Data Scientist Role Play: Profiling and Analyzing the Yelp Dataset Coursera Worksheet

This is a 2-part assignment. In the first part, you are asked a series of questions that will help you profile and understand the data just like a data scientist would. For this first part of the assignment, you will be assessed both on the correctness of your findings, as well as the code you used to arrive at your answer. You will be graded on how easy your code is to read, so remember to use proper formatting and comments where necessary.

In the second part of the assignment, you are asked to come up with your own inferences and analysis of the data for a particular research question you want to answer. You will be required to prepare the dataset for the analysis you choose to do. As with the first part, you will be graded, in part, on how easy your code is to read, so use proper formatting and comments to illustrate and communicate your intent as required.

For both parts of this assignment, use this "worksheet." It provides all the questions you are being asked, and your job will be to transfer your answers and SQL coding where indicated into this worksheet so that your peers can review your work. You should be able to use any Text Editor (Windows Notepad, Apple TextEdit, Notepad ++, Sublime Text, etc.) to copy and paste your answers. If you are going to use Word or some other page layout application, just be careful to make sure your answers and code are lined appropriately.
In this case, you may want to save as a PDF to ensure your formatting remains intact for you reviewer.

Part 1: Yelp Dataset Profiling and Understanding

1. Profile the data by finding the total number of records for each of the tables below:

i. Attribute table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords
From attribute
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

ii. Business table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From business
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
```

```
+---------------+
```

iii. Category table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From category
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

iv. Checkin table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From checkin
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

v. elite_years table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From elite_years
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

vi. friend table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From friend

+---------------+
```

```
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

vii. hours table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From hours
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

viii. photo table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From photo
```

```
 +---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

ix. review table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From review
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

x. tip table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From tip
```

```
+---------------+
```

```
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

xi. user table =

– Count the tot number of records in this table

```
Select count(*) as TotNumRecords

From user
```

```
+---------------+
| TotNumRecords |
+---------------+
|         10000 |
+---------------+
```

2. Find the total distinct records by either the foreign key or primary key f
or each table. If two foreign keys are listed in the table, please specify wh
ich foreign key.

i. Business =

```
Select count(distinct (id)) as TotNumdistRec
From business
```

```
+---------------+
| TotNumdistRec |
+---------------+
|         10000 |
+---------------+
```

ii. Hours =

```
Select count(distinct (business_id)) as TotNumdistRec
From hours
```

```
+---------------+
| TotNumdistRec |
+---------------+
|          1562 |
+---------------+
```

iii. Category =

```
Select count(distinct (business_id)) as TotNumdistRec
From category
```

```
+---------------+
| TotNumdistRec |
+---------------+
```

```
|          2643 |
+---------------+
```

iv. Attribute =

```
Select count(distinct (business_id)) as TotNumdistRec
From attribute
```

```
+---------------+
| TotNumdistRec |
+---------------+
|          1115 |
+---------------+
```

v. Review =
vi. Checkin =
vii. Photo =
viii. Tip =

```
Select count(distinct (user_id)) as TotNumdistRec
From tip
```

```
+---------------+
| TotNumdistRec |
+---------------+
|           537 |
+---------------+
```

ix. User =

```
Select count(distinct (id)) as TotNumdistRec
From user
```

```
+---------------+
| TotNumdistRec |
+---------------+
|         10000 |
+---------------+
```

x. Friend =

```
Select count(distinct (user_id)) as TotNumdistRec
From friend
```

```
+---------------+
| TotNumdistRec |
+---------------+
|            11 |
+---------------+
```

xi. Elite_years =

```
Select count(distinct (user_id)) as TotNumdistRec
From elite_years
```

```
+---------------+
```

```
| TotNumdistRec |
+---------------+
|          2780 |
+---------------+
```

Note: Primary Keys are denoted in the ER-Diagram with a yellow key icon.

3. Are there any columns with null values in the Users table? Indicate "yes," or "no."

Answer: No

SQL code used to arrive at answer:

