Homework 2
Project Report
Yelp Data Ananlysis

Udita Gupta
NetID: ung200

December 16, 2017

The dataset is freely available at https://www.yelp.com/dataset

We just need to fill out a small form and we can access the data. We get a .tar file and we can extract the json files from it. This homework did not require us to use the images dataset. But it made us use the other five datasets, which consisted of reviews, users, business, tips.

Here I did some very minimal statistics to get some sense out of the data. Mainly it was implemented in Spark and PIG.

Before performing any of the operations, I made sure that the HPC is working correctly, and flawlessly. Also, the most challenging task was to load the json dataset into Pig variables. This was resolved using the libraries by Twitter, called elephant-bird.
First 3 questions were done in %sh and last 2 in %pyspark.

The report contains each question described in sequence. Each section contains the respective questions and the code for each of them and the plots for them.

The assignment was done in collaboration with my project teammates:
1. Yashashwani Gupta – yg1568
2. Hitarthi Shah – hus206
3. Ilyas Habeeb - mih278

Following are the operations that were performed:

```
%sh
ls /shared/d/ung200/Downloads/dataset

val df = spark.read.json("/Users/ung200/Downloads/dataset/business.json")
```

1. The average # of reviews and the average # of stars grouped by city and business category.

   ```
   %sh
   #Question 1
   ```

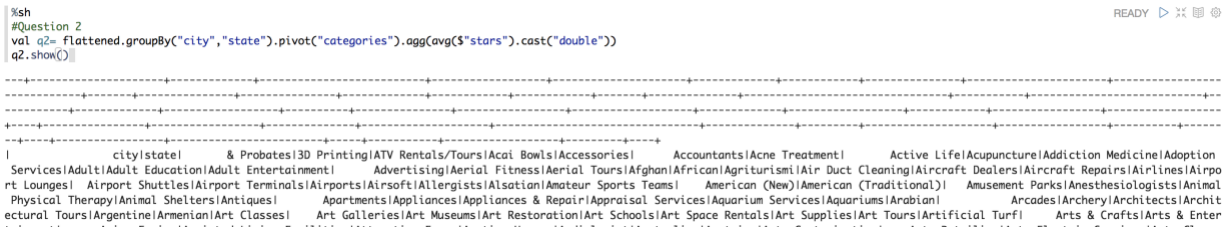
```
val flattened = df.withColumn("categories", explode($"categories"))
flattened.show()
flattened.createOrReplaceTempView("business")
val sqlDF=spark.sql("SELECT AVG(stars),city,AVG(review_count),categories FROM
business GROUP BY categories,city ORDER BY city DESC")
sqlDF.show()
```

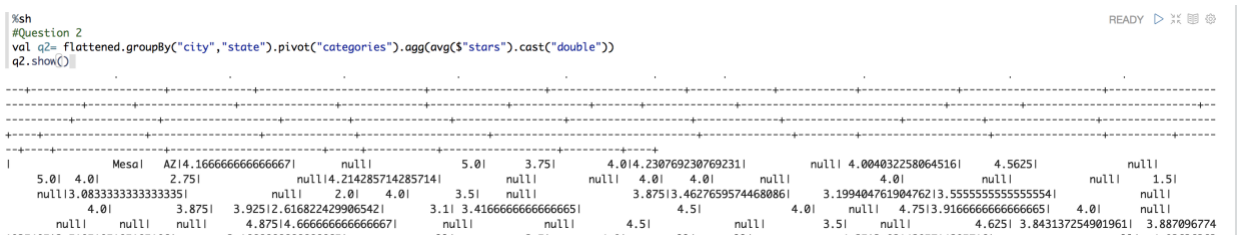
```
sqlDF: org.apache.spark.sql.DataFrame = [avg(stars): double, city: string ... 2 more fields]
+----------+--------------+-----------------+--------------------+
|avg(stars)|          city|avg(review_count)|          categories|
+----------+--------------+-----------------+--------------------+
|       3.5|Île-des-Soeurs|              3.0|         Restaurants|
|       3.5|Île-des-Soeurs|              3.0|          Sushi Bars|
|       4.0|Île des Soeurs|             38.0|            Day Spas|
|       4.0|Île des Soeurs|             38.0|       Beauty & Spas|
|       2.0|       toronto|              4.0|    Furniture Stores|
|       5.0|       toronto|              5.0|          Pet Stores|
|       5.0|       toronto|              5.0|Professional Serv...|
|       5.0|       toronto|              5.0|        Pet Services|
|       2.0|       toronto|              4.0|     Interior Design|
|       3.0|       toronto|             45.0|         Restaurants|
|       5.0|       toronto|              5.0|   Aquarium Services|
|       3.0|       toronto|             45.0|          Sushi Bars|
|       3.0|       toronto|             45.0|              Korean|
|       2.0|       toronto|              4.0|          Home Decor|
```

