Yelp Dataset Analysis Report

Assignment 2

BY-ASHISH BANSAL(AB6995)

Table of Contents

For this assignment i have used my local machine for completing my first three questions and for other questions i have used Dumbo in which i have used filezilla to import and export my files between locally and hadoop file system .

some hpc commands :

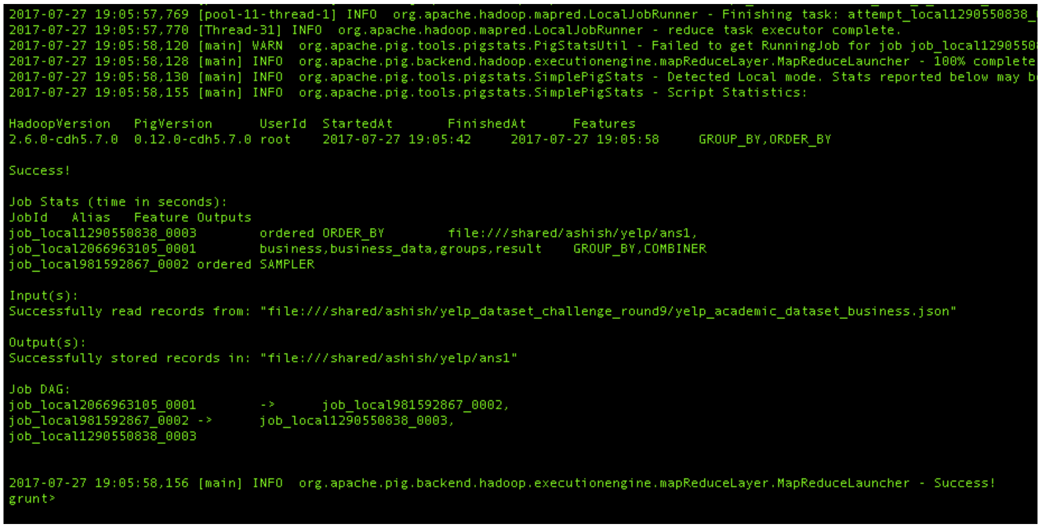
```
ssh ab6995@hpc2.nyu.edu

ssh dumbo

pig -x local

pyspark
```

1. Summarize the number of reviews by US city, by business category.

The pig script for the above question is as follows:

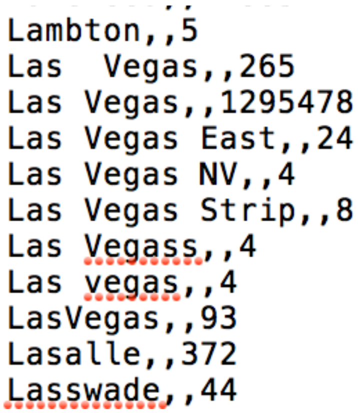
```
business_data =
       LOAD 'yelp_academic_dataset_business.json'
       USING JsonLoader(
       'business_id:chararray,
        name:chararray,
        neighborhood:chararray,
        address:chararray,
        city:chararray,
        state:chararray,
        postal_code:chararray,
        latitude:float,
        longitude:float,
        stars:float,
        review_count:int,
        is_open:int,
        attributes:bag{a:tuple(a:chararray)},
        categories:bag{a:tuple(a:chararray)},
        hours:bag{a:tuple(a:chararray)},
        type:chararray'
    );


business = FOREACH business_data GENERATE city, review_count,
FLATTEN(categories) as category;

groups = GROUP business BY (city, category);

result = FOREACH groups GENERATE FLATTEN(group) AS (city, category),
SUM(business.review_count);

ordered = ORDER result BY city;

STORE ordered INTO 'yelp/ans1' USING PigStorage(‘,');
```

I have used the yelp_academic_dataset_business.json file from the Yelp dataset to answer the above query. I loaded it into the pig using JSON loader. Filtered the data so that non-US cities are excluded from the data set. I then extracted the attributes city, review count and flattened the categories to generate a temporary table stored in variable business. Later grouped the tuples of this table and aggregated(sum) review count and later sorted it by city. Finally, stored the result into the destination specified above.



