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:

table_name	COUNT(*)
Attribute	10000
Business	10000
Category	10000

Checkin	10000
elite_years	10000
friend	10000
hours	10000
photo	10000
Review	10000
tip	10000
user	10000

SQL:

```
SELECT COUNT(*) FROM <table_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.

table_name	**column_name**	records
Business	id	10000
Hours	Business_id	1562
Category	Business_id	2643
Attribute	Business_id	1115
Review	id	10000
	Business_id	8090
	user_id	9581
Checkin	Business_id	493
Photo	Business_id	6493
	Id	10000
Tip	User_id	537
	Business_id	3979
User	Id	10000
Friend	User_id	11
Elite_years	User_id	2780

SQL:

```
SELECT COUNT(DISTINCT(<column_name>)) FROM <table_name>
```

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?

Answer: no

SQL code used to arrive at answer:

```
    SELECT COUNT(*)
    FROM USER
    WHERE id is null OR 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: max: avg:
   a.min: 1   max: 5   avg: 3.7082

ii. Table: Business, Column: Stars min: max: avg:
   a.min: 1.0   max: 5.0   avg: 3.6549

iii. Table: Tip, Column: Likes min: max: avg:
   a.min: 0   max: 2   avg: 0.0144

iv. Table: Checkin, Column: Count min: max: avg:
   a.min: 1   max: 53   avg: 1.9414

v. Table: User, Column: Review_count min: max: avg:
   a.min: 0   max: 2000   avg: 24.2995

SQL:

```
SELECT MIN(<column_name>), MAX(<column_name>),
AVG(<column_name>)
    FROM <table_name>;
```

5. List the cities with the most reviews in descending order:

SQL code used to arrive at answer:

```
SELECT b.city, SUM(b.review_count) as review_total
    FROM Business b
    GROUP BY b.city
    ORDER BY review_count DESC;
```

Copy and Paste the Result Below:

```
+-----------------+--------------+
| city            | review_total |
+-----------------+--------------+
| 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:
SQL code used to arrive at answer:

```
SELECT b.stars, COUNT(review_count) as count
FROM business b
WHERE b.city == 'Avon'
GROUP BY b.stars
ORDER BY b.stars DESC;
```

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

```
+-------+-------+
| stars | count |
+-------+-------+
|   5.0 |     1 |
|   4.5 |     1 |
|   4.0 |     2 |
|   3.5 |     3 |
|   2.5 |     2 |
|   1.5 |     1 |
+-------+-------+
```

ii. Beachwood SQL code used to arrive at answer:

```
SELECT b.stars, COUNT(review_count) as count
FROM business b
WHERE b.city == 'Beachwood'
GROUP BY b.stars
ORDER BY b.stars DESC;
```

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

```
+-------+-------+
   | stars | count |
   +-------+-------+
   |   5.0 |     5 |
   |   4.5 |     2 |
   |   4.0 |     1 |
   |   3.5 |     2 |
   |   3.0 |     2 |
   |   2.5 |     1 |
   |   2.0 |     1 |
+-------+-------+
```

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:

```
  +--------+--------------+
  | name   | review_count |
  +--------+--------------+
  | Gerald |         2000 |
  | Sara   |         1629 |
  | Yuri   |         1339 |
  +--------+--------------+
```

8. Does posing more reviews correlate with more fans?

SQL:

```
SELECT name, review_count, fans, strftime('%Y-%m-%d',
yelping_since) Since
FROM user
ORDER BY fans DESC;
```

Please explain your findings and interpretation of the results:

It's not correlated. Because Amy has 503 fans and only 609 reviews, while Gerald has only 253 fans and has 2000 reviews. Gerald has a little bit more than 50% fans and 328% more of reviews than Amy.

