Masterschool Mastery Project

NYC Short-Term Rental Insights

Tsatskina Ganna

12 June, 2023

SUMMARY

The analysis will try to discover important patterns in popular neighborhoods, rental prices, types of property and demand over a period time.

Short-term rental data grouped on geographic location can give us insights into market performance in areas we're interested in. Another opportunity is to compare information on neighborhoods side by side to find the region with the best investment potential and allocate our resources accordingly.

Pillow Palooza could use this information to figure out which borough to invest in, which types of property to prioritize and how to set competitive rental price.

The main insights are:

- Short-term rental demand in NYC has seasonal arising from April, highest demand is in June.
- The most attractive areas for investment are Manhattan and Brooklyn.
- Should be given priority to entire home/apartment and private rooms.
- Highest income is brought by objects at a price of up to $400.

CONTEXT

This dataset describes the listing activity and metrics in NYC for January – July 2019. These data files include all needed information to find out more about hosts, number of listings, types of property, geographical data, prices, number of booked and available days - necessary metrics to draw conclusions. For analysis were used tools:

- Python (code - Appendix 1)
- SQL (code - Appendix 2)
- Tableau

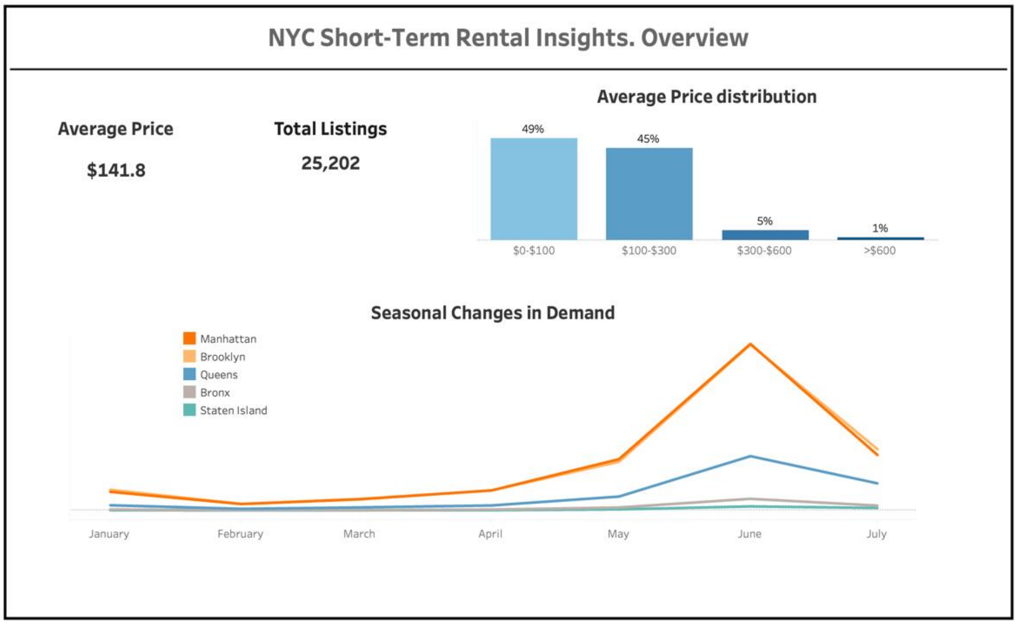
https://public.tableau.com/views/NYCShort-TermRentalInsights_16865833538660/NYCShort-TermRentalInsights_?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Using Python, we have cleaned the data (Appendix 1) and got three datasets:

prices
listing_id *the listing ID*
price *price in dollar*
borough *Name of the borough*
neighborhood *Name of the neighborhood*
price_per_mont *price per month in dollars*
latitude latitude coordinates
longitude longitude coordidantes
reviews
listing_id *the listing ID*
host_name *name of the host*
last_review *date of the last review*
number_of_reviews *number of reviews*
reviews_per_month *number of reviews per month*
calculated_host_listings_count *amount of listing per host*
availability_365 *number of days when listing is available for booking in the next 365 days*
booked_days_365 Amount of booked days in the next 365 days
room_types
listing_id *the listing ID*
description the description of the listing
room_type listing space type