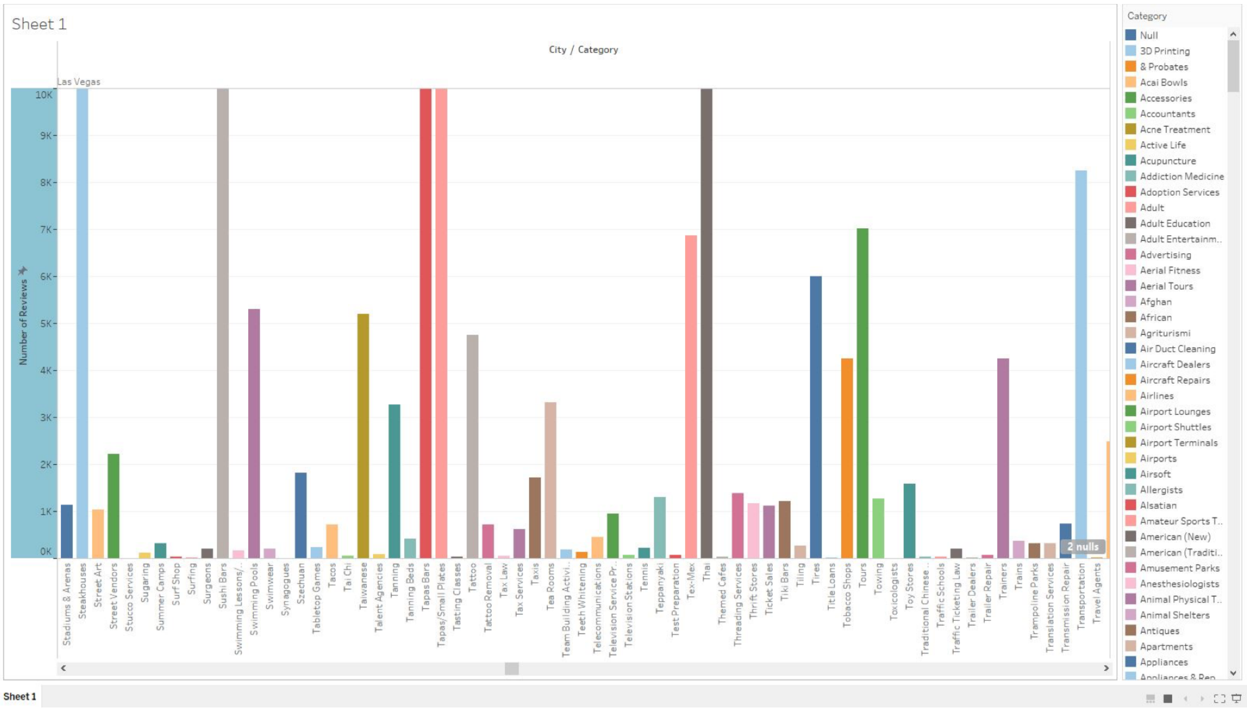
```
2017-07-27 19:05:57,769 [pool-11-thread-1] INFO  org.apache.hadoop.mapred.LocalJobRunner - Finishing task: attempt_local1290550838_
2017-07-27 19:05:57,770 [Thread-31] INFO  org.apache.hadoop.mapred.LocalJobRunner - reduce task executor complete.
2017-07-27 19:05:58,120 [main] WARN  org.apache.pig.tools.pigstats.PigStatsUtil - Failed to get RunningJob for job job_local12905508
2017-07-27 19:05:58,128 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2017-07-27 19:05:58,130 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Detected Local mode. Stats reported below may b
2017-07-27 19:05:58,155 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.7.0  0.12.0-cdh5.7.0 root    2017-07-27 19:05:42     2017-07-27 19:05:58     GROUP_BY,ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local1290550838_0003        ordered ORDER_BY        file:///shared/ashish/yelp/ans1,
job_local2066963105_0001        business,business_data,groups,result    GROUP_BY,COMBINER
job_local981592867_0002 ordered SAMPLER

Input(s):
Successfully read records from: "file:///shared/ashish/yelp_dataset_challenge_round9/yelp_academic_dataset_business.json"

Output(s):
Successfully stored records in: "file:///shared/ashish/yelp/ans1"

Job DAG:
job_local2066963105_0001        ->      job_local981592867_0002,
job_local981592867_0002 ->      job_local1290550838_0003,
job_local1290550838_0003


2017-07-27 19:05:58,156 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

There were a few problems which I observed in the output file which can cause fluctuations while analyzing the data and the plots might get affected by these. One of them is that some tuples have empty city attribute. Other is that same cities have been stored by different names like for example Las Vegas is stored as ‘Las Vegas East’, ‘Las Vegas NV’, ‘Las Vegas Strip’, ‘Las Vegass’, ‘LasVegas’ & ‘Las vegas’. Which causes inconsistency in the dataset and its analysis.

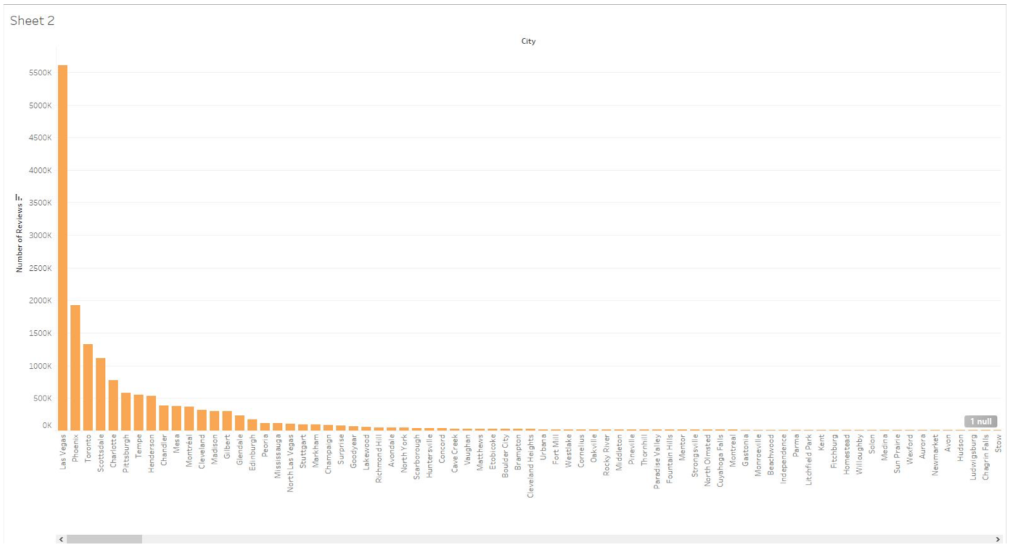


