Data Scientist Role Play: Profiling and Analyzing the Yelp Dataset Coursera Worksheet

Shahriar Ahmad Fahim
shahriarahmadf@gmail.com

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:

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

```
-- substitute attribute by  table_name
SELECT COUNT(*)
FROM attribute
```

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.

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

Secondary keys are denoted in the ER-Diagram with a red icon.

Table-1

	Primary key (#records)	Foreign key(s) (#records)
i. Business	id (10,000)	X
ii. Hours	X	business_id (1,562)
iii. Category	X	business_id (2,643)
iv. Attribute	X	business_id (1,115)
v. Review	id (10,000)	business_id (8,090), user_id (9,581)
vi. Checkin	X	business_id (493)
viii. Tip	X	user_id (537), business_id (3,979)
ix. User	id (10,000)	X
x. Friend	X	user_id (11)
xi. Elite_years	X	user_id (2,780)

```
-- substitute id by primary_key or forign_key and business by table_name
SELECT COUNT(DISTINCT id)
FROM business
```

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
    id IS NULL OR -- this line is excludable since id is a primary key
    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_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 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

```
-- substitute review_count by Column and user by Table
SELECT
    MIN(review_count)
    ,MAX(review_count)
    ,AVG(review_count)
FROM user
```

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

SQL code used to arrive at answer:

```
SELECT
    city
    ,review_count
FROM business
ORDER BY review_count DESC -- sorted in descending order of review_count
```

Copy and Paste the Result Below:

```
+------------+--------------+
| city       | review_count |
+------------+--------------+
| Las Vegas  |         3873 |
| Montréal   |         1757 |
| Gilbert    |         1549 |
| Las Vegas  |         1410 |
| Las Vegas  |         1389 |
| Las Vegas  |         1252 |
| Las Vegas  |         1116 |
| Las Vegas  |         1084 |
| Las Vegas  |          961 |
| Gilbert    |          902 |
| Las Vegas  |          864 |
| Scottsdale |          823 |
| Las Vegas  |          821 |
| Las Vegas  |          786 |
| Henderson  |          785 |
| Toronto    |          778 |
| Las Vegas  |          768 |
| Las Vegas  |          758 |
| Scottsdale |          726 |
| Cleveland  |          723 |
| Las Vegas  |          720 |
| Charlotte  |          715 |
| Phoenix    |          711 |
| Las Vegas  |          706 |
| Phoenix    |          700 |
+------------+--------------+
```

(Output limit exceeded, 25 of 10000 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 AS Star_Rating_For_City_Avon
    ,COUNT(stars) -- count how many times same star given for the city
FROM business
WHERE city = 'Avon'
GROUP BY stars -- group by same star ratings
```

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

```
+---------------------------+--------------+
| Star_Rating_For_City_Avon | COUNT(stars) |
+---------------------------+--------------+
|                       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 AS Star_Rating_For_City_Avon
    ,COUNT(stars) -- count how many times same star given for the city
FROM business
WHERE city = 'Beachwood'
GROUP BY stars -- group by same star ratings
```

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

```
+---------------------------+--------------+
| Star_Rating_For_City_Avon | COUNT(stars) |
+---------------------------+--------------+
|                       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
    name AS 'name_of_user'
    ,review_count
FROM user
ORDER BY review_count DESC -- sort in descending order of reviews
LIMIT 3 -- view only top 3
```

Copy and Paste the Result Below:

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

8. Does posing more reviews correlate with more fans?

Posing more reviews correlate positively very weakly with more fans.

Please explain your findings and interpretation of the results:

SQL Code used to show review_count vs fans record in descending order:

```
SELECT
    name AS 'name_of_user'
    ,review_count
FROM user
ORDER BY review_count DESC -- sort in descending order of reviews
LIMIT 15 OFFSET 0; -
- LIMIT = how many records to show, OFFSET = start from which record
```

LIMIT 15 OFFSET 0

```
+--------------+--------------+
| name_of_user | review_count |
+--------------+--------------+
| Gerald       |         2000 |
| Sara         |         1629 |
| Yuri         |         1339 |
| .Hon         |         1246 |
| William      |         1215 |
| Harald       |         1153 |
| eric         |         1116 |
| Roanna       |         1039 |
| Mimi         |          968 |
| Christine    |          930 |
| Ed           |          904 |
| Nicole       |          864 |
| Fran         |          862 |
| Mark         |          861 |
| Christina    |          842 |
+--------------+--------------+
```

By changing the OFFSET, or the starting position of record to show, we visualize two more intervals of records in the following Table.

LIMIT 15 OFFSET 75 (left) OFFSET 250 (right)

