

Project Title:

"Exploratory Data Analysis of NYC  
Airbnb Listings Using SQL  
(SQLite)"

By -Vidarshana

# Introduction:

- This project explores the New York City Airbnb Open Data using SQLite. The goal is to analyze patterns in pricing, availability, host behavior, and neighborhood trends to derive business insights.

# Data set overview:

kaggle - NYC Airbnb open data

Columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, room\_type, price, minimum\_nights, number\_of\_reviews, reviews\_per\_month, availability\_365, etc.



# Business Questions & SQL Solutions:

What are the top 5 most expensive listings?

```
SELECT id, name, price  
FROM airnb  
ORDER BY price DESC  
LIMIT 5;
```

The screenshot shows a SQLite database interface with the following details:

- Table:** AirNb
- Columns:** id, name, price, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month, calculated\_host\_listings\_count, availability\_365.
- Query:** --What are the top 5 most expensive listings?  
SELECT id, name, price  
FROM airnb  
ORDER BY price DESC  
LIMIT 5;
- Results:** A table showing the top 5 most expensive listings:

i	id	name	price
1	841211	Spacious Centrally Located Apt!!!	999
2	7726	Hip Historic Brownstone Apartment	99
3	16821	Large Room in Amazing East Village	99
4	27531	Eveland Private Bed & Living Room	99
5	39267	CENTRAL PARK LOFT all for YOU	99

Which neighbourhood group has the highest average price?

```
SELECT  
neighbourhood_group,  
AVG(price) AS avg_price  
FROM airnb  
GROUP BY  
neighbourhood_group  
ORDER BY avg_price DESC  
LIMIT 1;
```

The screenshot shows a SQLite database interface with the following details:

- Table:** AirNb
- Column:** id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month, calculated\_host\_listings\_count, availability\_365.
- Query:** SELECT neighbourhood\_group, AVG(price) AS average\_price FROM AirNb GROUP BY neighbourhood\_group ORDER BY average\_price DESC LIMIT 1;
- Result:** A single row is displayed: neighbourhood\_group: Manhattan, average\_price: 204.25542691751085.
- History:** Two previous queries are listed:
  - Which neighbourhood group has the highest average price?  
SELECT neighbourhood\_group, AVG(price) AS average\_price FROM AirNb GROUP BY neighbourhood\_group ORDER BY average\_price DESC LIMIT 1; (Run at 12:53:40)
  - What are the top 5 most expensive listings?  
SELECT id, name, price FROM airnb ORDER BY price DESC (Run at 12:44:27)

How many listings are there in each neighbourhood group?

```
SELECT neighbourhood_group, COUNT() AS total_listings  
FROM airbnb  
GROUP BY neighbourhood_group;
```

The screenshot shows a SQL editor interface with the following details:

- Pricing**, **Help**, **Run**, **SQLite**, **SQLite** (button), **History**, **Syntax**, **History**.
- Column** list: id (INTEGER), name (TEXT), host\_id (TEXT), host\_name (TEXT), neighbourhood\_grou... (truncated), neighbourhood (TEXT), latitude (TEXT), longitude (TEXT), room\_type (TEXT), price (TEXT), minimum\_nights (TEXT), number\_of\_reviews R... (truncated), last\_review (TEXT), reviews\_per\_month T... (truncated), calculated\_host\_listin... (truncated), availability\_365 INTE...
- Query**:

```
1 --How many listings are there in each neighbourhood group?  
2 SELECT neighbourhood_group,COUNT() AS total_listings FROM AirNb GROUP BY neighbourhood_group;
```
- Results** table:

neighbourhood_group	total_listings
Bronx	80
Brooklyn	2695
Manhattan	2764
Queens	424
Staten Island	37
- History** pane (partial):
  - 12:57:21 --How many listings are there in each neighbourhood group? (SQL)
  - 12:57:09 --How many listings are there in each neighbourhood group? (SQL)
  - 12:57:09 --How many listings are there in each neighbourhood group? (SQL)

