Business Data Analysis near CalStateLA, USC, UCLA in Los Angeles using Spark

Ram Dharan Donda (rdonda@calstatela.edu), Jongwook woo (jwoo5@calstatela.edu)
HiPIC (http://web.calstatela.edu/centers/hipic/)
July 2016

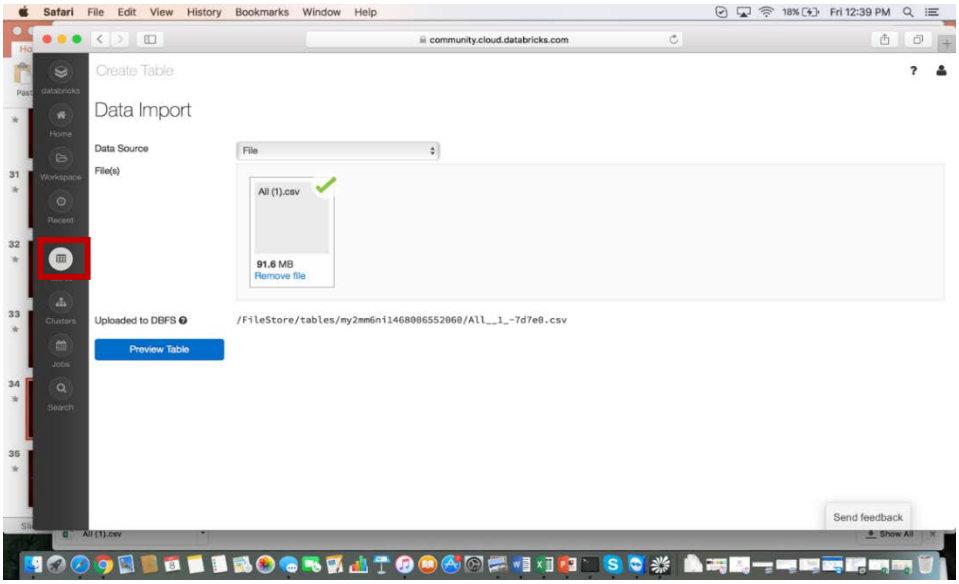
Objectives

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

- Analyze Yelp data set near California State University, USC, UCLA
- Learn how to use Databricks cloud computing service using Community Edition
- Create and upload the data to cluster
- Usage of Spark SQL, especially Data Frames and UDF
- Visualize data using ipython notebook and Excel Power view

Exercise 1: Create and upload the data to cluster

1) Download the data file (yelp_raw_fall_2016.csv) from S3 (https://s3.amazonaws.com/hipicdatasets/yelp_raw_fall_2016.csv) or you can directly load it from ipynb code given by us in the databricks' cloud computing.



2) Create a cluster and then select "Tables" in the left navigation bar and upload file as shown above. **NOTE:** Save the path after uploading, which should be used later.

3) Create a new ipython notebook or import yelp_review.dbc (or yelp_review.ipynb) file.

4) Load the data and print its schema with the following command

```
business_df_yelp=sqlContext.read.format('com.databricks.spark.csv').options(header='true',in
ferschema='true').load('path of the file')

business_df_yelp.printSchema()
```

Exercise 2: Creating and using UDFs

1) We created a function to calculate the distance and see the businesses around CSU LA, USC, UCLA. Lat2 and lon2 below can be modified as per your requirement.

```
import math
from math import sin, cos, sqrt, atan2, radians,asin

# radius in miles
RADIUS = 3965

# distance between (lat1, lon1) and (lat2, lon2)
def distance(lat1lon1, lat2, lon2, radius):
   try:
    # Geo of CalStateLA
    if lat2 is None:
      lat2 = 34.0651 #34.0512 #
    if lon2 is None:
      lon2 = -118.1701 #118.2437   #
    if radius is None:
      radius = 0

    dlat = math.radians(lat2-lat1)
    dlon = math.radians(lon2-lon1)
    a = math.sin(dlat/2) * math.sin(dlat/2) + math.cos(math.radians(lat1)) \
        * math.cos(math.radians(lat2)) * math.sin(dlon/2) * math.sin(dlon/2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
    d = radius * c

    return d
   except:
    return 0.0

# distance in the radius from California State University Los Angeles
def distanceCalStateLA(lat1, lon1) :
  return distance(lat1, lon1, None, None, RADIUS)

# distance in the radius from University of California Los Angeles
def distanceUCLA(lat1, lon1) :
  return distance(lat1, lon1, 34.0689, -118.4452, RADIUS)

# distance in the radius University of Southern California
def distanceUSC(lat1, lon1) :
  return distance(lat1, lon1, 34.0224, -118.2851, RADIUS)
```

