Airbnb Booking – Data Analysis & Business Intelligence Project

1. Project Overview

This project analyzes the Airbnb NYC 2019 dataset to uncover pricing patterns, demand trends, host performance, and revenue opportunities across different boroughs of New York City.

The project demonstrates a complete data analytics pipeline:

- Data Cleaning using Python (Pandas)
- Exploratory Data Analysis (EDA)
- Business Query Analysis using SQL (PostgreSQL)
- Interactive Dashboard creation using Power BI
- Business insights & strategic recommendations

This project highlights my ability to transform raw data into actionable business insights.

2. Dataset Summary

Dataset Name: Airbnb NYC

- Total Records: ~48,000 listings • Each row represents: **One Airbnb listing**
- Key Features:
  - Listing details (id, name) ○ Host details (host_id, host_name)
  - Location (neighbourhood_group, neighbourhood, latitude, longitude)
  - Pricing (price, minimum_nights)
  - Demand indicators (number_of_reviews, reviews_per_month) ○ Availability (availability_365)
  - Last review date

3. Data Cleaning & EDA in Python

3.1 Data

Inspection df.head()

df.info()

df.describe(include='all')

df.isnull().sum()

Observations:

- **Missing values in:**

- **Name**
- **host_name**
- **reviews_per_month**
- **last_review**

✅ 3.2 Handling Missing Values

```
df['name'] = df['name'].fillna('Unknown') df['host_name'] =
df['host_name'].fillna('Unknown') df['reviews_per_month']
= df['reviews_per_month'].fillna(0) df['last_review'] =
pd.to_datetime(    df['last_review'],    format='%d-%m-%Y',
errors='coerce')
df['has_review'] = df['last_review'].notnull().astype(int)
```

Cleaning Decisions:

- Replaced missing categorical values with "Unknown"
- Filled missing reviews_per_month with 0
- Converted last_review to datetime format
- Created new feature: has_review (review existence indicator)

This ensured data consistency before SQL loading.

3.3 Loading Data into PostgreSQL

Used SQLAlchemy to connect and push cleaned data into PostgreSQL:

```
df.to_sql('airbnb_data', engine, if_exists='replace', index=False)
```

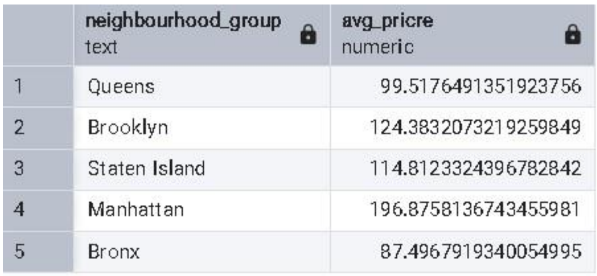
4. Data Analysis in SQL

After loading the dataset into PostgreSQL, I performed business-driven analysis.

```
select * from airbnb_data;
```

1. Average price by neighborhood group?

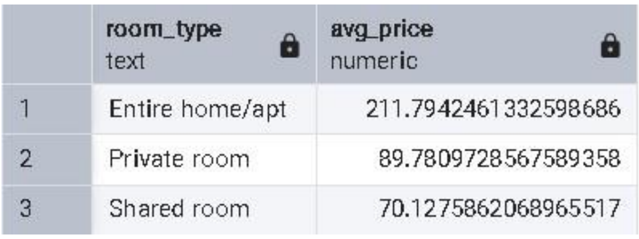
```
select neighbourhood_group, avg(price)as avg_pricre from airbnb_data group by neighbourhood_group;
```



	neighbourhood_group text	avg_pricre numeric
1	Queens	99.5176491351923756
2	Brooklyn	124.3832073219259849
3	Staten Island	114.8123324396782842
4	Manhattan	196.8758136743455981
5	Bronx	87.4967919340054995

--2. Which room type is most expensive on average?