```
-- Select all cols and check whether there is any null value
Select count(*)
From user
-- Eval all cols one by one for null, it will stop when it finds a null col
where id isnull or name isnull or review_count isnull or yelping_since isnull
or useful isnull or funny isnull or cool isnull or fans isnull or average_sta
rs isnull or compliment_hot isnull or compliment_more isnull or compliment_pr
ofile isnull or compliment_cute isnull or compliment_list isnull
or compliment_note isnull or compliment_plain isnull or compliment_cool isnul
l or compliment_funny isnull or compliment_writer isnull or compliment_photos
 isnull
```

4. For each table and column listed below, display the smallest (minimum), la rgest (maximum), and average (mean) value for the following fields:

i. Table: Review, Column: Stars

```
--Compute the min/max and average value of this col in this table and write t
hem down

Select min(stars) as minval
,max(stars) as maxval
,avg(stars) as average
From review
```

min: 1 max: 5 avg: 3.7082

ii. Table: Business, Column: Stars

```
-- Compute the min/max and average value of this col in this table and write
them down

Select min(stars) as minval
,max(stars) as maxval
,avg(stars) as average
From business
```

min: 1 max: 5 avg: 3.6589

iii. Table: Tip, Column: Likes

```
-- Compute the min/max and average value of this col in this table and write
them down
Select min(likes) as minval
,max(likes) as maxval
,avg(likes) as average
From tip
```

min: 0 max: 2 avg: 0.0144

iv. Table: Checkin, Column: Count

```
-- Compute the min/max and average value of this col in this table and write
them down
Select min(count) as minval
,max(count) as maxval
,avg(count) as average
From checkin
```

min: 1 max: 53 avg: 1.9414

v. Table: User, Column: Review_count

```
-- Compute the min/max and average value of this col in this table and write
them down
Select min(review_count) as minval
,max(review_count) as maxval
,avg(review_count) as average
From user
```

min: 0 max: 2000 avg: 24.2995

5. List the cities with the most reviews in descending order:

SQL code used to arrive at answer:

```
-- List the cities from table Business according to the # of reviews (review_
count column)
 Select city
 -- Create a new column with the sum of all business reviews grouped by city
 ,sum(review_count) as sumrevs_percity
 From business
 group by city
 order by sumrevs_percity desc;
```

Copy and Paste the Result Below:

city	sumrevs_percity
Las Vegas	82854
Phoenix	34503

```
| Toronto         |           24113 |
| Scottsdale      |           20614 |
| Charlotte       |           12523 |
| Henderson       |           10871 |
| Tempe           |           10504 |
| Pittsburgh      |            9798 |
| Montréal        |            9448 |
| Chandler        |            8112 |
| Mesa            |            6875 |
| Gilbert         |            6380 |
| Cleveland       |            5593 |
| Madison         |            5265 |
| Glendale        |            4406 |
| Mississauga     |            3814 |
| Edinburgh       |            2792 |
| Peoria          |            2624 |
| North Las Vegas |            2438 |
| Markham         |            2352 |
| Champaign       |            2029 |
| Stuttgart       |            1849 |
| Surprise        |            1520 |
| Lakewood        |            1465 |
| Goodyear        |            1155 |
+-----------------+-----------------+
(Output limit exceeded, 25 of 362 total rows shown)
```

6. Find the distribution of star ratings to the business in the following cities:

i. Avon

SQL code used to arrive at answer:

```
-- Find the distribution of star ratings to the business in the following city

 Select r.stars
 -- Creat a col that will contain the count the stars ratings/per rate
 ,count(r.stars) as countstars
 From review as r, business as b
 -- Filter by city of interest (city is linked to id from business and hence linked to business_id from review)
where (r.business_id = b.id
 and b.city='Avon')
 -- counting stars grouped by stars rating
 group by r.stars
```

Copy and Paste the Resulting Table Below (2 columns – star rating and count):

(Empty result)

ii. Beachwood

SQL code used to arrive at answer:

```
-- Find the distribution of star ratings to the business in the following cit
```

y