Find the most common room type in each neighbourhood group.

```
SELECT neighbourhood_group, room_type, COUNT() AS count  
FROM airbnb  
GROUP BY neighbourhood_group, room_type  
ORDER BY neighbourhood_group, count DESC;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**, **Help**, **Run**, **SQLite**, **History** buttons.
- Column** list: id (INTEGER), name (TEXT), host\_id (TEXT), host\_name (TEXT), neighbourhood\_group (TEXT), neighbourhood (TEXT), latitude (TEXT), longitude (TEXT), room\_type (TEXT), price (TEXT), minimum\_nights (TEXT), number\_of\_reviews (REAL), last\_review (TEXT), reviews\_per\_month (REAL), calculated\_host\_listings\_count (INTEGER), availability\_365 (INTEGER).
- History** panel: Syntax, History tab. Shows previous queries and their results.
- Results** panel: A table showing the count of room types by neighbourhood group and room type.

neighbourhood_group	room_type	count
Bronx	Private room	54
Bronx	Entire home/apt	23
Bronx	Shared room	3
Brooklyn	Entire home/apt	1611
Brooklyn	Private room	1060
Brooklyn	Shared room	24
Manhattan	Entire home/apt	1806
Manhattan	Private room	913
Manhattan	Shared room	45
Queens	Private room	233
Queens	Entire home/apt	179

# Which host has the most listings?

```
SELECT host_id, host_name, COUNT() AS total_listings  
FROM airnb  
GROUP BY host_id  
ORDER BY total_listings DESC  
LIMIT 1;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a free tier.
- Help**: Includes a search icon.
- Run**: A green play button icon.
- SQLite**: The active connection.
- History**: A sidebar with tabs for **Syntax** and **History**.
- Query Area**: Contains the SQL query and its execution results.
  - SQL code:

```
1 --Which host has the most listings?  
2 SELECT host_id,host_name, COUNT() AS total_listings FROM AirNb  
3 GROUP BY host_id ORDER BY total_listings DESC LIMIT 1;  
4
```
  - Execution results table:

host_id	host_name	total_listings
7503643	Vida	38
- History Panel**: Shows previous queries and their results.
  - Row 1: SQLite, --Which host has the most listings?, SELECT host\_id,host\_name, COUNT() AS tot, 13:08:13
  - Row 2: SQLite, --Which host has the most listings?, SELECT host\_id,host\_name, COUNT() AS tot, 13:08:04
  - Row 3: (partial) ...

# Total listings by price range

```
SELECT
CASE
WHEN price <= 100 THEN '0-100'
WHEN price <= 200 THEN '101-200'
WHEN price <= 300 THEN '201-300'
ELSE '300+'
END AS price_range,
COUNT(*) AS listings
FROM airnb
GROUP BY price_range;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a 0.1.4 beta (Memory) connection.
- Table**: AirNb
- Column**: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month, calculated\_host\_listings\_count, availability\_365.
- Query Bar**:

```
--Total listings by price range
SELECT
CASE
WHEN price <= 100 THEN '0-100'
WHEN price <= 200 THEN '101-200'
WHEN price <= 300 THEN '201-300'
ELSE '300+'
END AS price_range,
COUNT(*) AS listings
FROM airnb
GROUP BY price_range;
```
- Results Table**:

price_range	listings
0-100	242
101-200	2527
201-300	677
300+	2554
- History**: Shows previous queries and their results.

# Total number of reviews in each neighbourhood group

```
SELECT neighbourhood_group, SUM(number_of_reviews)
AS total_reviews
FROM airnb
GROUP BY neighbourhood_group;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**, **Help**, **Run**, **SQLite**, **NYC AIRBNB**, **History**, **Syntax**, **History**.
- SQLite**: Version 0.1.4 beta (Memory).
- Table**: **AirNb** (selected). Columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews.
- Query Result**:

neighbourhood_group	total_reviews
Bronx	5945
Brooklyn	150708
Manhattan	146042
Queens	25301
Staten Island	2641
- History**:
  - NYC AIRBNB --Total number of reviews in each neighbourgroup SELECT neighbourhood\_group,sum(number\_of\_reviews) AS total\_reviews FROM AirNb GROUP BY neighbourhood\_group; 13:30:22
  - NYC AIRBNB --Total number of reviews in each neighbourgroup SELECT neighbourhood\_group,COUNT() AS total\_reviews FROM AirNb GROUP BY neighbourhood\_group; 13:29:52

# Top 5 neighbourhoods by listing count

```
SELECT neighbourhood, COUNT() AS listing_count  
FROM airnb  
GROUP BY neighbourhood  
ORDER BY listing_count DESC  
LIMIT 5;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**, **Help**, **Run**, **SQLite**, **NYC AIRBNB** (selected tab).
- History** tab is open.
- Syntax** and **History** sub-tabs under History.
- Tables** section: **AirNb** table has columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price.
- SQL Editor**:

```
-- Top 5 neighbourhoods by listing count  
SELECT neighbourhood_group, COUNT() AS listing_count  
FROM AirNb  
GROUP BY neighbourhood_group  
ORDER BY listing_count DESC LIMIT 5;
```
- Results** table:

neighbourhood_group	listing_count
Manhattan	2764
Brooklyn	2695
Queens	424
Bronx	80
Staten Island	37
- Logs** section (orange border):

