

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

```
CODE: SELECT COUNT(*) AS TotalRecord  
FROM attribute;
```

- ii. Business table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM business;

iii. Category table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM category;

iv. Checkin table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM checkin;

v. elite_years table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM elite_years;

vi. friend table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM friend;

vii. hours table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM hours;

viii. photo table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM photo;

ix. review table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM review;

x. tip table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
FROM tip;

xi. user table = 10000

CODE: SELECT COUNT(*) AS TotalRecord
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 = 10000

CODE: SELECT COUNT(DISTINCT id) AS Total_Distinct_Primary_Key
FROM business; --Primary Key is id

ii. Hours = 1562

CODE: SELECT COUNT(DISTINCT business_id) AS Total_Distinct_Foreign_key
FROM hours; --Foreign key is business_id

iii. Category = 2643

CODE: SELECT COUNT(DISTINCT business_id) AS Total_Distinct_Foreign_key
FROM category; --Foreign key is business_id

iv. Attribute = 1115

CODE: SELECT COUNT(DISTINCT business_id) AS Total_Distinct_Foreign_key
FROM attribute; --Foreign key is business_id

v. Review = 10000

CODE: SELECT COUNT(DISTINCT id) AS Total_Distinct_Primary_Key
FROM review; --Primary Key is id

vi. Checkin = 493

CODE: SELECT COUNT(DISTINCT business_id) AS Total_Distinct_Foreign_key
FROM checkin; --Foreign key is business_id

vii. Photo = 10000

CODE: SELECT COUNT(DISTINCT id) AS Total_Distinct_Primary_Key
FROM photo; --Primary Key is id

viii. Tip = 3979

CODE: SELECT COUNT(DISTINCT business_id) AS Total_Distinct_Foreign_key
FROM tip; --Foreign key is business_id

ix. User = 10000

CODE: SELECT COUNT(DISTINCT id) AS Total_Distinct_Primary_Key
FROM user; --Primary Key is id

x. Friend = 11

CODE: SELECT COUNT(DISTINCT user_id) AS Total_Distinct_Foreign_key
FROM friend; --Foreign key is user_id

xi. Elite_years = 2780

CODE: SELECT COUNT(DISTINCT user_id) AS Total_Distinct_Foreign_key
FROM elite_years; --Foreign key is user_id

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 *  
FROM user  
WHERE name IS NULL OR -- DO NOT need to check id column because it is Primary key and  
-- Primary CAN NOT be NULL  
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:	max:	avg:
1	5	3.7082

CODE:

```
SELECT MIN(Stars) FROM Review;  
SELECT MAX(Stars) FROM Review;  
SELECT AVG(Stars) FROM Review;
```

ii. Table: Business, Column: Stars

min:	max:	avg:
1.0	5.0	3.6549

CODE:

```
SELECT MIN(Stars) FROM Business;
```

```
SELECT MAX(Stars) FROM Business;
```

```
SELECT AVG(Stars) FROM Business;
```

iii. Table: Tip, Column: Likes

min:	max:	avg:
0	2	0.0144

Similar code as above

iv. Table: Checkin, Column: Count

min:	max:	avg:
1	53	1.9414

v. Table: User, Column: Review_count

min:	max:	avg:
0	2000	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 Total_reviews
FROM Business
```

GROUP BY City

ORDER BY Total_reviews DESC;

Copy and Paste the Result Below:

+-----+-----+	
city	Total_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 as StarRating
,COUNT(Stars) as Count
FROM Business
WHERE city IN ('Avon') -- Can also replace this line with "WHERE city IN ('Avon') AND Stars BETWEEN 1
--AND 5"
GROUP BY Stars;
```

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

+-----+-----+		
StarRating	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 StarRating
,COUNT(Stars) as Count
FROM Business
WHERE city IN ('Beachwood')
GROUP BY Stars;
```

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

+-----+-----+				
	StarRating		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 name
, COUNT(review_count) AS Total_Reviews
FROM User
GROUP BY name
ORDER BY Total_Reviews DESC
LIMIT 3;
```

Copy and Paste the Result Below:

```
+-----+-----+
| name | Total_Reviews |
+-----+-----+
| Nicole | 2397 |
| Sara | 2253 |
| Gerald | 2034 |
+-----+-----+
```

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

No, after retrieving data from user table it IS NOT necessary that posting more reviews correlate with more fans