2. Pivot the business categories as columns, and show the average # stars for each category, by (city,state):

```
val tor=flattened.filter("city=='toronto'")
tor.groupBy("city","state").pivot("categories").agg(avg($"stars").cast("double")).show()

val q2= flattened.groupBy("city","state").pivot("categories").agg(avg($"stars").cast("double"))
q2.show()
```





3. What is the average rank (# stars) for businesses that are ‘Mexican’ category, AND offer takeout: (e.g. "attributes": {"RestaurantsTakeOut": true,…})

val q3=
flattened.withColumn("RestaurantsTakeOut",flattened("attributes.RestaurantsTakeOut"))
q3.filter("categories=='Mexican'").filter("RestaurantsTakeOut=='true'").agg(avg($"stars")).show()

4. For businesses within 15km of Toronto center, show the average # stars aand average # reviews by type of business category

Center: Toronto, CA
Latitude: 43.6532° N, 79.3832° W
The bounding circle for this problem is a ~15 km radius. A business falls in the region if it’s coordinates are within the circle.

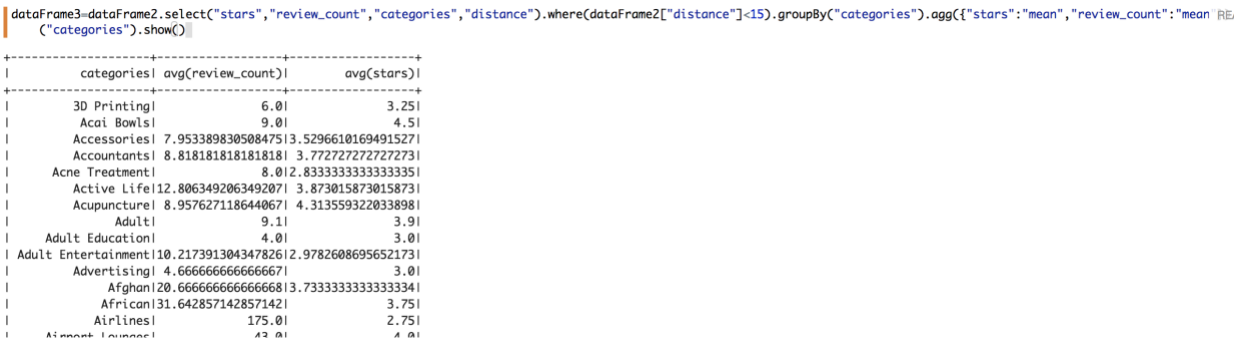
```
%pyspark
from pyspark.sql.functions import *
from math import sin, cos, radians, acos
dataFrame = sqlContext.read.json("/shared/d/ung200/Downloads/dataset/business.json")

dataFrame1 =
dataFrame.select(explode("categories").alias("categories"),"latitude","longitude",
"stars","city","name","review_count","business_id").orderBy("categories")
def dist(lat,long):
   if(lat is None or long is None):
      return 100;
   return acos(
      sin(radians(lat)) * sin(radians(43.6532)) +
      cos(radians(lat)) * cos(radians(43.6532)) *
      cos(radians(long) - radians(-79.3832))
   ) * 6371
```

```
from pyspark.sql.functions import udf
dist_udf = udf(dist)

dataFrame2=dataFrame1.select(dist_udf("latitude","longitude").alias("distance"),"stars","cit
y","review_count","categories","name","business_id").orderBy("city")

%pyspark
dataFrame3=dataFrame2.select("stars","review_count","categories","distance").where(data
Frame2["distance"]<15).groupBy("categories").agg({"stars":"mean","review_count":"mean"
}).orderBy("categories").show()
```



```
dataFrame3=dataFrame2.select("stars","review_count","categories","distance").where(dataFrame2["distance"]<15).groupBy("categories").agg({"stars":"mean","review_count":"mean RE
    ("categories").show()

+--------------------+-----------------+------------------+
|          categories| avg(review_count)|        avg(stars)|
+--------------------+-----------------+------------------+
|         3D Printing|              6.0|              3.25|
|          Acai Bowls|              9.0|               4.5|
|         Accessories| 7.9533898305084475|3.5296610169491527|
|         Accountants| 8.818181818181818| 3.772727272727273|
|      Acne Treatment|              8.0|2.8333333333333335|
|         Active Life|12.806349206349207| 3.873015873015873|
|         Acupuncture| 8.957627118644067| 4.313559322033898|
|               Adult|              9.1|               3.9|
|     Adult Education|              4.0|               3.0|
| Adult Entertainment|10.217391304347826|2.9782608695652173|
|         Advertising| 4.666666666666667|               3.0|
|             Afghan|20.666666666666668|3.7333333333333334|
|            African|31.642857142857142|              3.75|
|           Airlines|            175.0|              2.75|
|     Airport Lounges|             43.0|               4.0|
```

