TRANSACTION_ID INT ,
Product_Name STRING ,
Ingredient_Name STRING ,
Measure STRING ,
MME_Conversion_Factor INT ,
Combined_Labeler_Name STRING ,
Revised_Company_Name STRING ,
Reporter_family STRING ,
dos_str INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION "/user/fmamagh2/Group5"
TBLPROPERTIES ('skip.header.line.count'='1');
```

Note: The yellow highlights are fields that will be selected for further queries in Pig.

Use describe and select from function to check the table is created correctly:

```
describe pain_pills;

select * from pain_pills limit 3;
```

Connecting to pig interactive mode using Grunt Shell and utilizing HCatalog to share data between Hive and Pig

Open a new terminal and ssh to the Hadoop cluster
Connect to pig and load data from hive table:

```
pig -useHCatalog

data = LOAD 'fmamagh2.pain_pills' using org.apache.hive.hcatalog.pig.HCatLoader();

DESCRIBE data;
```

Extract the required columns for ETL processing:

```
Buyers = foreach data generate buyer_dea_no, buyer_bus_act, buyer_name, buyer_address1,
buyer_address2, buyer_city, buyer_state, buyer_zip, buyer_county, drug_code, drug_name, quantity,
transaction_date;

describe Buyers;
```

Buyers: {buyer_dea_no: chararray,buyer_bus_act: chararray,buyer_name: chararray,buyer_address1: chararray,buyer_address2: chararray,buyer_city: chararray,buyer_state: chararray,buyer_zip: chararray,buyer_county: chararray,drug_code: int,drug_name: chararray,quantity: int,transaction_date: chararray}

Categorize the drug names:

```
c = foreach Buyers generate drug_code, drug_name;

Drugname = DISTINCT c;

dump Drugname;
```

(9143,OXYCODONE)
(,DRUG_NAME)
(9193,HYDROCODONE)

Split the Buyers based on their drug category purchase:

```
SPLIT Buyers INTO B_OXY IF drug_name == 'OXYCODONE', B_HYDRO IF drug_name == 'HYDROCODONE' ;
```

(Optional) **Shorten the names of the buyers to the first 10 characters in the names:**

```
boN = foreach B_OXY generate buyer_dea_no, SUBSTRING(buyer_name, 0, 11) AS buyer_name, drug_name;

boName = DISTINCT boN;
```

Extracting the top 100 Buyers of Oxycodone

```
bo1 = foreach B_OXY generate buyer_dea_no, drug_name, quantity;

bo2 = group bo1 by buyer_dea_no;

describe bo2;

bo_total = foreach bo2 generate group, SUM(bo1.quantity) As total;

bo_sort = ORDER bo_total BY total desc;

bo100 = limit bo_sort 100;

-- The reason for this join is to add shortened buyer_name
oxy_total1 = JOIN bo100 BY group, boName BY buyer_dea_no;
oxy_total2 = foreach oxy_total1 generate boName::buyer_dea_no AS buyer_dea_no ,
boName::buyer_name AS buyer_name , boName::drug_name AS drug_name , bo100::total AS total;
oxy_total100 = ORDER oxy_total2 BY total DESC;
```

Store the results into HDSF:

```
STORE oxy_total100 INTO 'Buyers/oxytotal';
```

List of top 2 purchases per each buyer in top 100

```
Buyer_oxy1 = group B_OXY BY buyer_dea_no;

Buyer_oxy2 = foreach Buyer_oxy1{

        sorted = ORDER B_OXY BY quantity DESC;

        high_qty = limit sorted 2;

        generate group, FLATTEN(high_qty);};

Buyer_oxy3 = JOIN oxy_total100 BY buyer_dea_no , Buyer_oxy2 BY group;

Buyers_Detail1 = foreach Buyer_oxy3 generate oxy_total100::buyer_dea_no AS buyer_dea_no,
Buyer_oxy2::high_qty::buyer_bus_act AS buyer_bus_act, oxy_total100::buyer_name AS buyer_name,
Buyer_oxy2::high_qty::buyer_name AS buyer_name_dtl, Buyer_oxy2::high_qty::buyer_address1 AS
buyer_address1,Buyer_oxy2::high_qty::buyer_address2 As buyer_address2,
Buyer_oxy2::high_qty::buyer_city AS buyer_city,Buyer_oxy2::high_qty::buyer_state AS
buyer_state,Buyer_oxy2::high_qty::buyer_zip As buyer_zip ,Buyer_oxy2::high_qty::buyer_county AS
buyer_county, Buyer_oxy2::high_qty::drug_name AS drug_name ,Buyer_oxy2::high_qty::quantity AS
quantity, Buyer_oxy2::high_qty::transaction_date AS Date;