Code:

```
SELECT name
, SUM(Fans) AS Fan_Points
```

```
,SUM(review_count) AS Total_Reviews
FROM User
GROUP BY name
ORDER BY Total_Reviews DESC, Fan_Points DESC
LIMIT 7;
```

RESULT:

```
+-----+-----+-----+
| name   | Fan_Points | Total_Reviews |
+-----+-----+-----+
| Nicole | 200        | 2397          |
| Sara   | 72         | 2253          |
| Gerald | 256        | 2034          |
| Lisa   | 207        | 2021          |
| Mark   | 156        | 1945          |
| Jennifer | 86        | 1862          |
| Jen    | 148        | 1744          |
+-----+-----+-----+
```

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

Answer:

More Love according to my strategy

```
+-----+-----+
| LOVE | HATE |
+-----+-----+
| 8079 | 1046 |
+-----+-----+
```

SQL code used to arrive at answer:

```
SELECT (SELECT COUNT(average_stars) AS 'LOVE'  
FROM User  
WHERE average_stars >= 3) AS 'LOVE',  
(SELECT COUNT(average_stars) AS 'HATE'  
FROM User  
WHERE average_stars <= 2) AS 'HATE'
```

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

SQL code used to arrive at answer:

```
SELECT id  
,name  
,fans  
FROM USER  
ORDER BY Fans DESC  
LIMIT 10;
```

Copy and Paste the Result Below:

id	name	fans
-9I98YbNQnLdAmcYfb324Q	Amy	503
-8EnCioUmDygAbsYZmTeRQ	Mimi	497
--2vR0DIsmQ6WfcSzKWigw	Harald	311

```

| -G7Zkl1wIWBBmD0KRy_sCw | Gerald | 253 |
| -0liMAZI2SsQ7VmyzJjokQ | Christine | 173 |
| -g3XlcCb2b-BD0QBCcq2Sw | Lisa | 159 |
| -9bbDysuiWeo2VShFJJtcw | Cat | 133 |
| -FZBTkAZEXoP7CYvRV2ZwQ | William | 126 |
| -9da1xk7zggnfO1uTVYGkA | Fran | 124 |
| -lh59ko3dxChBSZ9U7LfUw | Lissa | 120 |
+-----+-----+-----+

```

```

38tScZkvRLoa5h-wNPyjkw |
| 38s4jUzBkei3Gy-U5mtEJA |
| 38rXDufRtJeGSMP6ducaCw

```

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?

Yes

CODE:

```

WHEN Stars >=2 AND Stars <= 3 THEN 'GROUP1'

WHEN Stars >=4 AND Stars <= 5 THEN 'GROUP2'

END Stars_Grouping

,Name

,City

,Hours

```

FROM Business INNER JOIN Hours ON Business.id = Hours.Business_ID
WHERE City = 'Toronto'

Result:

Stars_Grouping	name	city	hours
GROUP2	Cabin Fever	Toronto	Monday 16:00-2:00
GROUP2	Cabin Fever	Toronto	Tuesday 18:00-2:00
GROUP2	Cabin Fever	Toronto	Friday 18:00-2:00
GROUP2	Cabin Fever	Toronto	Wednesday 18:00-2:00
GROUP2	Cabin Fever	Toronto	Thursday 18:00-2:00
GROUP2	Cabin Fever	Toronto	Sunday 16:00-2:00
GROUP2	Cabin Fever	Toronto	Saturday 16:00-2:00
GROUP2	Gussied Up	Toronto	Tuesday 11:00-19:00
GROUP2	Gussied Up	Toronto	Friday 11:00-19:00
GROUP2	Gussied Up	Toronto	Wednesday 11:00-19:00
GROUP2	Gussied Up	Toronto	Thursday 11:00-19:00
GROUP2	Gussied Up	Toronto	Sunday 12:00-16:00
GROUP2	Gussied Up	Toronto	Saturday 11:00-17:00
GROUP2	Innercity MMA	Toronto	Friday 17:00-22:00
GROUP2	Innercity MMA	Toronto	Tuesday 18:00-22:00
GROUP2	Innercity MMA	Toronto	Thursday 18:00-22:00
GROUP2	Innercity MMA	Toronto	Wednesday 17:00-22:00
GROUP2	Innercity MMA	Toronto	Monday 17:00-22:00
GROUP1	Big Smoke Burger	Toronto	Monday 10:30-21:00
GROUP1	Big Smoke Burger	Toronto	Tuesday 10:30-21:00
GROUP1	Big Smoke Burger	Toronto	Friday 10:30-21:00
GROUP1	Big Smoke Burger	Toronto	Wednesday 10:30-21:00

