# [C:\Users\jwoo5\AppData\Local\Temp\templateTermTutorial.html](http://www.calstatela.edu/centers/hipic) CIS5200 Term Project Tutorial https://avatars2.githubusercontent.com/u/4156894?v=3&s=100

California State University, Los Angeles

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

#### Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

#### Date: 12/08/2020

**Lab Tutorial**

Fereshteh Mamaghani([fmamagh@calstatela.edu](mailto:fmamagh@calstatela.edu))

Adrian Marroquin(amarro15[@calstatela.edu](mailto:fmamagh@calstatela.edu))

Siying Chen([schen112@calstatela.edu](mailto:schen112@calstatela.edu))

Jinhui Liu (jliu2@calstatela.edu)

#### Aleksander Sekowski()

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 upload to the HDFS by my teammate, therefore, in this Objective, I will just get the data from my teammate in HDFS 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:**

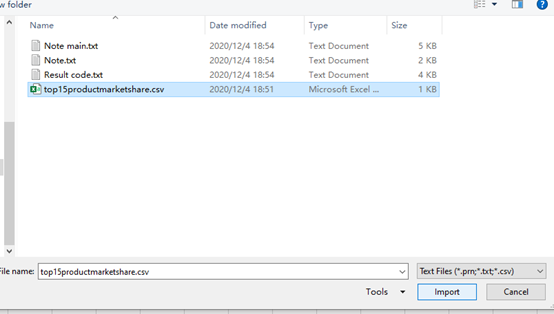
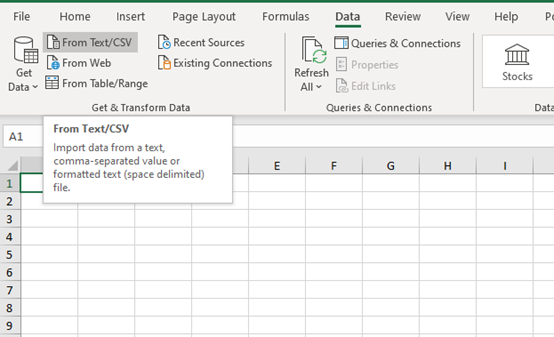
scp jliu2@129.150.69.91:/home/jliu2/top15productmarketshare top15productmarketshare.csv

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

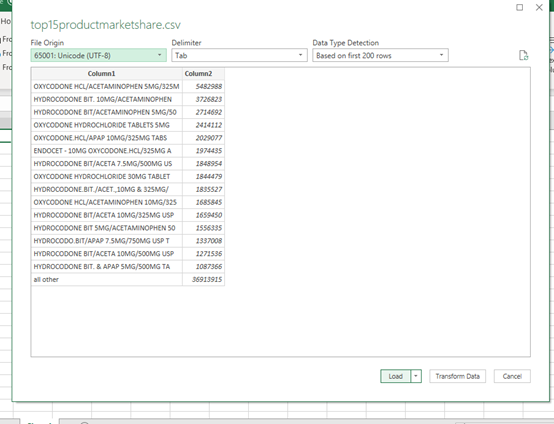
Step 3: Visualize the data in Excel and Power BI

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

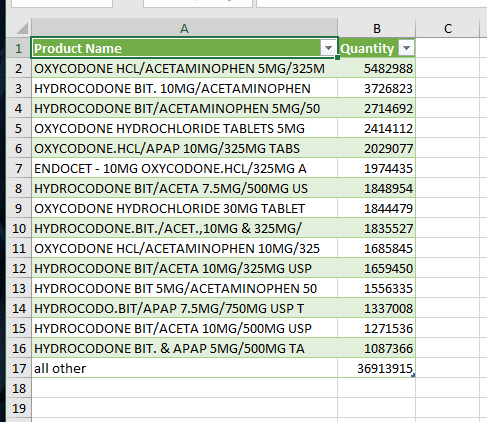
**Data-> From Text/CSV -> Import**

****

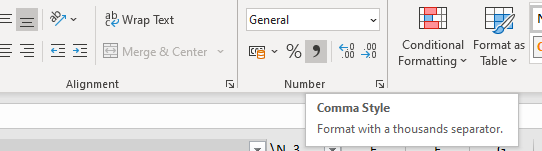
**2. Leave the default setting or you may change the file origin to Unicode (UTF-8) if you want and Load the data:**