5. For the top 10 and bottom 10 food businesses near Toronto (ranked by stars), summarize star rating for reviews in January through May.

```
%pyspark
dataFrame5=dataFrame2.select("name","categories","business_id","stars","distance").wher
e(dataFrame2.categories=="Food").where(dataFrame2.distance<15).groupBy("business_id",
"name").agg(avg("stars").alias("Rating"))
dataFrame5.show()
dataFrame6=dataFrame5.orderBy(desc("Rating")).limit(10)
dataFrame6.show()
dataFrame7=dataFrame5.orderBy("Rating").limit(10)
dataFrame7.show()
```

```
%pyspark
dataFrame5=dataFrame2.select("name","categories","business_id","stars","distance").where(dataFrame2.categories=="Food").where(dataFrame2.distance<15).groupBy("business_id","name").agg(avg
    ("stars").alias("Rating"))
dataFrame5.show()
dataFrame6=dataFrame5.orderBy(desc("Rating")).limit(10)
dataFrame6.show()
dataFrame7=dataFrame5.orderBy("Rating").limit(10)
dataFrame7.show()
```

```
+--------------------+--------------------+------+
|         business_id|                name|Rating|
+--------------------+--------------------+------+
|Zn6DD4bPZPp0haHQB...|Bake Sale at Six ...|   4.0|
|hupi6kGVTvEn8Q0VG...|  Bruno's Fine Foods|   2.5|
|W8oYWcvFrthDx_tbH...|              Costco|   3.5|
|330HJ6PP0faybQt5u...|Chef George Break...|   4.0|
|Ge1JTxN0ytKncaoUs...|        Petite Thuet|   3.0|
|hbp7ZML8BHykBztIG...|          Pizza Nova|   2.5|
|Io-S_cPy4ElhHUheH...|Old's Cool Genera...|   5.0|
|9xTx4vWlL2kkhUqNE...|      Keeffaa Coffee|   4.5|
|R3iom4lNTG-9A6qcz...|          Java House|   3.5|
|snFLgArPIv9-QbZZc...|           Starbucks|   2.5|
|1yjpuOmGbjirCJjik...|           DAVIDsTEA|   4.5|
|AfQjI9Wbu6_Hzy2b3...|  Cafe On the Square|   2.5|
|bE7zwbaBHFb_EKWL5...|Mana'ish Global F...|   4.5|
|G6aNyWVp2UjgKRidP...|         La Limonada|   4.5|
|GZo_8eDJOt3Fv6eMm...|Manni Wasserman's...|   4.5|
```

reviews= sqlContext.read.json("/shared/d/ung200/Downloads/dataset/review.json")

%pyspark
#top10
op=reviews.select(month('date').alias('date_month'),"business_id","stars")
op2=dataFrame6.join(op,'business_id').select(op.date_month,op.stars,dataFrame6.name,dataFrame6.business_id).where(op["date_month"]<6)
op2.show()
op2.groupBy("name","business_id").agg(avg("stars").alias("Rating")).show()

```
%pyspark
#top10
op=reviews.select(month('date').alias('date_month'),"business_id","stars")
op2=dataFrame6.join(op,'business_id').select(op.date_month,op.stars,dataFrame6.name,dataFrame6.business_id).where(op["date_month"]<6)
op2.show()
op2.groupBy("name","business_id").agg(avg("stars").alias("Rating")).show()
```

```
+----------+-----+--------------------+--------------------+
|date_month|stars|                name|         business_id|
+----------+-----+--------------------+--------------------+
|         5|    5|Goddard's Souveni...|-9zPSrzbZ81FismxD...|
|         4|    5|Goddard's Souveni...|-9zPSrzbZ81FismxD...|
|         3|    5|Road Grill Food T...|OQed6yRev3Jq6a-9B...|
|         4|    5|Road Grill Food T...|OQed6yRev3Jq6a-9B...|
|         1|    5|The Herbal Clinic...|g2fU5POyJ01a3-REv...|
|         2|    4|The Herbal Clinic...|g2fU5POyJ01a3-REv...|
|         3|    5|                LCBO|dy-qs8C-BuHtFDhvY...|
|         1|    5|  Hot Pot Restaurant|VBMJjX1rPuwVvzTAp...|
|         2|    5|  Hot Pot Restaurant|VBMJjX1rPuwVvzTAp...|
|         5|    5|  Hot Pot Restaurant|VBMJjX1rPuwVvzTAp...|
|         2|    5|       Global Cheese|0j3ScXP2pii16Y4oj...|
|         2|    5|       Global Cheese|0j3ScXP2pii16Y4oj...|
|         4|    5|Grinning Face Non...|rO77pOx3oQdzAhxFg...|
|         5|    5|Grinning Face Non...|rO77pOx3oQdzAhxFg...|
|         4|    5|           Starbucks|J58EVuAJqoPVwouGN...|
```