2) Registering UDFs

```
from pyspark.sql.functions import udf
from pyspark.sql.types import FloatType
distanceCalStateLA_udf = udf(distanceCalStateLA, FloatType())
distanceUCLA_udf = udf(distanceUCLA, FloatType())
distanceUSC_udf = udf(distanceUSC, FloatType())
```

Exercise 3: Usage of data frames and visualization

1) Below commands are used to calculate the distance for all the business from CSU LA

```
distance_df=business_df_yelp.withColumn('distance', distanceCalStateLA_udf('latitude'
,'longitude'))

display(distance_df)
```

A new column is added to *business_df_yelp* data frame by using **df.withColumn()** function

2) Similarly calculate the distance for each business from UCLA by using below code.

```
distance_ucla_df=business_df_yelp.withColumn('distance',
distanceUCLA_udf('latitude','longitude'))

display(distance_ucla_df)
```

3) Now get only businesses that are less than 5 miles from CalStateLA

```
business_within_5=distance_df.where(distance_df['distance']<5)

display(business_within_5)
```

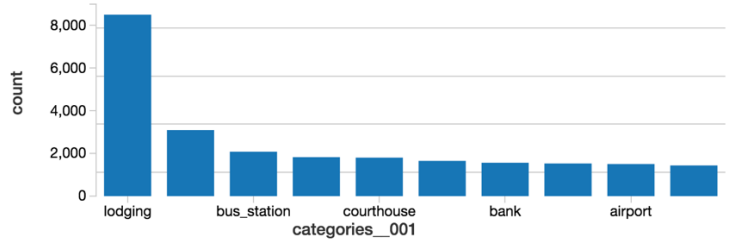
4) Get top 10 business categories in 5 miles from CalStateLA

```
from pyspark.sql.functions import col
top_business_csula=business_within_5.groupby('categories__001').count().sort(col("count").d
esc())

display(top_business_csula.take(10))
```

```
from pyspark.sql.functions import col
top_business_csula=business_within_5.groupby('categories__001').count().sort(col("count").desc())
display(top_business_csula.take(10))
```

(6) Spark Jobs



Command took 15.60s

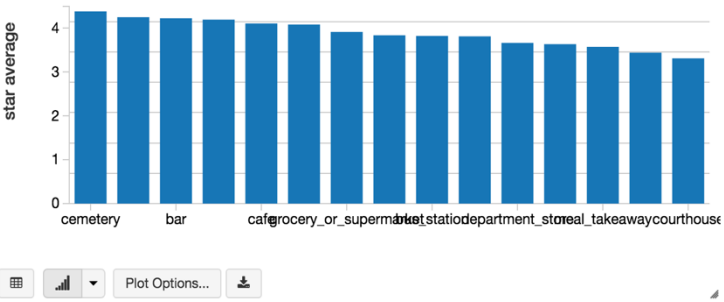
5) Get the businesses which are having more than 700 rating points where the ratings are higher than the overall average under 5 miles from CalStateLA

```
import pyspark.sql.functions as func

business_within_5=distance_df.where(distance_df['distance']<5)
display(business_within_5.groupBy('categories__001').agg(func.count('stars').alias('total')
,func.avg('stars').alias('star average')).where("total>700").orderBy(col('star
average').desc()))
```

```
business_within_5=distance_df.where(distance_df['distance']<5)
display(business_within_5.groupBy('categories__001')
        .agg(func.count('stars').alias('total'), func.avg('stars').alias('star average')).where("total>700").orderBy(col('star
average').desc()))
```

(1) Spark Jobs



Command took 12.63s

5) Get all the businesses under 5 miles from USC

```
business_df_usc=business_df_yelp.withColumn('distance',distanceUSC_udf('latitude','longitud
e'))
```

```
business_within_5_usc = distance_df_usc.where(distance_df_usc['distance']<5)
display(business_within_5_usc)
```