****

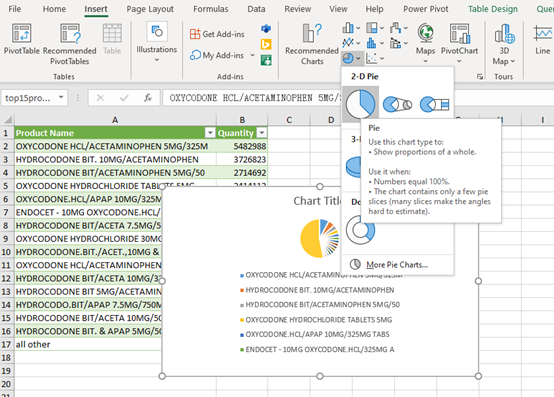
**3. Changing Column1 and Column2 to “Product Name” and “Quantity”:**

****

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

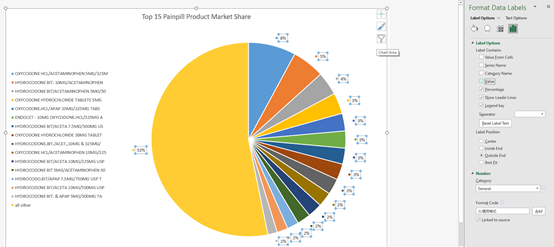
****

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

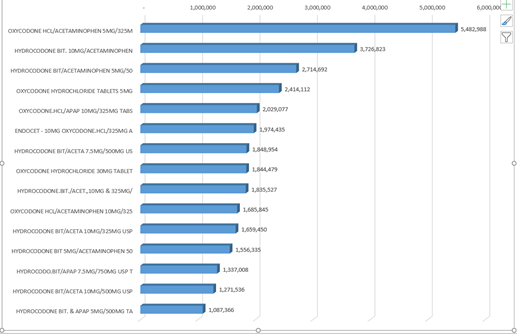
****

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

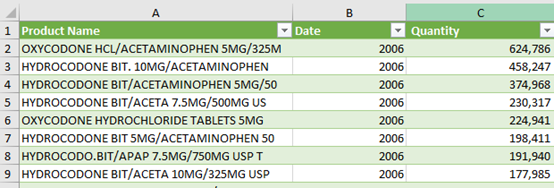
**And Now you can change with the Chart Design and Format option:**

****

**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:**

****

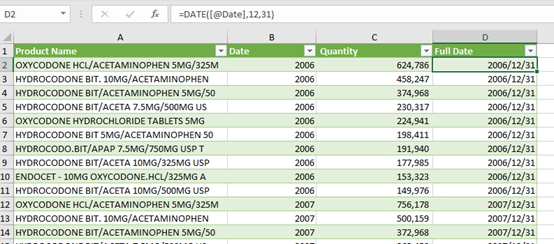
**8. Repeat the Steps to import the top10 product market share csv file in to Excel and adjust the value in C(Quantity) column and rename the columns as following: Product Name, Date, Quantity**

****

**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:**

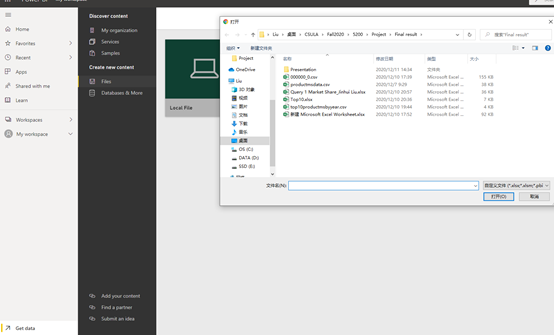
=DATE([@Date],12,31)