```
Lambton,,5
Las  Vegas,,265
Las Vegas,,1295478
Las Vegas East,,24
Las Vegas NV,,4
Las Vegas Strip,,8
Las Vegass,,4
Las vegas,,4
LasVegas,,93
Lasalle,,372
Lasswade,,44
```

For analyzing in R i have converted all the output files to csv.



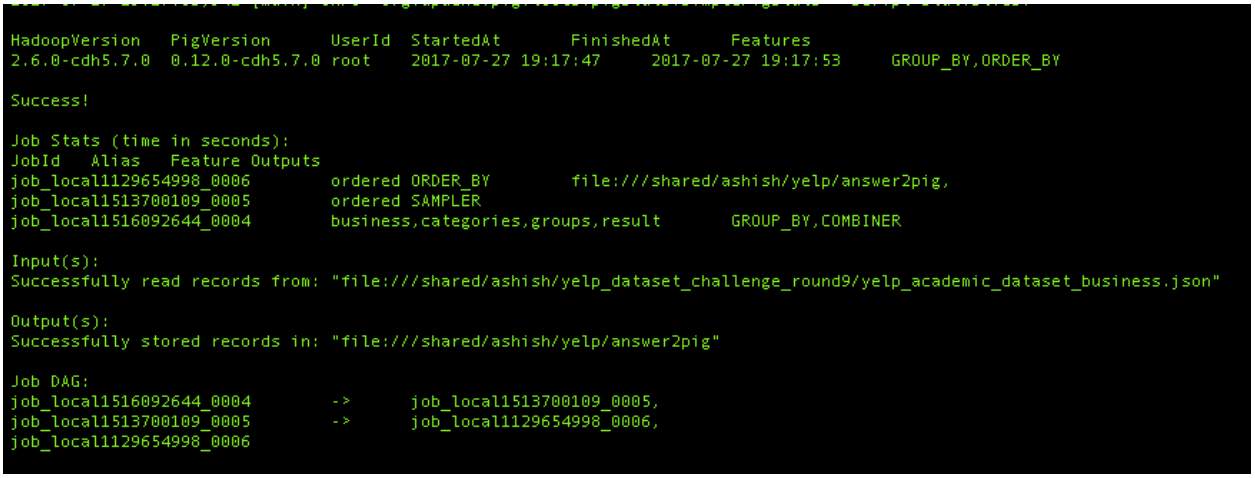
Bar Graph City/Category vs Number of Reviews



CITY(LAS VEGAS) VS NUMBER OF REVIEWS

2. Rank all cities by # of stars descending, for each category

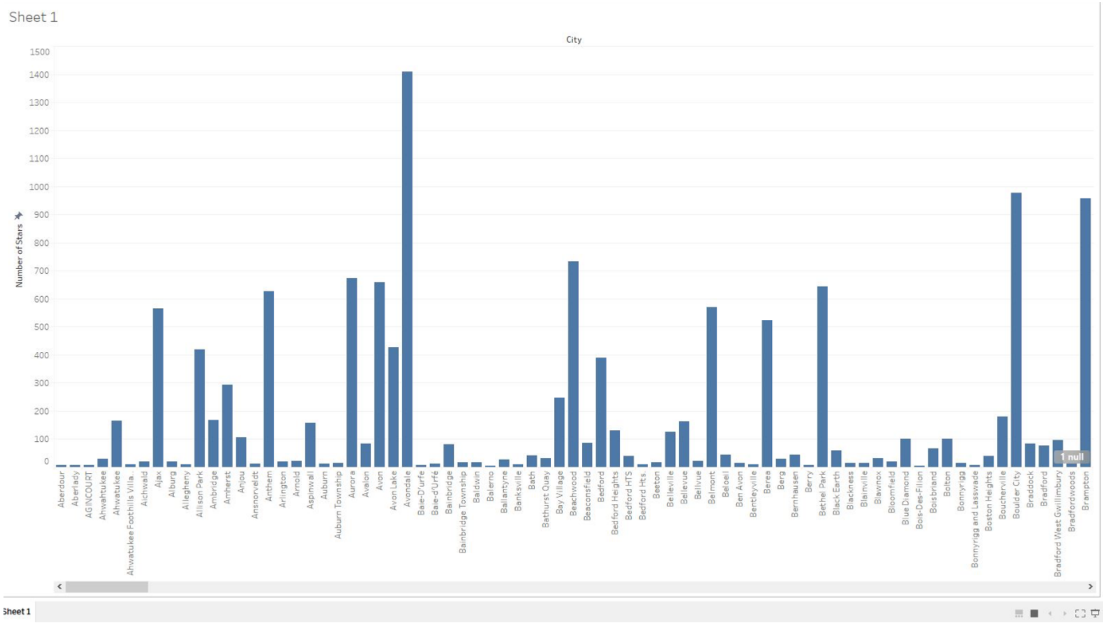
```
business =
        LOAD 'yelp_academic_dataset_business.json'
        USING JsonLoader(
        'business_id:chararray,
        name:chararray,
        neighborhood:chararray,
        address:chararray,
        city:chararray,
        state:chararray,
        postal_code:chararray,
        latitude:float,
        longitude:float,
        stars:float,
        review_count:int,
        is_open:int,
        attributes:bag{a:tuple(a:chararray)},
        categories:bag{a:tuple(a:chararray)},
        hours:bag{a:tuple(a:chararray)},
        type:chararray' );
categories = FOREACH business GENERATE city, stars, FLATTEN(categories) AS category;
groups = GROUP categories BY (category, city);
result = FOREACH groups
          GENERATE FLATTEN(group), AVG(categories.stars) AS rank;
ordered = ORDER result BY category, rank DESC;
STORE ordered INTO 'yelp/Answer2pig' USING PigStorage(‘,’);
```



```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.7.0  0.12.0-cdh5.7.0 root    2017-07-27 19:17:47     2017-07-27 19:17:53     GROUP_BY,ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local1129654998_0006        ordered ORDER_BY        file:///shared/ashish/yelp/answer2pig,
job_local1513700109_0005        ordered SAMPLER
job_local1516092644_0004        business,categories,groups,result       GROUP_BY,COMBINER

Input(s):
Successfully read records from: "file:///shared/ashish/yelp_dataset_challenge_round9/yelp_academic_dataset_business.json"

Output(s):
Successfully stored records in: "file:///shared/ashish/yelp/answer2pig"

