Hey, my name is Niv and this is my first peer-review assignment. I hope you'll find it to your liking.

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 = 10,000
ii. Business table = 10,000
iii. Category table = 10,000
iv. Checkin table = 10,000
v. elite_years table = 10,000
vi. friend table = 10,000
vii. hours table = 10,000
viii. photo table = 10,000
ix. review table = 10,000
x. tip table = 10,000
xi. user table = 10,000

SQL CODE: SELECT COUNT(*) AS row_count from [Column_Name]

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: 10,000
ii. Hours = business_id: 1,562
iii. Category = business_id: 2,643
iv. Attribute = business_id: 1,115
v. Review = id: 10,000
vi. Checkin = business_id: 493
vii. Photo = id: 10,000
viii. Tip = user_id: 537
ix. User = id: 10,000
x. Friend = user_id: 11
xi. Elite_years = user_id: 2,780

Note: Primary Keys are denoted in the ER-Diagram with a yellow key icon.

SQL CODE: SELECT COUNT(DISTINCT [Key_Name]) AS Key_Count From [Table_Name]

3. Are there any columns with null values in the Users table? Indicate "yes," or "no."

Answer: **no. surprisingly.**

SQL code used to arrive at answer:

```
SELECT count(*)-count(name)
,count(*)-count(review_count)
,count(*)-count(yelping_since)
,count(*)-count(useful)
,count(*)-count(funny)
,count(*)-count(cool)
,count(*)-count(fans)
,count(*)-count(average_stars)
,count(*)-count(compliment_hot)
,count(*)-count(compliment_more)
,count(*)-count(compliment_profile)
,count(*)-count(compliment_cute)
,count(*)-count(compliment_list)
,count(*)-count(compliment_note)
,count(*)-count(compliment_plain)
,count(*)-count(compliment_cool)
,count(*)-count(compliment_funny)
,count(*)-count(compliment_writer)
,count(*)-count(compliment_photos)
from user
```

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: 3.71

ii. Table: Business, Column: Stars

min: 1 max: 5 avg: 3.65

iii. Table: Tip, Column: Likes

min: 0 max: 2 avg: 0.01

iv. Table: Checkin, Column: Count

min: 1 max: 53 avg: 1.94

v. Table: User, Column: Review_count

min: 0 max: 2000 avg: 24.3

SQL code:

```
SELECT min([column])as Minimum
,max([column]) as Maximum
,round(avg([column]),2) as avg
from [Table]
```

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_review
FROM business
GROUP BY city
ORDER BY total_review DESC
```

Copy and Paste the Result Below:

city	total_review
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
,sum(review_count) AS stars_rating_count
FROM business
WHERE city = "Avon"
GROUP BY stars
```

Copy and Paste the Resulting Table Below (2 columns ı€" star rating and count):

stars	stars_rating_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) AS stars_rating_count
FROM business
WHERE city = "Beachwood"
GROUP BY stars
```

Copy and Paste the Resulting Table Below (2 columns ı€" star rating and count):

stars	stars_rating_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 id
,review_count
from user
GROUP BY id
ORDER BY review_count DESC
LIMIT 3
```

Copy and Paste the Result Below:

```
+------------------------+--------------+
| id                     | review_count |
+------------------------+--------------+
| -G7Zkl1wIWBBmD0KRy_sCw |         2000 |
| -3s52C4zL_DHRK0ULG6qtg |         1629 |
| -8lbUNlXVSoXqaRRiHiSNg |         1339 |
+------------------------+--------------+
```