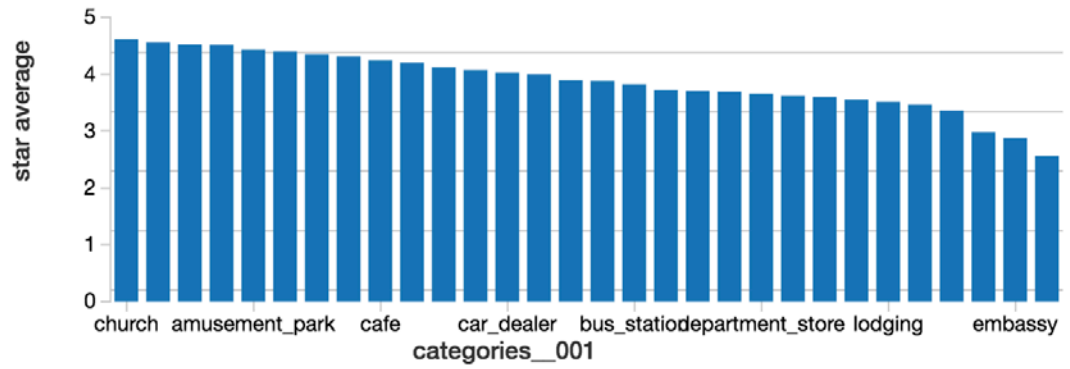
6) Get the businesses which are having more than 500 instances where the ratings are higher than the overall average under 5 miles from USC

```
Display(business_within_5_usc.groupBy('categories__001').agg(func.count('stars').alias('tot
    al'),func.avg('stars').alias('star average')).where("total>500").orderBy(col('star
                                    average').desc()))
```

You can download the results as csv and use it for visualization in excel. **NOTE:** You have an option to download more than 1,000 date.

```
> display(business_within_5_usc.groupBy('categories__001')
          .agg(func.count('stars').alias('total'), func.avg('stars').alias('star average'))
          .where("total>500").orderBy(col('star average').desc()))
```

▸ (1) Spark Jobs



r

Command took 12.87s

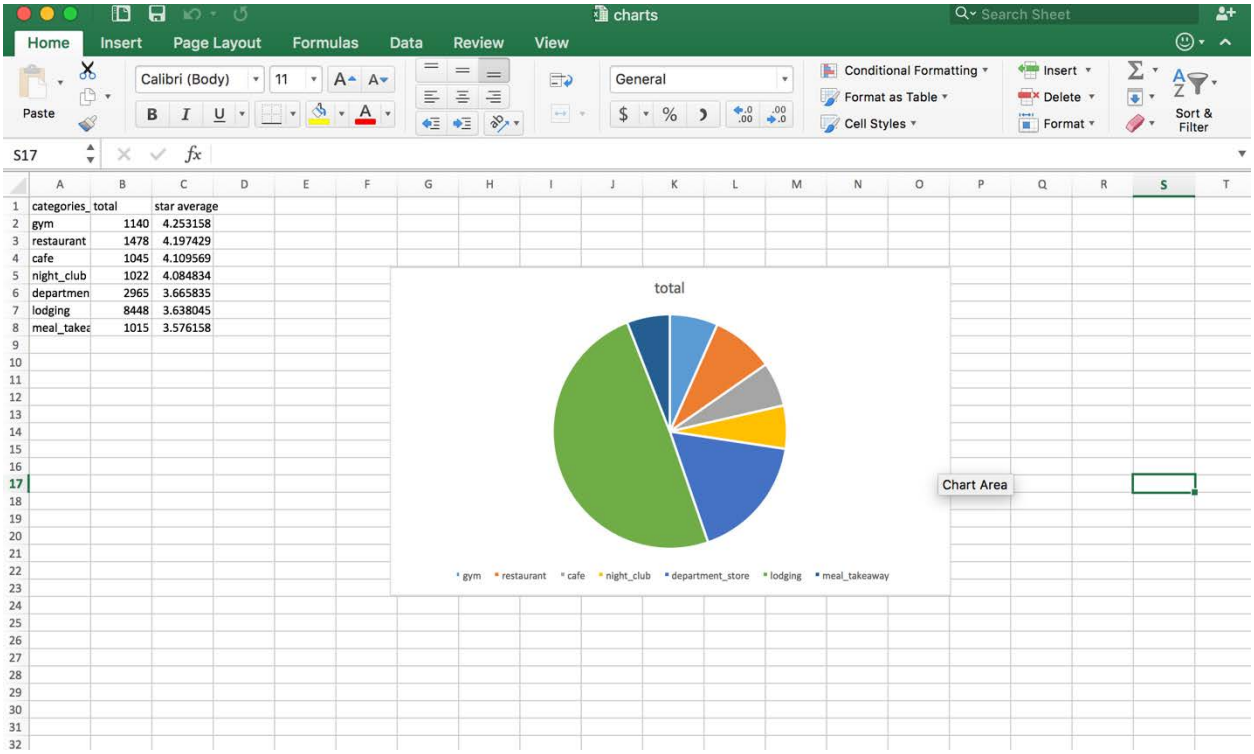
Repeat it for UCLA.