**The final result should be like this:**

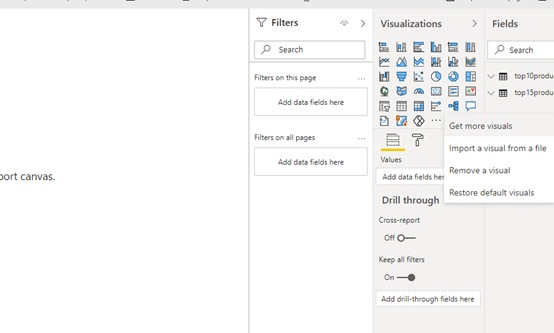


**10. Now save the Excel file(you may also want to rename the sheets at the bottom) and Open PowerBI (https://powerbi.microsoft.com/en-us/) and sign in PowerBI by using calstate email address**

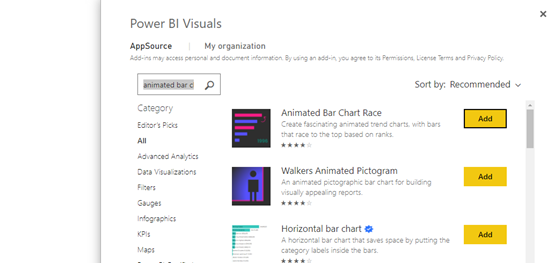
**Click “Get Data” at the left bottom corner to import the excel file we just created:**

****

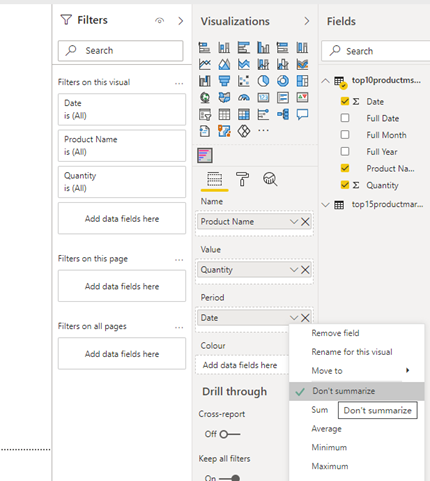
**11. Click Query 1 Market Share.xlsx or the name you give to the file in “My workspace” with type-Dashboard and click “Get more visuals” in Visualizations section:**

****

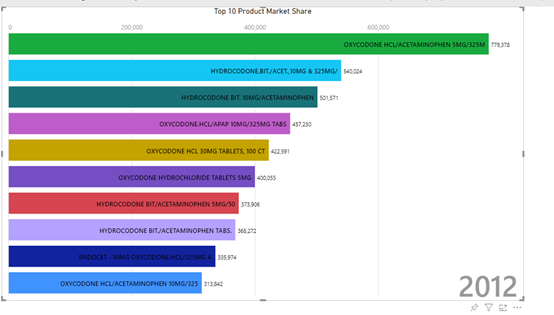
**12. Search “animated bar chart race” and click Add:**

****

**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

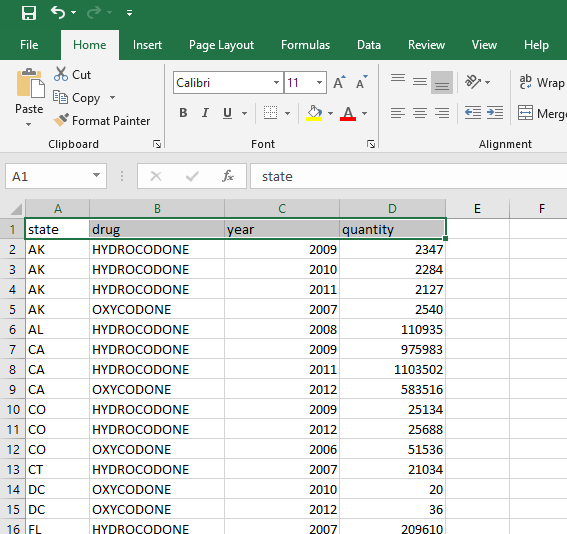
cat 000000\_0 000001\_0 000002\_0 000003\_0 000004\_0 000005\_0 000006\_0 000007\_0 > drugs\_out.csv

**open git bash 3:**

scp [asekows@129.150.69.91:/home/asekows/drugs\_out.csv drugs\_out.csv](about:blank)

