

## 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
-- SQL Code
SELECT *
FROM attribute;
```

```
ii. Business table = 10000
-- SQL Code
SELECT *
FROM business;
```

```
iii. Category table = 10000
-- SQL Code
SELECT *
FROM category;
```

```
iv. Checkin table = 10000
-- SQL Code
SELECT *
FROM checkin;
```

```
v. elite_years table = 10000
-- SQL Code
```

```
SELECT *  
FROM elite_years;
```

```
vi. friend table = 10000  
-- SQL Code  
SELECT *  
FROM friend;
```

```
vii. hours table = 10000  
-- SQL Code  
SELECT *  
FROM hours;
```

```
viii. photo table = 10000  
-- SQL Code  
SELECT *  
FROM photo;
```

```
ix. review table = 10000  
-- SQL Code  
SELECT *  
FROM review;
```

```
x. tip table = 10000
-- SQL Code
SELECT *
FROM tip;
```

```
xi. user table = 10000
-- SQL Code
SELECT *
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
-- Code
SELECT count(DISTINCT(id))
FROM business;
```

```
ii. Hours = business_id: 1562
-- Code
SELECT count(DISTINCT(business_id))
```

```
FROM hours;
```

```
iii. Category = business_id: 2643
```

```
-- Code
```

```
SELECT count(DISTINCT(business_id))
```

```
FROM category;
```

```
iv. Attribute = business_id: 1115
```

```
-- Code
```

```
SELECT count(DISTINCT(business_id))
```

```
FROM attribute;
```

```
v. Review = id: 10000; business_id: 8090(FK1); user_id: 9581(FK2)
```

```
-- Code
```

```
-- id
```

```
SELECT count(DISTINCT(id))
```

```
FROM review;
```

```
-- business_id
```

```
SELECT count(DISTINCT(business_id))
```

```
FROM review;
```

```
-- user_id
```

```
SELECT count(DISTINCT(user_id))  
FROM review;
```

```
vi. Checkin = business_id: 493  
-- Code  
SELECT count(DISTINCT(business_id))  
FROM checkin;
```

```
vii. Photo = id: 10000; business_id: 6493  
-- Code  
-- id  
SELECT count(DISTINCT(id))  
FROM photo;
```

```
-- business_id  
SELECT count(DISTINCT(business_id))  
FROM photo;
```

```
viii. Tip = business_id: 3979 (FK1); User_id: 537 (FK2)  
-- Code  
-- business_id  
SELECT count(DISTINCT(business_id))  
FROM tip;
```

```
-- user_id
SELECT count(DISTINCT(user_id))
FROM tip;
```

```
ix. User = id: 10000
-- Code
SELECT count(DISTINCT(id))
FROM user;
```

```
x. Friend = user_id: 11
-- Code
SELECT count(DISTINCT(user_id))
FROM friend;
```

```
xi. Elite_years = user_id: 2780
-- Code
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 [some_column] IS NULL;
```

-- some\_column refers to id, name, review\_count, yelping\_since ect. input all column names in user table to check if there is null value.

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: 1.0    max: 5.0    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 count(review_count), city
FROM business
GROUP BY city
ORDER BY count(review_count) DESC;
```

Copy and Paste the Result Below:

+-----+		
count(review_count)   city		
+-----+		
	1561	Las Vegas
	1001	Phoenix
	985	Toronto
	497	Scottsdale
	468	Charlotte
	353	Pittsburgh
	337	Montréal
	304	Mesa
	274	Henderson
	261	Tempe

	239		Edinburgh	
	232		Chandler	
	189		Cleveland	
	188		Gilbert	
	188		Glendale	
	176		Madison	
	150		Mississauga	
	141		Stuttgart	
	105		Peoria	
	80		Markham	
	71		Champaign	
	70		North Las Vegas	
	64		North York	
	60		Surprise	
	54		Richmond Hill	
+-----+-----+				

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)
FROM business
WHERE city = 'Avon'
GROUP BY stars

```

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

stars	sum(review_count)
1.5	10
2.5	6
3.5	88
4.0	21
4.5	31
5.0	3

SQL code used to arrive at answer:

```

SELECT stars, sum(review_count)
FROM business

```

```
WHERE city = 'Beachwood'
GROUP BY stars
```

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

stars	sum(review_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;
```

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:

Yes, but the number of reviews is not the only factor, but in general, the more time spend on the app, the more fans you will get.

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

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

Answer:

there are 1780 'love' appeared, and 232 'hate' appeared.

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

## 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?

No, the two groups I choose have pretty much the same distributions of hours.

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

In general, the group of 2-3 stars tend to have fewer number os reviews, but there is a small amount in 4-5 stars group have fewer reviews than the others.

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

The postal code of 2-3 stars group are all the same, maybe from the same area. But 4-5 stars group all have different postal code.