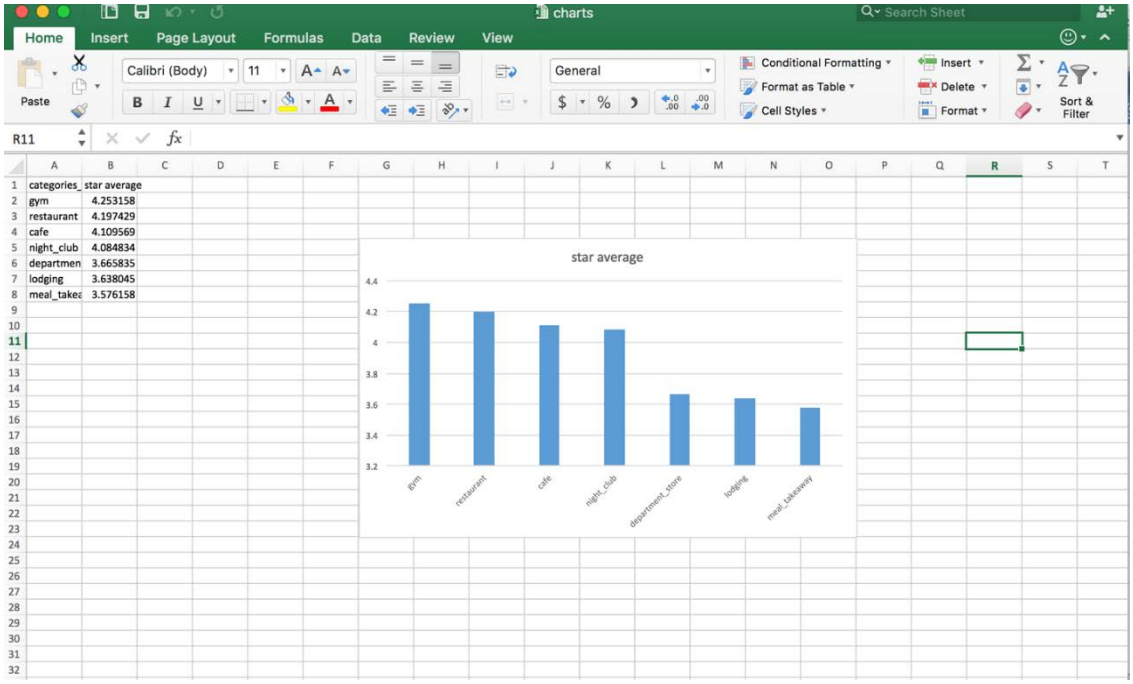
```
display(business_within_5_ucla.groupBy('categories__001')
        .agg(func.count('stars').alias('total'), func.avg('stars').alias('star average'))
        .where("total>500").orderBy(col('star average').desc()))
```

Exercise 4. Steps to visualize results in excel:

1). Download the output in csv format.

2). After opening the file, select recommended charts under insert tab and choose the chart style





Exercise 5: Using Spark SQL

1) Register data frame as table by using following command

```
business_df_yelp.registerTempTable("yelp_data")
```

Now Spark SQL queries can be adopted

2) Check areas which has more star ratings in and around Los Angeles

```
sqlContext.sql("SELECT count(stars) as total,city from business_data_yelp where city
IN('alhambra','Pasadena','Long beach','Santa monica','Beverly hills','burbank','West
hollywood','arcadia','El monte','Monterey park','San gabriel','downey','baldwin
park','Montebello','Los angeles') and stars=5 group by city order by total desc").show()
```