To support the key results, we provide analysis (using SQL and Tableau) and got further results:

1. Brief overview



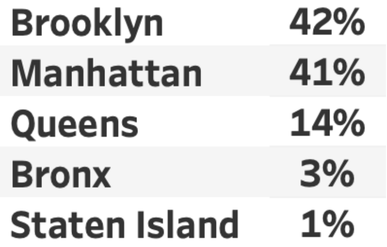
Vis.1 Overview of the main metrics in the data sets

There is brief overview (composed with Tableau) of our data set that highlight that in total we have 25202 listings during given period with average price $142.8. We have seasonal fluctuations: rental demand started arising in the April, got its maximum in the June and decreases in July. This change can be explained by the influence of the tourist season on demand (Vis.1)

2. Rental demand

2.1. Borough's analysis

This metric (made in Tableau) reflects the popularity of vacation rentals in the different boroughs during the given period.



Brooklyn	42%
Manhattan	41%
Queens	14%
Bronx	3%
Staten Island	1%

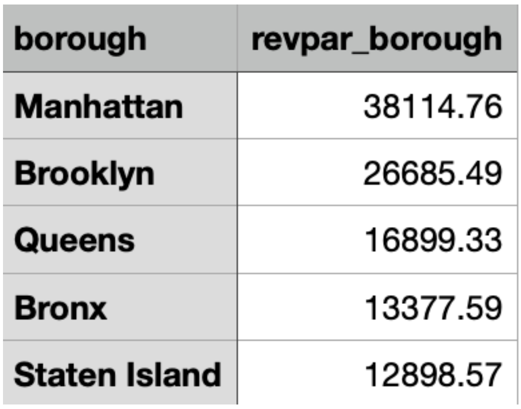
Tab.1. The most popular neighborhoods

So, Brooklyn and Manhattan keep leading positions on the demand in NYC and together represent 83% of total vacation rental (Tab.1).

Revenue per Available Room in the context of each borough

Let's analyze the boroughs of NYC in terms of revenue using KPI Revenue per Available Rooms (SQL calculation, Appendix 2).

RevPAR shows the amount of revenue generated by one room. Our aim is increasing the RevPAR since it reflects both the pricing for rooms and the ability to fill them.



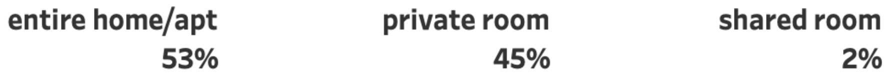
borough	revpar_borough
Manhattan	38114.76
Brooklyn	26685.49
Queens	16899.33
Bronx	13377.59
Staten Island	12898.57

Tab.2 RevPAR for each borough

The calculation of this KPI defines that the most attractive for Pillow Palooza is the area consisting of Manhattan and Brooklyn (Tab.2)