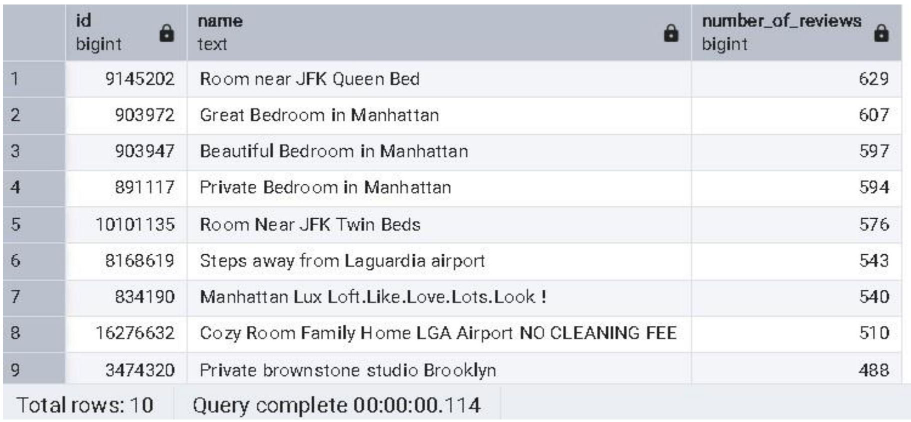
```
SELECT room_type, AVG(price) AS avg_price FROM airbnb_data GROUP BY room_type  ORDER BY avg_price DESC
```



	room_type text	avg_price numeric
1	Entire home/apt	211.7942461332598686
2	Private room	89.7809728567589358
3	Shared room	70.1275862068965517

--3. Top 10 most reviewed listings?

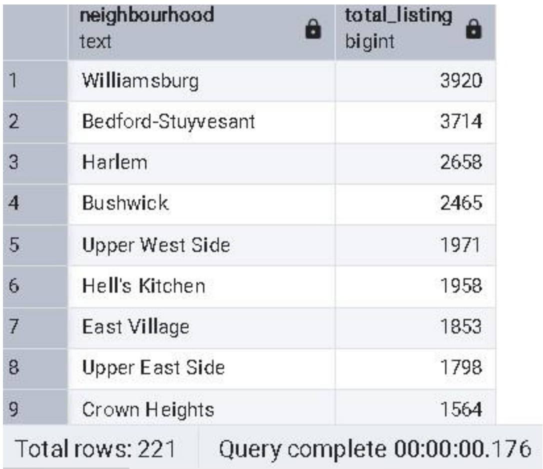
```
SELECT id, name, number_of_reviews
FROM airbnb_data ORDER BY number_of_reviews DESC LIMIT 10;
```



	id bigint	name text	number_of_reviews bigint
1	9145202	Room near JFK Queen Bed	629
2	903972	Great Bedroom in Manhattan	607
3	903947	Beautiful Bedroom in Manhattan	597
4	891117	Private Bedroom in Manhattan	594
5	10101135	Room Near JFK Twin Beds	576
6	8168619	Steps away from Laguardia airport	543
7	834190	Manhattan Lux Loft.Like.Love.Lots.Look !	540
8	16276632	Cozy Room Family Home LGA Airport NO CLEANING FEE	510
9	3474320	Private brownstone studio Brooklyn	488

Total rows: 10 Query complete 00:00:00.114

--4. Which neighborhoods have the most listings? select neighbourhood, count(*) as total_listing from airbnb_data group by neighbourhood order by total_listing desc;



	neighbourhood text	total_listing bigint
1	Williamsburg	3920
2	Bedford-Stuyvesant	3714
3	Harlem	2658
4	Bushwick	2465
5	Upper West Side	1971
6	Hell's Kitchen	1958
7	East Village	1853
8	Upper East Side	1798
9	Crown Heights	1564

Total rows: 221 Query complete 00:00:00.176

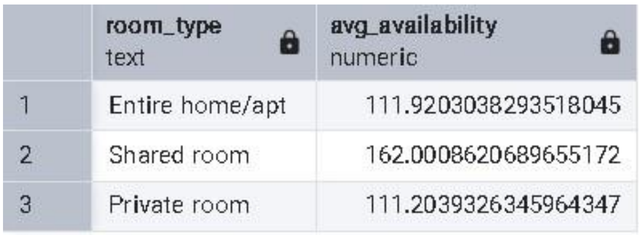
--

5. Hosts with multiple listings?

```
select host_id, host_name ,
count(*) as listing_count  from airbnb_data group by host_id,
host_name having count(*)>1  order by listing_count desc;
```

--6. Average availability by room type?