Job DAG:
job_local1516092644_0004        ->      job_local1513700109_0005,
job_local1513700109_0005        ->      job_local1129654998_0006,
job_local1129654998_0006
```

I have used the yelp_academic_dataset_business.json file from the Yelp dataset to answer the above query. I loaded it into the pig using JSON loader. I then extracted city, stars and flattened the categories from the data and stored it in the variable categories. Then grouped the category and city as we have to rank all the cities for each category. Lastly, sorted the extracted data in descending order based on stars and stored the out in the above destination.

Analyzing the output file, I saw the same problem as observed in the previous case with one additional problem. There are empty values for business categories for some tuples. These all will add to inconsistent analysis of the data and plotting.

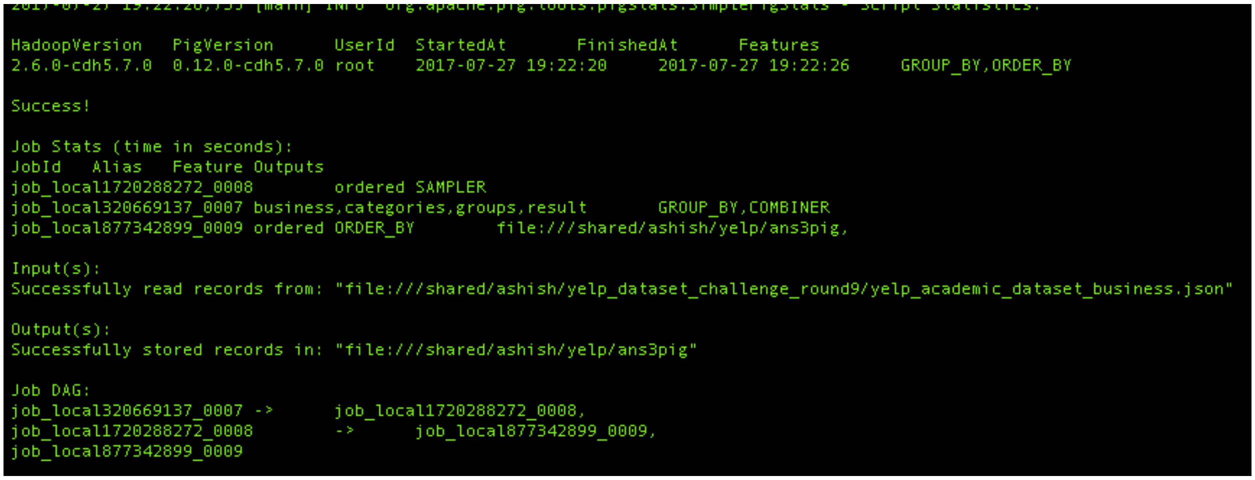


CITY VS NO. OF STARS

3. What is the average rank (# stars) for businesses within 10 miles of the University of Wisconsin - Madison, by type of business?

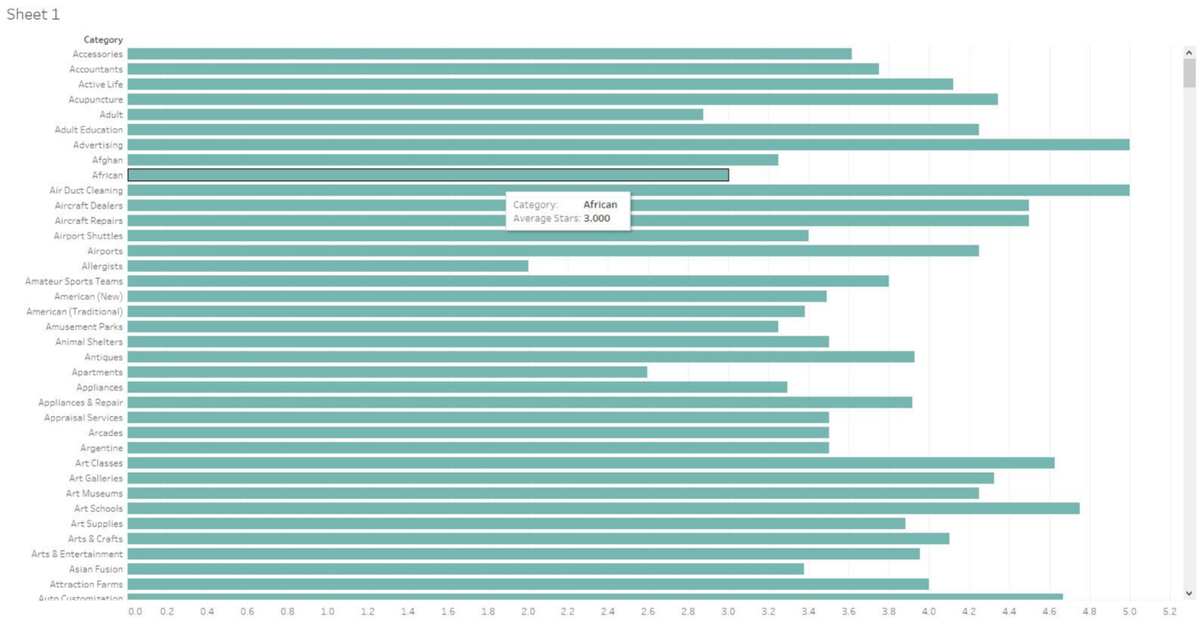
The pig script for the above question is as follows:

```
business =
    LOAD 'yelp_academic_dataset_business.json'
    USING JsonLoader(
    'business_id:chararray,
    name:chararray,
    neighborhood:chararray,
    address:chararray,
    city:chararray,
    state:chararray,
    postal_code:chararray,
    latitude:float,
    longitude:float,
        stars:float,
        review_count:int,
        is_open:int,
        attributes:bag{a:tuple(a:chararray)},
        categories:bag{a:tuple(a:chararray)},
        hours:bag{a:tuple(a:chararray)},
        type:chararray' );

filtered = FILTER business BY
  (latitude>42.908333) AND
  (latitude<43.241667) AND
  (longitude<-89.250556) AND
  (longitude>-89.583889);

categories = FOREACH filtered GENERATE stars, FLATTEN(categories) AS category;
groups = GROUP categories BY category;
result = FOREACH groups GENERATE FLATTEN(group), AVG(business_categories.stars) as rank;
ordered = ORDER result BY rank DESC;
STORE ordered INTO 'yelp/answer3pig' USING PigStorage(‘,');
```