2.2. Property type analyzing

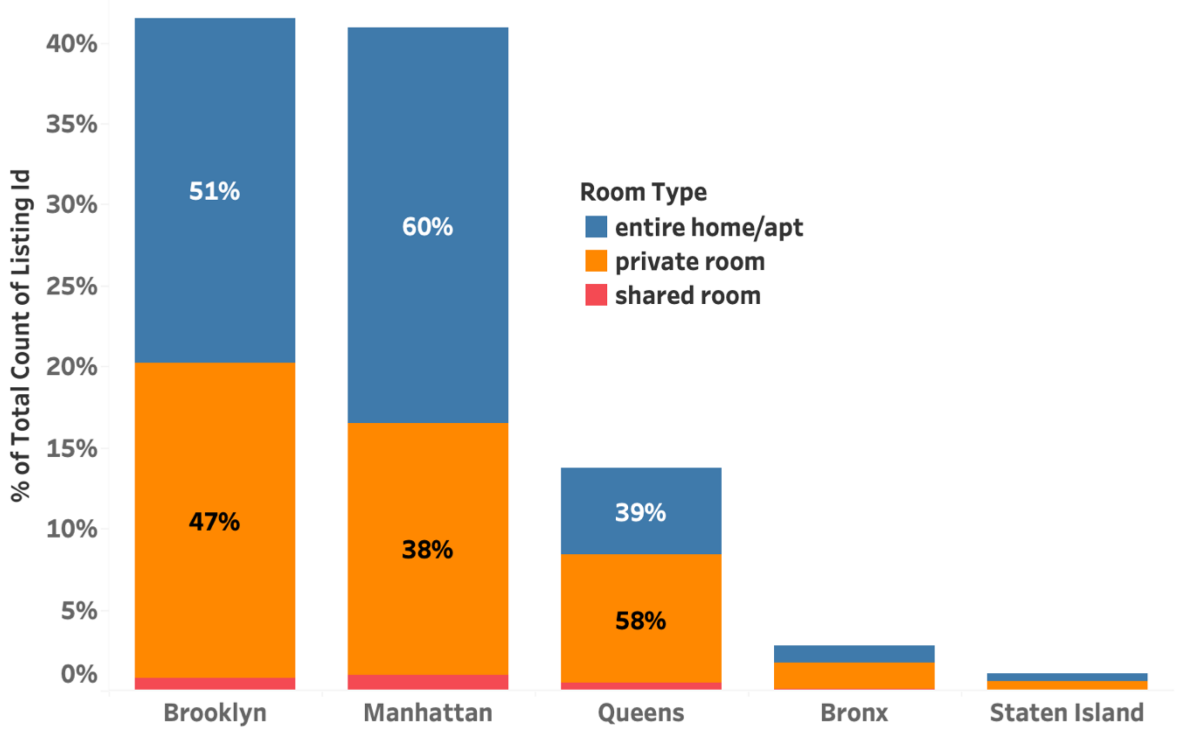
During the analysis, we found that the most popular types of property for rent in NYC are entire home/apartment (Tab.3, using Tableau)



entire home/apt	private room	shared room
53%	45%	2%

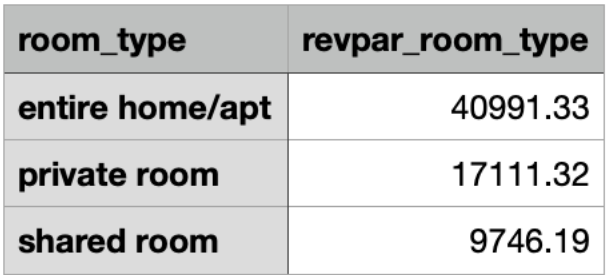
Tab.3 The most rented property types

The distribution of different types of listings across different boroughs highlights that entire home / apartment and private room are more popular in Manhattan and Brooklyn areas and provide more opportunities for certain types of listings (Vis.2, using Tableau).



Vis.2. Room type frequency

Analysis of revenue per room gives us clear understanding that entire homes / apartment type of property is more attractive for investment. The second position place private rooms



room_type	revpar_room_type
entire home/apt	40991.33
private room	17111.32
shared room	9746.19