```
+-----------+--------------+------+------------+
| name      | review_count | fans | Since      |
+-----------+--------------+------+------------+
| Amy       |          609 |  503 | 2007-07-19 |
| Mimi      |          968 |  497 | 2011-03-30 |
| Harald    |         1153 |  311 | 2012-11-27 |
| Gerald    |         2000 |  253 | 2012-12-16 |
| Christine |          930 |  173 | 2009-07-08 |
| Lisa      |          813 |  159 | 2009-10-05 |
| Cat       |          377 |  133 | 2009-02-05 |
| William   |         1215 |  126 | 2015-02-19 |
| Fran      |          862 |  124 | 2012-04-05 |
| Lissa     |          834 |  120 | 2007-08-14 |
| Mark      |          861 |  115 | 2009-05-31 |
| Tiffany   |          408 |  111 | 2008-10-28 |
| bernice   |          255 |  105 | 2007-08-29 |
| Roanna    |         1039 |  104 | 2006-03-28 |
| Angela    |          694 |  101 | 2010-10-01 |
| .Hon      |         1246 |  101 | 2006-07-19 |
| Ben       |          307 |   96 | 2007-03-10 |
| Linda     |          584 |   89 | 2005-08-07 |
| Christina |          842 |   85 | 2012-10-08 |
| Jessica   |          220 |   84 | 2009-01-12 |
| Greg      |          408 |   81 | 2008-02-16 |
| Nieves    |          178 |   80 | 2013-07-08 |
| Sui       |          754 |   78 | 2009-09-07 |
| Yuri      |         1339 |   76 | 2008-01-03 |
| Nicole    |          161 |   73 | 2009-04-30 |
+-----------+--------------+------+------------+
(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: Yes, there are 1780 reviews with the word 'love' and 232 with the word 'hate'.

Answer: SQL code used to arrive at answer:

```
  SELECT COUNT(id)
  FROM review
  WHERE UPPER(TEXT) LIKE '%LOVE%' ;
Result: 1780

  SELECT COUNT(id)
  FROM review
  WHERE UPPER(TEXT) LIKE '%HATE%' ;
Result: 232
```

10. Find the top 10 users with the most fans: SQL code used to arrive at answer:

```
SELECT name, fans, strftime('%Y-%m-%d', yelping_since)
Since
  FROM user
  ORDER BY fans DESC
  LIMIT 10;
```

Copy and Paste the Result Below:

```
+-----------+------+------------+
| name      | fans | Since      |
+-----------+------+------------+
| Amy       |  503 | 2007-07-19 |
| Mimi      |  497 | 2011-03-30 |
| Harald    |  311 | 2012-11-27 |
| Gerald    |  253 | 2012-12-16 |
| Christine |  173 | 2009-07-08 |
| Lisa      |  159 | 2009-10-05 |
| Cat       |  133 | 2009-02-05 |
| William   |  126 | 2015-02-19 |
| Fran      |  124 | 2012-04-05 |
| Lissa     |  120 | 2007-08-14 |
+-----------+------+------------+
```

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.

I am choosing: City: Toronto and Category: Restaurants.

Include your code.

```
# SQL To pick the city and category with more choices:
Select b.city, category, COUNT(b.id) total
FROM Business b
INNER JOIN category c ON b.id = c.business_id
GROUP BY b.city, c.category
HAVING COUNT(b.id) > 4;
```

Result:

```
+---------+-------------+-------+
| city    | category    | total |
+---------+-------------+-------+
| Phoenix | Restaurants |     6 |
| Toronto | Restaurants |    10 |
+---------+-------------+-------+
```

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

   Yes. The 2-3 starts opens 7days/week and start their shift earlier than the 4-5 stars. The 5 stars beside opening late, they open 5 days/week.

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

   Yes. The group with 4-5 stars (89 reviews) have almost double of reviews compared with the 2-3 stars restaurants with most reviews (47 reviews) and 1000% more reviews with the restaurants with less reviews (5).

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

No. They are in different neighbourhood and different and forward sortation area (FSA), FSA - the 3 first letters of postal code.