```
 Select r.stars
 -- Creat a col that will contain the count the stars ratings/per rate
 ,count(r.stars) as countstars
 From review as r, business as b
 -- Filter by city of interest (city is linked to id from business and hence
linked to business_id from review)
where (r.business_id = b.id
 and b.city='Beachwood')
 -- counting stars grouped by stars rating
 group by r.stars
```

Copy and Paste the Resulting Table Below (2 columns – star rating and count):

stars	countstars
3	1

7. Find the top 3 users based on their total number of reviews:

SQL code used to arrive at answer:

```
-- Find the top 3 users based on their total number of reviews
 Select name
 ,review_count
 From user
 order by review_count desc
 limit 3
```

Copy and Paste the Result Below:

name	review_count
Gerald	2000
Sara	1629
Yuri	1339

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

The number of fans does not seem to decay with the number of reviews written,
 hence giving a first look at this data I would say there is no positive corr
elation between both. Looking at the top 15 reviewers it can be seen that man
y of them have very few fans. Still for this to be fully analysed I could plo
t #review vs #fans for better visualisation of the data and then run a correl
ation statistical test

```
-- Explore whether higher #reviews correlates with higher #fans
```

```
 Select name
 ,review_count
 ,fans
 From user
 order by review_count desc
limit 15
```

name	review_count	fans
Gerald	2000	253
Sara	1629	50
Yuri	1339	76
.Hon	1246	101
William	1215	126
Harald	1153	311
eric	1116	16
Roanna	1039	104
Mimi	968	497
Christine	930	173
Ed	904	38
Nicole	864	43
Fran	862	124
Mark	861	115
Christina	842	85

9. Are there more reviews with the word "love" or with the word "hate" in the m?

Answer: More reviews with the word ‘love’. Here are the numbers

numTimes_love	numTimes_hate
1780	232

SQL code used to arrive at answer:

```
-- Find text with words 'love' and 'hate' in it and count them separately
 Select sum(text like '%love%') as numTimes_love
 ,sum(text like '%hate%') as numTimes_hate
 From review
```

10. Find the top 10 users with the most fans:

SQL code used to arrive at answer:

```
-- Find the top 10 users with the most fans
 Select name
```

```
 ,fans
 From user
 order by fans desc
 limit 10
```

Copy and Paste the Result Below:

name	fans
Amy	503
Mimi	497
Harald	311
Gerald	253
Christine	173
Lisa	159
Cat	133
William	126
Fran	124
Lissa	120

11. Is there a strong relationship (or correlation) between having a high number of fans and being listed as "useful" or "funny?" Out of the top 10 users with the highest number of fans, what percent are also listed as “useful” or “funny”?

Key:
0% - 25% - Low relationship
26% - 75% - Medium relationship
76% - 100% - Strong relationship

SQL code used to arrive at answer:

```
-- Explore whether higher #fans correlates with been highly tagged as useful
or funny (higher #useful or funny)
-- For this profiling analysis the code has a subquery in which users (table
user) are classified according to the number of fans. I set arbitrarily (and
exploratory) low_numbfans = 0-50, medium_numbfans = 51-99, high_numbfans > 10
0, and select name, fans, and useful+funny listings.
-- From this subquery, it sums the total number of fans,use_fun listings insi
de each group. But it also counts the number of users inside each class, beca
use sum of fans and numb of use_fun listings will have to be weighted to this
 values in order to have a fair comparison between #fans and #use_fun listing
s among the 3 different classes (low-medium-high amount of fans)
Select count(name) as numb_userinthis_class
,sum(fans) as sum_fans
,sum(use_fun) as sum_use_fun
,sum(fans)/(count(name)) as weighted_sum_fans
,sum(use_fun)/count(name) as weighted_sumuse_fun
,fans_class
from
     (Select name
     ,fans,
     case
     when fans<51 then 'low_numbfans'
```

```
     when (fans>=51 and fans<100) then 'medium_numbfans'
     when fans>=100 then 'high_numbfans'
     end as fans_class
     ,useful+funny as use_fun
     From user
     order by fans desc)
group by fans_class
```

Copy and Paste the Result Below:

```
----------------------+----------+-------------+-------------------+--------
-------------+-----------------+
| numb_userinthis_class | sum_fans | sum_use_fun | weighted_sum_fans | weight
ed_sumuse_fun | fans_class      |
+----------------------+----------+-------------+-------------------+-------
--------------+-----------------+
|                   16 |     3036 |      346790 |               189 |
        21674 | high_numbfans   |
|                 9952 |     9719 |      233264 |                 0 |
           23 | low_numbfans    |
|                   32 |     2141 |       48436 |                66 |
         1513 | medium_numbfans |
+----------------------+----------+-------------+-------------------+-------
--------------+-----------------+
```

Please explain your findings and interpretation of the results:

Weighting the sums is necessary because the amount of individuals inside each class is too different and this could affect the interpretation. For example, the sum_fans and sum_use_fun for low_numbfans class outreaches the numbers in medium_numbfans, but this is simply because 9952 individuals contributed to the final values. By dividing these numbers by the #individuals we get an objective measure.

From the results above observing the "weighted amounts of fans and useful+funny listings" of each group-class,i.e.low-medium-high number of fans, it is clear that both variables correlate positively: higher amount of fans correlate with being listed as useful or funny.

Part 2: Inferences and Analysis

1. Pick one city and category of your choice and group the businesses in that city or category by their overall star rating. Compare the businesses with 2-3 stars to the businesses with 4-5 stars and answer the following questions. Include your code.

i. Do the two groups you chose to analyze have a different distribution of hours?

After a qualitative inspection, low_rating businesses in Las Vegas tend to have a longer opening hours (8-22 for ex), while high-rated businesses have more office time hours (8-17 for ex).

Code:

—Go through a subquery that selects the city and stars from 2-5 dropping the 3.5 rating(as it is not asked to be included) and joins with table hours. sec

ond subquery goes through a case statement to classify the stars as low or hi
gh rating. Finally it extract from these two groups, hour,rating class column
s to visualise the distribution of hours both classes have. This is a qualita
tive inspection.

```
select hours
,ratingclass
from
     (select name
      ,stars
      ,hours,
      case -- classification according to #stars
      when stars in (2.0,2.5,3.0) then 'low_rating'
      else 'high_rating' end as ratingclass
      from
            (select name--subquery to take data from 1 city/leave out stars <
2
            ,stars
            ,hours
            from business inner join hours on business.id= hours.business_id
            where (city='Las Vegas' and stars >1.5 and stars!=3.5)
            order by stars)) --limit 25 offset 25
```

Result from the code:

hours	ratingclass
Monday|8:00-22:00	low_rating
Tuesday|8:00-22:00	low_rating
Friday|8:00-22:00	low_rating
Wednesday|8:00-22:00	low_rating
Thursday|8:00-22:00	low_rating
Sunday|8:00-22:00	low_rating
Saturday|8:00-22:00	low_rating
Monday|11:00-0:00	low_rating
Tuesday|11:00-0:00	low_rating
Friday|11:00-0:00	low_rating
Wednesday|11:00-0:00	low_rating
Thursday|11:00-0:00	low_rating
Sunday|11:00-0:00	low_rating
Saturday|11:00-0:00	low_rating
Monday|10:00-19:00	high_rating
Tuesday|10:00-19:00	high_rating
Friday|10:00-19:00	high_rating
Wednesday|10:00-19:00	high_rating
Thursday|10:00-19:00	high_rating
Saturday|10:00-19:00	high_rating
Monday|9:00-17:00	high_rating
Tuesday|9:00-17:00	high_rating
Friday|9:00-17:00	high_rating
Wednesday|9:00-17:00	high_rating
Thursday|9:00-17:00	high_rating

(Output limit exceeded, 25 of 69 total rows shown)

ii. Do the two groups you chose to analyze have a different number of reviews ?

Yes. they have a different number of reviews.

totrev_count	ratingclass
838	high_rating
403	low_rating

Code:

```
-- Go through two subqueries in which it is first selected the reviews_count
and stars (over 1.5 and not 3.5, as this is not included in the assignment) f
rom a chosen city ('Las Vegas'). Then classify stars by low_rating (2 to 3sta
rs)/high_rating (>3.5 stars). Finally, count reviews for these two classes.