Tab.4 RevPAR for each property type

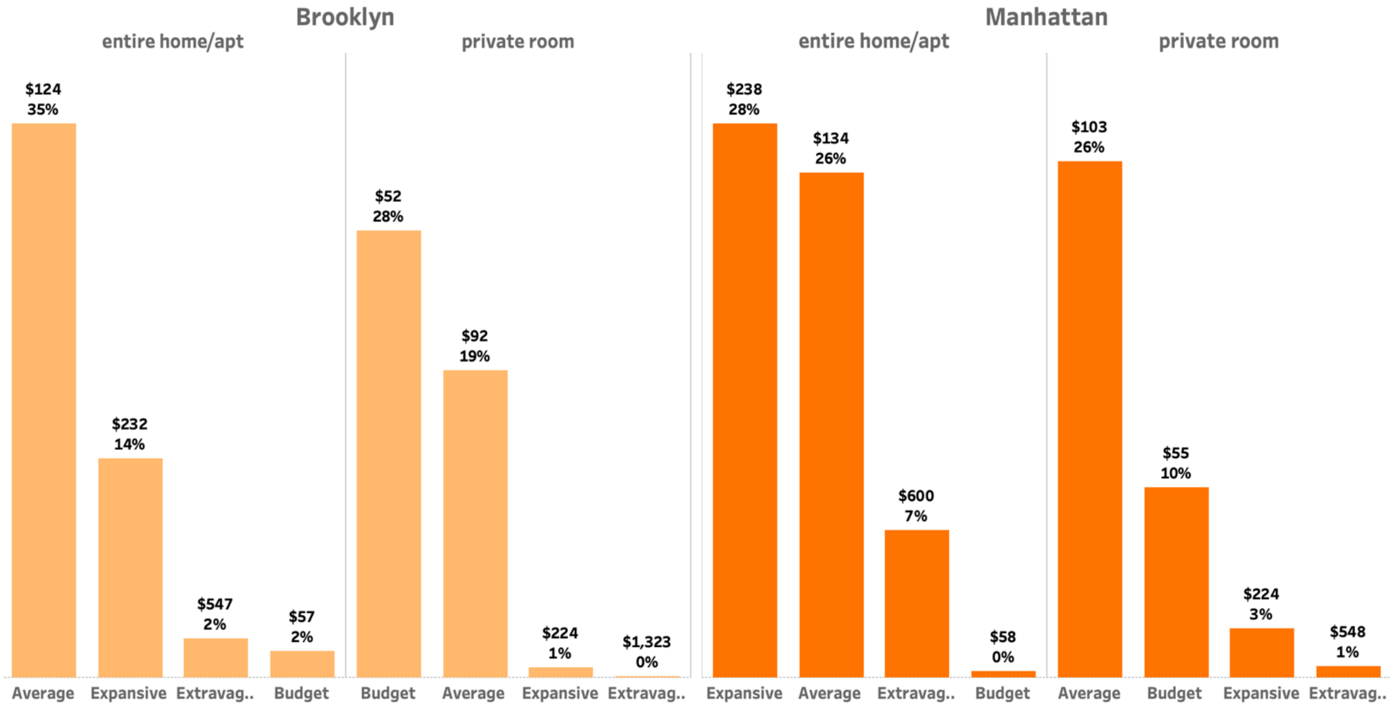
3. Competitive price

Next aim in our analysis is to identify competitive rental price range.
Firstly, we found how the estimated book days correlate with the price of an Airbnb listing in New York City.
Using SQL (Appendix 2) determined the correlation coefficient of -0.076. This indicates a weak negative correlation between price and booked days in the dataset. This means that there is a slight tendency for prices to decrease slightly as the number of booked days increases, although the relationship is not very strong.

For further analysis I propose to focus on two boroughs - Brooklyn and Manhattan and two types of property – entire home / apartment and private rooms which we define as more attractive for investment through previous analysis.
We can group the data and count frequencies for listings in each price range by room type. We will assign the following categories and price ranges:

label	price
Budget	$10-69
Average	$70-175
Expensive	$176-350
Extravagant	>$350

Price range



Vis.3. Price distribution in Brooklyn and Manhattan

Vis.3 (using Tableau) shows most appropriable price categories:

- in Brooklyn are
  average and expensive → entire home / apartment,
  budget and average → private rooms
- in Manhattan are
  expensive and average → entire home / apartment,
  budget and average → private rooms

RECOMMENDATIONS

Recommendations for investment in defined area, property types and competitive price can be summarized in the following table:

Borough	Room Type	Price
Brooklyn	Entyre home / apartment	\$70 – 175
Brooklyn	Private Room	\$10 – 175
Manhattan	Entyre home / apartment	\$70 – 350
Manhattan	Private Room	\$70 - 175

