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.

Grant Healy

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 records

```
Select *
From attribute;
```

ii. Business table = 10000 records

```
Select *
From business;
```

iii. Category table = 10000 records

```
Select *
From category;
```

iv. Checkin table = 10000 records

```
Select *
From checkin;
```

v. elite_years table = 10000 records

```
Select *
From elite_years;
```

vi. friend table = 10000 records

```
Select *
From friend;
```

vii. hours table = 10000 records

```
Select *
From hours;
```

viii. photo table = 10000 records

```
Select *
From photo;
```

ix. review table = 10000 records

```
Select *
```

From review;

x. tip table = 10000 records

Select *

From tip;

xi. user table = 10000 records

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 = 10000 distinct records

Select distinct(id)

From business;

ii. Hours = 1562 distinct records

Select distinct(business_id)

From hours;

iii. Category = 2643 distinct records

Select distinct(business_id)

From category;

iv. Attribute = 1115 distinct records

Select distinct(business_id)

From attribute;

v. Review = 10000 distinct records

```
Select distinct(id)
From review;
```

vi. Checkin = 493 distinct records

```
Select distinct(business_id)
From checkin;
```

vii. Photo = 10000 distinct records

```
Select distinct(id)
From photo;
```

viii. Tip = 537 distinct records (I used user_id as the foreign key for this table)

```
Select distinct(user_id)
From tip;
```

ix. User = 10000 distinct records

```
Select distinct(id)
From user;
```

x. Friend = 11 distinct records

```
Select distinct(user_id)
From friend;
```

xi. Elite_years = 2780 distinct records

```
Select 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 *
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: .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 city,
sum(review_count) as total_city_reviews
From business
group by city
order by total_city_reviews desc;
```

Copy and Paste the Result Below:

city	total_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

(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 city,
```

```
    count(stars) as Star_Count,
    stars
From business
Where city = 'Avon'
Group by stars;
```

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

```
+------+------------+-------+
| city | Star_Count | stars |
+------+------------+-------+
| Avon |          1 |   1.5 |
| Avon |          2 |   2.5 |
| Avon |          3 |   3.5 |
| Avon |          2 |   4.0 |
| Avon |          1 |   4.5 |
| Avon |          1 |   5.0 |
+------+------------+-------+
```

ii. Beachwood

SQL code used to arrive at answer:

```
Select city,
    count(stars) as Star_Count,
    stars
From business
Where city = 'Beachwood'
Group by stars;
```

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

```
+-----------+------------+-------+
| city      | Star_Count | stars |
```

```
+-----------+------------+-------+
| Beachwood |      1 |  2.0 |
| Beachwood |      1 |  2.5 |
| Beachwood |      2 |  3.0 |
| Beachwood |      2 |  3.5 |
| Beachwood |      1 |  4.0 |
| Beachwood |      2 |  4.5 |
| Beachwood |      5 |  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:

It would appear there is a correlation between the amount of reviews posted and the average amount of fans a user has. The results from my query show that users with 1001-2000 reviews have the highest average amount of fans, 501-1000 second highest, 51-500 third,

And the users with the least amount of reviews have the lowest average number of fans.

```
Select review_bins,
   avg(fans) as average_fans,
   avg(review_count) as average_reviews
From
   (Select case
   When review_count <= 50 then '0 - 50'
   When review_count between 51 and 500 then '51 - 500'
   When review_count between 501 and 1000 then '501 - 1000'
   When review_count between 1001 and 2000 then '1001 - 2000'
   else 'other' end as review_bins,
   review_count,
   fans
   From user) as subtable
group by subtable.review_bins
order by average_fans desc;
```

```
+-------------+---------------+----------------+
| review_bins |   average_fans | average_reviews |
+-------------+---------------+----------------+
| 1001 - 2000 |       129.625 |       1342.125 |
| 501 - 1000  |         76.4 |        677.85 |
| 51 - 500    |  8.89576547231 |   138.966340934 |
| 0 - 50      | 0.289004539918 |    8.5434614107 |
+-------------+---------------+----------------+
```

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

Answer: There are more reviews with the word love than hate

SQL code used to arrive at answer:

```
Select count(distinct(id)) as love_reviews
From review
Where lower(text) like '%love%'
;
```

```
+--------------+
| love_reviews |
+--------------+
|       1780 |
+--------------+
```

Reviews with the word love: 1780

```
Select count(distinct(id)) as hate_reviews
From review
Where lower(text) like '%hate%'
;
```

```
+--------------+
| hate_reviews |
+--------------+
|        232 |
+--------------+
```