```
2017-07-27 19:22:26,733 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.7.0  0.12.0-cdh5.7.0 root    2017-07-27 19:22:20     2017-07-27 19:22:26     GROUP_BY,ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local1720288272_0008        ordered SAMPLER
job_local320669137_0007 business,categories,groups,result       GROUP_BY,COMBINER
job_local877342899_0009 ordered ORDER_BY        file:///shared/ashish/yelp/ans3pig,

Input(s):
Successfully read records from: "file:///shared/ashish/yelp_dataset_challenge_round9/yelp_academic_dataset_business.json"

Output(s):
Successfully stored records in: "file:///shared/ashish/yelp/ans3pig"

Job DAG:
job_local320669137_0007 ->      job_local1720288272_0008,
job_local1720288272_0008        ->      job_local877342899_0009,
job_local877342899_0009
```

I have used the yelp_academic_dataset_business.json file from the Yelp dataset to answer the above query. I loaded it into the pig using JSON loader. I then filtered out the businesses not in the desired ranges of longitudes and latitudes. Flattened the categories and extracted stars from the filtered dataset. Stored the average rank of businesses within 20 miles of range from UWM by the type of business in sorted in descending order by average rank.

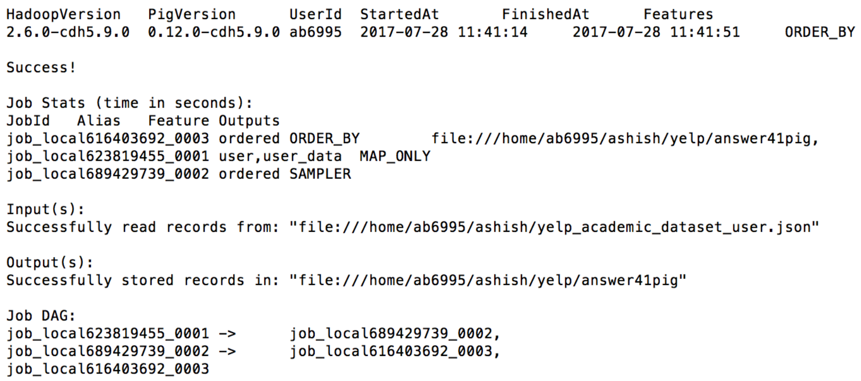


Average Stars vs Category

4. Rank reviewers by number of reviews. For the top 10 reviewers, show their average number of stars, by category.

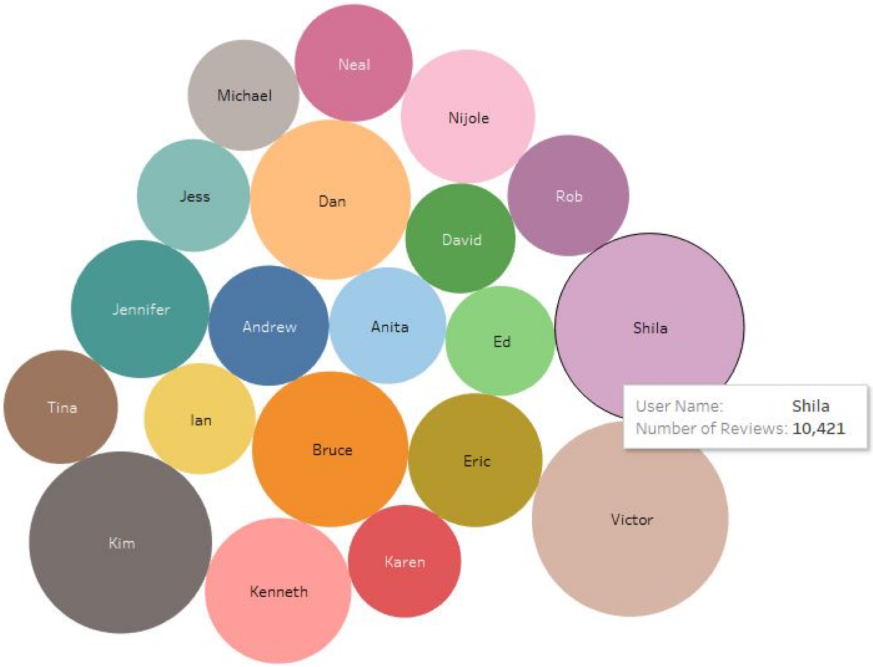
The pig script for the first part of question is as follows(rank reviewers by number of reviews):

```
user_data =
     LOAD ‘yelp_academic_dataset_user.json’
     USING JsonLoader(
     'user_id:chararray,
     name:chararray,
     review_count:int,
     yelping_since:chararray,
     friends:bag{a:tuple(a:chararray)},
     useful:int,
     funny:int,
     cool:int,
     fans:int,
     elite:bag{a:tuple(a:int)},
     average_stars:float,
     compliment_hot:int,
     compliment_more:int,
     compliment_profile:int,
     compliment_cute:int,
     compliment_list:int,
     compliment_note:int,
     compliment_plain:int,
     compliment_cool:int,
            compliment_funny:int,
            compliment_write:int,
            compliment_photos:int,
            type:chararray');

        user = FOREACH user_data GENERATE name, review_count;
        ordered = ORDER user BY review_count DESC;
STORE ordered INTO ‘yelp/answer41pig' USING PigStorage(‘,’);
```



```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.9.0  0.12.0-cdh5.9.0 ab6995  2017-07-28 11:41:14     2017-07-28 11:41:51     ORDER_BY

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local616403692_0003 ordered ORDER_BY        file:///home/ab6995/ashish/yelp/answer41pig,
job_local623819455_0001 user,user_data  MAP_ONLY
job_local689429739_0002 ordered SAMPLER

Input(s):
Successfully read records from: "file:///home/ab6995/ashish/yelp_academic_dataset_user.json"

