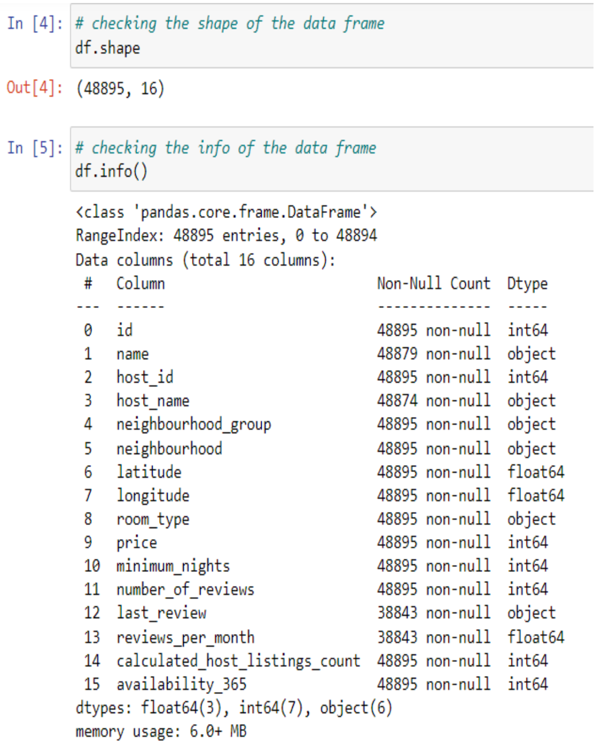
# Storytelling Case Study – Airbnb NYC

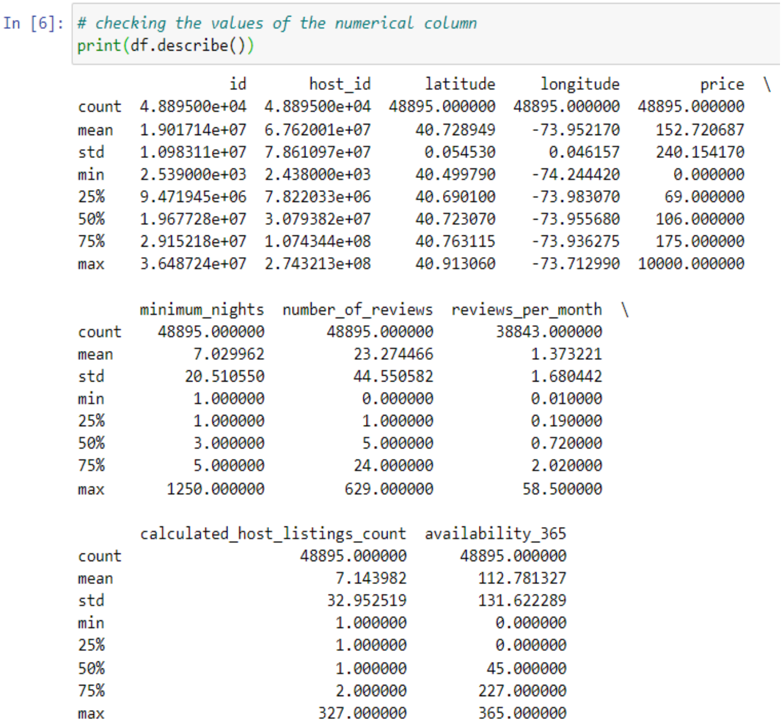
*By- Shreyas Dubey and Keerthi Adep*

## Methodology Document

### Data Wrangling



```
In [4]: # checking the shape of the data frame
        df.shape

Out[4]: (48895, 16)

In [5]: # checking the info of the data frame
        df.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 48895 entries, 0 to 48894
        Data columns (total 16 columns):
         #   Column                          Non-Null Count  Dtype
        ---  ------                          --------------  -----
         0   id                              48895 non-null  int64
         1   name                            48879 non-null  object
         2   host_id                         48895 non-null  int64
         3   host_name                       48874 non-null  object
         4   neighbourhood_group             48895 non-null  object
         5   neighbourhood                   48895 non-null  object
         6   latitude                        48895 non-null  float64
         7   longitude                       48895 non-null  float64
         8   room_type                       48895 non-null  object
         9   price                           48895 non-null  int64
         10  minimum_nights                  48895 non-null  int64
         11  number_of_reviews               48895 non-null  int64
         12  last_review                     38843 non-null  object
         13  reviews_per_month               38843 non-null  float64
         14  calculated_host_listings_count  48895 non-null  int64
         15  availability_365                48895 non-null  int64
        dtypes: float64(3), int64(7), object(6)
        memory usage: 6.0+ MB
```



```
In [6]: # checking the values of the numerical column
        print(df.describe())

                 id       host_id      latitude     longitude         price  \
count  4.889500e+04  4.889500e+04  48895.000000  48895.000000  48895.000000
mean   1.901714e+07  6.762001e+07     40.728949    -73.952170    152.720687
std    1.098311e+07  7.861097e+07      0.054530      0.046157    240.154170
min    2.539000e+03  2.438000e+03     40.499790    -74.244420      0.000000
25%    9.471945e+06  7.822033e+06     40.690100    -73.983070     69.000000
50%    1.967728e+07  3.079382e+07     40.723070    -73.955680    106.000000
75%    2.915218e+07  1.074344e+08     40.763115    -73.936275    175.000000
max    3.648724e+07  2.743213e+08     40.913060    -73.712990  10000.000000

       minimum_nights  number_of_reviews  reviews_per_month  \
count    48895.000000       48895.000000       38843.000000
mean         7.029962          23.274466           1.373221
std         20.510550          44.550582           1.680442
min          1.000000           0.000000           0.010000
25%          1.000000           1.000000           0.190000
50%          3.000000           5.000000           0.720000
75%          5.000000          24.000000           2.020000
max       1250.000000         629.000000          58.500000

       calculated_host_listings_count  availability_365
count                    48895.000000      48895.000000
mean                         7.143982        112.781327
std                         32.952519        131.622289
min                          1.000000          0.000000
25%                          1.000000          0.000000
50%                          1.000000         45.000000
75%                          2.000000        227.000000
max                        327.000000        365.000000
```

Explanation:

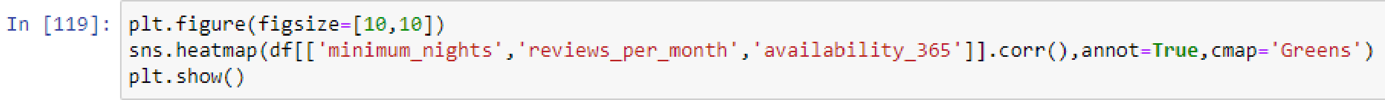
- First snapshot explains number of rows and columns in the dataset of Airbnb's where 'df' is our main dataframe. Another part of snapshot shows datatypes of all the attributes in data and number of missing values in different attributes. (Note: last_review and reviews_per_month columns have some missing values which can be ignored as these are of no use in our final visualization).

