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:

```
SELECT COUNT(*)
FROM Table
```

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.

```
SELECT COUNT(DISTINCT(key))
       FROM table
```

i. Business = id: 10000
ii. Hours = business_id: 1562
iii. Category = business_id: 2643
iv. Attribute = business_id: 1115

vii. Photo = id:10000 business_id:6493
viii. Tip = user_id: 537 business_id: 3979
ix. User = id: 10000
x. Friend = user_id: 11
xi. Elite_years = user_id: 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
```

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

```
SELECT  AVG(column)
FROM table
```

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 Reviews
FROM Business
GROUP BY City
ORDER BY Reviews DESC
```

Copy and Paste the Result Below:

```
+-----------------+---------+
| city            | Reviews |
+-----------------+---------+
| 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:

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

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

```
+-------+-------+
| stars | count |
+-------+-------+
|   1.5 |    10 |
|   2.5 |     6 |
|   3.5 |    88 |
|   4.0 |    21 |
|   4.5 |    31 |
|   5.0 |     3 |
+-------+-------+
```

ii. Beachwood

SQL code used to arrive at answer:

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

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

| stars | count |
|---|---|
| 2.0 | 8 |
| 2.5 | 3 |
| 3.0 | 11 |
| 3.5 | 6 |
| 4.0 | 69 |
| 4.5 | 17 |
| 5.0 | 23 |

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

SQL code used to arrive at answer:

```
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:

Based on the table below of users organized by number of fans from greatest to least. You can see there is not necessarily a positive correlation between the two variables. To double check we can order by review count descending aslo, and again find no correlation.

| name | review_count | fans |
|---|---|---|
| Amy | 609 | 503 |
| Mimi | 968 | 497 |
| Harald | 1153 | 311 |
| Gerald | 2000 | 253 |
| Christine | 930 | 173 |
| Lisa | 813 | 159 |

| Cat | 377 | 133 |
| --- | --- | --- |
| William | 1215 | 126 |
| Fran | 862 | 124 |
| Lissa | 834 | 120 |
| Mark | 861 | 115 |
| Tiffany | 408 | 111 |
| bernice | 255 | 105 |
| Roanna | 1039 | 104 |
| Angela | 694 | 101 |
| .Hon | 1246 | 101 |
| Ben | 307 | 96 |
| Linda | 584 | 89 |
| Christina | 842 | 85 |
| Jessica | 220 | 84 |
| Greg | 408 | 81 |
| Nieves | 178 | 80 |
| Sui | 754 | 78 |
| Yuri | 1339 | 76 |
| Nicole | 161 | 73 |

(Output limit exceeded, 25 of 10000 total rows shown)

```
SELECT
name, review_count, fans
FROM user
ORDER BY fans DESC
```

| 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 |

```
| Dominic   |          836 |   37 |
| Lissa     |          834 |  120 |
| Lisa      |          813 |  159 |
| Alison    |          775 |   61 |
| Sui       |          754 |   78 |
| Tim       |          702 |   35 |
| L         |          696 |   10 |
| Angela    |          694 |  101 |
| Crissy    |          676 |   25 |
| Lyn       |          675 |   45 |
+-----------+--------------+------+
(Output limit exceeded, 25 of 10000 total rows shown)
```

```
SELECT
name, review_count, fans
FROM user
ORDER BY review_count DESC
```

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

Answer: Love.

"Love" is used 1780 times while "hate" is only used 232

SQL code used to arrive at answer:

```
SELECT COUNT(*) AS Love
FROM review
WHERE text LIKE '%love%'

SELECT COUNT(*) AS Hate
FROM review
WHERE text LIKE '%hate%'
```

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.

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

Not Significant, for this exercise I chose Las Vegas and Shopping. Compared to businesses with 2-3 stars the 4-5 star group has nearly the same amount of working days: 12 (4-5 stars) and 13 (2-3 stars)

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

Absolutely, the 4-5 star group has more than double the number of reviews.
4-5 stars: 244
2-3 stars: 108

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

Businesses with 2-3 stars are located in the same area based on the postal code returned of 89121.

SQL code used for analysis:

i & ii

```sql
SELECT CASE
  WHEN stars >= 4 THEN "4-5 stars"
  WHEN stars >= 2 THEN "2-3 stars"
  ELSE "below 2"
  END star_rank,
  city,
  c.category,
  sum(review_count) AS reviews,
  count(distinct business.id) AS company_count,
  count(h.hours) AS working_days
FROM business
JOIN hours h ON business.id = h.business_id
```

```
JOIN category c ON business.id = c.business_id
WHERE city = "Las Vegas" AND c.category = "Shopping"
GROUP BY star_rank
```

iii

```
SELECT CASE
  WHEN stars >= 4 THEN "4-5 stars"
  WHEN stars >= 2 THEN "2-3 stars"
  ELSE "below 2"
  END star_rank,
address,
neighborhood,
city,
postal_code
FROM business
JOIN category c ON business.id = c.business_id
WHERE city = "Las Vegas" AND c.category = "Shopping"
ORDER BY star_rank
```

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:

The overall number of businesses still open is much higher than businesses that have closed 446/61

ii. Difference 2:

The average star rating for businesses still open is ~0.12 higher than businesses that have closed

SQL code used for analysis:

```
SELECT is_open,
      count(distinct business.id) AS businesses,
      count(distinct review.id) AS reviews,
      avg(review.stars) AS average_rating
FROM business
JOIN review ON business.id =  review.business_id
GROUP BY is_open
```

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:

Finding the best cities for shopping.

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 analysis I want to first have the city and state to identify the area. Next, I will use average star ratings of reviews for shopping in each city as the primary criteria for evaluation. I will also take into consideration the total number of open businesses in the Shopping category for city to rule out any closed businesses.
Total number reviews will go in to consideration as well to help boost the validity of the reviews.

iii. Output of your finished dataset:

| city | state | avg_stars | reviews | open_businesses |
|---|---|---|---|---|
| Middleton | WI | 5.0 | 8 | 1 |
| Pittsburgh | PA | 5.0 | 8 | 1 |
| Chandler | AZ | 4.5 | 19 | 3 |
| Cleveland | OH | 4.5 | 723 | 1 |
| Scarborough | ON | 4.5 | 3 | 1 |
| Toronto | ON | 4.375 | 63 | 4 |
| Scottsdale | AZ | 4.0 | 20 | 1 |
| Phoenix | AZ | 4.0 | 18 | 1 |
| Tempe | AZ | 4.0 | 14 | 1 |
| Strongsville | OH | 4.0 | 3 | 1 |
| Las Vegas | NV | 3.875 | 53 | 3 |
| Charlotte | NC | 3.66666666667 | 19 | 3 |
| Mississauga | ON | 3.5 | 10 | 1 |
| Edinburgh | EDH | 3.5 | 6 | 1 |
| Stuttgart | BW | 3.5 | 3 | 1 |
| Gilbert | AZ | 2.0 | 4 | 1 |
| Mesa | AZ | 2.0 | 3 | 0 |

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

```sql
SELECT
   city, state,
   AVG(stars) AS avg_stars,
   SUM(review_count) AS reviews,
   SUM(is_open) AS open_businesses
FROM business
   JOIN category ON id = business_id
WHERE category = 'Shopping'
GROUP BY city
ORDER BY AVG(stars) DESC, SUM(is_open) DESC, SUM(review_count) DESC;
```