```
-- Top 5 neighbourhoods by listing count  
SELECT neighbourhood_group, COUNT() AS listing_count  
FROM AirNb  
GROUP BY neighbourhood_group  
ORDER BY DESC LIMIT 5;
```

Help: SQLITE\_ERROR: sqlite3 result code 1:  
near "DESC": syntax error

# Average price of listings for each room type

```
SELECT room_type, ROUND(AVG(price), 2) AS avg_price  
FROM airnb  
GROUP BY room_type;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a 0.1.4 beta (Memory) connection.
- Table**: AirNb table with columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month, calculated\_host\_listings\_count.
- Query Bar**: The query entered is:

```
--Average price of listings for each room type  
SELECT room_type, ROUND(AVG(price), 2) AS avg_price  
FROM airnb  
GROUP BY room_type;
```
- Results Table**: A table showing the average price for each room type:

room_type	avg_price
Entire home/apt	218.65
Private room	95.64
Shared room	94.9
- History**: Shows the executed query and its results.
- Logs**: Shows an error message from the database:

```
--Average price of listings for each room type  
SELECT room_type, ROUND(AVG(price), 2)  
...
```

Help: SQLITE\_ERROR: sqlite3 result code 1:  
near "(" syntax error

# Average price of listings for each room type

```
SELECT room_type, ROUND(AVG(price), 2) AS avg_price  
FROM airnb  
GROUP BY room_type;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a 0.1.4 beta (Memory) connection.
- Table**: AirNb table with columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month, calculated\_host\_listings\_count.
- Query Bar**: The query entered is:

```
--Average price of listings for each room type  
SELECT room_type, ROUND(AVG(price), 2) AS avg_price  
FROM airnb  
GROUP BY room_type;
```
- Results Table**: A table showing the average price for each room type:

room_type	avg_price
Entire home/apt	218.65
Private room	95.64
Shared room	94.9
- History**: Shows the executed query and its results.
- Logs**: Shows an error message from the database:

```
--Average price of listings for each room type  
SELECT room_type, ROUND(AVG(price), 2)  
...
```

Help: SQLITE\_ERROR: sqlite3 result code 1:  
near "(" syntax error

How many listings have no host name (missing data)?

```
SELECT COUNT(*) AS no_host_name  
FROM airnb  
WHERE host_name IS NULL OR TRIM(host_name) = '';
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a 0.1.4 beta (Mem...).
- Help**: Standard help menu.
- Run**: A green play button icon.
- Cloud**: Cloud storage icon.
- SQLite**: Database icon.
- NYC AIRBN**: Current database connection.
- History**: History tab.
- Syntax**: Syntax tab.
- History**: History tab.

**Table:** AirNb

**Column:**

- id INTEGER
- name TEXT
- host\_id TEXT
- host\_name TEXT
- neighbourhood\_group\_id TEXT
- neighbourhood TEXT
- latitude TEXT
- longitude TEXT
- room\_type TEXT
- price TEXT
- minimum\_nights TEXT
- number\_of\_reviews R...
- last\_review TEXT
- reviews\_per\_month T...
- calculated\_host\_listin...
- availability\_365 INTE...

**Query:**

```
1 --How many listings have no host name (missing data)?  
2 SELECT COUNT(*) AS no_host_name  
3 FROM airnb  
4 WHERE host_name IS NULL OR TRIM(host_name) = '';  
5
```

**Result:**

no_host_name
3

**History:**

- NYC AIRBNB - How many listings have no host name (missing data)?  
SELECT COUNT(\*) AS no\_host\_name  
FROM airnb  
WHERE host\_name IS NULL OR TRIM(host\_name) = '';  
14:13:48
- NYC AIRBNB - Number of listings without any reviews  
SELECT COUNT() AS no\_number\_of\_reviews  
14:11:49
- NYC AIRBNB - Estimate revenue per listing (price x minimum\_nights)  
SELECT id, name, (price \* minimum\_nights)

Number of listings without any reviews

```
SELECT COUNT() AS no_review_listings  
FROM airnb  
WHERE number_of_reviews = 0;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**, **Help**, **Run**, **Cloud**, **SQLite**, **NYC AIRNB** buttons.
- Table**: **AirNb** table structure is displayed with columns: id (INTEGER), name (TEXT), host\_id (TEXT), host\_name (TEXT), neighbourhood\_group (TEXT), neighbourhood (TEXT), latitude (TEXT), longitude (TEXT), room\_type (TEXT).
- Query Result**:
  - Line 1: -- Number of listings without any reviews
  - Line 2: SELECT COUNT() AS no\_number\_of\_reviews FROM AirNb WHERE number\_of\_reviews=0;
  - Line 3: 546
  - Line 4:
  - Line 5:
- Output**: no\_number\_of\_reviews: 546

Estimate revenue per listing (price × minimum nights)