select count (review_count) as totrev_count
,ratingclass
from
     (select name
      ,stars
      ,review_count,
      case -- classification according to #stars
      when stars in (2.0,2.5,3.0) then 'low_rating'
      else 'high_rating' end as ratingclass
      from
            (select name--subquery to take data from 1 city/leave out stars <
2
            ,stars
            ,review_count
            from business
            where (city='Las Vegas' and stars >1.5 and stars!=3.5)
            order by stars))
group by ratingclass –counts the reviews by group ratingclass
```

iii. Are you able to infer anything from the location data provided between t hese two groups? Explain.

From the analysis below, it can be seen that both groups have ratings coming from the same neighbourhoods.

SQL code used for analysis:

```
-- As the question before: the code goes through two subqueries in which sele
cts firstly reviews and stars (over 1.5) from a chosen city ('Las Vegas'). Th
en, classify stars by low_rating (2 to 3.5 stars)/high_rating (>3.5 stars). F
inally, counts #times a neighborhood appears associated to that rating groupi
ng by ratingclass (low or high), gives the name of the neighborhood and ratin
gclass. It perfoms this twice making a UNION for the 2 grouping (low/high rat
ingclass) so that presence of a common neighbor is visible.
select neighborhood
```

```
,count(neighborhood) as count_neigh
,ratingclass
from
     (select name
      ,stars
      ,neighborhood,
      case -- classification according to #stars
      when stars in (2.0,2.5,3.0) then 'low_rating'
      else 'high_rating' end as ratingclass
      from
            (select name--subquery to take data from 1 city/leave out stars <
2
            ,stars
            ,neighborhood
            ,review_count
            from business
            where (city='Las Vegas' and stars >1.5 and stars!=3.5)
            order by stars))

where ratingclass= ‘high_rating'
group by neighborhood


union

select neighborhood
,count(neighborhood) as count_neigh
,ratingclass
from
     (select name
      ,stars
      ,neighborhood,
      case -- classification according to #stars
      when stars in (2.0,2.5,3.0) then 'low_rating'
      else 'high_rating' end as ratingclass
      from
            (select name--subquery to take data from 1 city/leave out stars <
2
            ,stars
            ,neighborhood
            ,review_count
            from business
            where (city='Las Vegas' and stars >1.5 and stars!=3.5)
            order by stars))
where ratingclass= 'low_rating'
group by neighborhood
limit 25 --offset 25
```

neighborhood	count_neigh	ratingclass
	74	low_rating
	154	high_rating
Anthem	2	high_rating
Centennial	19	low_rating
Centennial	24	high_rating
Chinatown	13	low_rating
Chinatown	27	high_rating
Downtown	26	low_rating
Downtown	58	high_rating

```
| Eastside        |          35 | low_rating  |
| Eastside        |          46 | high_rating |
| Northwest       |          11 | low_rating  |
| Northwest       |          27 | high_rating |
| South Summerlin |           6 | low_rating  |
| South Summerlin |          10 | high_rating |
| Southeast       |          46 | low_rating  |
| Southeast       |          97 | high_rating |
| Southwest       |          12 | low_rating  |
| Southwest       |          48 | high_rating |
| Spring Valley   |          26 | low_rating  |
| Spring Valley   |         107 | high_rating |
| Summerlin       |          11 | low_rating  |
| Summerlin       |          32 | high_rating |
| Sunrise         |          13 | high_rating |
| Sunrise         |          13 | low_rating  |
+-----------------+-------------+-------------+

+--------------+-------------+-------------+
| neighborhood | count_neigh | ratingclass |
+--------------+-------------+-------------+
| The Lakes    |           3 | low_rating  |
| The Lakes    |           4 | high_rating |
| The Strip    |          56 | low_rating  |
| The Strip    |          68 | high_rating |
| University   |           4 | low_rating  |
| University   |          14 | high_rating |
| Westside     |          48 | low_rating  |
| Westside     |         107 | high_rating |
+--------------+-------------+-------------+
```

2. Group business based on the ones that are open and the ones that are close
d. What differences can you find between the ones that are still open and the
 ones that are closed? List at least two differences and the SQL code you use
d to arrive at your answer.

i. Difference 1: Amount of reviews is higher (almost ten times) in the busine
sses that are open:

```
-- As a first profiling, check the amount of reviews for businesses open and
not open
select sum(review_count) as tot_numrevs
,is_open
from business
group by is_open
```

```
+-------------+---------+
| tot_numrevs | is_open |
+-------------+---------+
|       35261 |       0 |
|      269300 |       1 |
+-------------+---------+
```

ii. Difference 2: The amount of stars rating given to open business is much higher than for not_opened ones. This, as the difference described above could be simply due to the fact that open businesses keep accumulating data while the not_opened ones stopped at some point.

Difference 3 (with the same code as Difference 2): The distribution of stars ratings seems to be slightly better for the open business, as 4-5 stars have the higher percentage of rating while for not_opened ones 3.5-4.5 have the higher percentages (see table of results)

SQL code used for analysis:

```
-- As a second profiling, check what is/was amount of rating given and the distribution of stars rating (in percentage) for the business open/not_open

Select is_open
,stars
,sum(stars) -- sum the # of stars given by grouping them
,tot_stars -- the tot # of stars given to open
, round(100*(sum(stars)/tot_stars)) as perc_stars --round the perc for better visualization
from business cross join
                          (select sum(stars) as tot_stars -- cross join with subquery that picks the tot # of stars for open shops
                           from business
                            where is_open=1)
where is_open=1
group by stars

Union  -- Make a union with extracted sum of stars and percentage for not_open business

Select is_open
,stars
,sum(stars) -- sum the # of stars given by grouping them
,tot_stars -- the tot # of stars given to open
, round(100*(sum(stars)/tot_stars)) as perc_stars --round the perc for better visualization
from business cross join
                        (select sum(stars) as tot_stars-- cross join with subquery that picks the tot # of stars for open shops
                         from business
                         where is_open=0)
where is_open=0
group by stars
```

is_open	stars	sum(stars)	tot_stars	perc_stars
0	1.0	14.0	5351.0	0.0
0	1.5	36.0	5351.0	1.0
0	2.0	188.0	5351.0	4.0
0	2.5	420.0	5351.0	8.0
0	3.0	816.0	5351.0	15.0
0	3.5	1032.5	5351.0	19.0
0	4.0	1304.0	5351.0	24.0

0	4.5	850.5	5351.0	16.0
0	5.0	690.0	5351.0	13.0
1	1.0	142.0	31198.0	0.0
1	1.5	273.0	31198.0	1.0
1	2.0	944.0	31198.0	3.0
1	2.5	1805.0	31198.0	6.0
1	3.0	3372.0	31198.0	11.0
1	3.5	5190.5	31198.0	17.0
1	4.0	6716.0	31198.0	22.0
1	4.5	5620.5	31198.0	18.0
1	5.0	7135.0	31198.0	23.0

3. For this last part of your analysis, you are going to choose the type of analysis you want to conduct on the Yelp dataset and are going to prepare the data for analysis.

Ideas for analysis include: Parsing out keywords and business attributes for sentiment analysis, clustering businesses to find commonalities or anomalies between them, predicting the overall star rating for a business, predicting the number of fans a user will have, and so on. These are just a few examples to get you started, so feel free to be creative and come up with your own problem you want to solve. Provide answers, in-line, to all of the following:

i. Indicate the type of analysis you chose to do:

I'll prepare the data to do clustering of businesses to find commonalities or anomalies. I'll select columns/tables I consider informative for this type of future analysis

ii. Write 1-2 brief paragraphs on the type of data you will need for your analysis and why you chose that data:

For this I will need business table:I'll extract id, neighborhood, latitude, longitude, stars and review_count either for open business as well as not open (is_opne column). I'll join this table to category (category column). I chose this data because I could analyse for ex: rate of open/closed business according to neighborhood/stars/review_count. I could also analyse what's the effect of been close to similar businesses (by extracting latitude/longitude) for example on star rating or rev_count. This could lead to an analysis of the kind: how beneficial is the clustering of restaurants as customers have more choice.

iii. Output of your finished dataset:

id	neigh	city	lat	long	stars	rev_count	is_open	cat