3) List the star rating and its total counts

```
display(sqlContext.sql("select stars as score, count(*) as total from ( select case  when
stars between 0 and 1 then ' 0-1' when stars between 1 and 2 then '1-2' when stars between
2 and 3 then '2-3' when stars between 3 and 4 then '3-4' when stars between 4 and 5 then
'4-5'when stars is Null  then 'others' end as stars from yelp_data) yelp_data group by
yelp_data.stars order by score"))
```

4) Check the number of businesses opened during different time intervals of the day

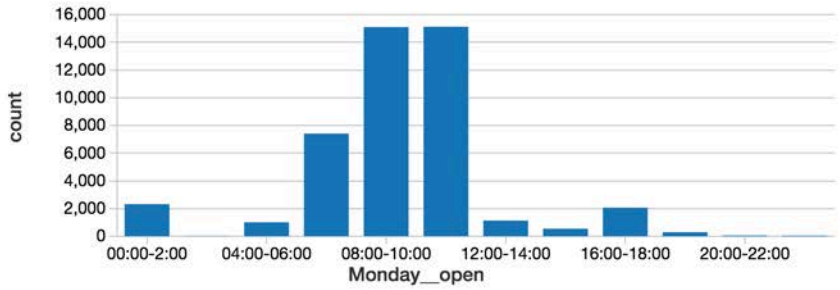
```
time_data=sqlContext.sql("select case when Monday__open between 0.0 and 1.99 then '00:00-
2:00' when Monday__open
between 2.0 and 3.59 then '02:00-04:00' when Monday__open between 4.0 and 5.59 then '04:00-
06:00' when Monday__open
between 6.0 and 7.59 then '06:00-08:00' when Monday__open between 8.0 and 9.59 then '08:00-
10:00' when Monday__open
between 10.0 and 11.59 then '10:00-12:00' when Monday__open between 12.0 and 13.59 then
'12:00-14:00' when Monday__open
between 14.0 and 15.39 then '14:00-16:00' when Monday__open between 16.0 and 17.59 then
'16:00-18:00' when Monday__open
 between 18.0 and 19.59 then '18:00-20:00' when Monday__open between 20.0 and 21.59 then
'20:00-22:00' when Monday__open
 between 22.0 and 23.59 then '22:00-00:00' end as Monday__open from yelp_data where
Monday__open is not null")
```

```
time_data=sqlContext.sql("select case when Monday__open between 0.0 and 1.99 then '00:00-2:00' when Monday__open between
2.0 and 3.59 then '02:00-04:00' when Monday__open between 4.0 and 5.59 then '04:00-06:00' when Monday__open between 6.0
and 7.59 then '06:00-08:00' when Monday__open between 8.0 and 9.59 then '08:00-10:00' when Monday__open between 10.0 and
11.59 then '10:00-12:00' when Monday__open between 12.0 and 13.59 then '12:00-14:00' when Monday__open between 14.0 and
15.39 then '14:00-16:00' when Monday__open between 16.0 and 17.59 then '16:00-18:00' when Monday__open between 18.0 and
19.59 then '18:00-20:00' when Monday__open between 20.0 and 21.59 then '20:00-22:00' when Monday__open between 22.0 and
23.59 then '22:00-00:00' end as Monday__open from yelp_data where Monday__open is not null")
```

```
Command took 0.12s
```

```
from pyspark.sql.functions import col
display(time_data.groupby('Monday__open').count().sort(col('Monday__open')))
```

