# 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

```
SELECT COUNT(*)
FROM attribute
```

ii. Business table = 10000

```
SELECT COUNT(*)
FROM business
```

iii. Category table = 10000

```
SELECT COUNT(*)
FROM category
```

iv. Checkin table = 10000

```
SELECT COUNT(*)
FROM checkin
```

v. elite_years table = 10000

```
SELECT COUNT(*)
FROM elite_years
```

vi. friend table = 10000

```
SELECT COUNT(*)
FROM friend
```

vii. hours table = 10000

```
SELECT COUNT(*)
FROM hours
```

viii. photo table = 10000

```
SELECT COUNT(*)
```

```
FROM photo
```

ix. review table = 10000

```
SELECT COUNT(*)
FROM review
```

x. tip table = 10000

```
SELECT COUNT(*)
FROM tip
```

xi. user table = 10000

```
SELECT COUNT(*)
FROM user
```

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

```
SELECT COUNT(DISTINCT(id))
FROM business
```

ii. Hours = business_id:1562

```
SELECT COUNT(DISTINCT(business_id))
FROM hours
```

iii. Category = business_id:2643

```
SELECT COUNT(DISTINCT(business_id))
FROM category
```

iv. Attribute = business_id:1115

```
SELECT COUNT(DISTINCT(business_id))
FROM attribute
```

v. Review = id: 10000, business_id: 8090, user_id: 9581

```
SELECT COUNT(DISTINCT(id)), COUNT(DISTINCT(business_id)), COUNT(DISTINCT(user_id))
FROM review
```

vi. Checkin = business_id:493

```
SELECT COUNT(DISTINCT(business_id))
FROM checkin
```

vii. Photo = id:10000 business_id:6493

```
SELECT COUNT(DISTINCT(id)), COUNT(DISTINCT(business_id))
FROM photo
```

viii. Tip = user_id:537 business_id:3979

```
SELECT COUNT(DISTINCT(user_id)), COUNT(DISTINCT(business_id))
FROM tip
```

ix. User = id:10000

```
SELECT COUNT(DISTINCT(id))
FROM user
```

x. Friend = user_id:11

```
SELECT COUNT(DISTINCT(user_id))
```

```
FROM friend
```

xi. Elite_years = user_id:2780

```
SELECT COUNT(DISTINCT(user_id))
FROM elite_years
```

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:

    i. Table: Review, Column: Stars

```
        min:    1   max:    5   avg: 3.7082

        SELECT MIN(stars), MAX(stars), AVG(stars)
        FROM review
```

    ii. Table: Business, Column: Stars

```
        min: 1.0    max: 5.0    avg: 3.6549

        SELECT MIN(stars), MAX(stars), AVG(stars)
        FROM business
```

    iii. Table: Tip, Column: Likes

```
        min:    0   max:    2   avg:  0.0144

        SELECT MIN(likes), MAX(likes), AVG(likes)
        FROM tip
```

    iv. Table: Checkin, Column: Count

```
        min:    1   max:    53  avg: 1.9414
        SELECT MIN(count), MAX(count), AVG(count)
        FROM checkin
```

v. Table: User, Column: Review_count

```
        min:    0   max:    2000    avg:  24.2995

        SELECT MIN(review_count), MAX(review_count), AVG(review_count)
        FROM user
```

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

SQL code used to arrive at answer:

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

Copy and Paste the Result Below:

```
    +-------------------+-----------------+
| SUM(review_count) | city            |
+-------------------+-----------------+
|             82854 | Las Vegas       |
|             34503 | Phoenix         |
|             24113 | Toronto         |
|             20614 | Scottsdale      |
|             12523 | Charlotte       |
|             10871 | Henderson       |
|             10504 | Tempe           |
|              9798 | Pittsburgh      |
|              9448 | Montréal        |
|              8112 | Chandler        |
|              6875 | Mesa            |
|              6380 | Gilbert         |
|              5593 | Cleveland       |
|              5265 | Madison         |
|              4406 | Glendale        |
|              3814 | Mississauga     |
|              2792 | Edinburgh       |
|              2624 | Peoria          |
|              2438 | North Las Vegas |
|              2352 | Markham         |
|              2029 | Champaign       |
|              1849 | Stuttgart       |
|              1520 | Surprise        |
|              1465 | Lakewood        |
|              1155 | Goodyear        |
+-------------------+-----------------+
(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 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):

```
+-------------------+-------+
| SUM(review_count) | stars |
+-------------------+-------+
|                10 |   1.5 |
|                 6 |   2.5 |
|                88 |   3.5 |
|                21 |   4.0 |
|                31 |   4.5 |
|                 3 |   5.0 |
+-------------------+-------+
```

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):

```
+-------------------+-------+
| SUM(review_count) | stars |
+-------------------+-------+
|                 8 |   2.0 |
|                 3 |   2.5 |
|                11 |   3.0 |
|                 6 |   3.5 |
|                69 |   4.0 |
|                17 |   4.5 |
|                23 |   5.0 |
+-------------------+-------+
```

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
FROM user
ORDER BY review_count DESC
LIMIT 3
```

Copy and Paste the Result Below:

```
+------------------------+--------+--------------+
| id                     | name   | review_count |
+------------------------+--------+--------------+
| -G7Zkl1wIWBBmD0KRy_sCw | Gerald |         2000 |
| -3s52C4zL_DHRK0ULG6qtg | Sara   |         1629 |
| -8lbUNlXVSoXqaRRiHiSNg | Yuri   |         1339 |
+------------------------+--------+--------------+
```

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

Posing more reviews does not correlate with more fans. Amy has the most fans with the number of 503 and she has 609 reviews. Yuri has only 76 fans while he has 1339 reviews. Jeb has 0 fans while he has 57 reviews.