Describe Buyers_Detail1;
```

Prepare the date format for 3D map in excel

```
Buyers_Detail = foreach Buyers_Detail1 generate buyer_dea_no, buyer_bus_act, buyer_name,
buyer_name_dtl, buyer_address1, buyer_address2, buyer_city, buyer_state, buyer_zip, buyer_county,
drug_name, quantity, CONCAT(SUBSTRING(Date,0,2),'/',SUBSTRING(Date,2,4),'/',SUBSTRING(Date,4,9))
AS Date;

describe Buyers_Detail;
```

Store the results into HDSF:

```
STORE Buyers_Detail INTO 'Buyers/oxydetail';
```

(Optional) change the name of the stored files to prevent overwriting the file in get process

```
hdfs dfs -mv Buyers/oxydetail/part-r-00000 Buyers/oxydetail/oxydetail

hdfs dfs -mv Buyers/oxytotal/part-r-00000 Buyers/oxytotal/oxytotal
```

Download the data into your PC:

hdfs dfs -get Buyers/oxytotal/oxytotal

hdfs dfs -get Buyers/oxydetail/oxydetail

scp fmamagh2@129.150.69.91:/home/fmamagh2/oxytotal oxytotal.tsv

scp fmamagh2@129.150.69.91:/home/fmamagh2/oxydetail oxydetail.tsv

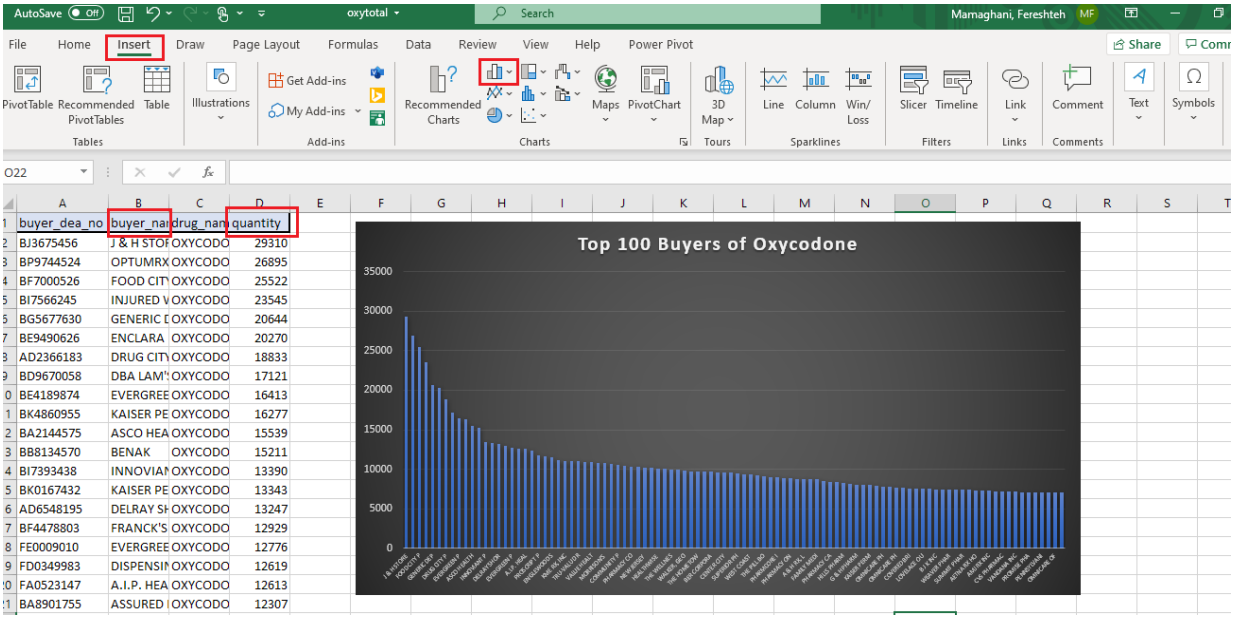
Loading Data into and Visualizing using Power Map in Excel

Open the oxydetail and oxytotal files in excel and separate the columns as Tab delimited