**Open downloaded file in Excel**

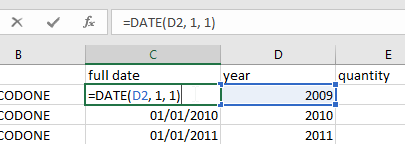
Add column headings



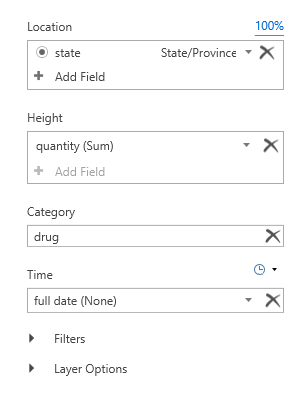
Save as xlsx

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

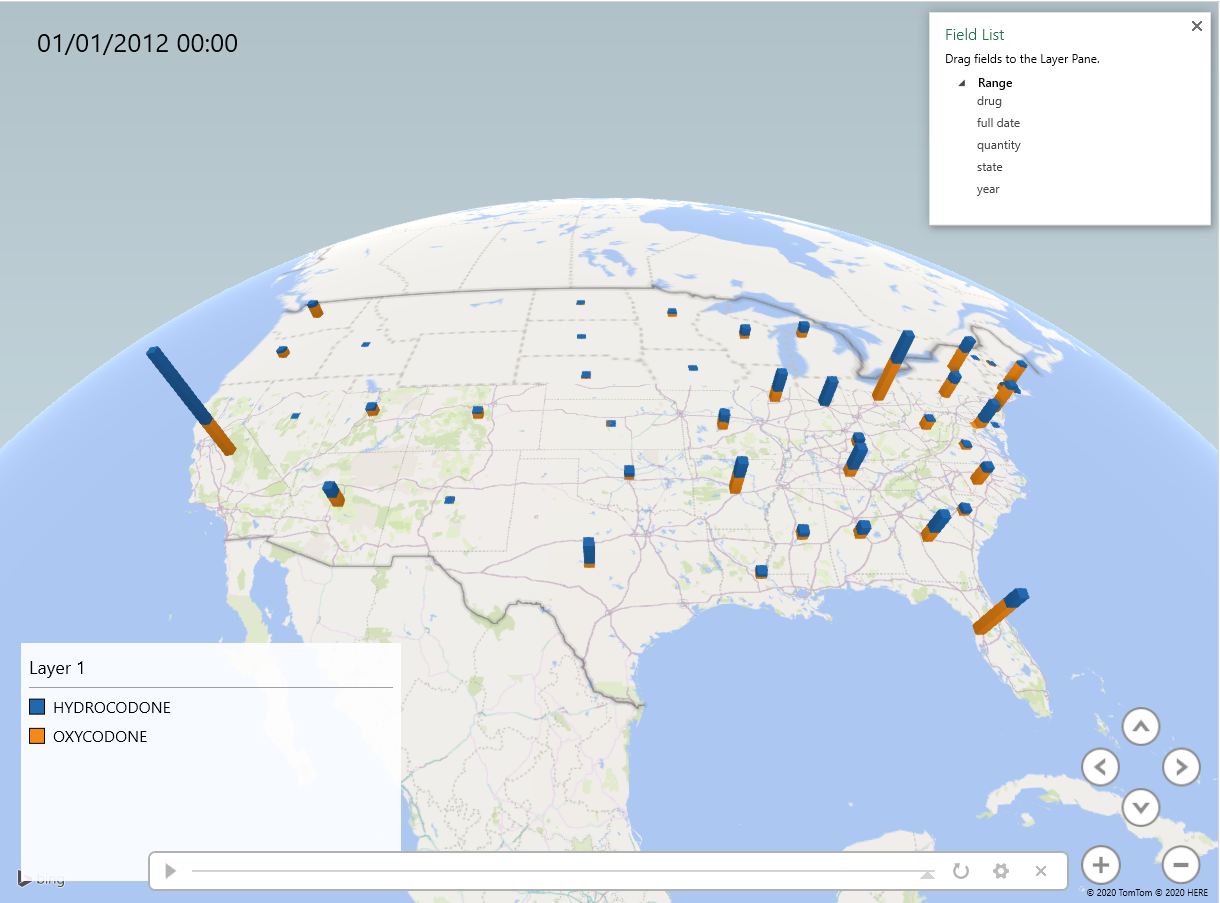
Use following formula for the “full date” column:



Create 3d Map and use:



**Result**



**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

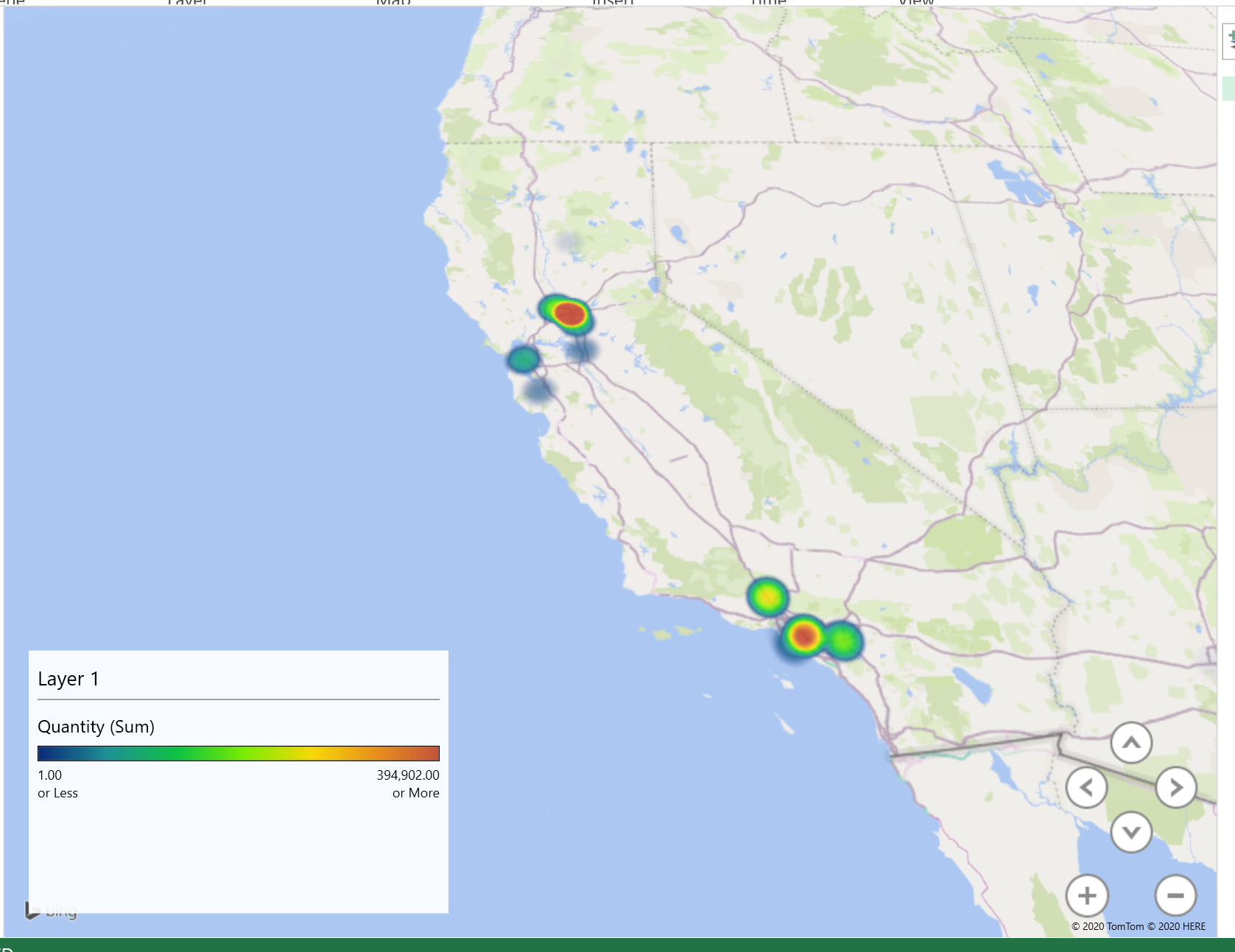
scp amarro15@129.150.69.91:/home/amarro15/top\_ten.csv .