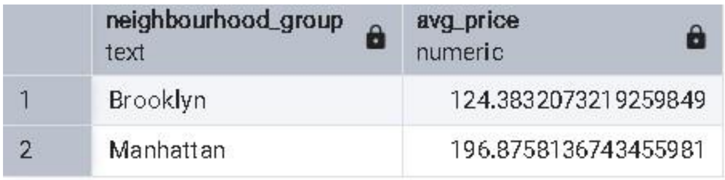
```
select room_type, avg(availability_365) as avg_availability FROM airbnb_data
GROUP BY room_type;
```



	room_type text	avg_availability numeric
1	Entire home/apt	111.9203038293518045
2	Shared room	162.0008620689655172
3	Private room	111.2039326345964347

--7.Price comparison: Manhattan vs Brooklyn?

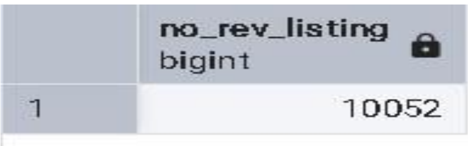
```
SELECT neighbourhood_group, AVG(price) AS avg_price FROM airbnb_data
WHERE neighbourhood_group IN ('Manhattan', 'Brooklyn') GROUP BY
neighbourhood_group;
```



	neighbourhood_group text	avg_price numeric
1	Brooklyn	124.3832073219259849
2	Manhattan	196.8758136743455981

--8.Listings with no reviews?

```
select   count(*) as no_rev_listing from airbnb_data where
number_of_reviews=0;
```



	no_rev_listing bigint
1	10052

9. Recently active listings?

```
--
SELECT * FROM airbnb_data

WHERE last_review IS NOT NULL;
```

--10. High price but low demand listings?

```
SELECT id, name, price, number_of_reviews FROM airbnb_data

WHERE price > (SELECT AVG(price) FROM airbnb_data)  AND
number_of_reviews < (SELECT AVG(number_of_reviews) FROM airbnb_data);
```

--11.Which listings have the highest revenue potential?

```
SELECT id, name, neighbourhood_group,price * (365 - availability_365) AS
revenue_potential FROM airbnb_data ORDER BY revenue_potential DESC LIMIT
10;
```

--12.Which listings are priced above market but receive fewer reviews?

```
SELECT id, name, neighbourhood, price, number_of_reviews FROM airbnb_data
WHERE price > (SELECT AVG(price) FROM airbnb_data) AND
number_of_reviews < (SELECT AVG(number_of_reviews) FROM airbnb_data)
ORDER BY price DESC;
```

--13.Which listings are underpriced but highly demanded?

```
SELECT id, name, neighbourhood, price, number_of_reviews FROM airbnb_data
WHERE price < (SELECT AVG(price) FROM airbnb_data) AND
number_of_reviews > (SELECT AVG(number_of_reviews) FROM
airbnb_data)ORDER BY number_of_reviews DESC;
```

--14.How unequal are prices within the same neighborhood?

```
SELECT neighbourhood, MAX(price) - MIN(price) AS price_gap FROM
airbnb_data GROUP BY neighbourhood ORDER BY price_gap DESC;
```

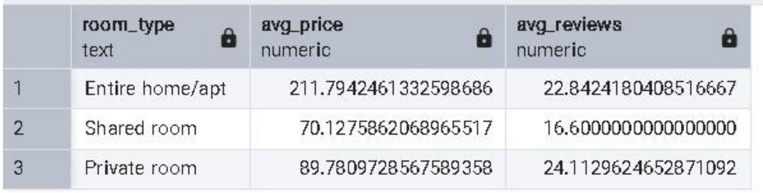
--15.Which hosts dominate a neighborhood?

```
SELECT neighbourhood, host_id, host_name, COUNT(*) AS listing_count FROM
airbnb_data GROUP BY neighbourhood, host_id, host_name HAVING COUNT(*)
>= 20 ORDER BY listing_count DESC;
```

16.Which room type gives best value for guests?