```
| -0DET7VdEQOJVJ_v6klEug | Brown's Corners | Markham             | 43.8484 |
-79.3487 |   3.0 |        25 |       1 | Asian Fusion            |
| -0DET7VdEQOJVJ_v6klEug | Brown's Corners | Markham             | 43.8484 |
-79.3487 |   3.0 |        25 |       1 | Restaurants             |
| -1H-8MO9uEyS9MGmPz3RQw |                 | Stuttgart-Vaihingen | 48.7264 |
 9.11306 |   2.0 |         4 |       1 | Transportation          |
| -1H-8MO9uEyS9MGmPz3RQw |                 | Stuttgart-Vaihingen | 48.7264 |
 9.11306 |   2.0 |         4 |       1 | Public Transportation   |
| -1H-8MO9uEyS9MGmPz3RQw |                 | Stuttgart-Vaihingen | 48.7264 |
 9.11306 |   2.0 |         4 |       1 | Hotels & Travel         |
| -1H-8MO9uEyS9MGmPz3RQw |                 | Stuttgart-Vaihingen | 48.7264 |
 9.11306 |   2.0 |         4 |       1 | Train Stations          |
| -1H-8MO9uEyS9MGmPz3RQw |                 | Stuttgart-Vaihingen | 48.7264 |
 9.11306 |   2.0 |         4 |       1 | Metro Stations          |
| -2bYV9zVtn2F5XpiAaHt5A |                 | Edinburgh           | 55.9526 |
-3.11324 |   3.0 |         4 |       1 | Restaurants             |
| -2bYV9zVtn2F5XpiAaHt5A |                 | Edinburgh           | 55.9526 |
-3.11324 |   3.0 |         4 |       1 | Delis                   |
| -2HjuT4yjLZ3b5f_abD87Q |                 | Charlotte           | 35.1727 |
-80.8755 |   3.5 |         8 |       1 | Electronics             |
| -2HjuT4yjLZ3b5f_abD87Q |                 | Charlotte           | 35.1727 |
-80.8755 |   3.5 |         8 |       1 | Shopping                |
| -2HjuT4yjLZ3b5f_abD87Q |                 | Charlotte           | 35.1727 |
-80.8755 |   3.5 |         8 |       1 | Automotive              |
| -2HjuT4yjLZ3b5f_abD87Q |                 | Charlotte           | 35.1727 |
-80.8755 |   3.5 |         8 |       1 | Car Stereo Installation |
| -2q4dnUw0gGJniGW2aPamQ |                 | Champaign           | 40.0941 |
-88.2458 |   2.0 |         4 |       0 | Restaurants             |
| -2q4dnUw0gGJniGW2aPamQ |                 | Champaign           | 40.0941 |
-88.2458 |   2.0 |         4 |       0 | Mexican                 |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Restaurants             |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Bars                    |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Italian                 |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Nightlife               |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Pizza                   |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Salad                   |
| -3oxnPPPU3YoxO9M1I2idg |                 | Mesa                | 33.3799 |
-111.806 |   4.0 |       129 |       0 | Gluten-Free             |
| -49WY_TEa9ZEcRk_GnuLog |                 | Sheffield Village   | 41.4259 |
 -82.081 |   3.5 |        27 |       1 | American (Traditional)  |
| -49WY_TEa9ZEcRk_GnuLog |                 | Sheffield Village   | 41.4259 |
 -82.081 |   3.5 |        27 |       1 | Restaurants             |
| -49WY_TEa9ZEcRk_GnuLog |                 | Sheffield Village   | 41.4259 |
 -82.081 |   3.5 |        27 |       1 | Southern                |
+------------------------+-----------------+---------------------+---------+-
---------+-------+-----------+---------+-------------------------+
(Output limit exceeded, 25 of 696 total rows shown)
```

iv. Provide the SQL code you used to create your final dataset:

```
Select b.id
,b.neighborhood
,b.city
,b.latitude
,b.longitude
,b.stars
,b.review_count
,b.is_open
,c.category
from business as b inner join category as c
on b.id=c.business_id
```