- Second snapshot explains if there are any outliers in the dataset. As we can see there are some outliers in attributes like price, minimum_nights, number_of_reviews, reviews_per_month, etc. These outliers will be managed during visualization, so no special treatment is required as of now.

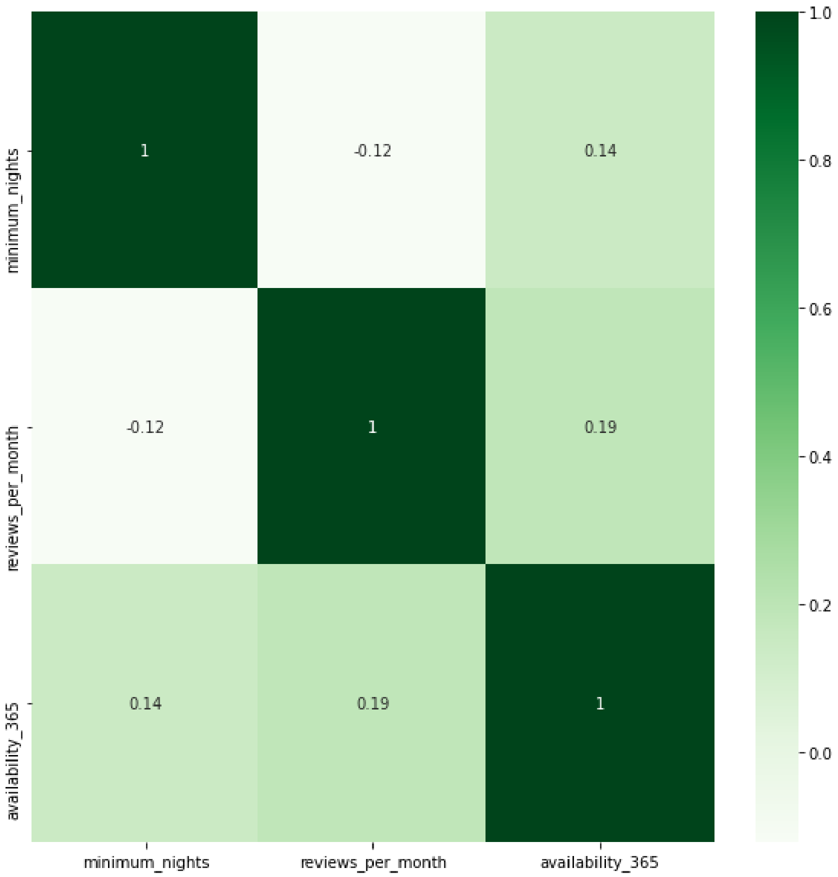
# Data Visualization

## *Visualizations for Head of Acquisitions and Operations & Head of User Experience*

### 1) Impact of Minimum Nights and Availability on Airbnb bookings



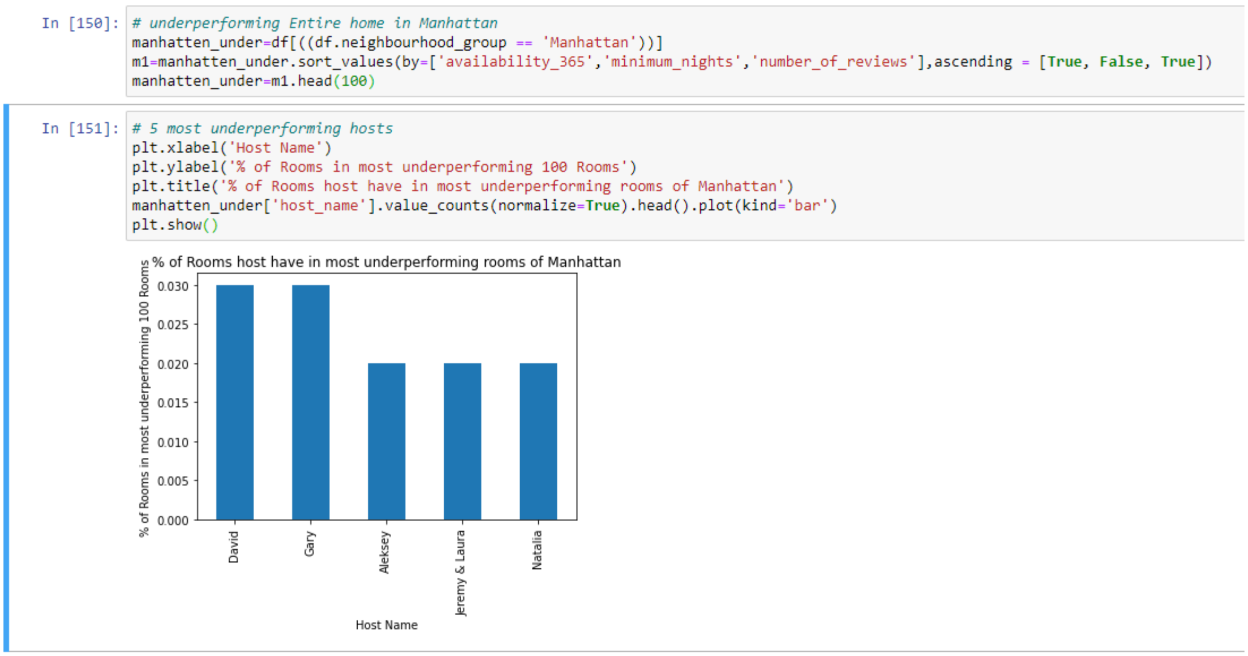
```
In [119]: plt.figure(figsize=[10,10])
          sns.heatmap(df[['minimum_nights','reviews_per_month','availability_365']].corr(),annot=True,cmap='Greens')
          plt.show()
```



Explanation:

- *df* – master dataframe – contains Airbnb .csv file
- Used seaborn for making a heatmap.
- This plot shows that there is a positive correlation between reviews per month and availability and negative correlation between reviews per month and minimum nights so we can say that people prefer less number of minimum nights and higher number of availability.