I used 2 SQL codes for this analysis.

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

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

```
+---------+--------------+------+
| name    | review_count | fans |
+---------+--------------+------+
| Joe     |            2 |    0 |
| Jeb     |           57 |    0 |
| Jed     |            8 |    0 |
| Rae     |            2 |    0 |
| Ryan    |            2 |    0 |
| Joe     |            1 |    0 |
| Scott   |            7 |    0 |
| John    |            3 |    0 |
| Ron     |            9 |    0 |
| Bryan   |            5 |    0 |
| Patti   |            2 |    0 |
| Gary    |           23 |    0 |
| Kristin |           28 |    0 |
| Cynthia |            4 |    0 |
| Mrme    |            2 |    0 |
| Austin  |            2 |    0 |
| Mesut   |           25 |    0 |
| Lissa   |            3 |    0 |
| Tara    |            3 |    0 |
| Lyndsey |            1 |    0 |
| Annie   |           11 |    0 |
| Daniece |            2 |    0 |
| Alex    |            7 |    0 |
| Mary    |            2 |    0 |
| Garen   |            3 |    0 |
+---------+--------------+------+
(Output limit exceeded, 25 of 10000 total rows shown)
```

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

Answer:

Yes. There are more rewievs with the word "love" than with the word "hate". There are 1780 reviews with the word 'love' and 232 reviews with the word 'hate'.

SQL code used to arrive at answer:

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

SELECT COUNT(*)
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 |
+-----------+------+
```

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:

```
SELECT name, fans, useful, funny
FROM user
ORDER BY fans DESC
LIMIT 10
```

Copy and Paste the Result Below:

```
+-----------+------+--------+--------+
| name      | fans | useful |  funny |
+-----------+------+--------+--------+
| Amy       |  503 |   3226 |   2554 |
| Mimi      |  497 |    257 |    138 |
| Harald    |  311 | 122921 | 122419 |
| Gerald    |  253 |  17524 |   2324 |
| Christine |  173 |   4834 |   6646 |
| Lisa      |  159 |     48 |     13 |
| Cat       |  133 |   1062 |    672 |
| William   |  126 |   9363 |   9361 |
| Fran      |  124 |   9851 |   7606 |
| Lissa     |  120 |    455 |    150 |
+-----------+------+--------+--------+
```

Please explain your findings and interpretation of the results:

All of the top 10 users with the highest number of fans are also listed as “useful” and “funny”. I believe there is a strong correlation (76% - 100% - Strong relationship) between having a high number of fans and 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?

I analyzed Las Vegas and Food on this question. Yes, two groups have different distribution of hours. The food place with 2.5 stars open between 8:00-22:00 on Saturday while the food place with higher rating 4.0 opens late on Saturday.

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

I analyzed Las Vegas and Food on this question. Yes, two groups have different number of reviews. The food place with 2.5 stars have 6 reviews while the food place with higher rating 4.0 have 30 reviews.

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

The food places in two different groups are located in different postal codes.

SQL code used for analysis:

```
SELECT business.name, business.city, category.category, business.stars ,hours.hours, business.review_count, business.address, business.postal_code
FROM (business INNER JOIN category ON business.id =
category.business_id) INNER JOIN hours ON hours.business_id =
business.id
WHERE business.city = 'Las Vegas' AND category.category = "Food"
GROUP BY business.stars;
```

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 businesses that are open have more reviews on average than the businesses that are closed.

ii. Difference 2:have more reviews on average than

The businesses that are open listed as 'funny' compared to the businesses that are closed.

```
+---------------------+----------------------------+----------------------------+---------------------+---------+
| AVG(business.stars) | SUM(business.review_count) | AVG(business.review_count) | COUNT(review.funny) | is_open |
+---------------------+----------------------------+----------------------------+---------------------+---------+
|       3.52039473684 |                      35261 |              23.1980263158 |                   1 |       0 |
|       3.67900943396 |                     269300 |              31.7570754717 |                  13 |       1 |
+--------------------+----------------------------+----------------------------+---------------------+---------+
```

SQL code used for analysis:

```
SELECT AVG(business.stars),SUM(business.review_count),AVG(business.review_count),COUNT(review.funny), business.is_open
FROM business
LEFT JOIN review
ON business.id = review.id
GROUP BY business.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:

Comparison of the average number of reviews and stars of the restaurants among different cities.

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

Getting a decision to open a restaurant in one of the cities used in the analysis. The city which has lowest average number of reviews and stars would be a target city to open a restaurant.

iii. Output of your finished dataset:

```
     +-----------+-------------+----------------------------+---------------------+
| city      | category    | AVG(business.review_count) | AVG(business.stars) |
+-----------+-------------+----------------------------+---------------------+
| Charlotte | Restaurants |                        5.5 |                4.25 |
| Las Vegas | Restaurants |                      265.5 |               3.875 |
| Phoenix   | Restaurants |              126.166666667 |                 3.5 |
| Tempe     | Restaurants |                        5.0 |                 2.5 |
| Toronto   | Restaurants |                       29.9 |                 3.4 |
+-----------+-------------+----------------------------+---------------------+
```

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

```
SELECT business.city, category.category, AVG(business.review_count), AVG(business.stars)
FROM business
LEFT JOIN category
ON category.business_id=business.id
WHERE business.city IN ('Phoenix', 'Toronto', 'Charlotte', 'Las Vegas') AND category='Restaurants'
GROUP BY business.city
```