```
+--------------+--------------+------+  +--------------+--------------+------+
| name_of_user | review_count | fans |  | name_of_user | review_count | fans |
+--------------+--------------+------+  +--------------+--------------+------+
| Steven       |          400 |   35 |  | Akansha      |          195 |   12 |
| Justin       |          399 |   17 |  | Kelly        |          195 |   10 |
| Jessica      |          398 |   14 |  | Marisa       |          195 |   18 |
| Jen          |          388 |   27 |  | Brey         |          194 |    9 |
| Alias        |          379 |   14 |  | Paula        |          193 |    2 |
| Katrina      |          378 |   18 |  | John         |          191 |   20 |
| Cat          |          377 |  133 |  | Melissa      |          189 |    9 |
| Princeton    |          376 |   64 |  | Becca        |          189 |   21 |
| Sara         |          373 |   15 |  | Nomadic      |          188 |    2 |
| Jennifer     |          365 |    5 |  | Jacqueline   |          188 |    2 |
| Linda        |          365 |   40 |  | Mira         |          186 |    6 |
| Renee        |          364 |   54 |  | Tri          |          185 |    5 |
| Song         |          363 |   33 |  | Allison      |          184 |   22 |
| C            |          363 |   17 |  | Vrati        |          183 |    1 |
| G            |          359 |    0 |  | Erika        |          183 |   12 |
 +-------------+--------------+-----+   +--------------+--------------+------+
```

In record visualization, the records are viewed in the sorted descending order of review_count. It is observed that the number of fans in the intervals appear a bit randomly with no general increasing or decreasing trend of fan number with decrease of review_count. However, if we find the average of fan number for each of these intervals using the code below, we get more interesting observations.

SQL Code used to find average of number of fans for the interval of review_count:

```sql
SELECT
    AVG(fans) AS average_fans_from_1_to_15
FROM ( -- from the subquery
    SELECT fans, review_count
    FROM user
    ORDER BY review_count DESC -- sorted in desc order of reviews
    LIMIT 15 OFFSET 0 -- show 15 records starting from 1);
```

```
average_fans_from_1_to_15 = 140.8
average_fans_from_76_to_90 = 32.4
average_fans_from_251_to_265 = 10.067 (approx.)
average_fans_from_451_to_465 = 7.267 (approx.)
average_fans_from_751_to_765 = 3.667 (approx.)
average_fans_from_1501_to_1515 = 1.533 (approx.)
average_fans_from_3001_to_3015 = 0.067 (approx.)
```

It is easy to observe that, with interval of 15, the average of each of these intervals are decreasing, meaning- the number of fans is decreasing on average with decreasing intervals of review_count with review_count being in descending order in all intervals. So, we interpret that there is a weak positive correlation between review_count and number of fans. However, further concrete conclusion on this correlation can be obtained investigating the correlation coefficient between these two columns, which is outside the scope of this report.

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” in them than “hate”.

```
+----------------------+   +----------------------+
| Reviews Having love |   | Reviews Having hate |
+----------------------+   +----------------------+
|                 1780 |   |                  232 |
+----------------------+   +----------------------+
```

SQL code used to arrive at answer:

```
SELECT
    COUNT(text) AS 'Reviews Having love' -- count how many texts are returned
FROM review
WHERE text like '%love%' -- checking if text has love in it using wildcard
```

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

SQL code used to arrive at answer:

```
SELECT
    name AS 'UserName'
    ,fans AS 'NumberOfFans'
from user
ORDER BY fans DESC -- sort in descending order of fan number
LIMIT 10 -- show top ten users
```

Copy and Paste the Result Below:

```
+-----------+--------------+
| UserName  | NumberOfFans |
+-----------+--------------+
| Amy       |          503 |
| Mimi      |          497 |
| Harald    |          311 |
| Gerald    |          253 |
| Christine |          173 |
| Lisa      |          159 |
| Cat       |          133 |
| William   |          126 |
| Fran      |          124 |
| 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.