2) plots of underperforming hosts in various neighborhood :



```
In [150]: # underperforming Entire home in Manhattan
          manhatten_under=df[((df.neighbourhood_group == 'Manhattan'))]
          m1=manhatten_under.sort_values(by=['availability_365','minimum_nights','number_of_reviews'],ascending = [True, False, True])
          manhatten_under=m1.head(100)
```

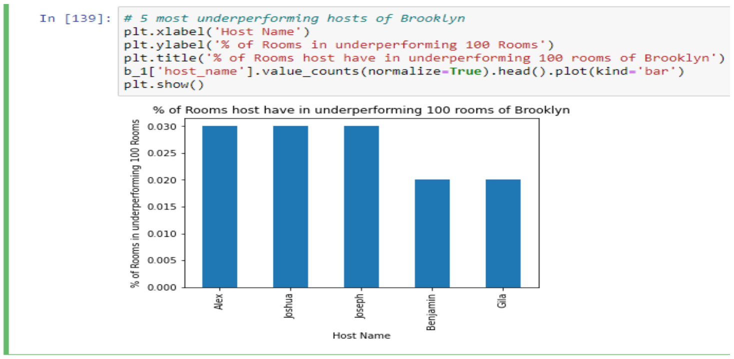
```
In [151]: # 5 most underperforming hosts
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in most underperforming 100 Rooms')
          plt.title('% of Rooms host have in most underperforming rooms of Manhattan')
          manhatten_under['host_name'].value_counts(normalize=True).head().plot(kind='bar')
          plt.show()
```

% of Rooms host have in most underperforming rooms of Manhattan
% of Rooms in most underperforming 100 Rooms
0.030
0.025
0.020
0.015
0.010
0.005
0.000
David
Gary
Aleksey
Jeremy & Laura
Natalia
Host Name

Explanation:

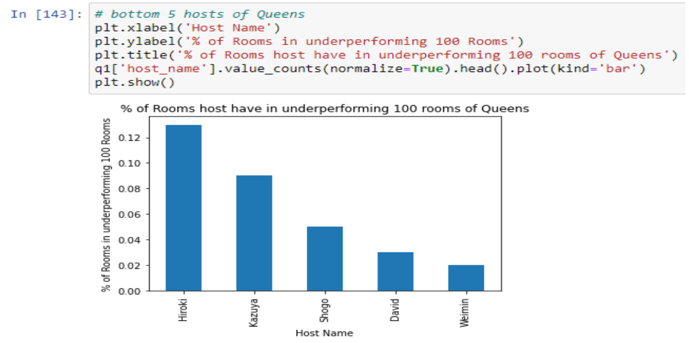
- *df* dataframe – contains Airbnb .csv file
- First selected all the rows from Manhattan neighborhood, sorted it such that rooms with least availability, highest minimum nights and least number of reviews are in the top and selected top 100 rows.
- Then made bar chart of those hosts which have maximum number of rooms in the least 100 list.

Similarly underperforming hosts of other neighborhood is also found and plotted as bar plot :



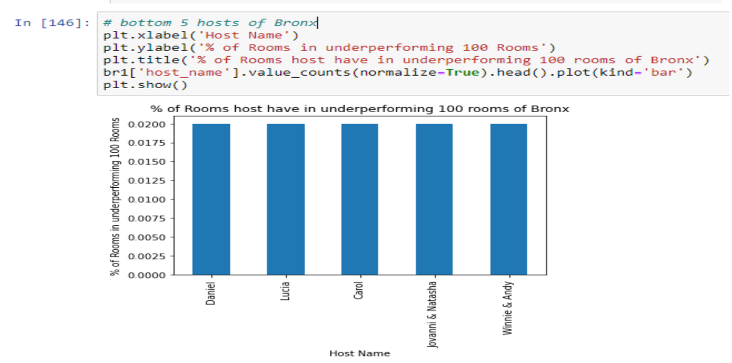
```
In [139]: # 5 most underperforming hosts of Brooklyn
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in underperforming 100 Rooms')
          plt.title('% of Rooms host have in underperforming 100 rooms of Brooklyn')
          b_1['host_name'].value_counts(normalize=True).head().plot(kind='bar')
          plt.show()
```

% of Rooms host have in underperforming 100 rooms of Brooklyn
% of Rooms in underperforming 100 Rooms
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Alex
Joshua
Joseph
Benjamin
Gila
Host Name

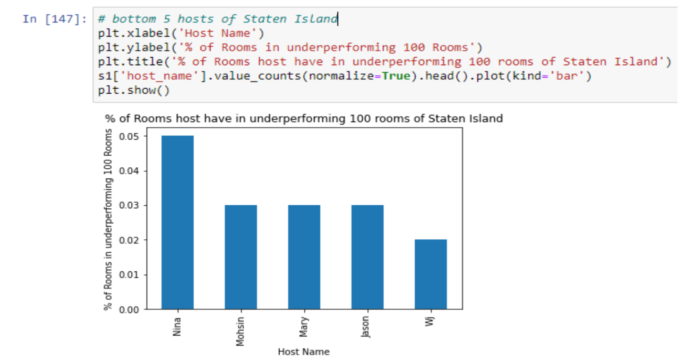


```
In [143]: # bottom 5 hosts of Queens
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in underperforming 100 Rooms')
          plt.title('% of Rooms host have in underperforming 100 rooms of Queens')
          q1['host_name'].value_counts(normalize=True).head().plot(kind='bar')
          plt.show()
```