%pyspark
#bottom10
op3=reviews.select(month('date').alias('date_month'),"business_id","stars")
op3=dataFrame7.join(op3,'business_id').select(op3.date_month,op3.stars,dataFrame7.name,dataFrame7.business_id).where(op3["date_month"]<6)
op3.show()
op3.groupBy("name","business_id").agg(avg("stars").alias("Rating")).show()

```
%pyspark
#bottom10
op3=reviews.select(month('date').alias('date_month'),"business_id","stars")
op3=dataFrame7.join(op3,'business_id').select(op3.date_month,op3.stars,dataFrame7.name,dataFrame7.business_id).where(op3["date_month"]<6)
op3.show()
op3.groupBy("name","business_id").agg(avg("stars").alias("Rating")).show()
```

```
+----------+-----+--------------------+--------------------+
|date_month|stars|                name|         business_id|
+----------+-----+--------------------+--------------------+
|         4|    2|         Tim Hortons|OaKWXPZl3yfEbhcGW...|
|         5|    1|         Tim Hortons|OaKWXPZl3yfEbhcGW...|
|         3|    1|         Tim Hortons|OaKWXPZl3yfEbhcGW...|
|         5|    1|  Parkway Fine Foods|CHf_Uk6x6pF740PA6...|
|         4|    1|  Parkway Fine Foods|CHf_Uk6x6pF740PA6...|
|         4|    1|  Parkway Fine Foods|CHf_Uk6x6pF740PA6...|
|         5|    3|  Parkway Fine Foods|CHf_Uk6x6pF740PA6...|
|         5|    1|         Coffee Time|85atsQTkgBz5tl_F2...|
|         4|    1|         Coffee Time|85atsQTkgBz5tl_F2...|
|         2|    1|         RealEat Inc|WvMkxBdYLTh8ikog8...|
|         1|    1|         RealEat Inc|WvMkxBdYLTh8ikog8...|
|         3|    1|         RealEat Inc|WvMkxBdYLTh8ikog8...|
|         4|    1|    Smoke's Weinerie|7T1XVTSocHcOu2H5v...|
|         3|    1|Swiss Chalet Roti...|u7bjH0lJcE7Q4BFlK...|
|         2|    1|  Alfredos Fine Food|JdEnsUNTkDbSRYnsn...|
```

PIG:

%sh
ls /Users/ung200/Downloads/dataset/

%pig
REGISTER '/Users/ung200/Downloads/dataset/elephant-bird-core-4.15.jar'
REGISTER '/Users/ung200/Downloads/dataset/elephant-bird-hadoop-compat-4.15.jar'
REGISTER '/Users/ung200/Downloads/dataset/elephant-bird-pig-4.15.jar'
REGISTER '/Users/ung200/Downloads/dataset/json-simple-1.1.1.jar'

%pig
a = LOAD '/Users/ung200/Downloads/dataset/business.json' USING
com.twitter.elephantbird.pig.load.JsonLoader('-nestedLoad') as (json:map[]);