First, let us investigate how many unique (distinct) categories and unique (distinct) cities are there in the category table and business city respectively following the below code structure:

```
select count(DISTINCT category) --replace category by column_name
from category -- replace category by table_name
```

```
+--------------------------+   +----------------------+
| COUNT(DISTINCT category) |   | COUNT(DISTINCT city) |
+--------------------------+   +----------------------+
|                      712 |   |                  362 |
+--------------------------+   +----------------------+
```

```
select (city), count(city)
from business
GROUP BY city
ORDER BY count(city) DESC
LIMIT 10
```

We visualize the highest category and highest city in the whole dataset.

```
+------------------+-----------------+   +------------+-------------+
| category         | count(category) |   | city       | count(city) |
+------------------+-----------------+   +------------+-------------+
| Restaurants      |             912 |   | Las Vegas  |        1561 |
| Food             |             425 |   | Phoenix    |        1001 |
| Shopping         |             405 |   | Toronto    |         985 |
| Beauty & Spas    |             224 |   | Scottsdale |         497 |
| Nightlife        |             214 |   | Charlotte  |         468 |
| Home Services    |             203 |   | Pittsburgh |         353 |
| Health & Medical |             196 |   | Montréal   |         337 |
| Bars             |             185 |   | Mesa       |         304 |
| Local Services   |             166 |   | Henderson  |         274 |
| Automotive       |             163 |   | Tempe      |         261 |
+------------------+-----------------+   +------------+-------------+
```

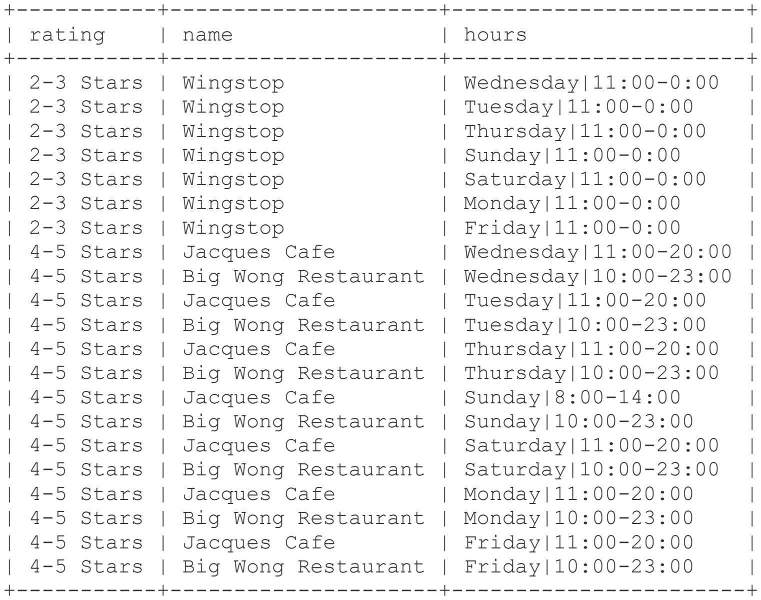
We choose the city Las Vegas and the category Restaurants.

```
SELECT
    CASE
        WHEN b.stars >= 4 THEN '4-5 Stars'
        WHEN (b.stars >= 2 AND b.stars <= 3) THEN '2-3 Stars'
        END AS rating

    , COUNT(*) AS rating_count
FROM business b

WHERE b.city LIKE '%las%'
GROUP BY rating
```

```
+-----------+--------------+
|    rating | rating_count |
+-----------+--------------+
|      None |          337 |
| 2-3 Stars |          430 |
| 4-5 Stars |          869 |
+-----------+--------------+
```