% of Rooms host have in underperforming 100 rooms of Queens
% of Rooms in underperforming 100 Rooms
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Hiroki
Kazuya
Shogo
David
Weimin
Host Name



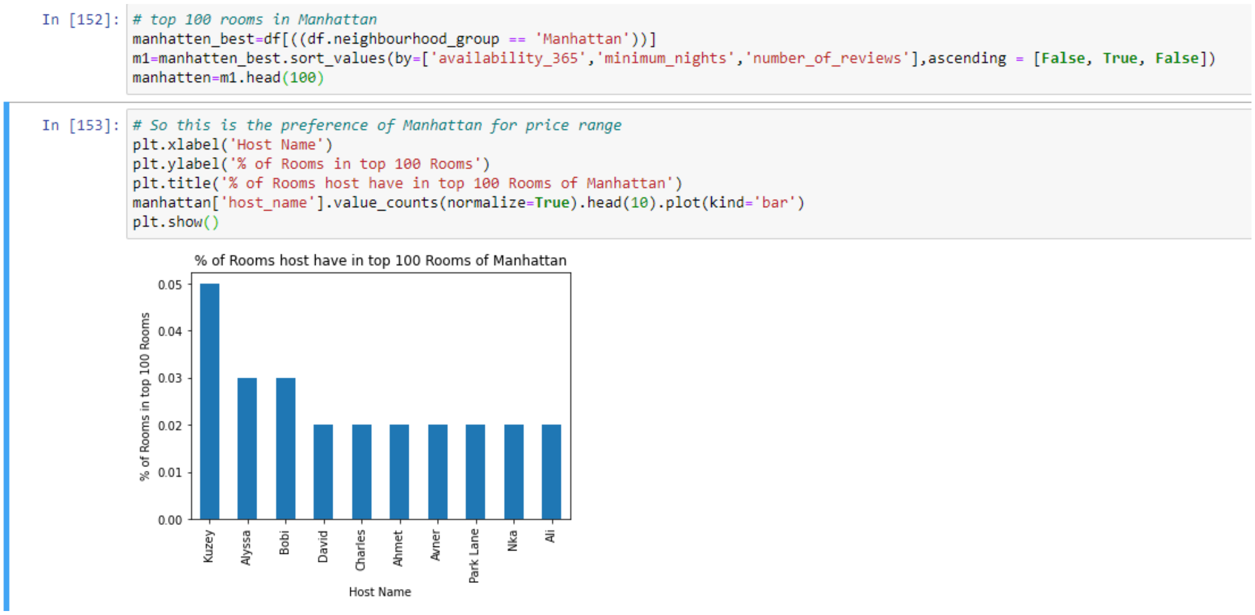
```
In [146]: # bottom 5 hosts of Bronx
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in underperforming 100 Rooms')
          plt.title('% of Rooms host have in underperforming 100 rooms of Bronx')
          br1['host_name'].value_counts(normalize=True).head().plot(kind='bar')
          plt.show()
```



```
In [147]: # bottom 5 hosts of Staten Island
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in underperforming 100 Rooms')
          plt.title('% of Rooms host have in underperforming 100 rooms of Staten Island')
          s1['host_name'].value_counts(normalize=True).head().plot(kind='bar')
          plt.show()
```

% of Rooms host have in underperforming 100 rooms of Staten Island
% of Rooms in underperforming 100 Rooms
0.05
0.04
0.03
0.02
0.01
0.00
Nina
Mohsin
Mary
Jason
Wj
Host Name

## 3) plots of best hosts in various neighborhood:

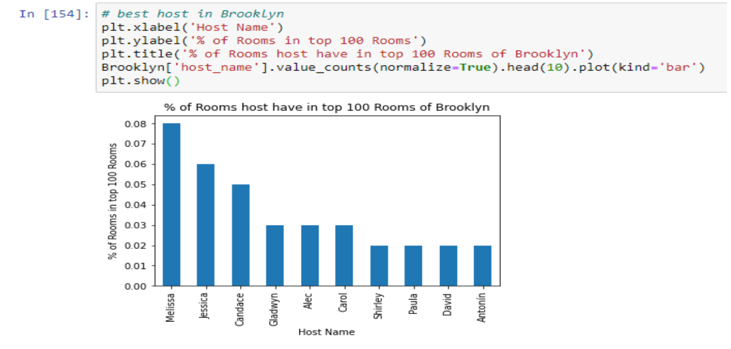


```
In [152]: # top 100 rooms in Manhattan
          manhatten_best=df[((df.neighbourhood_group == 'Manhattan'))]
          m1=manhatten_best.sort_values(by=['availability_365','minimum_nights','number_of_reviews'],ascending = [False, True, False])
          manhatten=m1.head(100)

In [153]: # So this is the preference of Manhattan for price range
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in top 100 Rooms')
          plt.title('% of Rooms host have in top 100 Rooms of Manhattan')
          manhattan['host_name'].value_counts(normalize=True).head(10).plot(kind='bar')
          plt.show()
```

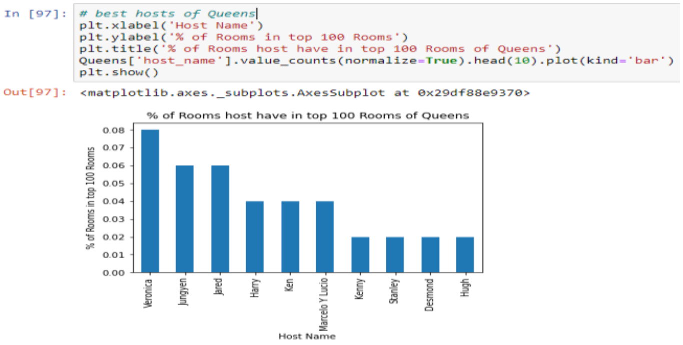
### Explanation:

- *df* dataframe – contains Airbnb .csv file
- First selected all the rows from Manhattan neighborhood, sorted it such that rooms with highest availability, least minimum nights and highest number of reviews are in the top and selected top 100 rows.
- Then made bar chart of those hosts which have maximum number of rooms in the best 100 list.

