Data Scientist Role Play: Profiling and Analyzing the Yelp Dataset Coursera Worksheet

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
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

```
select count(*) as total
from attribute;
+-------+
| total |
+-------+
| 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
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 = business_id `3979` user_id `537`
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.

```
select count(distinct id)
from business;
```

```
+----------------------+
| count(distinct id)  |
+----------------------+
|                10000 |
+----------------------+
```

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 [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

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

```
select min(stars),
max(stars),
avg(stars)
from review;
+------------+------------+------------+
| min(stars) | max(stars) | avg(stars) |
+------------+------------+------------+
|          1 |          5 |     3.7082 |
+------------+------------+------------+
```

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 |
```

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

ii. Beachwood

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 â€“ 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,
sum(review_count)
from user
group by id
order by review_count desc
limit 3;
```

Copy and Paste the Result Below:

```
+--------+--------------------+
| name   | sum(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:

No, because according to result Gerald has more reviews however, Mimi has the most fans.

```
 select name,
fans,
sum(review_count)
from user
group by id
order by review_count desc;
```

name	fans	sum(review_count)
Gerald	253	2000
Sara	50	1629
Yuri	76	1339
.Hon	101	1246
William	126	1215
Harald	311	1153
eric	16	1116
Roanna	104	1039
Mimi	497	968
Christine	173	930
Ed	38	904
Nicole	43	864
Fran	124	862
Mark	115	861
Christina	85	842
Dominic	37	836
Lissa	120	834
Lisa	159	813
Alison	61	775
Sui	78	754
Tim	35	702
L	10	696
Angela	101	694
Crissy	25	676
Lyn	45	675

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

Answer: There is more reviews with word love than hate.

SQL code used to arrive at answer:

```
 select count(*) as hate
from review
where text like '%hate%';
```

```
+------+
| hate |
+------+
|  232 |
+------+
```

```
select count(*) as love
from review
where text like '%love%';
```

```
+------+
| love |
+------+
| 1780 |
+------+
```

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?

Yes.

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

Yes

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

Yes, there is two business in the same neighborhood in different categories with star rates 4-5. And they don't have much reviews.

SQL code used for analysis:

```
select
business.name
, business.city
, category.category
, business.stars
, hours.hours
, business.review_count
, business.postal_code
from (business inner join category on business.id = category.business_id)
```

```
inner join hours on hours.business_id = category.business_id
where business.city = 'Las Vegas'
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 review count for open business is higher than the closed ones.

ii. Difference 2: The star rates for open business is higher than the closed ones.

SQL code used for analysis:

```
SELECT avg(stars),
avg(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:

Finding a correlation between the total number of businesses with high star rates and the most visited cities.

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

First I need two different databases to compare businesses in different cities. Then I have to find the total number of businesses with each star rate. In order to do the analysis, I join business and review tables. After that, I add a case statement to easily see the high rates and the low rates.
The reason I choose this analysis is to understand whether the review stars of the businesses in the more visited cities are higher or lower. Because when people see that a business is crowded, they tend to give a better rating.

iii. Output of your finished dataset:

city	total_business	postal_code	stars	star_rate
Las Vegas	193	89118	5	high
Phoenix	65	85019	1	low
Toronto	51	M2M 3W5	5	high
Scottsdale	37	85251	4	high
Henderson	30	89123	2	low
Tempe	28	85281	5	high
Pittsburgh	23	15235	5	high
Chandler	22	85225	3	high
Charlotte	21	28202	3	high
Montréal	18	H2X 1S3	5	high
Madison	16	53719	2	low
Gilbert	13	85234	5	high
Mesa	13	85209	1	low
Cleveland	12	44106	5	high
North Las Vegas	6	89030	1	low
Edinburgh	5	EH12 6AW	4	high
Glendale	5	85308	1	low
Lakewood	5	44107	3	high
Cave Creek	4	85331	4	high
Champaign	4	61820	5	high
Markham	4	L3R 5G5	1	low
North York	4	M2N 5P9	4	high
Mississauga	3	L5B 4C1	2	low
Surprise	3	85379	5	high
Avondale	2	85323	5	high

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

```sql
select city,
count(name) as total_business,
postal_code,
review.stars,
case when review.stars <= 2 then 'low'
when review.stars >= 3 then 'high'
else 'other'
end star_rate
from business
inner join review on business.ID = review.business_ID
group by city
order by total_business desc;
```