▸ (1) Spark Jobs



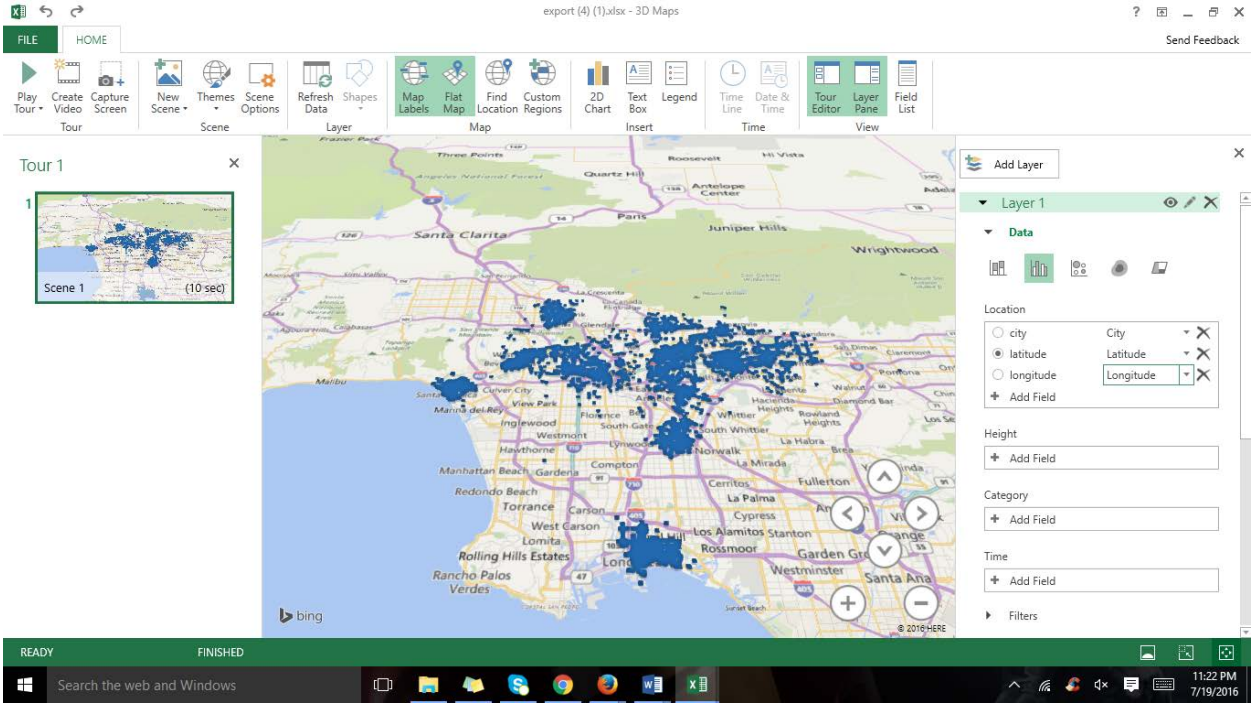
5) Spatial visualization of all the businesses in different cities around Los Angeles

```
display(sqlContext.sql("SELECT latitude,longitude,city from yelp_data where city
IN('alhambra','Pasadena','Long beach','Santa monica','Beverly hills','burbank','West
hollywood','arcadia','El monte','Monterey park','San gabriel','downey','baldwin
park','Montebello','Los angeles') "))
```

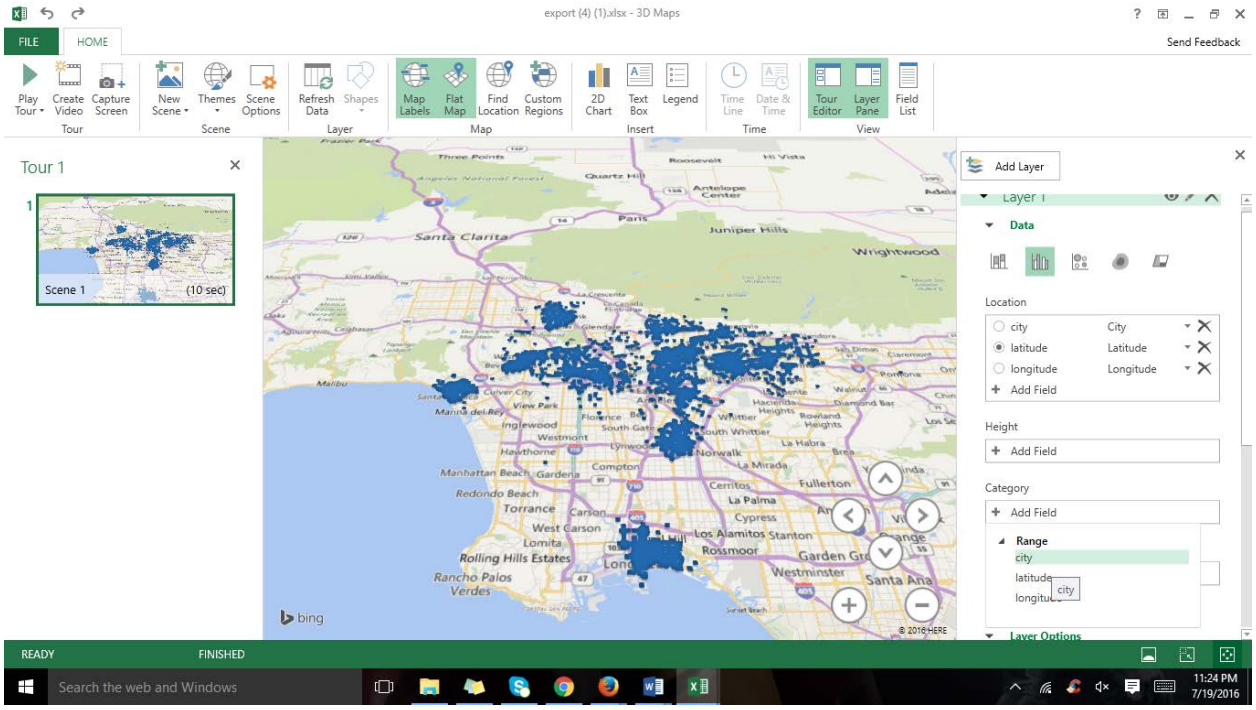
Exercise 6: Download the results and follow the below steps for Spatial visualization