SQL code used for analysis:

```
SELECT b.city, category, h.hours, b.review_count,
CASE
WHEN stars BETWEEN 2 and 3 THEN '2-3 stars'
WHEN stars BETWEEN 4 and 5 THEN '4-5 stars'
END AS stars_group,
CASE
WHEN UPPER(hours) LIKE '%MONDAY%' THEN 'Monday'
WHEN UPPER(hours) LIKE '%TUESDAY%' THEN 'Tuesday'
WHEN UPPER(hours) LIKE '%WEDNESDAY%' THEN
'Wednesday'
WHEN UPPER(hours) LIKE '%THURSDAY%' THEN 'Thursday'
WHEN UPPER(hours) LIKE '%FRIDAY%' THEN 'Friday'
WHEN UPPER(hours) LIKE '%SATURDAY%' THEN 'Saturday'
WHEN UPPER(hours) LIKE '%SUNDAY%' THEN 'Sunday'
END AS week_day, b.neighborhood,
b.address, b.postal_code
FROM Business b
INNER JOIN category c ON b.id = c.business_id
INNER JOIN hours h ON b.id = h.business_id
WHERE b.city = 'Toronto' and c.category =
'Restaurants' AND b.stars IN (2,3,4,5)
GROUP BY b.id, h.hours
ORDER BY stars_group, week_day;
```

```
+---------+-------------+-----------------------+---
-----------+-------------+-----------+--------------
----------+-----------------------+-------------+
| city    | category    | hours                 |
review_count | stars_group | week_day  |
neighborhood           | address               |
postal_code |
+---------+-------------+-----------------------+---
-----------+-------------+-----------+--------------
----------+------------------------+------------+
| Toronto | Restaurants | Friday|10:30-21:00    |
47 | 2-3 stars   | Friday    | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Friday|9:00-4:00      |
34 | 2-3 stars   | Friday    | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Friday|11:00-23:00    |
5 | 2-3 stars   | Friday    | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Monday|10:30-21:00    |
47 | 2-3 stars   | Monday    | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Monday|9:00-23:00     |
34 | 2-3 stars   | Monday    | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Monday|11:00-23:00    |
5 | 2-3 stars   | Monday    | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Saturday|10:30-21:00  |
47 | 2-3 stars   | Saturday  | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Saturday|10:00-4:00   |
34 | 2-3 stars   | Saturday  | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Saturday|11:00-23:00  |
5 | 2-3 stars   | Saturday  | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Sunday|11:00-19:00    |
47 | 2-3 stars   | Sunday    | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Sunday|10:00-23:00    |
34 | 2-3 stars   | Sunday    | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Sunday|11:00-23:00    |
5 | 2-3 stars   | Sunday    | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Thursday|10:30-21:00  |
47 | 2-3 stars   | Thursday  | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Thursday|9:00-23:00   |
34 | 2-3 stars   | Thursday  | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Thursday|11:00-23:00  |
5 | 2-3 stars   | Thursday  | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Tuesday|10:30-21:00   |
47 | 2-3 stars   | Tuesday   | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Tuesday|9:00-23:00    |
34 | 2-3 stars   | Tuesday   | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Tuesday|11:00-23:00   |
5 | 2-3 stars   | Tuesday   | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Wednesday|10:30-21:00 |
47 | 2-3 stars   | Wednesday | Downtown Core
| 260 Yonge Street      | M4B 2L9     |
| Toronto | Restaurants | Wednesday|9:00-23:00  |
34 | 2-3 stars   | Wednesday | Entertainment
District | 270 Adelaide Street W | M5H 1X6     |
| Toronto | Restaurants | Wednesday|11:00-23:00 |
5 | 2-3 stars   | Wednesday | Downtown Core
| 389 Church Street     | M5B 2E5     |
| Toronto | Restaurants | Friday|18:00-23:00    |
89 | 4-5 stars   | Friday    | Niagara
| 169 Niagara Street    | M5V         |
| Toronto | Restaurants | Saturday|18:00-23:00  |
89 | 4-5 stars   | Saturday  | Niagara
| 169 Niagara Street    | M5V         |
| Toronto | Restaurants | Sunday|12:00-16:00    |
89 | 4-5 stars   | Sunday    | Niagara
| 169 Niagara Street    | M5V         |
| Toronto | Restaurants | Thursday|18:00-23:00  |
89 | 4-5 stars   | Thursday  | Niagara
| 169 Niagara Street    | M5V         |
+---------+-------------+-----------------------+---
-----------+-------------+-----------+--------------
----------+---------------------+-------------+
(Output limit exceeded, 25 of 26 total rows shown)
```

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 opened business have average stars and average funny bigger than the closed ones.

ii. Difference 2:

Closed business have average useful and cool greater than opened ones.