Tab.5 Recommendations

I would suggest continuing collecting and analyzing data based on the following considerations:

1. Analysis showed seasonality in the demand of short-term rental, but we have only data for half a year. So, it is quite important to observe changes for longer period
2. As defined correlation between price and booked days is weak, I would recommend continuing analyzing data during longer period for correct appropriable price of property and choose the best hosts for our start-up

Appendix

Appendix 1. Cleaning data with Python

```
import numpy as np
import pandas as pd
import datetime as dt

# Importing the data
# Load airbnb_price.csv, prices
prices = pd.read_csv("/content/prices.csv")

# Load airbnb_room_type.xlsx, xls
xls = pd.ExcelFile("//content/room_typs.xlsx")
```

```python
# Parse the first sheet from xls, room_types
room_types = xls.parse(0)

# Load airbnb_last_review.tsv, reviews
reviews = pd.read_csv("/content/reviews.tsv", sep="\t")

# Cleaning the price column
# Remove whitespace and string characters from prices column
prices["price"] = prices["price"].str.replace(" dollars", "")

# Convert prices column to numeric datatype
prices["price"] = pd.to_numeric(prices["price"])

# Calculating average price
# Subset prices for listings costing $0, free_listings
free_listings = prices["price"] == 0

# Update prices by removing all free listings from prices
prices = prices.loc[~free_listings]

# Calculate the average price
avg_price = round(prices["price"].mean(), 2)

# Comparing costs to the private rental market
# Add a new column to the prices DataFrame, price_per_month
prices["price_per_month"] = prices["price"] * 365 / 12

# Calculate average_price_per_month
average_price_per_month = round(prices["price_per_month"].mean(), 2)
difference = round((average_price_per_month - 3100),2)

# Cleaning the room_type column
# Convert the room_type column to lowercase
room_types["room_type"] = room_types["room_type"].str.lower()

# Update the room_type column to category data type
room_types["room_type"] = room_types["room_type"].astype("category")

# What timeframe are we working with?
# Change the data type of the last_review column to datetime
reviews["last_review"] = pd.to_datetime(reviews["last_review"])

# Create first_reviewed, the earliest review date
first_reviewed = reviews["last_review"].dt.date.min()

# Create last_reviewed, the most recent review date
last_reviewed = reviews["last_review"].dt.date.max()

# Joining the DataFrames
# Merge prices and room_types to create rooms_and_prices
rooms_and_prices = prices.merge(room_types, how="outer", on="listing_id")

# Merge rooms_and_prices with the reviews DataFrame to create airbnb_merged
airbnb_merged = rooms_and_prices.merge(reviews, how="outer", on="listing_id")

# Drop missing values from airbnb_merged
airbnb_merged.dropna(inplace=True)
```

```python
# Analyzing listing prices by NYC borough
# Extract information from the nbhood_full column and store as a new column, borough
airbnb_merged["borough"] = airbnb_merged["nbhood_full"].str.partition(",")[0]

# Group by borough and calculate summary statistics
boroughs = airbnb_merged.groupby("borough")["price"].agg(["sum", "mean", "median", "count"])

# Round boroughs to 2 decimal places, and sort by mean in descending order
boroughs = boroughs.round(2).sort_values("mean", ascending=False)

#  Price range by borough
# Create labels for the price range, label_names
label_names = ["Budget", "Average", "Expensive", "Extravagant"]

# Create the label ranges, ranges
ranges = [0, 69, 175, 350, np.inf]

# Insert new column, price_range, into DataFrame
airbnb_merged["price_range"] = pd.cut(airbnb_merged["price"], bins=ranges, labels=label_names)

# Calculate occurence frequencies for each label, prices_by_borough
prices_by_borough = airbnb_merged.groupby(["borough", "price_range"])["price_range"].count()

# Storing the final result

solution = {'average_price_per_month': average_price_per_month,
            'prices_by_borough':prices_by_borough}
print(solution)
```