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

Yes, 2-3 Stars restaurant named Wingstop in Las Vegas everyday have the same opening to closing hours - 11:00 to 0:00. On the other hand, the 4-5 stars restaurants Jacques Café and Big Wong Restaurant have varying range of opening hours. So, the distribution of hours is different.



```
+-----------+---------------------+-----------------------+
| rating    | name                | hours                 |
+-----------+---------------------+-----------------------+
| 2-3 Stars | Wingstop            | Wednesday|11:00-0:00  |
| 2-3 Stars | Wingstop            | Tuesday|11:00-0:00    |
| 2-3 Stars | Wingstop            | Thursday|11:00-0:00   |
| 2-3 Stars | Wingstop            | Sunday|11:00-0:00     |
| 2-3 Stars | Wingstop            | Saturday|11:00-0:00   |
| 2-3 Stars | Wingstop            | Monday|11:00-0:00     |
| 2-3 Stars | Wingstop            | Friday|11:00-0:00     |
| 4-5 Stars | Jacques Cafe        | Wednesday|11:00-20:00 |
| 4-5 Stars | Big Wong Restaurant | Wednesday|10:00-23:00 |
| 4-5 Stars | Jacques Cafe        | Tuesday|11:00-20:00   |
| 4-5 Stars | Big Wong Restaurant | Tuesday|10:00-23:00   |
| 4-5 Stars | Jacques Cafe        | Thursday|11:00-20:00  |
| 4-5 Stars | Big Wong Restaurant | Thursday|10:00-23:00  |
| 4-5 Stars | Jacques Cafe        | Sunday|8:00-14:00     |
| 4-5 Stars | Big Wong Restaurant | Sunday|10:00-23:00    |
| 4-5 Stars | Jacques Cafe        | Saturday|11:00-20:00  |
| 4-5 Stars | Big Wong Restaurant | Saturday|10:00-23:00  |
| 4-5 Stars | Jacques Cafe        | Monday|11:00-20:00    |
| 4-5 Stars | Big Wong Restaurant | Monday|10:00-23:00    |
| 4-5 Stars | Jacques Cafe        | Friday|11:00-20:00    |
| 4-5 Stars | Big Wong Restaurant | Friday|10:00-23:00    |
+-----------+---------------------+-----------------------+
```

```
SELECT
    CASE
        WHEN b.stars >= 4 THEN '4-5 Stars'
        WHEN (b.stars >= 2 AND b.stars <= 3) THEN '2-3 Stars'
        END AS rating
    , b.name
    , h.hours

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 = 'Las Vegas'
    AND c.category = 'Restaurants'
    AND rating IS NOT NULL
ORDER BY rating, hours DESC
```

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

Yes, the number of reviews is different for two groups- 2-3 stars and 4-5 stars. 4-5 stars have a much higher number of ratings for each restaurants.

```
+-----------+---------------------+--------------+
| rating    | name                | review_count |
+-----------+---------------------+--------------+
| 2-3 Stars | Wingstop            |          123 |
| 4-5 Stars | Jacques Cafe        |          168 |
| 4-5 Stars | Big Wong Restaurant |          768 |
+-----------+---------------------+--------------+
```

```
SELECT
    CASE
        WHEN b.stars >= 4 THEN '4-5 Stars'
        WHEN (b.stars >= 2 AND b.stars <= 3) THEN '2-3 Stars'
        END AS rating
    , b.name
    , b.review_count

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 = 'Las Vegas'
    AND c.category = 'Restaurants'
    AND rating IS NOT NULL
GROUP BY b.review_count
```

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

It can be inferred that, when we no longer care about hours data, another restaurant of 4-5 stars can be seen. Mainly, it can be observed that not many 2-3 stars or 4-5 stars restaurants are there in Las Vegas city.

```
+-----------+---------------------+--------------+--------------------------------+--------------+-------------+----------+-----------+-------+---------+
| rating    | name                | review_count | address                        | neighborhood | postal_code | latitude | longitude | state | is_open |
+-----------+---------------------+--------------+--------------------------------+--------------+-------------+----------+-----------+-------+---------+
| 4-5 Stars | Big Wong Restaurant |          768 | 5040 Spring Mountain Rd        | Chinatown    | 89146       |  36.1267 |   -115.21 | NV    |       1 |
| 4-5 Stars | Hibachi-San         |            3 | 3480 S Maryland Pkwy           | Eastside     | 89169       |  36.1259 |  -115.135 | NV    |       0 |
| 4-5 Stars | Jacques Cafe        |          168 | 1910 Village Center Cir, Unit 1 | Summerlin    | 89134       |  36.1933 |  -115.304 | NV    |       0 |
| 2-3 Stars | Wingstop            |          123 | 5045 W Tropicana Ave           |              | 89103       |  36.1003 |   -115.21 | NV    |       1 |
+-----------+---------------------+--------------+--------------------------------+--------------+-------------+----------+-----------+-------+---------+
```