%pig
--Question 1
query1_1 = FOREACH a GENERATE (int)json#'review_count' as
review_count,(double)json#'stars' as stars, json#'city' as city, json#'categories' as categories;
flattenedQuery1Data = FOREACH query1_1 GENERATE review_count,stars, city,
FLATTEN(categories);
groupedQuery1Data = GROUP flattenedQuery1Data BY (city,categories);
finalData = FOREACH groupedQuery1Data GENERATE group.city as city , group.categories as
category, AVG(flattenedQuery1Data.review_count), AVG(flattenedQuery1Data.stars);
dump finalData;

```
%pig
query1_1 = FOREACH a GENERATE (int)json#'review_count' as review_count,(double)json#'stars' as stars, json#'city' as city, json#'categories' as categories;
flattenedQuery1Data = FOREACH query1_1 GENERATE review_count,stars, city, FLATTEN(categories);
groupedQuery1Data = GROUP flattenedQuery1Data BY (city,categories);
finalData = FOREACH groupedQuery1Data GENERATE group.city as city , group.categories as category, AVG(flattenedQuery1Data.review_count), AVG(flattenedQuery1Data.stars);
dump finalData;
```

```
(,Pizza,4.0,3.5)
(,Fashion,5.0,4.0)
(,Italian,4.0,3.5)
(,Shopping,6.0,3.25)
(,Restaurants,4.0,3.5)
(,Sports Wear,5.0,4.0)
(,Sporting Goods,5.0,4.0)
(,Shopping Centers,7.0,2.5)
(Oka,Food,3.0,4.5)
(Oka,Thai,3.0,2.5)
(Oka,Parks,10.0,4.0)
(Oka,Beaches,10.0,4.0)
(Oka,Active Life,10.0,4.0)
(Oka,Campgrounds,10.0,4.0)
(Oka,Restaurants,3.0,2.5)
(Oka,Specialty Food,3.0,4.5)
(Oka,Hotels & Travel,10.0,4.0)
(Oka,Fruits & Veggies,3.0,4.5)
```

%pig
--Question 2
query2_3 = FOREACH a GENERATE (double)json#'stars' as stars, json#'city' as city, json#'state' as state, json#'categories' as categories;
query2_2 = FOREACH query2_3 GENERATE stars, city, state, FLATTEN(categories);
query2_1 = GROUP query2_2 BY (city,state,categories);
query2 = FOREACH query2_1 GENERATE group.city as city,group.state as state, group.categories as category, AVG(query2_2.stars);
dump query2;

```
%pig
query2_3 = FOREACH a GENERATE (double)json#'stars' as stars, json#'city' as city, json#'state' as state, json#'categories' as categories;
query2_2 = FOREACH query2_3 GENERATE stars, city, state, FLATTEN(categories);
query2_1 = GROUP query2_2 BY (city,state,categories);
query2 = FOREACH query2_1 GENERATE group.city as city,group.state as state, group.categories as category, AVG(query2_2.stars);
dump query2;
```

```
(,HH,Pizza,3.5)
(,HH,Italian,3.5)
(,HH,Restaurants,3.5)
(,EDH,Fashion,4.0)
(,EDH,Shopping,4.0)
(,EDH,Sports Wear,4.0)
(,EDH,Sporting Goods,4.0)
(,MLN,Shopping,2.5)
(,MLN,Shopping Centers,2.5)
(Oka,QC,Food,4.5)
(Oka,QC,Thai,2.5)
(Oka,QC,Parks,4.0)
(Oka,QC,Beaches,4.0)
(Oka,QC,Active Life,4.0)
(Oka,QC,Campgrounds,4.0)
(Oka,QC,Restaurants,2.5)
(Oka,QC,Specialty Food,4.5)
(Oka,QC,Hotels & Travel,4.0)
```

```
%pig
--Question 3
ques3_1 = FOREACH a GENERATE (double)json#'stars' as stars,json#'attributes' as attributes,
json#'categories' as categories:bag{a:tuple(b:chararray)} ;
ques3_2 = FOREACH ques3_1 GENERATE stars as
stars,FLATTEN(attributes#'RestaurantsTakeOut') as takeout, FLATTEN(categories) as categories;
ques3_3 = FILTER ques3_2 BY (categories=='Mexican') AND (takeout matches '.*true.*');
ques3_4 = GROUP ques3_3 BY (takeout,categories);
ques3 = FOREACH ques3_4 GENERATE group.takeout as takeout , group.categories as category,
AVG(ques3_3.stars);
dump ques3;
```