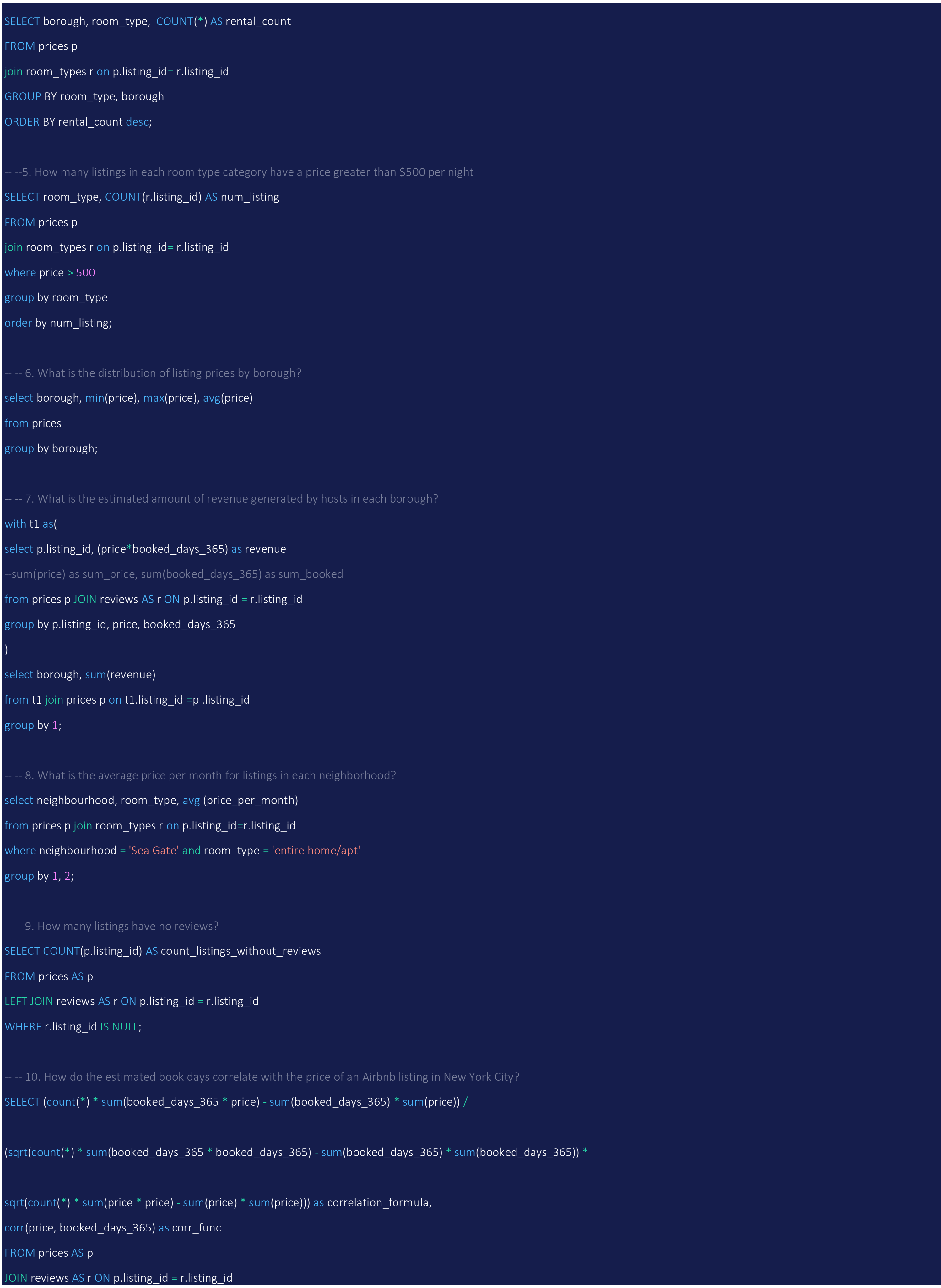
Appendix 2. Analise data with SQL

```sql
-- 1.What are the most popular
-- neighborhoods for short-term rentals in New York City?
SELECT room_type, COUNT(*) AS rental_count
FROM prices p
join room_types r on p.listing_id= r.listing_id
GROUP BY room_type
ORDER BY room_type;

-- 2.What is the average price of the listings by room type "shared room"?
SELECT room_type, round(AVG(price), 2) AS average_price
FROM prices p
join room_types r on p.listing_id= r.listing_id
WHERE room_type = 'shared room'
GROUP BY room_type
ORDER BY room_type;

-- -- 3. Which borough has the highest average price per month?
SELECT borough, avg(price_per_month) as average_price
FROM prices
GROUP BY borough
ORDER BY average_price DESC;

-- -- 4. How many listings of each room type are in each borough?
```



```
SELECT borough, room_type,  COUNT(*) AS rental_count
FROM prices p
join room_types r on p.listing_id= r.listing_id
GROUP BY room_type, borough
ORDER BY rental_count desc;

-- --5. How many listings in each room type category have a price greater than $500 per night
SELECT room_type, COUNT(r.listing_id) AS num_listing
FROM prices p
join room_types r on p.listing_id= r.listing_id
where price > 500
group by room_type
order by num_listing;

-- -- 6. What is the distribution of listing prices by borough?
select borough, min(price), max(price), avg(price)
from prices
group by borough;

-- -- 7. What is the estimated amount of revenue generated by hosts in each borough?
with t1 as(
select p.listing_id, (price*booked_days_365) as revenue
--sum(price) as sum_price, sum(booked_days_365) as sum_booked
from prices p JOIN reviews AS r ON p.listing_id = r.listing_id
group by p.listing_id, price, booked_days_365
)
select borough, sum(revenue)
from t1 join prices p on t1.listing_id =p .listing_id
group by 1;

