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

```
1. SELECT
2. count(*) as answer
3. FROM attribute;
```

All other queries for this task are almost the same.

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 = id: 10000

Example of query:

```
1. SELECT
2. count(DISTINCT(id))
3. FROM business;
```

All other ones are analogical.

ii. Hours = business_id: 1562
iii. Category = business_id: 2643
iv. Attribute = business_id: 1115
v. Review = id: 10000, business_id: 8090, user_id: 9581
vi. Checkin = business_id: 493
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:

SQL code used to arrive at answer:

```
1.        SELECT COUNT(*)
2.          FROM user
3.          WHERE id IS NULL OR
4.            name IS NULL OR
5.            review_count IS NULL OR
6.            yelping_since IS NULL OR
7.            useful IS NULL OR
8.            funny IS NULL OR
9.            cool IS NULL OR
10.           fans IS NULL OR
11.           average_stars IS NULL OR
12.           compliment_hot IS NULL OR
```

```
13.           compliment_more IS NULL OR
14.           compliment_profile IS NULL OR
15.           compliment_cute IS NULL OR
16.           compliment_list IS NULL OR
17.           compliment_note IS NULL OR
18.           compliment_plain IS NULL OR
19.           compliment_cool IS NULL OR
20.           compliment_funny IS NULL OR
21.           compliment_writer IS NULL OR
22.           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:

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:

```
1.       SELECT
2.  city,
3.  SUM(review_count) as number_of_reviews
4.  FROM business
5.  GROUP BY city
6.  order by number_of_reviews desc;
```

Copy and Paste the Result Below:

```
+-----------------+-------------------+
| city            | number_of_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:

```sql
SELECT
stars,
SUM(review_count) as number_of_reviews
FROM business
WHERE city == 'Avon'
GROUP BY stars;
```

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

```
+-------+--------------------+
| stars | number_of_reviews |
+-------+--------------------+
|   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:

```sql
SELECT
```

```
2. stars,
3. SUM(review_count) as number_of_reviews
4. FROM business
5. WHERE city == 'Beachwood'
6. GROUP BY stars;
```

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

```
+-------+------------------+
| stars | number_of_reviews |
+-------+------------------+
|   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:

```
1. SELECT
2. name,
3. review_count as number_of_reviews
4. FROM user
5. ORDER BY number_of_reviews desc
6. LIMIT 3;
```

Copy and Paste the Result Below:

```
+--------+------------------+
| name   | number_of_reviews |
+--------+------------------+
| Gerald |             2000 |
| Sara   |             1629 |
| Yuri   |             1339 |
+--------+------------------+
```

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

There is a certain dependency, but number of reviews does not seem to be the most significant factor.

```
+------------------------+--------------+------+
| id                     | review_count | fans |
+------------------------+--------------+------+
| -9I98YbNQnLdAmcYfb324Q |          609 |  503 |
| -8EnCioUmDygAbsYZmTeRQ |          968 |  497 |
| --2vR0DIsmQ6WfcSzKWigw |         1153 |  311 |
| -G7Zkl1wIWBBmD0KRy_sCw |         2000 |  253 |
| -0IiMAZI2SsQ7VmyzJjokQ |          930 |  173 |
| -g3XIcCb2b-BD0QBCcq2Sw |          813 |  159 |
| -9bbDysuiWeo2VShFJJtcw |          377 |  133 |
| -FZBTkAZEXoP7CYvRV2ZwQ |         1215 |  126 |
```

```
| -9da1xk7zgnnfO1uTVYGkA |          862 |  124 |
| -lh59ko3dxChBSZ9U7LfUw |          834 |  120 |
| -B-QEUESGWHPE_889WJaeg |          861 |  115 |
| -DmqnhW4Omr3YhmnigaqHg |          408 |  111 |
| -cv9PPT7IHux7XUc9dOpkg |          255 |  105 |
| -DFCC64NXgqrxlO8aLU5rg |         1039 |  104 |
| -IgKkE8JvYNWeGu8ze4P8Q |          694 |  101 |
| -K2Tcgh2EKX6e6HqqIrBIQ |         1246 |  101 |
| -4viTt9UC44lWCFJwleMNQ |          307 |   96 |
| -3i9bhfvrM3F1wsC9XIB8g |          584 |   89 |
| -kLVfaJytOJY2-QdQoCcNQ |          842 |   85 |
| -ePh4Prox7ZXnEBNGKyUEA |          220 |   84 |
| -4BEUkLvHQntN6qPfKJP2w |          408 |   81 |
| -C-l8EHSLXtZZVfUAUhsPA |          178 |   80 |
| -dw8f7FLaUmWR7bfJ_Yf0w |          754 |   78 |
| -8lbUNlXVSoXqaRRiHiSNg |         1339 |   76 |
| -0zEEaDFIjABtPQni0XlHA |          161 |   73 |
+------------------------+--------------+------+
```

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

Answer:

There are 1780 reviews with word love, and there are only 232 reviews with word hate, which means that there are more with ‘love’.