SQL code used for analysis:

```
SELECT
CASE WHEN b.is_open = 0 THEN 'Closed'
ELSE 'Open'
END AS 'Is Open?', count(b.is_open) AS 'Qty',
SUM(review_count) AS 'Qty Reviews',
AVG(review_count) AS 'AVG_Reviews',
AVG(r.stars) AS 'AVG_stars', AVG(r.useful),
AVG(r.funny), AVG(r.cool)
FROM business b
INNER JOIN review r ON b.id = r.business_id
GROUP BY b.is_open;
```

```
+----------+-----+-------------+---------------+----
-----------+----------------+----------------+------
----------+
| Is Open? | Qty | Qty Reviews |   AVG_Reviews |
AVG_stars |  AVG(r.useful) |   AVG(r.funny) |
AVG(r.cool) |
+----------+-----+-------------+---------------+----
-----------+----------------+----------------+------
----------+
| Closed   |  71 |        9217 | 129.816901408 |
3.64788732394 | 0.971830985915 | 0.211267605634 |
0.422535211268 |
| Open     | 565 |      175821 | 311.187610619 |
3.7610619469 | 0.856637168142 | 0.269026548673 |
0.387610619469 |
+----------+-----+-------------+---------------+----
-----------+----------------+----------------+------
----------+
```

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:

How the users are engaged with number of fans, review_count and other attributes.

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

The users with most fans are not related with the numbers of review_count or how many times the user votes: useful, funny or cool or how many years as Yelp's user.

Sometimes the engagement of one user is coming from other platforms or the user is engaged in other activities or way that it is not transparent with the data collected here at Yelp dataset.

iii. Output of your finished dataset: iv. Provide the SQL code you used to create your final dataset:

```
SELECT u.name
       ,u.fans
       ,u.review_count
       ,(2017 - strftime('%Y', u.yelping_since)) AS
Years_at_Yelp
       ,u.useful
       ,u.funny
       ,u.cool
       ,u.average_stars
FROM user u
ORDER BY u.fans DESC, u.review_count DESC;
```

```
+-----------+------+--------------+---------------+--------+--------+-----
---+---------------+
| name      | fans | review_count | Years_at_Yelp | useful |  funny |
cool | average_stars |
+-----------+------+--------------+---------------+--------+--------+-----
---+---------------+
| Amy       |  503 |          609 |            10 |   3226 |   2554 |
2751 |          3.21 |
| Mimi      |  497 |          968 |             6 |    257 |    138 |
159 |          4.05 |
| Harald    |  311 |         1153 |             5 | 122921 | 122419 |
122890 |           4.4 |
| Gerald    |  253 |         2000 |             5 |  17524 |   2324 |
15008 |           3.6 |
| Christine |  173 |          930 |             8 |   4834 |   6646 |
4321 |          3.69 |
| Lisa      |  159 |          813 |             8 |     48 |     13 |
6 |          4.09 |
| Cat       |  133 |          377 |             8 |   1062 |    672 |
1076 |          3.99 |
| William   |  126 |         1215 |             2 |   9363 |   9361 |
9370 |          4.41 |
| Fran      |  124 |          862 |             5 |   9851 |   7606 |
9344 |           4.1 |
| Lissa     |  120 |          834 |            10 |    455 |    150 |
342 |          3.68 |
| Mark      |  115 |          861 |             8 |   4008 |    570 |
2765 |          3.36 |
| Tiffany   |  111 |          408 |             9 |   1366 |    984 |
1279 |          4.09 |
| bernice   |  105 |          255 |            10 |    120 |    112 |
109 |          3.95 |
| Roanna    |  104 |         1039 |            11 |   2995 |   1188 |
636 |          3.71 |
| .Hon      |  101 |         1246 |            11 |   7850 |   5851 |
5104 |          3.14 |
| Angela    |  101 |          694 |             7 |    158 |    164 |
105 |          3.89 |
| Ben       |   96 |          307 |            10 |   1180 |   1155 |
1143 |           3.7 |
| Linda     |   89 |          584 |            12 |   3177 |   2736 |
3019 |          4.06 |
| Christina |   85 |          842 |             5 |    158 |     34 |
102 |           4.1 |
| Jessica   |   84 |          220 |             8 |   2161 |   2091 |
2067 |           4.1 |
| Greg      |   81 |          408 |             9 |    820 |    753 |
746 |          3.67 |
| Nieves    |   80 |          178 |             4 |   1091 |    774 |
940 |          3.64 |
| Sui       |   78 |          754 |             8 |      9 |     18 |
2 |          3.62 |
| Yuri      |   76 |         1339 |             9 |   1166 |    220 |
561 |          4.11 |
| Nicole    |   73 |          161 |             8 |     13 |     10 |
6 |          3.87 |
+-----------+------+--------------+---------------+--------+--------+-----
---+---------------+
(Output limit exceeded, 25 of 10000 total rows shown)
```