SQL code used for analysis:

```
SELECT
    CASE
        WHEN b.stars >= 4 THEN '4-5 Stars'
        WHEN (b.stars >= 2 AND b.stars <= 3) THEN '2-3 Stars'
        END AS rating
    , b.name
    , b.review_count
    , b.address
    , b.neighborhood
    , b.postal_code
    , b.latitude
    , b.longitude
    , b.state
    , b.is_open

FROM business b
    INNER JOIN category c
        ON b.id = c.business_id
WHERE b.city = 'Las Vegas'
    AND c.category = 'Restaurants'
    AND rating IS NOT NULL
GROUP BY b.name
```

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: Most businesses are open (8,480 out of 10,000), and comparing open and closed businesses that have reviews, open businesses had their latest review on July 2017 whereas closed businesses got last review on July 2016 by latest, which is one year before.

```
+--------------------------------------+  +----------------------------------------+
| Latest_reviews_of_open_business_dates |  | Latest_reviews_of_closed_business_dates |
+--------------------------------------+  +----------------------------------------+
| 2017-07-24 00:00:00                  |  | 2016-07-09 00:00:00                    |
| 2017-07-18 00:00:00                  |  | 2016-04-16 00:00:00                    |
| 2017-07-18 00:00:00                  |  | 2016-03-30 00:00:00                    |
| 2017-07-12 00:00:00                  |  | 2015-12-31 00:00:00                    |
| 2017-07-12 00:00:00                  |  | 2015-11-15 00:00:00                    |
+--------------------------------------+  +----------------------------------------+
```

ii. Difference 2: Open businesses have a higher average reviews than closed ones.

```
+---------+--------------------+  +---------+--------------------+
| is_open |  AVG(review_count) |  | is_open |  AVG(review_count) |
+---------+--------------------+  +---------+--------------------+
|       1 | 31.757075471698112 |  |       0 | 23.198026315789473 |
+---------+--------------------+  +---------+--------------------+
```

SQL code used for analysis:

```
SELECT
    r.date AS 'Latest_reviews_of_open_business_dates'
FROM business b
INNER JOIN review r -- businesses that has reviews
    ON b.id = r.business_id
WHERE is_open = 1 -- open business
ORDER BY r.date DESC
LIMIT 5
```

```
SELECT
    is_open
    ,AVG(review_count)
    -- count average of review_count for open business
FROM business
WHERE is_open = 1
```

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: Find the noisiest restaurant with its characteristics

ii. Write 1-2 brief paragraphs on the type of data you will need for your analysis and why you chose that data:
First, different categories of businesses were visualized in descending order. Restaurants were found to be highest in terms of number of businesses. Noisy restaurants may often be attractive to young people looking for a hangout, or one to avoid for people who want a calm environment. So, the noisiest restaurant was found through querying the attribute with name NoiseLevel. The only restaurant that was found was Big Smoke Burger. Then by making inner joins with some other tables, all characteristics of this very loud noisy restaurant was explored.

iii. Output of your finished dataset:

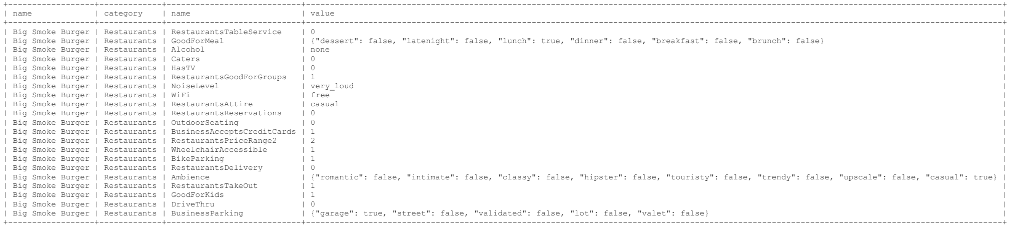
Visualization of different categories of businesses with number

```
+----------------------+------------------------+
| Number of businesses | category               |
+----------------------+------------------------+
|                   71 | Restaurants            |
|                   30 | Shopping               |
|                   23 | Food                   |
|                   20 | Nightlife              |
|                   17 | Bars                   |
|                   17 | Health & Medical       |
|                   16 | Home Services          |
|                   13 | Beauty & Spas          |
|                   12 | Local Services         |
|                   11 | American (Traditional) |
+----------------------+------------------------+
```

