Profiling and Analyzing the Yelp Dataset Coursera Worksheet

Part 1: Yelp Dataset Profiling and Understanding

1. Profiling the data by finding the total number of records for each of the tables below:

```
SELECT COUNT(*)
FROM Attribute
```

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. Finding the total distinct records by either the foreign key or primary key for each table.

```
SELECT  COUNT(DISTINCT business_id)
FROM Hours
```

i. Business = id(PK): 10000
ii. Hours = business_id(FK): 1562
iii. Category = business_id(FK): 2643
iv. Attribute = business_id(FK): 1115
v. Review = id(PK):10000, business_id(FK): 8090, user_id(FK): 9581
vi. Checkin = business_id(FK): 493
vii. Photo = id(PK): 10000, business_id(FK): 6493
viii. Tip = user_id(FK): 537, business_id(FK): 3979
ix. User = id(PK): 10000
x. Friend = user_id(FK): 11
xi. Elite_years = user_id(FK): 2780

3. Checking the Null values in the Users table.

=> "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

/*
PRAGMA table_info(user)
*/
```

4. Displaying the smallest (minimum), largest (maximum), and average (mean) value for the following fields:

```
  SELECT AVG(column)
  FROM table
```

i. Table: Review, Column: Stars

min: 1 max: 5 avg: 3.7082

ii. Table: Business, Column: Stars

min: 1   max: 5 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

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

SQL code used to arrive at answer:

```
SELECT city,SUM(review_count) TOTAL_REVIEWS
FROM business
GROUP BY city
ORDER BY TOTAL_REVIEWS DESC
```

```
+-----------------+---------------+
| city            | TOTAL_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. Finding the distribution of star ratings to the business in the following cities:

i. Avon

SQL code used to arrive at answer:

```
SELECT stars, COUNT(*) AS COUNT
FROM business
WHERE city='Avon'
GROUP BY stars
```

```
+-------+-------+
| 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(*) AS COUNT
FROM business
WHERE city='Beachwood'
GROUP BY stars
```

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

7. Finding the top 3 users based on their total number of reviews:

SQL code used to arrive at answer:

```
SELECT id,review_count
FROM user
ORDER BY review_count DESC
LIMIT 3
```

```
+--------+------------------------+--------------+
| name   | id                     | REVIEW_COUNT |
+--------+------------------------+--------------+
| Gerald | -G7Zkl1wIWBBmD0KRy_sCw |         2000 |
| Sara   | -3s52C4zL_DHRK0ULG6qtg |         1629 |
| Yuri   | -8lbUNlXVSoXqaRRiHiSNg |         1339 |
+--------+------------------------+--------------+
```

8. Does posing more reviews correlate with more fans.

```
SELECT fans,review_count AS REVIEW_COUNT
FROM user
ORDER BY review_count ASC
LIMIT 25
```

```
+------+--------------+
| fans | REVIEW_COUNT |
+------+--------------+
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            0 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
|    0 |            1 |
+------+--------------+
```

INTERPRETATION: It can be seen that for the people having less number of reviews, there are less fans.

```
SELECT fans,review_count AS REVIEW_COUNT
FROM user
ORDER BY review_count ASC
LIMIT 25
```

```
+------+--------------+
| fans | REVIEW_COUNT |
+------+--------------+
|  253 |         2000 |
|   50 |         1629 |
|   76 |         1339 |
|  101 |         1246 |
|  126 |         1215 |
| 311 |         1153 |
|  16 |         1116 |
| 104 |         1039 |
| 497 |          968 |
| 173 |          930 |
|  38 |          904 |
|  43 |          864 |
| 124 |          862 |
| 115 |          861 |
|  85 |          842 |
|  37 |          836 |
| 120 |          834 |
| 159 |          813 |
|  61 |          775 |
|  78 |          754 |
|  35 |          702 |
|  10 |          696 |
| 101 |          694 |
|  25 |          676 |
|  45 |          675 |
+------+--------------+
```

INTERPRETATION: For the people having large number of reviews, the fans are not correlated. It can be seen in the above output.

Overall, There is no correlation between the two attributes.

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

There are more reviews containing love. Love appears approximately 9 times more than the hate in the given reviews.

SQL code used to arrive at answer:

```
SELECT COUNT(*) AS LOVE_COUNT
FROM review
WHERE LOWER(text) LIKE '%love%'

+------------+
| LOVE_COUNT |
+------------+
|       1780 |
+------------+

SELECT COUNT(*) AS HATE_COUNT
FROM review
WHERE LOWER(text) LIKE '%hate%'

+------------+
| HATE_COUNT |
+------------+
|        232 |
+------------+
```

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

SQL code used to arrive at answer:

```
SELECT name, id, fans
FROM user
ORDER BY  fans DESC
LIMIT 10
```

```
+-----------+------------------------+------+
| name      | id                     | fans |
+-----------+------------------------+------+
| Amy       | -9I98YbNQnLdAmcYfb324Q |  503 |
| Mimi      | -8EnCioUmDygAbsYZmTeRQ |  497 |
| Harald    | --2vR0DIsmQ6WfcSzKWigw |  311 |
| Gerald    | -G7Zkl1wIWBBmD0KRy_sCw |  253 |
| Christine | -0IiMAZI2SsQ7VmyzJjokQ |  173 |
| Lisa      | -g3XIcCb2b-BD0QBCcq2Sw |  159 |
| Cat       | -9bbDysuiWeo2VShFJJtcw |  133 |
| William   | -FZBTkAZEXoP7CYvRV2ZwQ |  126 |
| Fran      | -9da1xk7zgnnfO1uTVYGkA |  124 |
| Lissa     | -lh59ko3dxChBSZ9U7LfUw |  120 |
+-----------+------------------------+------+
```

Part 2: Inferences and Analysis

1. Picking one city and category of choice and group the businesses in that city or category by their overall star rating. Then Comparing the businesses with 2-3 stars to the businesses with 4-5 stars.