GROUP1	Big Smoke Burger Toronto Thursday 10:30-21:00	
GROUP1	Big Smoke Burger Toronto Sunday 11:00-19:00	
GROUP1	Big Smoke Burger Toronto Saturday 10:30-21:00	

+-----+-----+-----+-----+

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

Yes

CODE:

```
SELECT CASE
WHEN Stars >=2 AND Stars <= 3 THEN 'GROUP1'
WHEN Stars >=4 AND Stars <= 5 THEN 'GROUP2'
END Stars_Grouping
,COUNT(Review_count)
FROM Business
WHERE City = 'Toronto'
GROUP BY Stars_Grouping
```

Result:

+-----+-----+	
Stars_Grouping	COUNT(Review_count)
+-----+-----+	
None	239
GROUP1	307
GROUP2	439
+-----+-----+	

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

Yes, we can get the House num and street name with the help of trimming tools.

SQL code used for analysis:

```
SELECT CASE  
WHEN Stars >=2 AND Stars <= 3 THEN 'GROUP1'  
WHEN Stars >=4 AND Stars <= 5 THEN 'GROUP2'  
END Stars_Grouping  
,Address  
FROM Business  
WHERE City = 'Toronto'  
GROUP BY Stars_Grouping
```

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: More reviews that business is still open

ii. Difference 2: Closed Business Stars average is less than Opened Business

SQL code used for analysis:

```
SELECT AVG(Stars)  
,SUM(Review_count)  
,is_open  
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:

I will use the following functions;

Business Categorizied by stars (getting information from reviews);

Used INNER JOIN, CASE Statement, Aliasing, Date Strings, Filtering Condition, Group By Business Name, Order By stars

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

Most of the time I would choose the String and integer/float

Why String;

If I am looking for state, country, id then I would use String data type. 70% of the time we deal with strings.

Why Integer;

For calculation Like Average, Maximum, Minimum, I would like to use Integer data type

iii. Output of your finished dataset:

name	stars	Stars_Grouping	Review_Year	Review_Month	Review_Day
808 Sushi	4	GOOD	2010	12	13
99 Ranch Market	4	GOOD	2014	01	04
AG Jeans	4	GOOD	2007	12	28
ARA Of Madison	4	GOOD	2016	06	23
Abuelo's	4	GOOD	2017	04	16
Akahana Asian Bistro	4	GOOD	2017	06	20
Art of Flavors	4	GOOD	2015	10	24
Ashbridges Bay Park	4	GOOD	2009	08	02
Asian Taste Restaurant	4	GOOD	2017	05	05
Baja Fresh Mexican Grill	4	GOOD	2010	07	30
Barro's Pizza	4	GOOD	2013	11	23
Basil Box	4	GOOD	2016	12	14
Beaver Choice	4	GOOD	2014	09	30
Big Earl's Greasy Eats	4	GOOD	2016	09	18
Blasted Barley Beer Company	4	GOOD	2017	01	14
Boba Tea House	4	GOOD	2008	02	24
Buca di Beppo	4	GOOD	2017	06	20
Budget Rent A Car	4	GOOD	2017	06	14
Burrito Bandito	4	GOOD	2011	01	16
CN Tower	4	GOOD	2015	04	01
Cafe Tandoor	4	GOOD	2016	05	19
Capital Espresso	4	GOOD	2016	02	17
Carson Kitchen	4	GOOD	2015	05	10
Cathedral of Learning	4	GOOD	2010	07	21
Chef Flemming's BakeShop	4	GOOD	2015	09	25

(Output limit exceeded, 25 of 348 total rows shown)

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

```
SELECT B.name
,R.Stars
,CASE
WHEN R.Stars >=2 AND R.Stars <= 3 THEN 'AVERAGE'
WHEN R.Stars >=4 AND R.Stars <= 5 THEN 'GOOD'
ELSE 'BAD'
END Stars_Grouping
,STRFTIME('%Y',date(R.Date)) AS Review_Year
,STRFTIME('%m', date(R.date)) AS Review_Month
,STRFTIME('%d',date(R.date)) AS Review_Day
FROM Business B INNER JOIN Review R ON B.id = R.Business_ID
WHERE Stars_Grouping = 'GOOD'
GROUP BY B.name
ORDER BY R.stars
```