Finding the noisiest restaurant

```
+------------------------+------------------------+----------------------------------------+-------------+------------+-----------+

| id                     | business_id            | name                                   | category    | name       | value     |
+------------------------+------------------------+----------------------------------------+-------------+------------+-----------+
| 1AxEmgv8Dsr3iU9Aa40jPw | 1AxEmgv8Dsr3iU9Aa40jPw | Nabers Music, Bar & Eats               | Restaurants | NoiseLevel | average   |
| -2q4dnUw0gGJniGW2aPamQ | -2q4dnUw0gGJniGW2aPamQ | Fiesta Ranchera                        | Restaurants | NoiseLevel | quiet     |
| 01xXe2m_z048W5gcBFpoJA | 01xXe2m_z048W5gcBFpoJA | Five Guys                              | Restaurants | NoiseLevel | average   |
| 0u2xwNy07-z3gv0tpuJX2Q | 0u2xwNy07-z3gv0tpuJX2Q | Subway                                 | Restaurants | NoiseLevel | average   |
| 1CP8aJa8ILlfM5deroar0Q | 1CP8aJa8ILlfM5deroar0Q | Wingstop                               | Restaurants | NoiseLevel | average   |
| -xEGQKqXZzvkdyVQilQtrg | -xEGQKqXZzvkdyVQilQtrg | Royal Dumpling                         | Restaurants | NoiseLevel | average   |
| 0kyhbUW6NkpYjJzFBZ64vQ | 0kyhbUW6NkpYjJzFBZ64vQ | The Cider Mill                         | Restaurants | NoiseLevel | quiet     |
| 2rcrwnlPd_w5oieGVyDgpw | 2rcrwnlPd_w5oieGVyDgpw | Masamune Japanese Restaurant           | Restaurants | NoiseLevel | average   |
| 1Ds8V2c7LlwSAA3O-9f4cA | 1Ds8V2c7LlwSAA3O-9f4cA | McDonald's                             | Restaurants | NoiseLevel | average   |
| 0kzPQQL8wVcHlBQzMdRdWQ | 0kzPQQL8wVcHlBQzMdRdWQ | Thirsty Goat                           | Restaurants | NoiseLevel | average   |
| 0B3W6KxkD3o4W4l6cq735w | 0B3W6KxkD3o4W4l6cq735w | Big Smoke Burger                       | Restaurants | NoiseLevel | very_loud |
| 1_y5e1u-o93EKOigXgR3LQ | 1_y5e1u-o93EKOigXgR3LQ | Hermanos Mexican Grill                 | Restaurants | NoiseLevel | average   |
| 2LVuwl-eH-8PYikyFmqcTQ | 2LVuwl-eH-8PYikyFmqcTQ | Papa Da Vinci                          | Restaurants | NoiseLevel | average   |
| 37kk0IW6jL7ZlxZF6k2QBg | 37kk0IW6jL7ZlxZF6k2QBg | Edulis                                 | Restaurants | NoiseLevel | average   |
| 0hBGwOLU2UfiYXkM8wc8Hw | 0hBGwOLU2UfiYXkM8wc8Hw | Otto Onkel                             | Restaurants | NoiseLevel | average   |
| 1NyHpXJqSLHnvDCOW0nJDg | 1NyHpXJqSLHnvDCOW0nJDg | Pizzaiolo                              | Restaurants | NoiseLevel | average   |
| -2bYV9zVtn2F5XpiAaHt5A | -2bYV9zVtn2F5XpiAaHt5A | Deli 194                               | Restaurants | NoiseLevel | average   |
| 2skQeu3C36VCiB653MIfrw | 2skQeu3C36VCiB653MIfrw | Bootleggers Modern American Smokehouse | Restaurants | NoiseLevel | average   |
| 1mkDrJRu3VABKy95gxD-Hg | 1mkDrJRu3VABKy95gxD-Hg | Charlie D's Catfish & Chicken          | Restaurants | NoiseLevel | average   |
| -0DET7VdEQOJVJ_v6klEug | -0DET7VdEQOJVJ_v6klEug | Flaming Kitchen                        | Restaurants | NoiseLevel | average   |
+------------------------+------------------------+----------------------------------------+-------------+------------+-----------+
```