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

I found the maximum number and average for both variables,
Then found that people with more reviews tend to have much more than the average 1.4896 fans, so I assume that the 2 variables are positively correlated.

```
SELECT id, name
,review_count
,fans
from user
GROUP BY id
ORDER BY review_count desc;
```

id	name	review_count	fans
-G7Zkl1wIWBBmD0KRy_sCw	Gerald	2000	253
-3s52C4zL_DHRK0ULG6qtg	Sara	1629	50
-8lbUNlXVSoXqaRRiHiSNg	Yuri	1339	76
-K2Tcgh2EKX6e6HqqIrBIQ	.Hon	1246	101
-FZBTkAZEXoP7CYvRV2ZwQ	William	1215	126
--2vR0DIsmQ6WfcSzKWigw	Harald	1153	311
-gokwePdbXjfS0iF7NsUGA	eric	1116	16
-DFCC64NXgqrxlO8aLU5rg	Roanna	1039	104
-8EnCioUmDygAbsYZmTeRQ	Mimi	968	497
-0IiMAZI2SsQ7VmyzJjokQ	Christine	930	173
-fUARDNuXAfrOn4WLSZLgA	Ed	904	38
-hKniZN2OdshWLHYuj21jQ	Nicole	864	43
-9da1xk7zgnnfO1uTVYGkA	Fran	862	124
-B-QEUESGWHPE_889WJaeg	Mark	861	115
-kLVfaJytOJY2-QdQoCcNQ	Christina	842	85
-kO6984fXByyZm3_6z2JYg	Dominic	836	37
-lh59ko3dxChBSZ9U7LfUw	Lissa	834	120
-g3XIcCb2b-BD0QBCcq2Sw	Lisa	813	159
-l9giG8TSDBG1jnUBUXp5w	Alison	775	61
-dw8f7FLaUmWR7bfJ_Yf0w	Sui	754	78
-AaBjWJYiQxXkCMDlXfPGw	Tim	702	35
-jt1ACMiZljnBFvS6RRvnA	L	696	10
-IgKkE8JvYNWeGu8ze4P8Q	Angela	694	101
-hxUwfo3cMnLTv-CAaP69A	Crissy	676	25
-H6cTbVxeIRYR-atxdielQ	Lyn	675	45

(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: Love.

SQL code used to arrive at answer:

```
SELECT
SUM(CASE WHEN review.text LIKE '%love%' THEN 1 ELSE 0 END) as love_co
unt
,SUM(CASE WHEN review.text LIKE '%hate%' THEN 1 ELSE 0 END) as hate_c
ount
,SUM(CASE WHEN review.text LIKE '%love%hate%' OR review.text LIKE '%h
ate%love%' THEN 1 ELSE 0 END) as both

FROM review;
```

love_count	hate_count	both
1780	232	54

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
-0IiMAZI2SsQ7VmyzJjokQ	Christine	173
-g3XIcCb2b-BD0QBCcq2Sw	Lisa	159
-9bbDysuiWeo2VShFJJtcw	Cat	133
-FZBTkAZEXoP7CYvRV2ZwQ	William	126
-9da1xk7zgnnfO1uTVYGkA	Fran	124
-lh59ko3dxChBSZ9U7LfUw	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, by quite a lot.
2-3 stars businesses have 7 working hours while 4-5 stars have 13 hours, almost double.

I Chose “Toronto” and obviously “Food”.
I first split into categories using CASE, then joined for hours.

CODE:

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
WHEN stars >= 2 THEN "2-3 stars"
ELSE "under 2"
END star_rank,
city,
c.category,
```

```
count(distinct business.id) AS business_count,
count(h.hours) AS hours_work
FROM business
JOIN hours h ON business.id = h.business_id
JOIN category c ON business.id = c.business_id
WHERE city = "Toronto" AND c.category = "Food"
GROUP BY star_rank
```

star_rank	city	category	business_count	hours_work
2-3 stars	Toronto	Food	1	7
4-5 stars	Toronto	Food	2	13

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

Yes, by quite a lot. 2-3 stars have 13 reviews while 4-5 stars have 41 reviews.

CODE:

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
WHEN stars >= 2 THEN "2-3 stars"
ELSE "under 2"
END star_rank,
city,
c.category,
count(distinct business.id) AS business_count,
sum(review_count) AS total_reviews
FROM business
JOIN category c ON business.id = c.business_id
WHERE city = "Toronto" AND c.category = "Food"
GROUP BY star_rank
```

star_rank	city	category	business_count	total_reviews
2-3 stars	Toronto	Food	2	13
4-5 stars	Toronto	Food	2	41

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

Im not familiar with postal codes in Canada, or neighborhoods, but it seems all locations are not far from each other so I'd assume the lower postal codes are closer to each other or have certain food level that the upper 2 do not have.