**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.

Chart

Description automatically generated



**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'; legend icon

**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](about:blank)

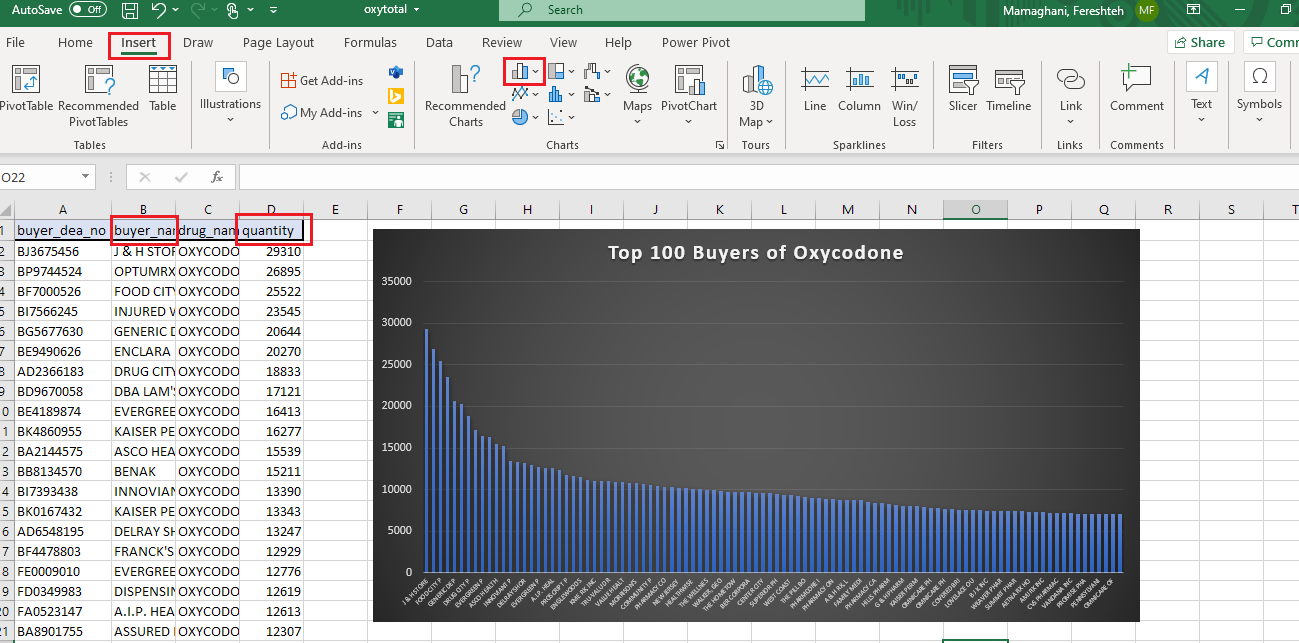
**Loading Data into and Visualizing using Power Map in Excel**

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

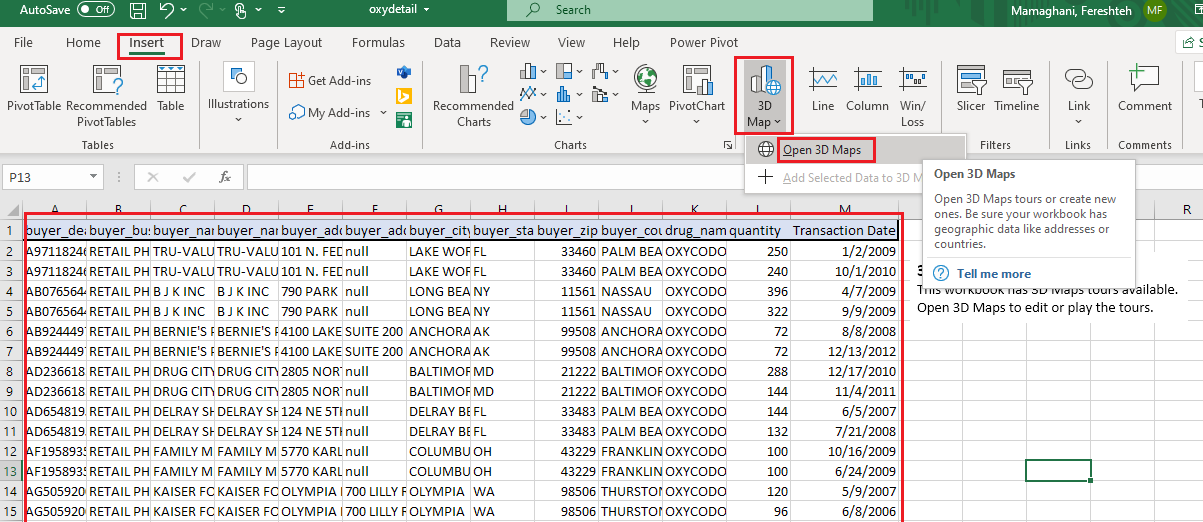
Add the header names to columns of each file oxy detail and oxytotal based on Buyers\_Detail and oxy\_total100 headers, repectively.