Different attributes of the restaurant Big Smoke Burger



```
+------------------+-------------+----------------------------+-------------------------------------------------------------------------------------------------------------------------------------------------+
| name             | category    | name                       | value
+------------------+-------------+----------------------------+-------------------------------------------------------------------------------------------------------------------------------------------------+
| Big Smoke Burger | Restaurants | RestaurantsTableService    | 0
| Big Smoke Burger | Restaurants | GoodForMeal                | {"dessert": false, "latenight": false, "lunch": true, "dinner": false, "breakfast": false, "brunch": false}
| Big Smoke Burger | Restaurants | Alcohol                    | none
| Big Smoke Burger | Restaurants | Caters                     | 0
| Big Smoke Burger | Restaurants | HasTV                      | 0
| Big Smoke Burger | Restaurants | RestaurantsGoodForGroups   | 1
| Big Smoke Burger | Restaurants | NoiseLevel                 | very_loud
| Big Smoke Burger | Restaurants | WiFi                       | free
| Big Smoke Burger | Restaurants | RestaurantsAttire          | casual
| Big Smoke Burger | Restaurants | RestaurantsReservations    | 0
| Big Smoke Burger | Restaurants | OutdoorSeating             | 0
| Big Smoke Burger | Restaurants | BusinessAcceptsCreditCards | 1
| Big Smoke Burger | Restaurants | RestaurantsPriceRange2     | 2
| Big Smoke Burger | Restaurants | WheelchairAccessible       | 1
| Big Smoke Burger | Restaurants | BikeParking                | 1
| Big Smoke Burger | Restaurants | RestaurantsDelivery        | 0
| Big Smoke Burger | Restaurants | Ambience                   | {"romantic": false, "intimate": false, "classy": false, "hipster": false, "touristy": false, "trendy": false, "upscale": false, "casual": true} |
| Big Smoke Burger | Restaurants | RestaurantsTakeOut         | 1
| Big Smoke Burger | Restaurants | GoodForKids                | 1
| Big Smoke Burger | Restaurants | DriveThru                  | 0
| Big Smoke Burger | Restaurants | BusinessParking            | {"garage": true, "street": false, "validated": false, "lot": false, "valet": false}
+------------------+-------------+----------------------------+-------------------------------------------------------------------------------------------------------------------------------------------------+
```

Opening hours of Big Smoke Burger

```
+------------------------+------------------------+
| hours                  | business_id            |
+------------------------+------------------------+
| Monday|10:30-21:00     | 0B3W6KxkD3o4W4l6cq735w |
| Tuesday|10:30-21:00    | 0B3W6KxkD3o4W4l6cq735w |
| Friday|10:30-21:00     | 0B3W6KxkD3o4W4l6cq735w |
| Wednesday|10:30-21:00  | 0B3W6KxkD3o4W4l6cq735w |
| Thursday|10:30-21:00   | 0B3W6KxkD3o4W4l6cq735w |
| Sunday|11:00-19:00     | 0B3W6KxkD3o4W4l6cq735w |
| Saturday|10:30-21:00   | 0B3W6KxkD3o4W4l6cq735w |
+------------------------+------------------------+
```

One photo available of this restaurant

```
+------------------------+------------------------+---------+---------+
| id                     | business_id            | caption | label   |
+------------------------+------------------------+---------+---------+
| 0KUkwx7Wk46bMNXVQV_sNw | 0B3W6KxkD3o4W4l6cq735w |         | outside |
+------------------------+------------------------+---------+---------+
```

Big Smoke Burger is still open with 3.0 stars and 47 reviews

```
+---------+-------+--------------+
| is_open | stars | review_count |
+---------+-------+--------------+
|       1 |   3.0 |           47 |
+---------+-------+--------------+
```

But the tables review, and checkin did not have any data related to this restaurant.

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

```
SELECT
    b.name
    ,c.category
    ,a.name
    ,a.value
    ,p.label
    ,h.hours
FROM (((business b
    INNER JOIN category c
        ON b.id = c.business_id )

    INNER JOIN attribute a
        ON b.id = a.business_id)

    INNER JOIN photo p
        ON b.id = p.business_id)

    INNER JOIN hours h
        ON b.id = h.business_id

WHERE c.category = 'Restaurants' AND a.business_id = '0B3W6KxkD3o4W4l6cq735w'
```