1) Open Excel, select the data and go to “Insert” tab to select PowerMap for 3D visualization: Insert >> Map >> Launch PoweMap
   NOTE: Make sure to download PowerMap add-in.

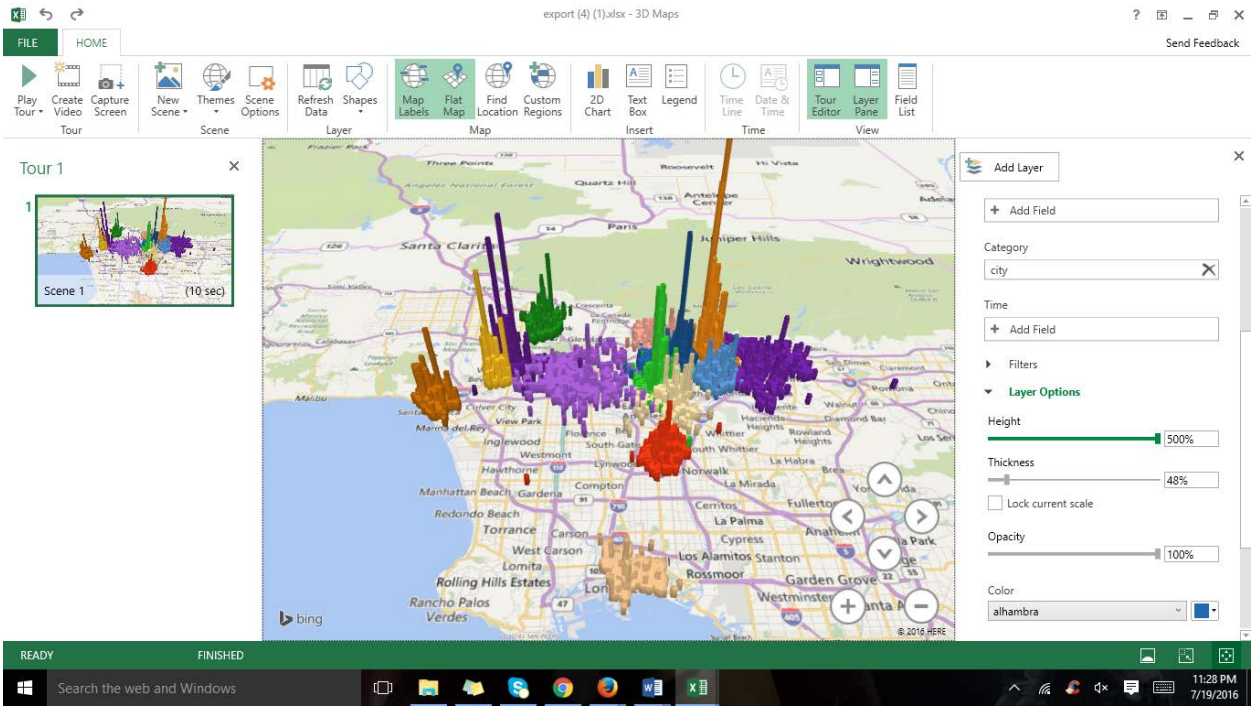
2) Open 3D maps and choose latitude, longitude and city respectively for rows having latitude, longitude and city values.