Output(s):
Successfully stored records in: "file:///home/ab6995/ashish/yelp/answer41pig"

Job DAG:
job_local623819455_0001 ->      job_local689429739_0002,
job_local689429739_0002 ->      job_local616403692_0003,
job_local616403692_0003
```

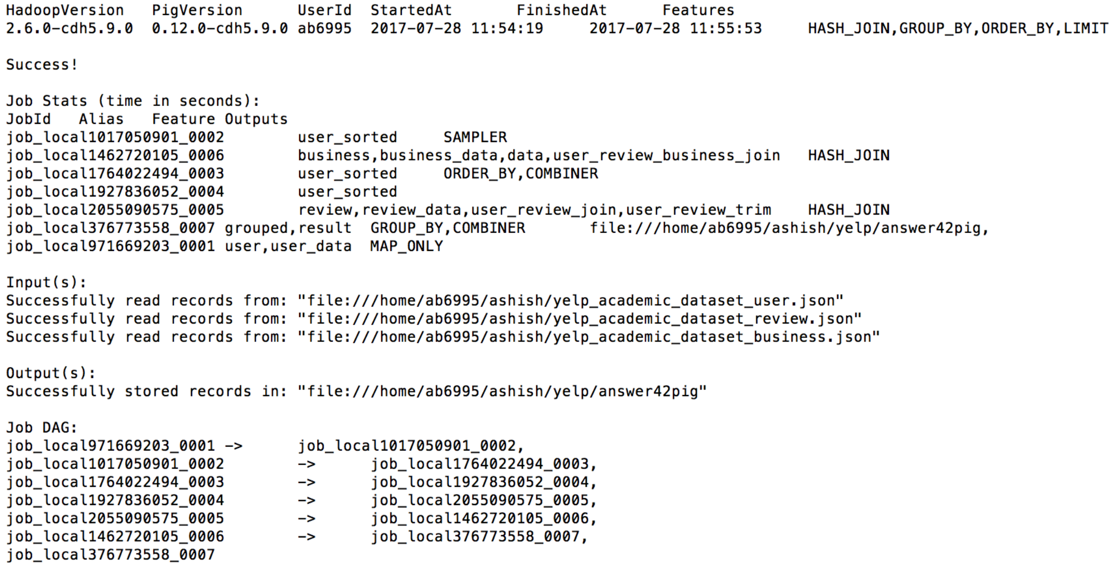
I have used the yelp_academic_dataset_user.json file from the Yelp dataset to answer the above query. I loaded it into the pig using JSON loader. For each tuple in the data set I extracted name and review count of the user and stored it in the variable user. Later sorted the tuples of extracted data by descending order of review count.



In the above graph, we can see that the size of the circle depicts the number of reviews made by the user. The lager the circle the more the number of reviews. Victor has the highest number of reviews(11284). One drawback of my output CSV was that I didn’t include the unique user IDs, which created an issue while plotting the data as there were many users with the same name.

The pig script for the second part of question is as follows(top 10 reviewers, show their average number of stars, by category):

```
business_data =
        LOAD ‘yelp_academic_dataset_business.json’
        USING JsonLoader(
        'business_id:chararray,
     name:chararray,
     neighborhood:chararray,
     address:chararray,
     city:chararray,
     state:chararray,
     postal_code:chararray,
     latitude:float,
     longitude:float,
     stars:float,
     review_count:int,
     is_open:int,
     attributes:bag{a:tuple(a:chararray)},
     categories:bag{a:tuple(a:chararray)},
     hours:bag{a:tuple(a:chararray)},
     type:chararray');

user_data =
        LOAD 'yelp_academic_dataset_user.json'
        USING JsonLoader(
                'user_id:chararray,
                name:chararray,
                review_count:int,
                yelping_since:chararray,
     friends:bag{a:tuple(a:chararray)},
     useful:int,
     funny:int,
     cool:int,
     fans:int,
     elite:bag{a:tuple(a:int)},
     average_stars:float,
     compliment_hot:int,
     compliment_more:int,
     compliment_profile:int,
     compliment_cute:int,
          compliment_list:int,
          compliment_note:int,
          compliment_plain:int,
          compliment_cool:int,
          compliment_funny:int,
          compliment_write:int,
          compliment_photos:int,
          type:chararray');

    review_data =
            LOAD 'yelp_academic_dataset_review.json'
            USING JsonLoader(
                    'review_id:chararray,
                    user_id:chararray,
                    business_id:chararray,
                    stars:float,
                    date:chararray,
                    text:chararray,
                    useful:int,
                    funny:int,
                    cool:int,
                    type:chararray');

    user = FOREACH user_data GENERATE user_id, name, review_count;
    business = FOREACH business_data GENERATE business_id, categories;
    review = FOREACH review_data GENERATE business_id, user_id, stars;
    user_sorted = ORDER user BY review_count DESC;
    top_10_users = LIMIT user_sorted 10;

    user_review_join = JOIN top_10_users BY user_id, review BY user_id;
    user_review_trim = FOREACH user_review_join GENERATE top_10_users::user_id AS user_id,
    name, stars, business_id;

    user_review_business_join = JOIN user_review_trim BY business_id, business BY business_id;

    data = FOREACH user_review_business_join GENERATE user_id, name, FLATTEN(categories)
    AS category, stars;