--

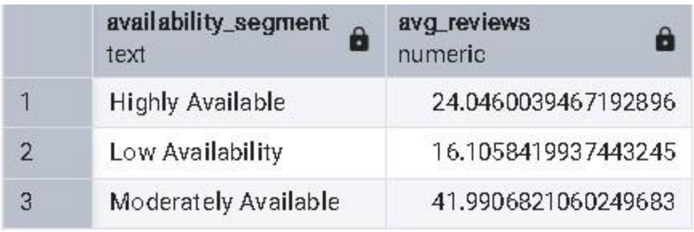
```
SELECT room_type, AVG(price) AS avg_price, AVG(number_of_reviews) AS
avg_reviews FROM airbnb_data GROUP BY room_type;
```



	room_type text	avg_price numeric	avg_reviews numeric
1	Entire home/apt	211.7942461332598686	22.8424180408516667
2	Shared room	70.1275862068965517	16.6000000000000000
3	Private room	89.7809728567589358	24.1129624652871092

--17.Are highly available listings actually less popular?

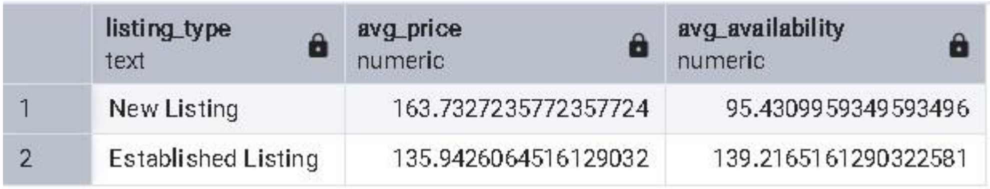
```
SELECT CASE  WHEN availability_365 > 300 THEN 'Highly Available' WHEN
availability_365 BETWEEN 100 AND 300 THEN 'Moderately Available' ELSE 'Low
Availability'  END AS availability_segment, AVG(number_of_reviews) AS
avg_reviews FROM airbnb_data GROUP BY availability_segment;
```



	availability_segment text	avg_reviews numeric
1	Highly Available	24.0460039467192896
2	Low Availability	16.1058419937443245
3	Moderately Available	41.9906821060249683

--18.Do newer listings struggle with visibility?

```
SELECT CASE WHEN number_of_reviews < 10 THEN 'New Listing'  ELSE
'Established Listing'  END AS listing_type,  AVG(price) AS avg_price,
AVG(availability_365) AS avg_availability FROM airbnb_data GROUP BY
listing_type;
```

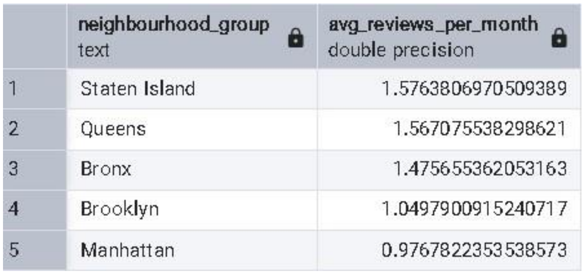


	listing_type text	avg_price numeric	avg_availability numeric
1	New Listing	163.7327235772357724	95.4309959349593496
2	Established Listing	135.9426064516129032	139.2165161290322581

19.Which areas show more frequent guest turnover?

```
SELECT neighbourhood_group, AVG(reviews_per_month) AS
avg_reviews_per_month FROM airbnb_data GROUP BY neighbourhood_group
ORDER BY avg_reviews_per_month DESC;
```