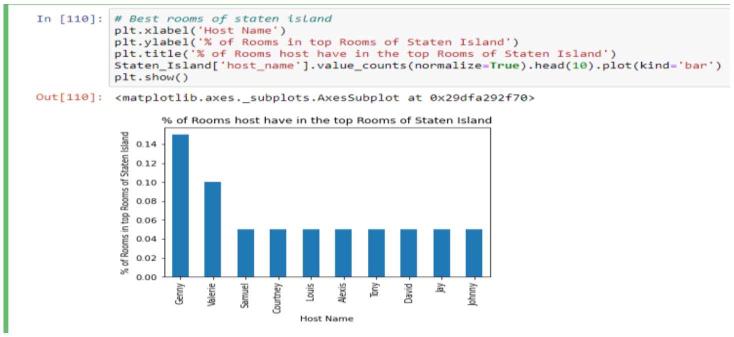
### Similarly best hosts of other neighborhood is also found and plotted as bar plot :



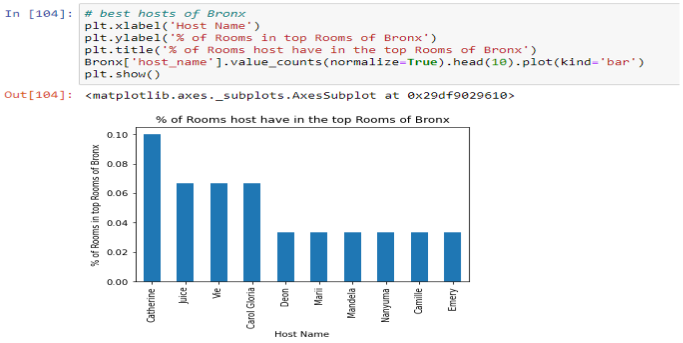
```
In [154]: # best host in Brooklyn
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in top 100 Rooms')
          plt.title('% of Rooms host have in top 100 Rooms of Brooklyn')
          Brooklyn['host_name'].value_counts(normalize=True).head(10).plot(kind='bar')
          plt.show()
```



```
In [97]: # best hosts of Queens
         plt.xlabel('Host Name')
         plt.ylabel('% of Rooms in top 100 Rooms')
         plt.title('% of Rooms host have in top 100 Rooms of Queens')
         Queens['host_name'].value_counts(normalize=True).head(10).plot(kind='bar')
         plt.show()

Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x29df88e9370>
```

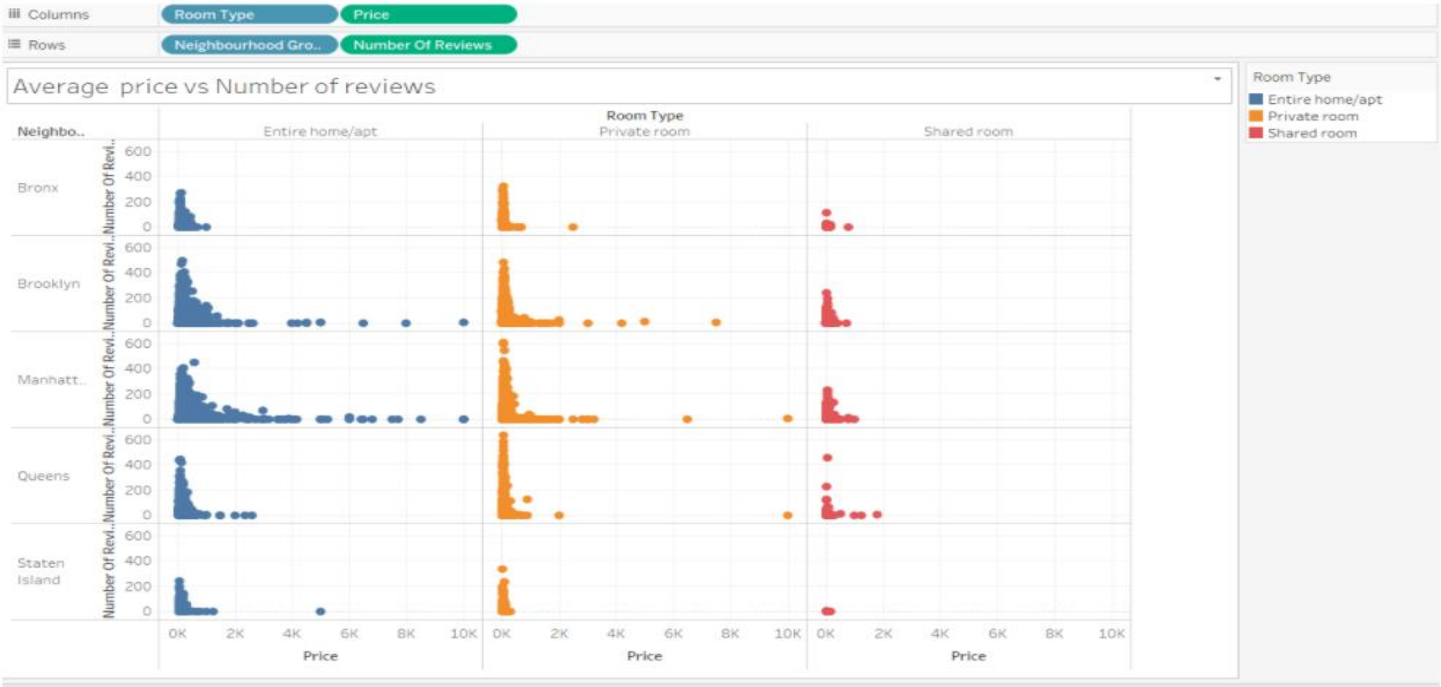


```
In [110]: # Best rooms of staten island
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in top Rooms of Staten Island')
          plt.title('% of Rooms host have in the top Rooms of Staten Island')
          Staten_Island['host_name'].value_counts(normalize=True).head(10).plot(kind='bar')
          plt.show()

Out[110]: <matplotlib.axes._subplots.AxesSubplot at 0x29dfa292f70>
```