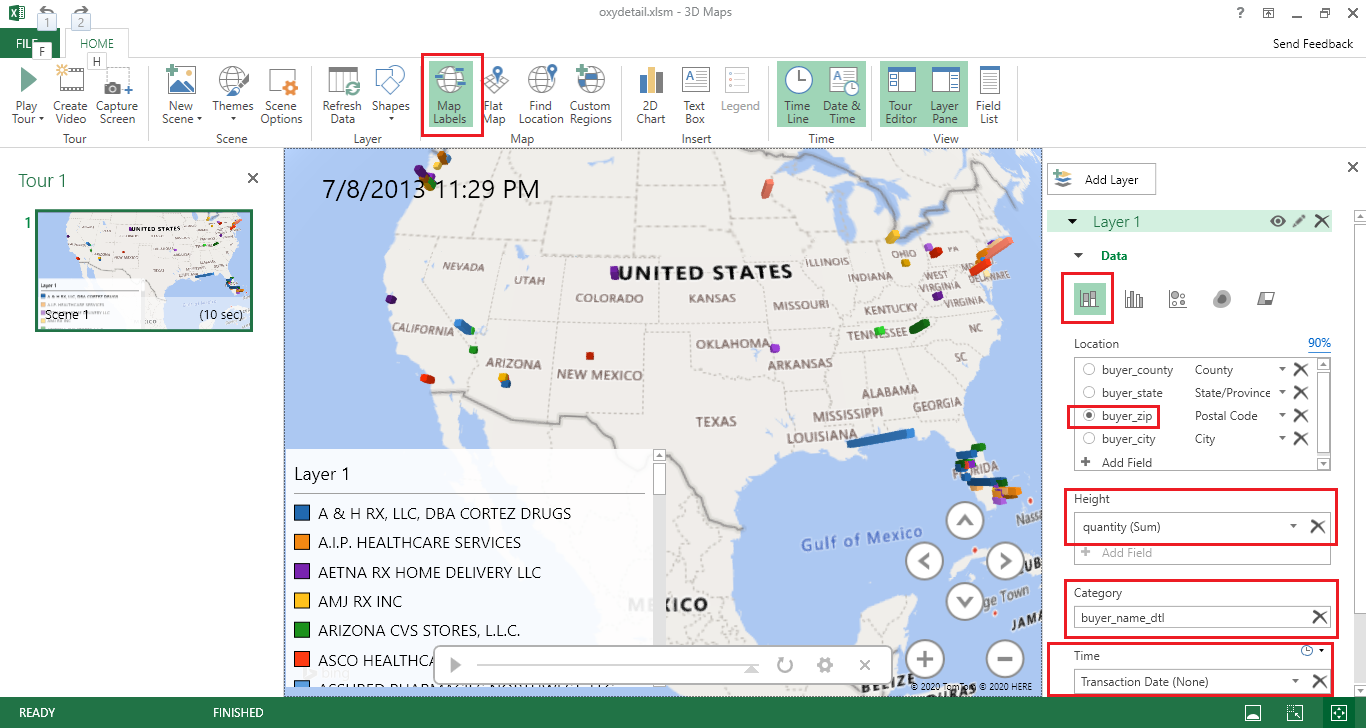
Use the Pig Latins describe Buyers\_Detail; and describe oxy\_total100; for the header names.

**2-D Bar chart for top 100 Buyers:**

****

**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