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.

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

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

ii. Hours = 1562

iii. Category = 2643

iv. Attribute = 1115

v. Review = 10000

vi. Checkin = 493

vii. Photo = 10000

viii. Tip = 3979 (using business_id), 537 (using user_id)

ix. User = 10000

x. Friend = 11

xi. Elite_years = 2780

\

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 count(*)-count(id),
       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: 3.6549

ii. Table: Business, Column: Stars

min: 1 max: 5 avg: 3.7082

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

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

SQL code used to arrive at answer:

"""

```
select city,
       sum(review_count) as src
from business
group by city
order by src desc
```

"""

Copy and Paste the Result Below:

""

```
+-----------------+-------+
| city            |   src |
+-----------------+-------+
| 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,
       count(*)
from business b
where city='Avon'
group by stars

"""
```

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

```
"""
+-------+----------+
| stars | 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,
       count(*)
from business b
where city='Beachwood'
group by stars

"""
```

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

```
"""
+-------+----------+
| stars | 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 id,
       name,
       review_count
from user
order by review_count desc

"""
```

-->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:

ans)

No, number of reviews of a particular user does not strongly correlate to the number of fans. Though the number of fans roughly increase with the review_count,

but after review_count surpassing a high thresold(1000 approx.) the fans of some users were not nearly as high enough as their neighbors when arranged by

descending order of review_count.

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 count(*)
from review
--where text like '%hate%'
--where text like '%love%'
(executed twice, one for each)
```

"""

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

SQL code used to arrive at answer:

```
"""
select name,
       review_count,
       fans
from user
order by fans desc

"""
```

Copy and Paste the Result Below:

```
"""
+-----------+--------------+------+
| name      | review_count | fans |
+-----------+--------------+------+
| Amy       |        609 |  503 |
| Mimi      |        968 |  497 |
| Harald    |       1153 |  311 |
| Gerald    |       2000 |  253 |
| Christine |        930 |  173 |
| Lisa      |        813 |  159 |
| Cat       |        377 |  133 |
| William   |       1215 |  126 |
| Fran      |        862 |  124 |
| Lissa     |        834 |  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.

City - Las Vegas

Category - Food

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

-- Yes, the 2-3 star ones have a larger span of time in which they are open compared to the 4-5 stars ones.

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

-- Yes, the 2-3 star ones have lesser reviews compared to the 4-5 stars ones.

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

-- Yes, the 4-5 stars ones lie in the Southeast neighborhood whereas the 2-3 stars ones lie in the Eastside neighborhood.

SQL code used for analysis:

```
"""
select
      case
          when b.stars between 2 and 3 then 1
          when b.stars between 4 and 5 then 2
          else 0
```

```
      end as c,
      --sum(review_count),
      h.hours,
      b.neighborhood
from business b
inner join hours h
on b.id=h.business_id
where b.id in
      (
       select business_id
       from category
       where category='Food'
      )
      and
      b.city='Las Vegas'
--group by c
--order by c
```

(commented code was used selectively to answer the three questions --> add only the 2nd commented code for i., add all three for ii., add only the 3rd for iii.)

"""

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 ones that are open have average stars slightly higher than those closed.

ii. Difference 2:

-- The ones that are open have a very high total number of review_count, the ones closed have comparatively very low total number of review_count.

SQL code used for analysis:

```
"""
select is_open,
       avg(stars),
       sum(review_count)
from business b
group by b.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.

i. Indicate the type of analysis you chose to do:

-- How the distribution in sentiments, total reviews of a review affect the overall star rating and determines the (open or closed) of a business.

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

-- Count of different sentiments per review per business for stars correlation, sum of review counts, sum of is_open - sum of all in that star category.

(above data is grouped by stars)

The sentiment if reviews is important as for example a funny review has much more impact than a plain review, even if the funny review didn't talk about the business

in much detail compared to a plain review , it leaves a bigger impact on the one reading it.

The review count is a direct indicator of the popularity etc.

iii. Output of your finished dataset:

-- Some insights:-

-- The reviews where sentiments were mentioned correspondingly led to popular(high review count) businesses.

-- 5 star businesses had much lesser number of reviews.

-- The count for sentiments and reviews peaked for moderately good businesses(3.5 or 4.0 stars).

-- The businesses in 2.5 star category tend to close down more.

"""

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

| count(*) - sum(b.is_open) | sbr | src | sru | srf | abs |

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

| 0 | 6 | 0 | 2 | 0 | 1.0 |

| 0 | 34 | 0 | 2 | 0 | 1.5 |

| 1 | 851 | 2 | 12 | 3 | 2.0 |

| 1 | 1598 | 10 | 22 | 1 | 5.0 |

| 14 | 7138 | 14 | 49 | 13 | 2.5 |

| 5 | 14480 | 20 | 64 | 22 | 3.0 |

| 12 | 24791 | 53 | 101 | 22 | 4.5 |

| 15 | 69319 | 70 | 154 | 42 | 4.0 |

| 23 | 66821 | 80 | 147 | 64 | 3.5 |

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

"""

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

1)

"""

```
select count(*) - sum(b.is_open), -- gives the total businesses in
the particular category which were closed
       sum(b.review_count) as sbr,
       sum(r.cool) as src,
       sum(r.useful) as sru,
       sum(r.funny) as srf,
       avg(b.stars) as abs
from business b
inner join review r
on b.id=r.business_id
group by b.stars
order by src,sru,srf desc

"""
```