    grouped = GROUP data BY (user_id, name, category);
    result = FOREACH grouped GENERATE FLATTEN(group), AVG(data.stars) AS AVG_STARS;
STORE result INTO 'yelp/answer42pig' USING PigStorage(‘,’);
```

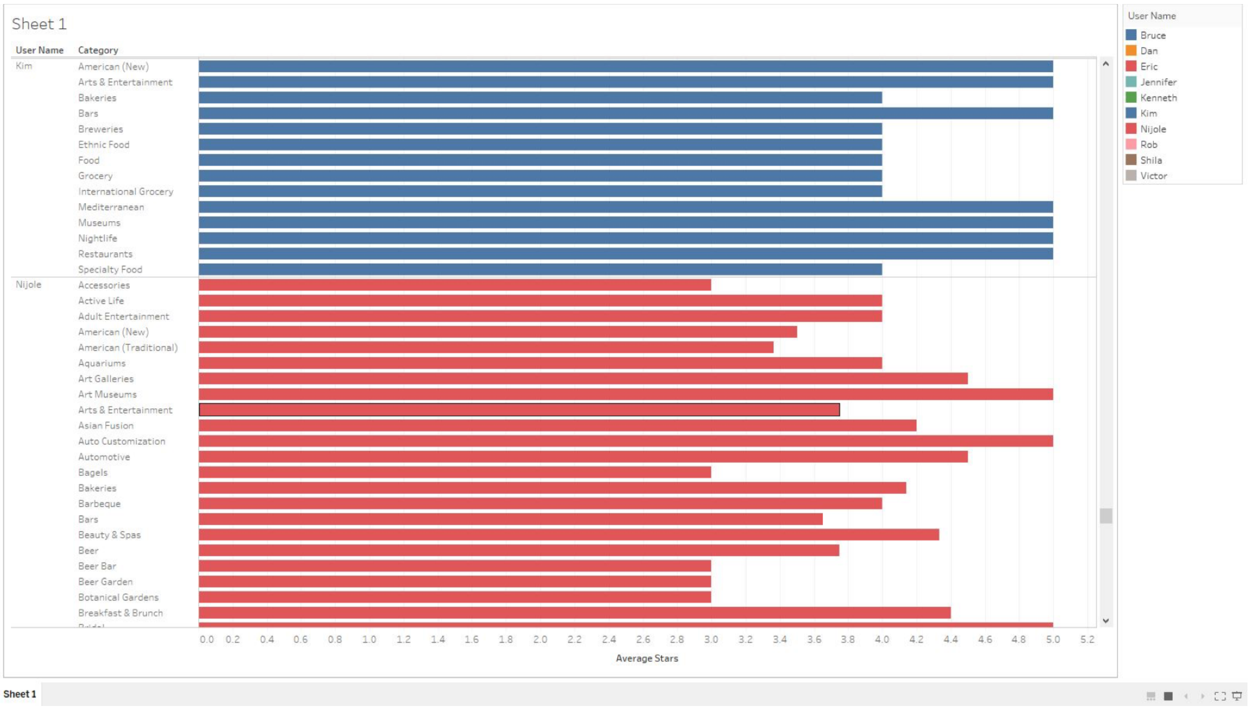


```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.9.0  0.12.0-cdh5.9.0 ab6995  2017-07-28 11:54:19     2017-07-28 11:55:53     HASH_JOIN,GROUP_BY,ORDER_BY,LIMIT

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local1017050901_0002        user_sorted     SAMPLER
job_local1462720105_0006        business,business_data,data,user_review_business_join   HASH_JOIN
job_local1764022494_0003        user_sorted     ORDER_BY,COMBINER
job_local1927836052_0004        user_sorted
job_local2055090575_0005        review,review_data,user_review_join,user_review_trim    HASH_JOIN
job_local376773558_0007 grouped,result  GROUP_BY,COMBINER       file:///home/ab6995/ashish/yelp/answer42pig,
job_local971669203_0001 user,user_data  MAP_ONLY

Input(s):
Successfully read records from: "file:///home/ab6995/ashish/yelp_academic_dataset_user.json"
Successfully read records from: "file:///home/ab6995/ashish/yelp_academic_dataset_review.json"
Successfully read records from: "file:///home/ab6995/ashish/yelp_academic_dataset_business.json"

Output(s):
Successfully stored records in: "file:///home/ab6995/ashish/yelp/answer42pig"

Job DAG:
job_local971669203_0001 ->      job_local1017050901_0002,
job_local1017050901_0002        ->      job_local1764022494_0003,
job_local1764022494_0003        ->      job_local1927836052_0004,
job_local1927836052_0004        ->      job_local2055090575_0005,
job_local2055090575_0005        ->      job_local1462720105_0006,
job_local1462720105_0006        ->      job_local376773558_0007,
job_local376773558_0007
```

I have used the yelp_academic_dataset_user.json, yelp_academic_dataset_business.json and yelp_academic_dataset_review.json file from the Yelp dataset to answer the above query.

I loaded it into the pig using JSON loader. First of all, I extracted user id, name, review count from user dataset. Similarly, extracted business id and categories from business dataset. Lastly, business id, user id and stars from the review dataset.

Then sorted the user data based on review count and selected top 10 reviewers and stored them in variable top_10_users. Later joined the data from top_10_users, review and business to get a final data table. Grouped it on user id, name and category and appended the average stars for each user.



Average Stars vs User Name/Category

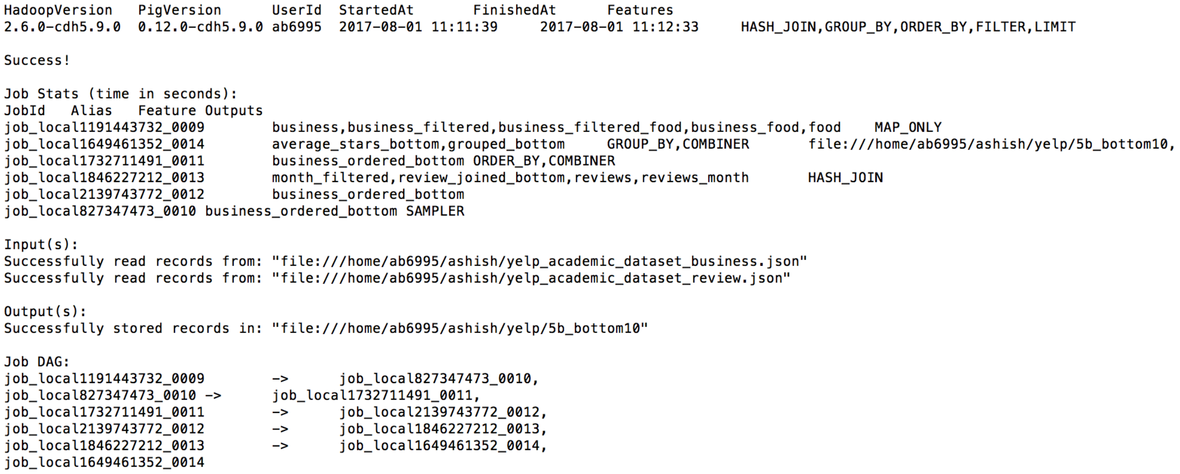
5. For the top 10 and bottom 10 food business near UWM (in terms of stars), summarize star rating for reviews in June through December.

The pig script for the second part of question is as follows(top 10 reviewers, show their average number of stars, by category):