--



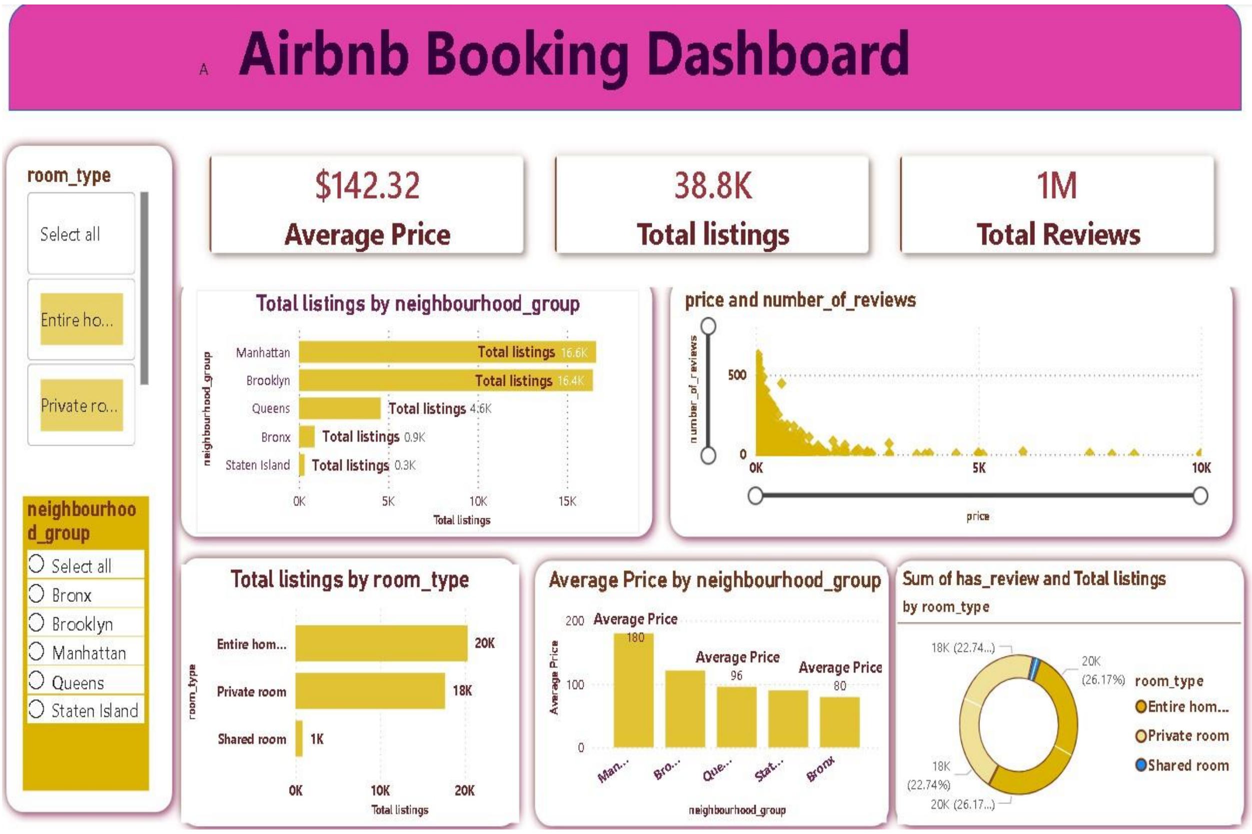
	neighbourhood_group text	avg_reviews_per_month double precision
1	Staten Island	1.5763806970509389
2	Queens	1.567075538298621
3	Bronx	1.475655362053163
4	Brooklyn	1.0497900915240717
5	Manhattan	0.9767822353538573

--20.Room type dominance by borough?

SELECT neighbourhood_group, room_type, COUNT(*) AS listing_count FROM airbnb_data GROUP BY neighbourhood_group, room_type ORDER BY neighbourhood_group, listing_count DESC;

5. Power BI Dashboard

The final stage involved building an interactive Powerbi



6.Key Insights from Dashboard

1️⃣ Manhattan is the most expensive and most supplied borough.
2️⃣ Mid-priced properties receive maximum engagement.
3️⃣ Entire homes dominate the market supply.
4️⃣ Many high-priced listings show low demand.
5️⃣ Brooklyn is a strong competitive alternative to Manhattan.
6️⃣ Outer boroughs are under-penetrated markets.

7. Business Recommendations

Based on dashboard insights:

◈ 1. Implement Dynamic Pricing

Adjust prices according to borough averages and demand trends.

Overpriced listings receive fewer reviews and lower engagement**.**

◈ 2. Focus on Mid-Range Pricing Strategy

Listings priced in the mid-range ($80–$150) attract higher reviews and occupancy.
Optimize for occupancy rather than high nightly rates.

◈ 3. Improve Low-Demand Listings

High availability + low reviews indicate weak demand.
Enhance photos, descriptions, and offer introductory discounts.

◈ 4. Expand in Underserved Boroughs

Bronx and Staten Island have low listing supply.
Encourage new host onboarding to capture untapped market potential.

◈ 5. Promote New Listings

Boost visibility for listings with zero reviews through promotional campaigns.

◈ 6. Monitor Professional Hosts

Ensure fair competition in neighbourhoods dominated by multi-listing hosts.