Reviews with the word hate: 232

```
Select count(distinct(id)) as both_reviews
From review
Where lower(text) like '%hate%'
and lower(text) like '%love%'
;
```

Their are additionally 54 reviews that feature both love and hate

```
+--------------+
| both_reviews |
+--------------+
|       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 chose to use Las Vegas as my city and restaurants as my category

i. Do the two groups you chose to analyze have a different distribution of hours?

The groups I selected did not appear to have a different distribution of hours

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

The restaurants with more stars appear to have a higher number of reviews on average

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

The only thing I can infer is that higher star restaurants likely have more total reviews. The small amount of establishments returned by my
Query may mean that these inferences may not hold true with a larger sample of data.

SQL code used for analysis:

For part 1

```
Select Case
        When Stars between 2 and 3 then '2 - 3 Stars'
        When Stars > 3 then '4 - 5 Stars'
        Else 'Less than 2 stars'
end as star_bins,
        Count(distinct(business.id)) as number_of_businesses,
        count(hours) as Hour_open,
        count(hours) / count(distinct(business.id)) as average_hours_open
```

```
From((business inner join hours on business.id = hours.business_id)
        inner join category on business.id = category.business_id)
Where city like 'Las Vegas' and category.category = 'Restaurants'
Group by star_bins;
```

```
+------------+--------------------+----------+------------------+
| star_bins  | number_of_businesses | Hour_open | average_hours_open |
+------------+--------------------+----------+------------------+
| 2 - 3 Stars |                  1 |       7 |                7 |
| 4 - 5 Stars |                  2 |      14 |                7 |
+------------+--------------------+----------+------------------+
```

For part 2

```
Select Case
        When Stars between 2 and 3 then '2 - 3 Stars'
        When Stars > 3 then '4 - 5 Stars'
        Else 'Less than 2 stars'
end as star_bins,
        Count(distinct(business.id)) as number_of_businesses,
        sum(review_count) as Review_count,
        sum(review_count) / count(distinct(business.id)) as average_review_count
From((business inner join hours on business.id = hours.business_id)
        inner join category on business.id = category.business_id)
Where city like 'Las Vegas' and category.category = 'Restaurants'
Group by star_bins;
```

```
+------------+--------------------+--------------+--------------------+
| star_bins  | number_of_businesses | Review_count | average_review_count |
+------------+--------------------+--------------+--------------------+
| 2 - 3 Stars |                  1 |        861 |                861 |
| 4 - 5 Stars |                  2 |       6552 |               3276 |
+------------+--------------------+--------------+--------------------+
```

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 total number of reviews for businesses that are open is much higher than those that are closed.

ii. Difference 2:

Slightly bigger average number of stars on reviews from business which are open, also more businesses are opened than closed.

SQL code used for analysis:

```
Select is_open,
    count(distinct(business.id)) as open_closed_count,
    count(distinct(review.id)) as total_reviews,
    avg(review.stars) as average_stars
From business join review on business.id = review.business_id
Group by is_open;
```

is_open	open_closed_count	total_reviews	average_stars
0	61	71	3.64788732394
1	446	565	3.7610619469

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:

In which 25 cities do businesses have the highest average star rating

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

For this analysis I need to collect the average number of stars that businesses receive and group them by city. After grouping them by city I then need to limit the results to 25
To see which 25 cities have the highest average stars on reviews.

iii. Output of your finished dataset:

city	total_reviews	average_stars
Ajax	1	5.0
Burton	1	5.0
Cornelius	1	5.0
Enterprise	1	5.0
Fort Mill	1	5.0
Goodyear	2	5.0
Laveen	1	5.0
Matthews	2	5.0
Middleton	1	5.0
Oakmont	1	5.0
San Tan Valley	1	5.0
Stow	1	5.0
Strongsville	1	5.0
Urbana	2	5.0
Westlake	2	4.5
Gilbert	13	4.38461538462
Scottsdale	37	4.16216216216
Pittsburgh	23	4.08695652174
Ahwatukee	1	4.0
Allison Park	1	4.0
Aurora	1	4.0
Edinburgh	5	4.0
Harrisburg	1	4.0

```
| Lakewood      |         5 |       4.0 |
| McMurray      |         1 |       4.0 |
+---------------+-----------+-----------+
```

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

```
Select city,
    count(distinct(review.id)) as total_reviews,
    avg(review.stars) as average_stars
From business join review on business.id = review.business_id
Group by city
Order by average_stars desc
Limit 25;
```