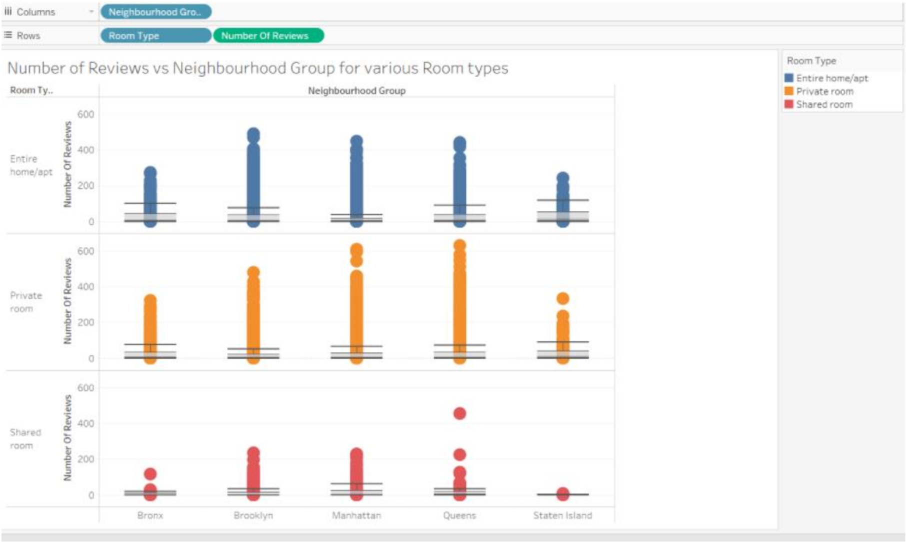
```
In [104]: # best hosts of Bronx
          plt.xlabel('Host Name')
          plt.ylabel('% of Rooms in top Rooms of Bronx')
          plt.title('% of Rooms host have in the top Rooms of Bronx')
          Bronx['host_name'].value_counts(normalize=True).head(10).plot(kind='bar')
          plt.show()

Out[104]: <matplotlib.axes._subplots.AxesSubplot at 0x29df9029610>
```

4) plot of cost vs Type vs number of reviews :



Explanation:

- This plot is made on Tableau with Room type and price in column and neighborhood group and number of reviews in rows.
- Room type is tagged with color.
- This plot shows how price varies for each room type for each neighborhood group.

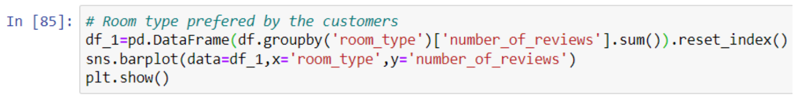


Explanation:

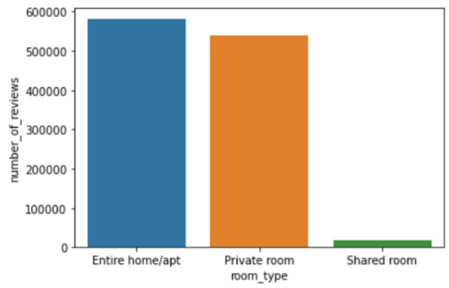
- This plot is made on Tableau with neighborhood group in column and Room type and number of reviews in rows.
- Room type is tagged with color.
- This plot shows how number of Reviews varies for each room type for each neighborhood group.

# *Visualizations For Data Analysis Manager & Lead Data Analyst*

## *1)Distribution of rooms in NYC :*



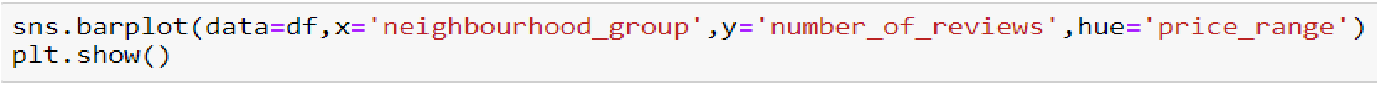
```
In [85]: # Room type prefered by the customers
         df_1=pd.DataFrame(df.groupby('room_type')['number_of_reviews'].sum()).reset_index()
         sns.barplot(data=df_1,x='room_type',y='number_of_reviews')
         plt.show()
```



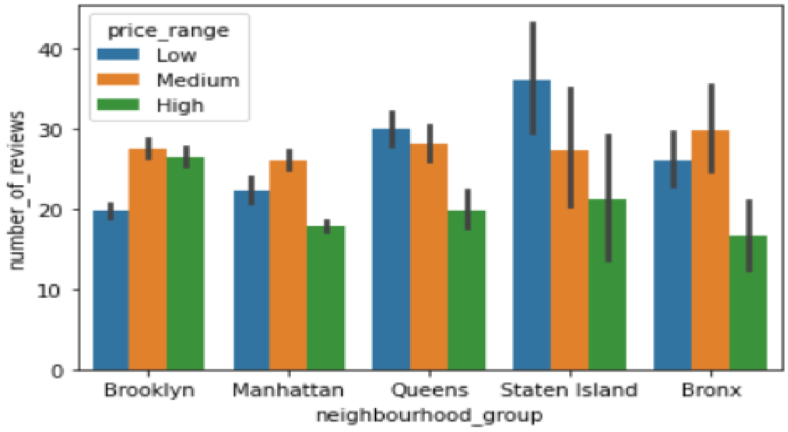
Explanation:

- *df* dataframe – contains Airbnb .csv file
- Grouped the data frame by room type and found sum of reviews for each room type.
- This plot gives you the plot of room type vs number of reviews.