CODE:

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
WHEN stars >= 2 THEN "2-3 stars"
ELSE "under 2"
END star_rank,
```

```
address,
neighborhood,
city,
postal_code
FROM business
JOIN category c ON business.id = c.business_id
WHERE city = "Toronto" AND c.category = "Food"
ORDER BY star_rank
```

star_rank	address	neighborhood	city	postal_code
2-3 stars	2280 Dundas Street W	Roncesvalles	Toronto	M6R 1X3
2-3 stars	3003 Bathurst Street		Toronto	M6B
4-5 stars	1669 Bloor Street W	High Park	Toronto	M6P 1A6
4-5 stars	247 Wallace Avenue	Wallace Emerson	Toronto	M6H 1V5

SQL code used for analysis:

I.

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
WHEN stars >= 2 THEN "2-3 stars"
ELSE "under 2"
END star_rank,
city,
c.category,
count(distinct business.id) AS business_count,
count(h.hours) AS hours_work
FROM business
JOIN hours h ON business.id = h.business_id
JOIN category c ON business.id = c.business_id
WHERE city = "Toronto" AND c.category = "Food"
GROUP BY star_rank
```

II.

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
WHEN stars >= 2 THEN "2-3 stars"
ELSE "under 2"
END star_rank,
city,
c.category,
count(distinct business.id) AS business_count,
sum(review_count) AS total_reviews
FROM business
JOIN category c ON business.id = c.business_id
WHERE city = "Toronto" AND c.category = "Food"
GROUP BY star_rank
```

III.

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
WHEN stars >= 2 THEN "2-3 stars"
ELSE "under 2"
END star_rank,
address,
neighborhood,
city,
postal_code
FROM business
JOIN category c ON business.id = c.business_id
WHERE city = "Toronto" AND c.category = "Food"
ORDER BY star_rank
```

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:

Businesses that are still open have a lot more reviews, even though open businesses might mean new businesses, I suppose that “old” and successful businesses make up for that. Or perhaps reviews help keep businesses open.

ii. Difference 2:

The stars given are nearly the same, which could mean that its not necessarily the quality of the business that made them close, but perhaps our former assumption about reviews helping to keep businesses open.

SQL code used for analysis:

```
SELECT  CASE WHEN is_open = 1 THEN "OPEN"
WHEN is_open = 0 THEN "CLOSED"
END status,
count(distinct id) AS businesses,
sum(review_count) AS total_review,
avg(review_count) AS avg_review,
avg(stars) AS avg_stars
FROM business
GROUP BY is_open
ORDER BY status DESC
```

status	businesses	total_review	avg_review	avg_stars
OPEN	8480	269300	31.7570754717	3.67900943396
CLOSED	1520	35261	23.1980263158	3.52039473684

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:

The analysis is to find out in which category we have the most businesses, and based on total reviews find out which categories have the highest avg rating. Also checking to see if more reviews mean lower average stars.

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

Im going to count the number of businesses per category (not regarding place for simplicity), and choose only categories with over 10 businesses.
Also, im going to get the number of reviews over 150 to avoid irregularities, and look at the average stars.

iii. Output of your finished dataset:

category	num_businesses	avg_stars	total_reviews
Health & Medical	17	4.088	203
Shopping	30	3.983	977
American (Traditional)	11	3.818	1128
Food	23	3.783	1781
Bars	17	3.5	1322
Nightlife	20	3.475	1351
Restaurants	71	3.458	4504

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

```
SELECT category,
count(distinct id) AS num_businesses,
round(avg(stars),3) AS avg_stars,
sum(review_count) total_reviews
FROM business
JOIN category ON business.id = category.business_id
GROUP BY category
HAVING num_businesses >= 10 AND total_reviews >= 150
ORDER BY avg_stars DESC
```