```
    business_data =
            LOAD 'yelp_academic_dataset_business.json'
            USING JsonLoader(
            'business_id:chararray,
         name:chararray,
         neighborhood:chararray,
         address:chararray,
         city:chararray,
         state:chararray,
         postal_code:chararray,
         latitude:float,
         longitude:float,
         stars:float,
         review_count:int,
         is_open:int,
         attributes:bag{a:tuple(a:chararray)},
         categories:bag{a:tuple(a:chararray)},
         hours:bag{a:tuple(a:chararray)},
         type:chararray'
```

```
   );

business_location_filtered = FILTER business_data BY
          (latitude<43.241667)AND
   (latitude>42.908333) AND
   (longitude>-89.583889) AND
   (longitude<-89.250556);

business_category_not_null = FILTER business_location_filtered BY (categories IS NOT NULL);

business_food = FOREACH business_category_not_null {
          food = FILTER categories BY (a MATCHES '.*Food.*');
   GENERATE business_id, name, stars, (IsEmpty(food.$0) ? NULL : food) AS food;
}

business_filtered_food = FILTER business_food BY (food IS NOT NULL);


business_ordered = ORDER business_filtered_food BY stars DESC;
top_10 = LIMIT business_ordered 10;

reviews_data =
          LOAD 'yelp_academic_dataset_review.json'
          USING JsonLoader(
          'review_id:chararray,
      user_id:chararray,
      business_id:chararray,
      stars:float,
      date:chararray,
      text:chararray,
      useful:int,
      funny:int,
      cool:int,
      type:chararray'
   );

reviews_month = FOREACH reviews_data GENERATE business_id, stars AS review_stars,
GetMonth(ToDate(date, 'yyyy-MM-dd')) AS month;
month_filtered = FILTER reviews_month BY (month >= 1) AND (month <= 10);
review_joined = JOIN top_10 BY business_id, month_filtered BY business_id;
grouped = GROUP review_joined BY (top_10::business_id, name, month);
average_stars = FOREACH grouped GENERATE
                      FLATTEN(group), AVG(review_joined.review_stars);

STORE average_stars INTO 'yelp/5a_top10’ USING PigStorage(',');

business_ordered_bottom = ORDER business_filtered_food BY stars ASC;
bottom_10 = LIMIT business_ordered_bottom 10;
review_joined_bottom = JOIN bottom_10 BY business_id, month_filtered BY business_id;
```

```
grouped_bottom = GROUP review_joined_bottom BY (bottom_10::business_id, name, month);
average_stars_bottom = FOREACH grouped_bottom GENERATE
                        FLATTEN(group), AVG(review_joined_bottom.review_stars);

STORE average_stars_bottom INTO 'yelp/5b_bottom10' USING PigStorage(',');
```



```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.9.0  0.12.0-cdh5.9.0 ab6995  2017-08-01 11:11:39     2017-08-01 11:12:33     HASH_JOIN,GROUP_BY,ORDER_BY,FILTER,LIMIT

Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local1191443732_0009        business,business_filtered,business_filtered_food,business_food,food    MAP_ONLY
job_local1649461352_0014        average_stars_bottom,grouped_bottom     GROUP_BY,COMBINER       file:///home/ab6995/ashish/yelp/5b_bottom10,
job_local1732711491_0011        business_ordered_bottom ORDER_BY,COMBINER
job_local1846227212_0013        month_filtered,review_joined_bottom,reviews,reviews_month       HASH_JOIN
job_local2139743772_0012        business_ordered_bottom
job_local827347473_0010 business_ordered_bottom SAMPLER

Input(s):
Successfully read records from: "file:///home/ab6995/ashish/yelp_academic_dataset_business.json"
Successfully read records from: "file:///home/ab6995/ashish/yelp_academic_dataset_review.json"

Output(s):
Successfully stored records in: "file:///home/ab6995/ashish/yelp/5b_bottom10"

Job DAG:
job_local1191443732_0009        ->      job_local827347473_0010,
job_local827347473_0010 ->      job_local1732711491_0011,
job_local1732711491_0011        ->      job_local2139743772_0012,
job_local2139743772_0012        ->      job_local1846227212_0013,
job_local1846227212_0013        ->      job_local1649461352_0014,
job_local1649461352_0014
```

I have used the yelp_academic_dataset_business.json and yelp_academic_dataset_reviews.json file from the Yelp dataset to answer the above query. I loaded it into the pig using JSON loader. First of all, I filtered the businesses based on the longitude and latitude ranges. Filtered out the tuples with null categories. Separated the businesses associated with category food and sorted by stars. The top 10 rows are extracted and stored in the variable top_10. The review data is loaded and filtered for months between june to december by converting the date to month.

This result is later joined with top_10 businesses found earlier by business id. The group is flattened and average review stars of each group. Later stored in the file.
Similar procedure is followed for finding out the bottom 10 but just instead of top 10 businesses we extract the bottom 10 and joint to get the desired result.