3. What is the average rank (# stars) for businesses that are 'Mexican' category, AND offer takeout: (e.g. "attributes": {"RestaurantsTakeOut": true,…})

```
%pig
ques3_1 = FOREACH a GENERATE (double)json#'stars' as stars,json#'attributes' as attributes, json#'categories' as categories:bag{a:tuple(b:chararray)} ;
ques3_2 = FOREACH ques3_1 GENERATE stars as stars,FLATTEN(attributes#'RestaurantsTakeOut') as takeout, FLATTEN(categories) as categories;
ques3_3 = FILTER ques3_2 BY (categories=='Mexican') AND (takeout matches '.*true.*');
ques3_4 = GROUP ques3_3 BY (takeout,categories);
ques3 = FOREACH ques3_4 GENERATE group.takeout as takeout , group.categories as category, AVG(ques3_3.stars);
dump ques3;

(true,Mexican,3.436754507628294)
```

```
%pig
--Question 4
ques4_1 = FOREACH a GENERATE (double)json#'stars' as stars, json#'latitude' as latitude,
json#'longitude' as longitude, (double)json#'review_count' as reviews, json#'categories' as
categories:bag{a:tuple(b:chararray)} ;
ques4_2 = FOREACH ques4_1 GENERATE stars,latitude,longitude,reviews, FLATTEN(categories)
as categories;

%pig
REGISTER '/Users/ung200/Downloads/dataset/piggybank.jar'
ques4_3 = FOREACH ques4_2 GENERATE *, (111.045*
org.apache.pig.piggybank.evaluation.math.toDegrees(ACOS(COS(org.apache.pig.piggybank.eval
uation.math.toRadians(43.6532))*
COS(org.apache.pig.piggybank.evaluation.math.toRadians(latitude))*
COS(org.apache.pig.piggybank.evaluation.math.toRadians(-79.3832) -
org.apache.pig.piggybank.evaluation.math.toRadians(longitude))+
SIN(org.apache.pig.piggybank.evaluation.math.toRadians(43.6532))*
SIN(org.apache.pig.piggybank.evaluation.math.toRadians(latitude))))) as distance:double;
ques4_4 = FILTER ques4_3 BY (distance<15);
ques4_5 = GROUP ques4_4 BY categories;
ques4 = FOREACH ques4_5 GENERATE group as category, AVG(ques4_4.stars),
AVG(ques4_4.reviews);
dump ques4;
```

```
%pig
--Question 4
ques4_1 = FOREACH a GENERATE (double)json#'stars' as stars, json#'latitude' as latitude, json#'longitude' as longitude, (double)json#'review_count' as reviews, json#'categories' as
    categories:bag{a:tuple(b:chararray)} ;
ques4_2 = FOREACH ques4_1 GENERATE stars,latitude,longitude,reviews, FLATTEN(categories) as categories;
```

READY

```
%pig
REGISTER '/Users/ung200/Downloads/dataset/piggybank.jar'
ques4_3 = FOREACH ques4_2 GENERATE *, (111.045* org.apache.pig.piggybank.evaluation.math.toDegrees(ACOS(COS(org.apache.pig.piggybank.evaluation.math.toRadians(43.6532))* COS(org.apache.pig
    .piggybank.evaluation.math.toRadians(latitude))* COS(org.apache.pig.piggybank.evaluation.math.toRadians(-79.3832) - org.apache.pig.piggybank.evaluation.math.toRadians(longitude))+ SIN
    (org.apache.pig.piggybank.evaluation.math.toRadians(43.6532))* SIN(org.apache.pig.piggybank.evaluation.math.toRadians(latitude))))) as distance:double;
ques4_4 = FILTER ques4_3 BY (distance<15);
ques4_5 = GROUP ques4_4 BY categories;
ques4 = FOREACH ques4_5 GENERATE group as category, AVG(ques4_4.stars), AVG(ques4_4.reviews);
dump ques4;
```