SQL code used for analysis:

```
SELECT b.stars, b.review_count, h.hours, b.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 hour_dis,
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_rate
FROM business b
    INNER JOIN hours h ON h.business_id = b.id
    INNER JOIN category c ON c.business_id = b.id
WHERE (b.city IS 'Toronto'
    AND c.category LIKE 'Food' )
    AND (b.stars BETWEEN 2 AND 3
    OR b.stars BETWEEN 4 AND 5)
GROUP BY b.stars, hour_dis
ORDER BY star_rate 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:

The businesses that are closed have lower average number of reviews and less total number of reviews than those who are still open.

ii. Difference 2:

The businesses that are closed have lower average star rates than those who are still open.

SQL code used for analysis:

```
SELECT is_open, AVG(review_count), sum(review_count), AVG(stars)
FROM business
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:

To predict the number of fans a user will have.

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

The factors that might help a user to have more fans are: number of reviews wrote, time on yelp, we will as well gather the information on whether it's funny, cool, or useful.

iii. Output of your finished dataset:

id	name	review_count	fans	average_stars	useful	funny	cool	yelping_time
-9I98YbNQnLdAmcYfb324Q	Amy	609	503	3.21	3226	2554	2751	over 5 years
-8EnCioUmDygAbsYZmTeRQ	Mimi	968	497	4.05	257	138	159	over 5 years
--2vRODIsmQ6WfcSzKWigw	Harald	1153	311	4.4	122921	122419	122890	over 5 years
-G7Zkl1wIWBBmDOKRy_sCw	Gerald	2000	253	3.6	17524	2324	15008	over 5 years
-0IiMAZI2SsQ7VmyzJJokQ	Christine	930	173	3.69	4834	6646	4321	over 5 years
-g3XIcCb2b-BD0QBCcq2Sw	Lisa	813	159	4.09	48	13	6	over 5 years
-9bbDysuiWeo2VShFJJtcw	Cat	377	133	3.99	1062	672	1076	over 5 years
-FZBTkAZEXoP7CYvRV2ZwQ	William	1215	126	4.41	9363	9361	9370	1-5 years
-9da1xk7zggnf0luTVYGkA	Fran	862	124	4.1	9851	7606	9344	over 5 years
-1h59ko3dxChBSZ9U7LfUw	Lissa	834	120	3.68	455	150	342	over 5 years
-B-QEUESGWHPE_889WJaeg	Mark	861	115	3.36	4008	570	2765	over 5 years

-DmqnhW4Omr3YhmnigaqHg	Tiffany		408	111		4.09	1366	984	1279	over 5 years	
-cv9PPT7IHux7XUc9d0pkg	bernice		255	105		3.95	120	112	109	over 5 years	
-DFCC64NXgqrxl08aLU5rg	Roanna		1039	104		3.71	2995	1188	636	over 5 years	
-IgKkE8JvYNWeGu8ze4P8Q	Angela		694	101		3.89	158	164	105	over 5 years	
-K2Tcgh2EKX6e6HqqIrBIQ	.Hon		1246	101		3.14	7850	5851	5104	over 5 years	
-4viTt9UC44lWCFJwleMNQ	Ben		307	96		3.7	1180	1155	1143	over 5 years	
-3i9bhfvrm3F1wsC9XIB8g	Linda		584	89		4.06	3177	2736	3019	over 5 years	
-kLVfaJytOJY2-QdQoCcNq	Christina		842	85		4.1	158	34	102	over 5 years	
-ePh4Prox7ZXnEBNGKyUEA	Jessica		220	84		4.1	2161	2091	2067	over 5 years	
-4BEUkLvHQntN6qPfKJP2w	Greg		408	81		3.67	820	753	746	over 5 years	
-C-18EHS�XtZZVfUAUhsPA	Nieves		178	80		3.64	1091	774	940	over 5 years	
-dw8f7FLaUmWR7bfJ_Yf0w	Sui		754	78		3.62	9	18	2	over 5 years	
-81bUN1XVSoXqaRRiHiSNg	Yuri		1339	76		4.11	1166	220	561	over 5 years	
-OzEEaDFIjABtPQniOXlHA	Nicole		161	73		3.87	13	10	6	over 5 years	
+-----+-----+-----+-----+-----+-----+-----+-----+											

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

```

SELECT u.id, u.name, u.review_count,u.fans, u.average_stars, u.useful, u.funny, u.cool,
CASE
    WHEN DATE('now')-u.yelping_since < 1 THEN 'less than a year'
    WHEN DATE('now')-u.yelping_since BETWEEN 1 AND 5 THEN '1-5 years'
    WHEN DATE('now')-u.yelping_since > 5 THEN 'over 5 years'

```

```
    END AS yelping_time  
FROM user u  
GROUP BY u.id  
ORDER BY fans DESC
```