-- -- 8. What is the average price per month for listings in each neighborhood?
select neighbourhood, room_type, avg (price_per_month)
from prices p join room_types r on p.listing_id=r.listing_id
where neighbourhood = 'Sea Gate' and room_type = 'entire home/apt'
group by 1, 2;

-- -- 9. How many listings have no reviews?
SELECT COUNT(p.listing_id) AS count_listings_without_reviews
FROM prices AS p
LEFT JOIN reviews AS r ON p.listing_id = r.listing_id
WHERE r.listing_id IS NULL;

-- -- 10. How do the estimated book days correlate with the price of an Airbnb listing in New York City?
SELECT (count(*) * sum(booked_days_365 * price) - sum(booked_days_365) * sum(price)) /

(sqrt(count(*) * sum(booked_days_365 * booked_days_365) - sum(booked_days_365) * sum(booked_days_365)) *

sqrt(count(*) * sum(price * price) - sum(price) * sum(price))) as correlation_formula,
corr(price, booked_days_365) as corr_func
FROM prices AS p
JOIN reviews AS r ON p.listing_id = r.listing_id
```

```
-- KPI's for NYC short-term rental insidhts
-- 1.1 Revenue per Available Room by borough
WITH t1 as (
    select borough, count(p.listing_id) as total_number_of_av_room, booked_days_365,
    sum(price*booked_days_365) as revenue
    from prices p join reviews r on p.listing_id = r.listing_id
    group by 1, booked_days_365
)
select borough, round(sum(revenue)/sum(total_number_of_av_room),2) as RevPAR_borough
from t1
group by 1
order by RevPAR_borough desc;

-- 1.2 Revenue per Available Room by room type
WITH t1 as (
    select room_type, count(p.listing_id) as total_number_of_av_room,
    sum(price*booked_days_365) as revenue
    from prices p join room_types rt on p.listing_id = rt.listing_id
    join reviews r on p.listing_id=r.listing_id
    group by 1, booked_days_365
)
select room_type, round(sum(revenue)/sum(total_number_of_av_room),2) as RevPAR_room_type
from t1
group by 1
order by RevPAR_room_type desc;
```