Add the header names to columns of each file oxy detail and oxytotal based on Buyers_Detail and oxy_total100 headers, repectively.

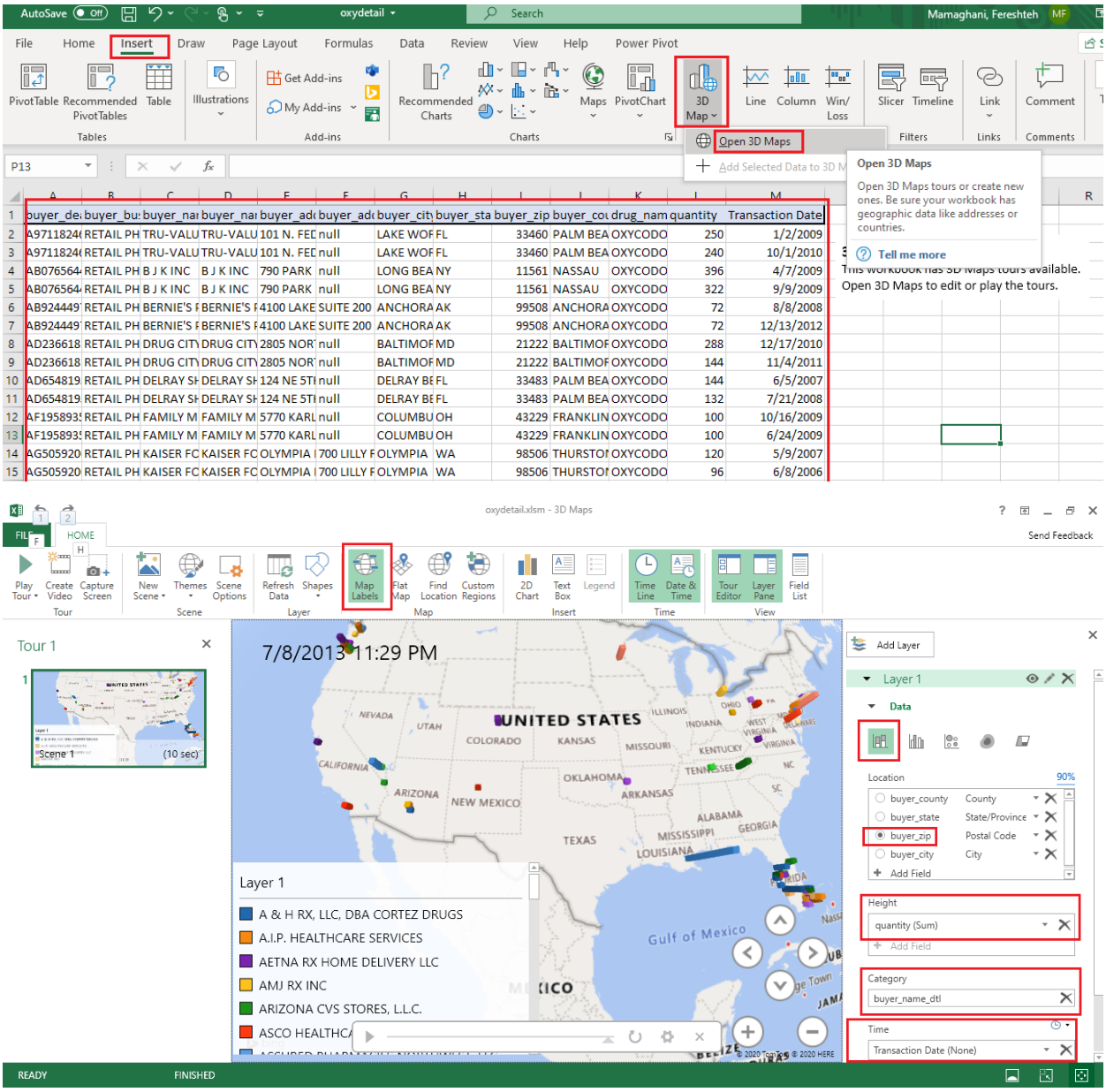
Use the Pig Latins describe Buyers_Detail; and describe oxy_total100; for the header names.

2-D Bar chart for top 100 Buyers:



buyer_dea_no	buyer_nam	drug_nam	quantity
BJ3675456	J & H STOI	OXYCODO	29310
BP9744524	OPTUMRX	OXYCODO	26895
BF7000526	FOOD CIT	OXYCODO	25522
BI7566245	INJURED V	OXYCODO	23545
BG5677630	GENERIC I	OXYCODO	20644
BE9490626	ENCLARA	OXYCODO	20270
AD2366183	DRUG CIT	OXYCODO	18833
BD9670058	DBA LAM	OXYCODO	17121
BE4189874	EVERGREE	OXYCODO	16413
BK4860955	KAISER PE	OXYCODO	16277
BA2144575	ASCO HEA	OXYCODO	15539
BB8134570	BENAK	OXYCODO	15211
BI7393438	INNOVIAN	OXYCODO	13390
BK0167432	KAISER PE	OXYCODO	13343
AD6548195	DELRAY SH	OXYCODO	13247
BF4478803	FRANCK'S	OXYCODO	12929
FE0009010	EVERGREE	OXYCODO	12776
FD0349983	DISPENSIN	OXYCODO	12619
FA0523147	A.I.P. HEA	OXYCODO	12613
BA8901755	ASSURED	OXYCODO	12307

3-D MAP for Buyers details and the amount of their 2 transactions:



References

1. URL of Data Source: https://www.kaggle.com/paultimothymooney/pain-pills-in-the-usa
2. Github: https://github.com/aleksUIX/CIS5200-team5 URL of References