## 2) Distribution of price in NYC :

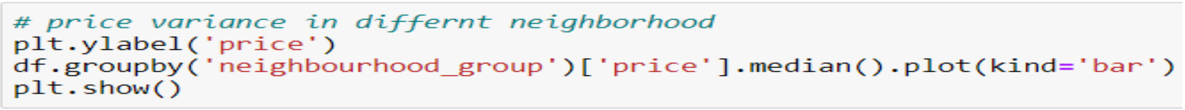


```
sns.barplot(data=df,x='neighbourhood_group',y='number_of_reviews',hue='price_range')
plt.show()
```

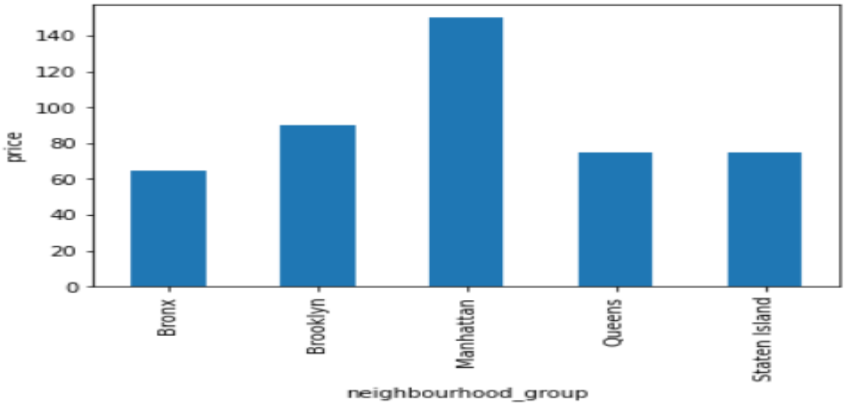


Explanation:

- *df* dataframe – contains Airbnb .csv file
- Used seaborn to plot barplot.
- This plot gives you the plot of neighborhood group vs number of reviews vs price range.



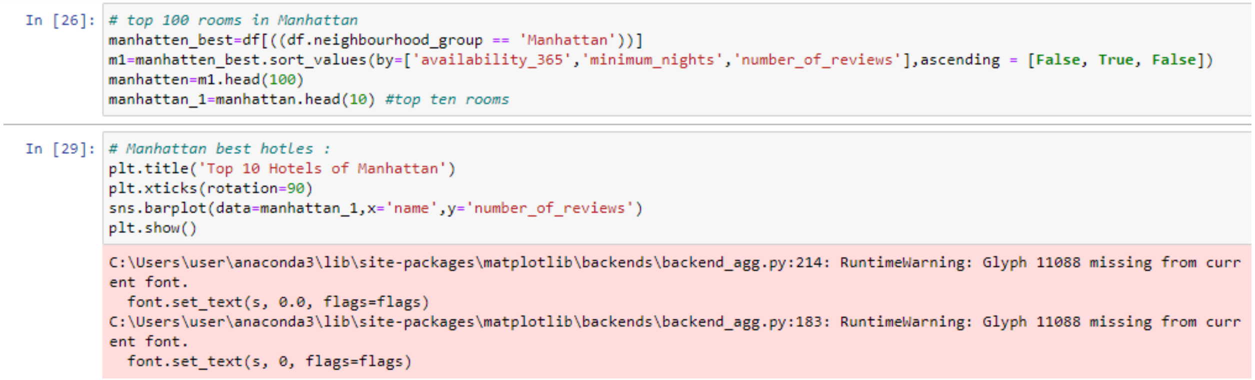
```
# price variance in differnt neighborhood
plt.ylabel('price')
df.groupby('neighbourhood_group')['price'].median().plot(kind='bar')
plt.show()
```



Explanation:

- *df* dataframe – contains Airbnb .csv file
- Used Matplotlib to plot barplot.
- This plot gives you the plot of neighborhood group vs price.

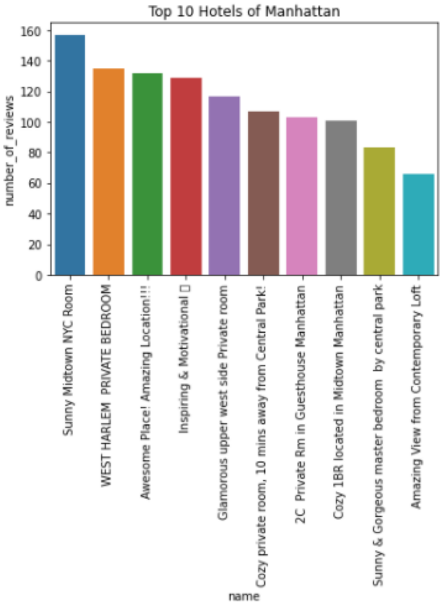
3) Top rated Rooms of various neighborhood :



```
In [26]: # top 100 rooms in Manhattan
         manhatten_best=df[((df.neighbourhood_group == 'Manhattan'))]
         m1=manhatten_best.sort_values(by=['availability_365','minimum_nights','number_of_reviews'],ascending = [False, True, False])
         manhatten=m1.head(100)
         manhattan_1=manhattan.head(10) #top ten rooms
```

```
In [29]: # Manhattan best hotles :
         plt.title('Top 10 Hotels of Manhattan')
         plt.xticks(rotation=90)
         sns.barplot(data=manhattan_1,x='name',y='number_of_reviews')
         plt.show()
```

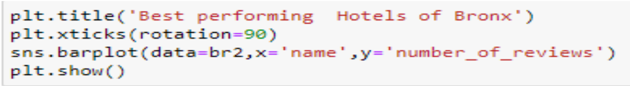
```
C:\Users\user\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:214: RuntimeWarning: Glyph 11088 missing from curr
ent font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\user\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:183: RuntimeWarning: Glyph 11088 missing from curr
ent font.
  font.set_text(s, 0, flags=flags)
```



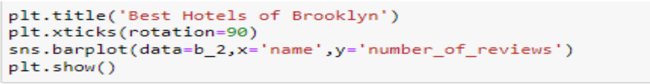
Explanation:

- *df* dataframe – contains Airbnb .csv file
- First selected all the rows from Manhattan neighborhood, sorted it such that rooms with highest availability, least minimum nights and highest number of reviews are in the top and selected top 100 rows.
- Then made bar chart of top 10 rooms vs number of reviews.