3) In the right side options panel of map, press add field under category and select city.



4) Increase the height under layer options to maximum.



5) List the businesses within 5 miles from CalStateLA, USC, UCLA using cache

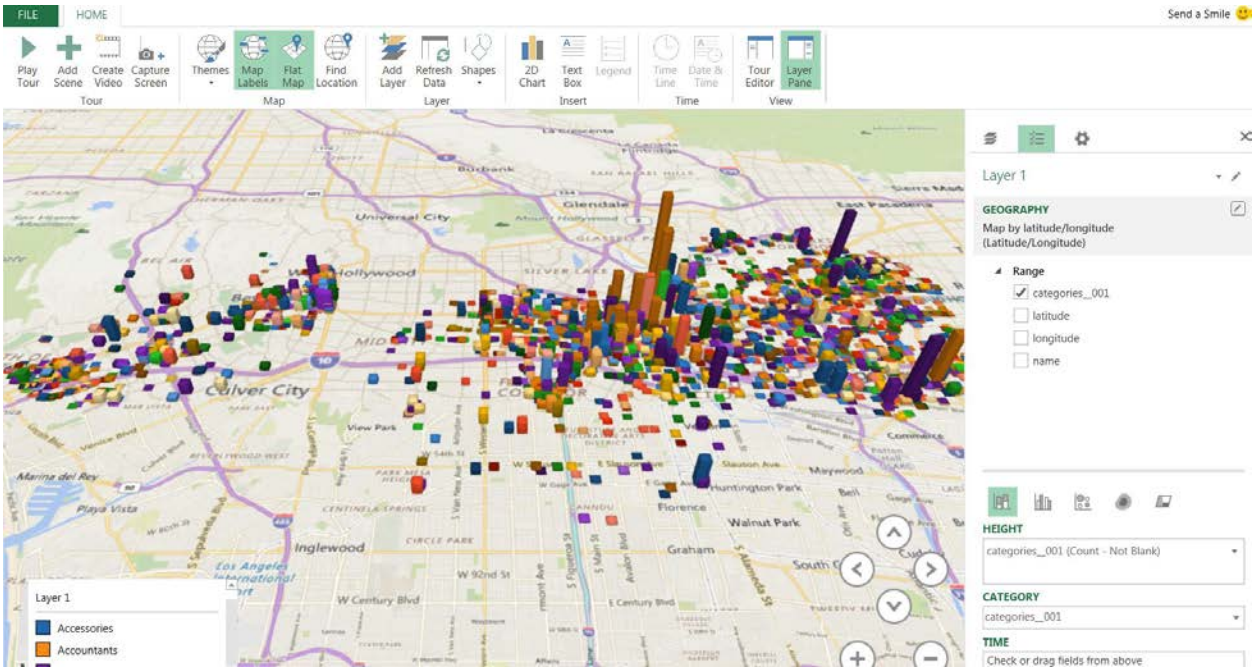
```
biz_near_schools =
business_within_5.unionAll(business_within_5_ucla.unionAll(business_within_5_usc))

biz_near_schools.cache()
```

6) List the businesses with 'latitude','longitude','categories__001', 'name'

```
display(biz_near_schools.select('latitude','longitude','categories__001', 'name'))
```

7) Download the csv file from the display cell result and repeat the Excel PowerMap to display the business categories near campuses as above; drag and drop “categories_001” to Height and Category fields in the PowerMap in the below to display 3D map.


FILE
HOME
Send a Smile
Play Tour
Add Scene
Create Video
Capture Screen
Tour
Themes
Map Labels
Flat Map
Find Location
Map
Add Layer
Refresh Data
Shapes
Layer
2D Chart
Text Box
Legend
Insert
Time Line
Date & Time
Time
Tour Editor
Layer Pane
View
Universal City
Glendale
Culver City
Inglewood
Florence
Graham
Walnut Park
Huntington Park
Marina del Rey
Playa Vista
W Century Blvd
W 92nd St
Layer 1
Accessories
Accountants
Layer 1
GEOGRAPHY
Map by latitude/longitude
(Latitude/Longitude)
Range
categories_001
latitude
longitude
name
HEIGHT
categories_001 (Count - Not Blank)
CATEGORY
categories_001
TIME
Check or drag fields from above