READY

```
(DJs,3.9,9.7)
(Bars,3.4200757575757574,47.38787878787879)
(Beer,3.5767045454545454,30.96590909090909)
(Food,3.6204998512347517,27.216602201725678)
(Golf,3.6428571428571143,4.714285714285714)
(Gyms,3.6614906832298137,9.77639751552795)
(Hats,4.125,7.0)
(Mags,3.8484848484848486,11.075757575757576)
(Pets,3.8634868421052633,9.233552631578947)
(Poke,3.8333333333333335,45.333333333333336)
(Pubs,3.3282967032967035,48.03846153846154)
(Rugs,3.7,9.2)
(Soup,3.6544117647058822,50.94117647058823)
(Thai,3.2542662116040955,47.13310580204778)
(Udon,3.5,258.0)
(Used,3.5617283950617282,11.135802469135802)
(Wigs,3.0,21.692307692307693)
```

```
%pig
ques5_1 = FOREACH a GENERATE (double)json#'stars' as stars,json#'business_id' as id,
json#'latitude' as latitude, json#'longitude' as longitude, json#'categories' as
categories:bag{a:tuple(b:chararray)} ;
ques5_2 = FOREACH ques5_1 GENERATE stars,id,latitude,longitude, FLATTEN(categories) as
categories;
REGISTER '/Users/ung200/Downloads/dataset/piggybank.jar'
ques5_3 = FOREACH ques5_2 GENERATE *, (111.045*
org.apache.pig.piggybank.evaluation.math.toDegrees(ACOS(COS(org.apache.pig.piggybank.eval
uation.math.toRadians(43.6532))*
COS(org.apache.pig.piggybank.evaluation.math.toRadians(latitude))*
COS(org.apache.pig.piggybank.evaluation.math.toRadians(-79.3832) -
org.apache.pig.piggybank.evaluation.math.toRadians(longitude))+
SIN(org.apache.pig.piggybank.evaluation.math.toRadians(43.6532))*
SIN(org.apache.pig.piggybank.evaluation.math.toRadians(latitude))))) as distance:double;
ques5_4 = FILTER ques5_3 BY (distance<15) AND (categories=='Food');
ques5_5 = GROUP ques5_4 BY id;
ques5_6 = FOREACH ques5_5 GENERATE group as id, AVG(ques5_4.stars) as stars;
dump ques5_6;
```

```
%pig
ques5_1 = FOREACH a GENERATE (double)json#'stars' as stars,json#'business_id' as id, json#'latitude' as latitude, json#'longitude' as longitude, json#'categories' as categories:bag{a:tuple
    (b:chararray)} ;
ques5_2 = FOREACH ques5_1 GENERATE stars,id,latitude,longitude, FLATTEN(categories) as categories;
REGISTER '/Users/ung200/Downloads/dataset/piggybank.jar'
ques5_3 = FOREACH ques5_2 GENERATE *, (111.045* org.apache.pig.piggybank.evaluation.math.toDegrees(ACOS(COS(org.apache.pig.piggybank.evaluation.math.toRadians(43.6532))* COS(org.apache.pig
    .piggybank.evaluation.math.toRadians(latitude))* COS(org.apache.pig.piggybank.evaluation.math.toRadians(-79.3832) - org.apache.pig.piggybank.evaluation.math.toRadians(longitude))+ SIN
    (org.apache.pig.piggybank.evaluation.math.toRadians(43.6532))* SIN(org.apache.pig.piggybank.evaluation.math.toRadians(latitude))))) as distance:double;
ques5_4 = FILTER ques5_3 BY (distance<15) AND (categories=='Food');
ques5_5 = GROUP ques5_4 BY id;
ques5_6 = FOREACH ques5_5 GENERATE group as id, AVG(ques5_4.stars) as stars;
dump ques5_6;
```

```
(-0DwB6Swi349EKfbAOF7A,3.5)
(-0NrB58jqKqJfuUCDupcsw,3.5)
(-25X5v1q3WU6s-craJSvTw,3.5)
(-4eA7UmZeiOKsGLmcXNx_w,4.0)
(-6CGECRbeyTCeyU4oHeXHQ,2.5)
(-76didnxGiiMO80BjSpYsQ,3.0)
(-9zPSrzbZ81FismxDSGLtA,5.0)
(-BAUrljU90RNV-hkokLhXA,3.5)
(-BJ0Z28LoETB_ZsdA4Ikeg,3.0)
(-BSf1LtnOrtCWAzmumxUpw,4.0)
(-Btu8zliXgeSH4eBm1u9Rw,4.0)
(-BvRroH7Q2CbJnp-IygX3Q,4.5)
(-EFDz-s9QUWJbFIp16O_3g,2.0)
(-Ej5bLw_bYuKt20k8FQi3w,4.0)
(-FHjXYCSizyNgUv-EXn6Yg,4.0)
(-IXNFjtECsn8fqF047tYFw,3.5)
(-IvATB9k2qNz19Gy0Q2NQw,3.5)
```

%pig
ques5_top = ORDER ques5_6 BY stars DESC;
top10 = limit ques5_top 10;
dump top10;

```
%pig
ques5_top = ORDER ques5_6 BY stars DESC;
top10 = limit ques5_top 10;
dump top10;
```

```
(xUsYf7lB0bi1zGEw41-aTw,5.0)
(oN8pqCTXY4ac4DmUXUfdvQ,5.0)
(aEA010Lba1mOCq976XThyw,5.0)
(6AeBlimS00y7CdhuhjpjRg,5.0)
(6T8YFkN7xkGlBfLMqdp28w,5.0)
(6aOnrzf15RMqFNOQ-_ElIA,5.0)
(V92rbUoSYcebJx42d10GZw,5.0)
(0j3ScXP2pii16Y4ojtKdSQ,5.0)
(6kIlmP82sIq2jxhNGUkEtg,5.0)
(DwG7_vYztZP-AMXEIvGgFA,5.0)
```