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

The groups having four-five stars have less number of hours than groups having two-three stars.

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

No, No correlation is found.
Some groups have more number of reviews than the other group and some have similar number of reviews to the other groups.

iii. Inference from the location data provided between these two groups?

There is nothing that can be infered from the location data provided between these two groups because of the distinct zip codes.

SQL code used for analysis:

```
SELECT Biz.name,
   Biz.review_count,
   Hr.hours,
   postal_code,
     CASE
    WHEN hours LIKE "%monday%" THEN 001
    WHEN hours LIKE "%tuesday%" THEN 002
    WHEN hours LIKE "%wednesday%" THEN 003
    WHEN hours LIKE "%thursday%" THEN 004
    WHEN hours LIKE "%friday%" THEN 005
    WHEN hours LIKE "%saturday%" THEN 006
    WHEN hours LIKE "%sunday%" THEN 007
     END AS day_no,
     CASE
     WHEN Biz.stars BETWEEN 2 AND 3 THEN '2-3 stars'
     WHEN Biz.stars BETWEEN 4 AND 5 THEN '4-5 stars'
     END AS star_rating
 FROM business Biz INNER JOIN hours Hr
 ON Biz.id = Hr.business_id
 INNER JOIN category C
 ON C.business_id = Biz.id
 WHERE (Biz.city == 'Las Vegas'
 AND
 C.category LIKE 'shopping')
 AND
 (Biz.stars BETWEEN 2 AND 3
 OR
 Biz.stars BETWEEN 4 AND 5)
 GROUP BY stars,day_no
 ORDER BY day_no,star_rating ASC
```

2. Group business based on the ones that are open and the ones that are closed. Differences between the ones that are still open and the ones that are closed.

i. SUM of the review count and the average of the review count for closed businesses is much lower than the open ones.

ii. AVG of the stars is almost similar for both the set of businesses with a negligible difference of 0.15 i.e. open businesses have average star rating exceeding the closed businesses by 0.15.

SQL code used for analysis:

```
SELECT is_open, COUNT(*), SUM(review_count), AVG(review_count), AVG(Stars)
FROM business
GROUP BY is_open
```

```
+---------+----------+-------------------+-------------------+---------------+
| is_open | COUNT(*) | SUM(review_count) | AVG(review_count) |    AVG(Stars) |
+---------+----------+-------------------+-------------------+---------------+
|       0 |     1520 |             35261 |     23.1980263158 | 3.52039473684 |
|       1 |     8480 |            269300 |     31.7570754717 | 3.67900943396 |
+---------+----------+-------------------+-------------------+---------------+
```

3. ANALYSIS:

i.
We are predicting the sentiment for each category.

ii.
We predict the sentiment of each category for all the businesses/irrespective of the businesses.
Reason behind choosing the dataset is to define the sentiment about the partiicular category in the country.
This can eventually help the new businesses enter the market providing better services and the old businesses to improve the services
in the category.
stars, category and business id are used to find the average star rating for each category.

Rule base used for classifying the category is:
If 0<= Avg_stars <=2 : Poor
If 2< Avg_stars <=3 : Average
If 3< Avg_stars <=4 : Good
If 4< Avg_stars <=5 : Best

iii. Output:

```
   +------------------------+---------------+-----------+
 | category               | AVERAGE_STARS | SENTIMENT |
 +------------------------+---------------+-----------+
 | Accessories            |           4.0 | GOOD      |
 | Active Life            |          4.15 | BEST      |
 | Acupuncture            |           4.5 | BEST      |
 | American (New)         | 3.33333333333 | GOOD      |
 | American (Traditional) | 3.81818181818 | GOOD      |
 | Apartments             |           3.5 | GOOD      |
 | Arabian                |           5.0 | BEST      |
 | Arcades                |           4.0 | GOOD      |
 | Architects             |           4.5 | BEST      |
 | Architectural Tours    |           4.5 | BEST      |
 | Art Galleries          | 4.33333333333 | BEST      |
 | Arts & Crafts          |          4.25 | BEST      |
 | Arts & Entertainment   |           4.0 | GOOD      |
 | Asian Fusion           |           3.5 | GOOD      |
 | Auto Detailing         |           5.0 | BEST      |
 | Auto Repair            |         4.625 | BEST      |
 | Automotive             |           4.5 | BEST      |
 | Bagels                 |           3.0 | AVERAGE   |
 | Bakeries               |           4.1 | BEST      |
 | Banks & Credit Unions  |           1.5 | POOR      |
 | Barbeque               |          3.75 | GOOD      |
 | Bars                   |           3.5 | GOOD      |
 | Beaches                |           3.5 | GOOD      |
 | Beauty & Spas          | 3.88461538462 | GOOD      |
 | Beer                   |           4.0 | GOOD      |
 +------------------------+---------------+-----------+
 (Output limit exceeded, 25 of 257 total rows shown)
```

iv. QUERY:

```
select
category, AVG(stars) AS AVERAGE_STARS,
CASE
WHEN AVG(stars)>=0 AND AVG(stars)<=2 THEN 'POOR'
WHEN AVG(stars)>2 AND AVG(stars)<=3 THEN 'AVERAGE'
WHEN AVG(stars)>3 AND AVG(stars)<=4 THEN 'GOOD'
WHEN AVG(stars)>4 AND AVG(stars)<=5 THEN 'BEST'
ELSE 'INVALID CALCULATION !!!'
END SENTIMENT
FROM category INNER JOIN business
ON business.id=category.business_id
GROUP BY category
```