```
SELECT id, name, (price * minimum_nights) AS estimated_revenue
FROM airbnb
ORDER BY estimated_revenue DESC
LIMIT 5;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a 0.1.4 beta (Memory) plan.
- Help**: Standard help menu.
- Run**: Run button.
- Cloud**: Cloud storage icon.
- SQLite**: Database connection icon.
- NYC AIRBNB**: Current database selected.
- Table**: AirNb table selected.
- Column**: List of columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month.
- Query Area**: Contains the SQL query:

```
1 --Estimate revenue per listing (price x minimum nights)
2 SELECT id, name, (price * minimum_nights) AS estimated_revenue
3 FROM airbnb
4 ORDER BY estimated_revenue DESC
5 LIMIT 5;
```
- Results Table**: Displays the top 5 listings with their estimated revenue:

id	name	estimated_revenue
1615764		400000
4204302	Prime W. Village location 1 bdrm	225000
2037951	Modern West Village Apartment	182500
3979611	Located at the heart of Manhattan	126000
1586935	Luxury Gramercy Lg 1Bd w Balcony	91250
- History**: Shows three previous queries with their results and execution times (14:08:27, 14:08:20, 14:08:27).

Estimate revenue per listing (price × minimum nights)

```
SELECT id, name, (price * minimum_nights) AS  
estimated_revenue  
FROM airbnb  
ORDER BY estimated_revenue DESC  
LIMIT 5;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**: Shows a 0.1.4 beta (Memory) plan.
- Help**: Standard help menu.
- Run**: Run button.
- Cloud**: Cloud storage icon.
- SQLite**: Database connection icon.
- NYC AIRBNB**: Database name.
- Table**: AirNb table.
- Column**: List of columns: id, name, host\_id, host\_name, neighbourhood\_group, neighbourhood, latitude, longitude, room\_type, price, minimum\_nights, number\_of\_reviews, last\_review, reviews\_per\_month.
- Query Area**: Contains the SQL query:1 --Estimate revenue per listing (price x minimum nights)  
2 SELECT id, name, (price \* minimum\_nights) AS estimated\_revenue  
3 FROM airbnb  
4 ORDER BY estimated\_revenue DESC  
5 LIMIT 5;
- Results Table**: Displays the top 5 listings with their estimated revenue:

id	name	estimated_revenue
1615764		400000
4204302	Prime W. Village location 1 bdrm	225000
2037951	Modern West Village Apartment	182500
3979611	Located at the heart of Manhattan	126000
1586935	Luxury Gramercy Lg 1Bd w Balcony	91250
- History**: Shows three previous queries with their results and execution times (14:08:27, 14:08:20, 14:08:27).

# Hosts operating in more than one neighbourhood

```
SELECT host_id, host_name, COUNT(DISTINCT neighbourhood)
AS areas
FROM airnb
GROUP BY host_id
HAVING areas > 1;
```

The screenshot shows a SQLite database interface with the following details:

- Pricing**, **Help**, **Run**, **SQLite**, **NYC AIRBNB** buttons.
- Table**: **AirNb** table selected.
- Column**: **host\_id**, **host\_name**, **areas**.
- SQL Query**:

```
1 --Hosts operating in more than one neighbourhood
2 SELECT host_id, host_name, COUNT(DISTINCT neighbourhood) AS areas
3 FROM airnb
4 GROUP BY host_id
5 HAVING areas > 1;
```
- Results Table**:

host_id	host_name	areas
10243387	Mirlet	2
10384906	Susan	3
10575680	David	2
10657357	Ivan	2
1146958	Liz	2
119588	Vero	2
12221	Lori	2
12485770	Raanan	2
126607	Laurine	2
1318137	Adriano	2
1364042	Jake	2
- History** tab with previous queries.



# Key Insights



- Manhattan has the highest average price.
- Private rooms are more affordable and common in Bronx and Queens.
- Some hosts manage listings across multiple neighborhoods.
- Many listings are overpriced relative to their neighborhood average.



# Tools Used

- SQLite for querying
- DB Browser for SQLite
- Kaggle for data
- Canva for visualizations

# Conclusion:

Through this analysis, I gained deeper insights into host behaviors, pricing patterns, and neighborhood trends in NYC Airbnb listings. SQL is a powerful tool for slicing and interpreting structured data.