SQL code used to arrive at answer:

Code number 1:

```sql
SELECT
COUNT(*) as number_of_love_reviews
FROM review
WHERE text LIKE '%love%';
```

Code number 2:

```sql
SELECT
COUNT(*) as number_of_hate_reviews
FROM review
WHERE text LIKE '%hate%';
```

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

SQL code used to arrive at answer:

```sql
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?

4-5 stars Bars tend to open significantly later, so there is difference in distribution, but sample size is still quite small.

ii. Do the two groups you chose to analyze have a different number of reviews?
4-5 star bars have a little less review, but it still might be distortion because of small sample size.

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

Postal code is different and sample size is small, so I don't think it is possible to make any assumptions.

SQL code used for analysis:

```sql
SELECT
B.name,
B.review_count,
H.hours,
postal_code,
CASE
    WHEN hours LIKE "%monday%" THEN 1
    WHEN hours LIKE "%tuesday%" THEN 2
    WHEN hours LIKE "%wednesday%" THEN 3
    WHEN hours LIKE "%thursday%" THEN 4
    WHEN hours LIKE "%friday%" THEN 5
    WHEN hours LIKE "%saturday%" THEN 6
    WHEN hours LIKE "%sunday%" THEN 7
END AS work_day,
CASE
    WHEN B.stars BETWEEN 2 AND 3 THEN '2-3 stars'
    WHEN B.stars BETWEEN 4 AND 5 THEN '4-5 stars'
END AS star_rating
FROM business B INNER JOIN hours H ON B.id = H.business_id
```

```
INNER JOIN category C ON C.business_id = B.id
WHERE (B.city == 'Toronto' AND C.category LIKE 'Bars') AND
(B.stars BETWEEN 2 AND 3 OR B.stars BETWEEN 4 AND 5)
GROUP BY
stars,
work_day
ORDER BY
work_day,
star_rating ASC;
```

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:

Closed businesses are significantly less reviewed.

ii. Difference 2:

Closed businesses have less stars.

SQL code used for analysis:

```
SELECT
B.name,
B.review_count,
H.hours,
postal_code,
stars,
CASE
    WHEN hours LIKE "%monday%" THEN 1
    WHEN hours LIKE "%tuesday%" THEN 2
    WHEN hours LIKE "%wednesday%" THEN 3
    WHEN hours LIKE "%thursday%" THEN 4
    WHEN hours LIKE "%friday%" THEN 5
    WHEN hours LIKE "%saturday%" THEN 6
    WHEN hours LIKE "%sunday%" THEN 7
END AS work_day,
CASE
    WHEN B.is_open == 0 THEN 'closed'
    WHEN B.is_open == 1 THEN 'open'
END AS closure
FROM business B INNER JOIN hours H ON B.id = H.business_id
INNER JOIN category C ON C.business_id = B.id
GROUP BY
stars,
work_day
ORDER BY
work_day ASC;
```

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:

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

I'm going to predict the amount of fans for certain user. I guess features that I need are number of reviews, time of yelping and tags for his reviews (cool, funny, etc).
Target feature is fans.

iii. Output of your finished dataset:

fans	name	review_count	years_on_yelp	useful	funny	cool
15	Monera	245	13	67	22	9
0	Joe	2	4	0	0	0
0	Rae	2	5	1	0	0
0	Jeb	57	11	34	14	0
0	Jed	8	9	2	3	1
1	Carolyn	43	6	1	0	0
0	Ryan	2	7	0	0	0
0	Scott	7	7	0	0	0
2	Talia	26	10	10	2	0
0	Joe	1	7	0	0	0
0	John	3	4	0	0	2
0	Ron	9	10	0	0	0
0	Bryan	5	9	0	0	0
0	Patti	2	6	15	13	9
0	Kristin	28	4	7	1	0
0	Gary	23	5	0	0	0
311	Harald	1153	8	122921	122419	122890
0	Cynthia	4	4	0	0	0
10	Kristie	213	10	63	6	2
2	Benjamin	111	7	97	57	32
0	Mrme	2	10	1	0	0
23	Tamaki	239	9	64	15	3
0	Austin	2	7	0	0	0
23	Kiristen	400	12	405	313	72
0	Mesut	25	7	12	5	1

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

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

```sql
SELECT
fans,
name,
review_count,
strftime('%Y', 'now') - strftime('%Y', yelping_since) as years_on_yelp,
useful,
funny,
cool
FROM user
ORDER BY id asc;
```