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 = 10000
ii. Business table = 10000
iii. Category table = 10000
iv. Checkin table = 10000
v. elite_years table = 10000
vi. friend table = 10000
vii. hours table = 10000
viii. photo table = 10000
ix. review table = 10000
x. tip table = 10000
xi. user table = 10000

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

i. Business = 10000
ii. Hours = 1562

iii. Category = 2643
iv. Attribute = 1115
v. Review = 10000
vi. Checkin = 493
vii. Photo = 10000
viii. Tip = 537 (user_id)
ix. User = 10000
x. Friend = 11
xi. Elite_years = 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 COUNT(*)

FROM user
WHERE
id IS NULL OR
name IS NULL OR
review_count IS NULL OR
yelping_since IS NULL OR
useful IS NULL OR
funny IS NULL OR
cool IS NULL OR
fans IS NULL OR
average_stars IS NULL OR
compliment_hot IS NULL OR
compliment_more IS NULL OR
compliment_profile IS NULL OR
compliment_cute IS NULL OR
compliment_list IS NULL OR
compliment_note IS NULL OR
compliment_plain IS NULL OR
compliment_cool IS NULL OR
compliment_funny IS NULL OR
compliment_writer IS NULL OR
compliment_photos IS NULL
+----------+
| COUNT(*) |
+----------+
|        0 |
+----------+
```

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

i. Table: Review, Column: Stars

min: 1 max: 5 avg: 3.7082

ii. Table: Business, Column: Stars

min: 1 max: 5 avg: 3.6549

iii. Table: Tip, Column: Likes

min: 0 max: 2 avg: 0.0144

iv. Table: Checkin, Column: Count

min: 1 max: 53 avg: 1.9414

v. Table: User, Column: Review_count

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:

```
SELECT city,SUM(review_count) AS review_num
FROM business
GROUP BY city
ORDER BY review_num DESC
```

Copy and Paste the Result Below:

city	review_num
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

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

i. Avon

SQL code used to arrive at answer:

```
SELECT stars AS star_rating,COUNT(stars) AS count
FROM business
WHERE city = "Avon"
GROUP BY stars
```

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

star_rating	count
1.5	1
2.5	2
3.5	3
4.0	2
4.5	1
5.0	1

ii. Beachwood

SQL code used to arrive at answer:

```
SELECT stars AS star_rating,COUNT(stars) AS count
FROM business
WHERE city = "Beachwood"
GROUP BY stars
```

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

star_rating	count
2.0	1
2.5	1
3.0	2
3.5	2
4.0	1
4.5	2
5.0	5

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

SQL code used to arrive at answer:

```
SELECT id,name,review_count AS total_review_num
FROM user
ORDER BY  total_review_num DESC
LIMIT 3
```

id	name	total_review_num
-G7Zkl1wIWBBmD0KRy_sCw	Gerald	2000
-3s52C4zL_DHRK0ULG6qtg	Sara	1629
-8lbUNlXVSoXqaRRiHiSNg	Yuri	1339

8. Does posing more reviews correlate with more fans?
Yes. There is a positive correlation between two variables.

Please explain your findings and interpretation of the results:

In order to answer this question, we need to calculate Pearson correlation coefficient, but SQLite doesn't provide build-in function. So, we can do something like this:

```
SELECT avg( (review_count - avg_x) * (fans - avg_y) )*avg( (review_count - avg_
x) * (fans - avg_y) )/(var_x*var_y) as R2
FROM user, (SELECT
      avg_x,
      avg_y,
      avg((review_count - avg_x)*(review_count - avg_x)) as var_x,
      avg((fans - avg_y)*(fans - avg_y)) as var_y
      FROM user, (select
          avg(review_count) as avg_x,
          avg(fans) as avg_y
          FROM user)
  );
```

```
+----------------+
|             R2 |
+----------------+
| 0.437136492915 |
+----------------+
```

Or

```
SELECT fans_num,
       COUNT(*) AS user_num,
       AVG(review_count) AS avg_review_num,
       AVG(fans) AS avg_fans_num
FROM (SELECT CASE
             WHEN fans <=10 THEN "0-10"
             WHEN fans  BETWEEN 11 AND 100 THEN "11-100"
             ELSE ">100"
             END fans_num,
             review_count,
             fans
     FROM user) AS table_1
GROUP BY  fans_num
```

```
+----------+----------+----------------+----------------+
| fans_num | user_num | avg_review_num |   avg_fans_num |
+----------+----------+----------------+----------------+
| 0-10     |     9722 |  15.4956799013 | 0.478708084756 |
| 11-100   |      262 |  298.022900763 |  27.5038167939 |
| >100     |       16 |          891.5 |         189.75 |
+----------+----------+----------------+----------------+
```

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

Answer: “love”

SQL code used to arrive at answer:

```
SELECT CASE
       WHEN LOWER(text) LIKE "%love%" THEN "love"
       WHEN LOWER(text) LIKE "%hate%" THEN "hate"
       ELSE "no_such_category"
       END category,
       COUNT(text)
FROM review
GROUP BY category
```

```
+------------------+-------------+
| category         | COUNT(text) |
+------------------+-------------+
| hate             |         178 |
| love             |        1780 |
| no_such_category |        8042 |
+------------------+-------------+
```

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

SQL code used to arrive at answer:

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

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.

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

```
SELECT CASE WHEN stars BETWEEN 2 AND 3 THEN 'Group 2-3 stars'
            WHEN stars BETWEEN 4 AND 5 THEN 'Group 4-5 stars'
            ELSE 'other'
            END 'Stars category', --
divide business into categories based on their ratings and create a new c
ategorical variable
        COUNT(DISTINCT b.id) AS count,--
count the number of distinct businnes
        COUNT(hours) AS days_open,--
count the number of entries in hours variable
        COUNT(hours)/COUNT(DISTINCT b.id) AS open_days_aver--
finding the average number of days opened for each distinct businnes
FROM business AS b INNER JOIN hours AS h ON b.id = h.business_id
WHERE city = 'Las Vegas'
GROUP BY stars
```

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

```
SELECT CASE WHEN stars BETWEEN 2 AND 3 THEN 'Group 2-3 stars'
            WHEN stars BETWEEN 4 AND 5 THEN 'Group 4-5 stars'
            ELSE 'other'
            END 'Stars category',
             --
divide business into categories based on their ratings and create a new c
ategorical variable
        SUM(review_count)--final number of reviews for each group
FROM business
WHERE city = 'Las Vegas'
GROUP BY stars
```

```
+-----------------+-------------------+
| Stars category  | SUM(review_count) |
+-----------------+-------------------+
| other           |               135 |
| other           |               896 |
| Group 2-3 stars |              1855 |
| Group 2-3 stars |              4846 |
| Group 2-3 stars |              8564 |
| other           |             19606 |
| Group 4-5 stars |             25128 |
| Group 4-5 stars |             17803 |
| Group 4-5 stars |              4021 |
+-----------------+-------------------+
```

Group with 2-3 star rating has less reviews than group with 4-5 star rating.

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

SQL code used for analysis:

```
SELECT neighborhood,stars

FROM business
WHERE city = 'Las Vegas'
GROUP BY stars
HAVING stars BETWEEN 2 AND 5
```

```
---------------+-------+
| neighborhood  | stars |
+---------------+-------+
|               |   2.0 |
| Chinatown     |   2.5 |
| Westside      |   3.0 |
| Summerlin     |   3.5 |
| Downtown      |   4.0 |
| Spring Valley |   4.5 |
| Spring Valley |   5.0 |
+---------------+-------+
```

Businesses with 4-5 star rating are located in different neighborhood than businesses with 2-3 star rating.

2. Group business based on the ones that are open and the ones that are closed. 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 used to arrive at your answer.

i. Difference 1: There are more open businesses than closed ones.

ii. Difference 2: Open businesses have significantly more reviews and a higher star rating.

SQL code used for analysis:

```
SELECT is_open,
       COUNT(DISTINCT b.id) AS num_business ,
       COUNT(DISTINCT r.id) AS num_reviews,
       AVG(r.stars)
FROM business AS b  JOIN review AS r ON b.id = r.business_id
GROUP BY is_open
```

```
+---------+--------------+-------------+---------------+
| is_open | num_business | num_reviews |  AVG(r.stars) |
+---------+--------------+-------------+---------------+
|       0 |           61 |          71 | 3.64788732394 |
|       1 |          446 |         565 |  3.7610619469 |
+---------+--------------+-------------+---------------+
```

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'm going to find out what categories of business are successful.

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

iii. I'm going to calculate proportion of open businesses, also numbers of business in each category and average star rating category got.

iii. Output of your finished dataset:

category	num_business	business_open	avg_star
Health & Medical	17	0.94	4.09
Home Services	16	0.94	4.0
Beauty & Spas	13	0.92	3.88
Food	23	0.87	3.78
Local Services	12	0.83	4.21
Shopping	30	0.83	3.98
Restaurants	71	0.75	3.46
American (Traditional)	11	0.73	3.82
Bars	17	0.65	3.5
Nightlife	20	0.6	3.48

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

```sql
SELECT category,
       COUNT(b.id) AS num_business,
       ROUND(AVG(b.is_open),2) AS business_open,
       ROUND(AVG(b.stars),2) AS avg_star

FROM business AS b
JOIN category AS c ON b.id = c.business_id

GROUP BY category
HAVING num_business > 10
ORDER BY business_open DESC
```