%pig
q5_bottom= ORDER ques5_6 BY stars ASC;
bottom10 = limit q5_bottom 10;
dump bottom10;

```
%pig
q5_bottom= ORDER ques5_6 BY stars ASC;
bottom10 = limit q5_bottom 10;
dump bottom10;
```

```
(V4226pZ4bN0mtEGeT7xkuQ,1.0)
(ldFrsUNIkDbSPYnspn8Pxg,1.0)
(WvMkxBdYLTh8ikog84ghCA,1.0)
(wl5lXES4GqDKJ00Z_c7ZVg,1.0)
(CHf_Uk6x6pF740PA6amvXw,1.0)
(u7bjH0lJcE7Q4BFlKTPJcg,1.0)
(85atsQTkgBz5tl_F2M4ZtA,1.0)
(UaoAGXDJPcpP7yObGZGD_Q,1.0)
(OaKWXPZl3yfEbhcGWFGTCw,1.0)
(7T1XVTSocHcOu2H5v2Iqog,1.0)
```

```
%pig
r = LOAD '/Users/ung200/Downloads/dataset/review.json' USING
com.twitter.elephantbird.pig.load.JsonLoader('-nestedLoad') as (json:map[]);
r1= FOREACH r GENERATE json#'stars' as stars,json#'business_id' as id, json#'date' as date;
top10bottom10 = UNION top10, bottom10;
joined_r = JOIN r1 by id, top10bottom10 by id;

%pig
final_required_Data = FOREACH joined_r GENERATE top10bottom10::id as bid,
(double)r1::stars as star,SUBSTRING(r1::date,5,7) as month;
filtered_data_by_month = FILTER final_required_Data BY (month matches '01|02|03|04|05');
dump filtered_data_by_month;
```

```
%pig
final_required_Data = FOREACH joined_r GENERATE top10bottom10::id as bid, (double)r1::stars as star,SUBSTRING(r1::date,5,7) as month;
filtered_data_by_month = FILTER final_required_Data BY (month matches '01|02|03|04|05');
dump filtered_data_by_month;
```

```
(0j3ScXP2pii16Y4ojtKdSQ,5.0,02)
(0j3ScXP2pii16Y4ojtKdSQ,5.0,02)
(6aOnrzf15RMqFNOQ-_ElIA,5.0,02)
(6aOnrzf15RMqFNOQ-_ElIA,5.0,01)
(6aOnrzf15RMqFNOQ-_ElIA,5.0,02)
(6aOnrzf15RMqFNOQ-_ElIA,5.0,05)
(6aOnrzf15RMqFNOQ-_ElIA,5.0,01)
(6aOnrzf15RMqFNOQ-_ElIA,5.0,01)
(7T1XVTSocHcOu2H5v2Iqog,1.0,04)
(CHf_Uk6x6pF740PA6amvXw,1.0,05)
(CHf_Uk6x6pF740PA6amvXw,3.0,05)
(CHf_Uk6x6pF740PA6amvXw,1.0,04)
(CHf_Uk6x6pF740PA6amvXw,1.0,04)
(WvMkxBdYLTh8ikog84ghCA,1.0,03)
(WvMkxBdYLTh8ikog84ghCA,1.0,02)
(WvMkxBdYLTh8ikog84ghCA,1.0,01)
(6T8YFkN7xkGlBfLMqdp28w,5.0,02)
(6T8YFkN7xkGlBfLMqdp28w,5.0,05)
```

%pig
grouped_data_by_business = GROUP filtered_data_by_month by bid;
avg_rating = FOREACH grouped_data_by_business GENERATE group,
AVG(filtered_data_by_month.star) as avg_stars;
dump avg_rating;

```
%pig
grouped_data_by_business = GROUP filtered_data_by_month by bid;
avg_rating = FOREACH grouped_data_by_business GENERATE group, AVG(filtered_data_by_month.star) as avg_stars;
dump avg_rating;
```

```
(0j3ScXP2pii16Y4ojtKdSQ,5.0)
(6T8YFkN7xkGlBfLMqdp28w,5.0)
(6aOnrzf15RMqFNOQ-_ElIA,5.0)
(7T1XVTSocHcOu2H5v2Iqog,1.0)
(85atsQTkgBz5tl_F2M4ZtA,1.0)
(CHf_Uk6x6pF740PA6amvXw,1.5)
(OaKWXPZl3yfEbhcGWFGTCw,1.3333333333333333)
(UaoAGXDJPcpP7yObGZGD_Q,1.0)
(V42Z6pZ4bN0mtEGeT7xkuQ,1.0)
(WvMkxBdYLTh8ikog84ghCA,1.0)
(aEA010Lba1mOCq976XThyw,5.0)
(ldFrsUNIkDbSPYnspn8Pxg,1.0)
(oN8pqCTXY4ac4DmUXUfdvQ,5.0)
(u7bjH0lJcE7Q4BFlKTPJcg,1.0)
(wl5lXES4GqDKJ00Z_c7ZVg,1.0)
```