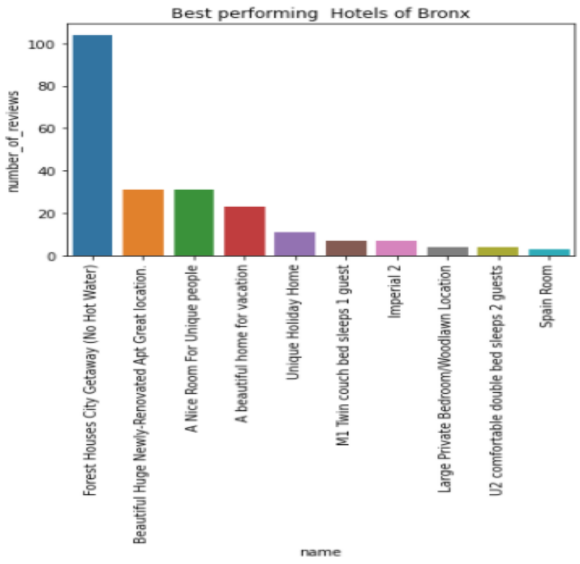
Similarly best rooms of other neighborhood is also found and plotted as bar plot :

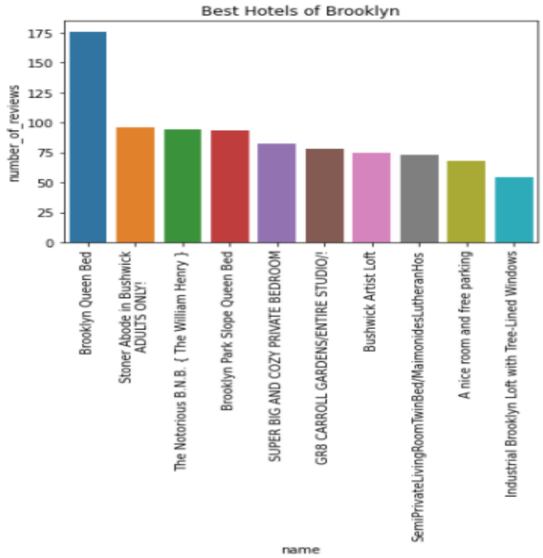


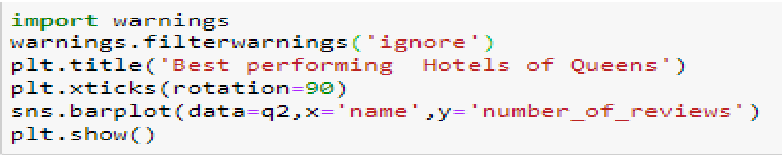
```
plt.title('Best performing  Hotels of Bronx')
plt.xticks(rotation=90)
sns.barplot(data=br2,x='name',y='number_of_reviews')
plt.show()
```



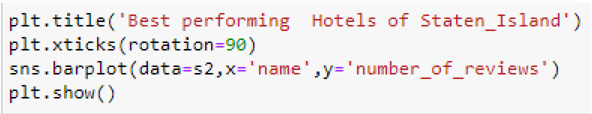
```
plt.title('Best Hotels of Brooklyn')
plt.xticks(rotation=90)
sns.barplot(data=b_2,x='name',y='number_of_reviews')
plt.show()
```



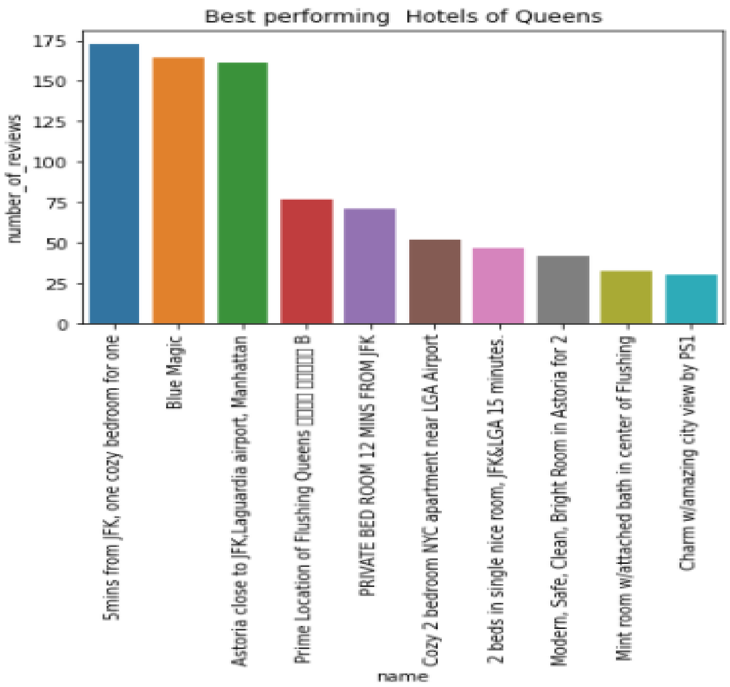


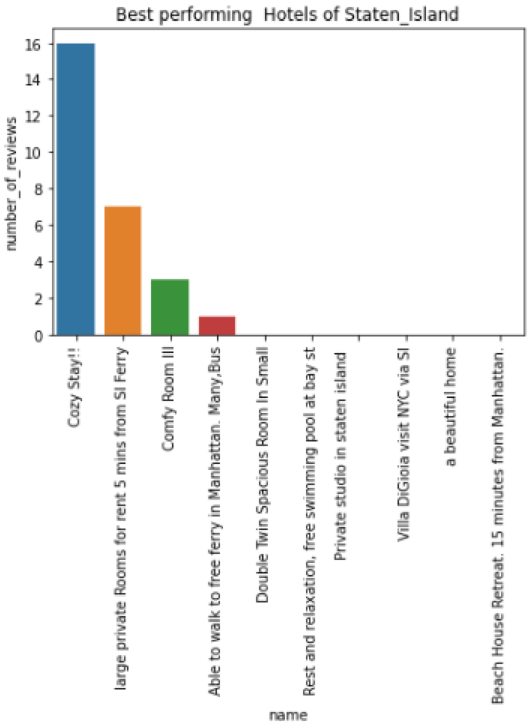


```
import warnings
warnings.filterwarnings('ignore')
plt.title('Best performing  Hotels of Queens')
plt.xticks(rotation=90)
sns.barplot(data=q2,x='name',y='number_of_reviews')
plt.show()
```

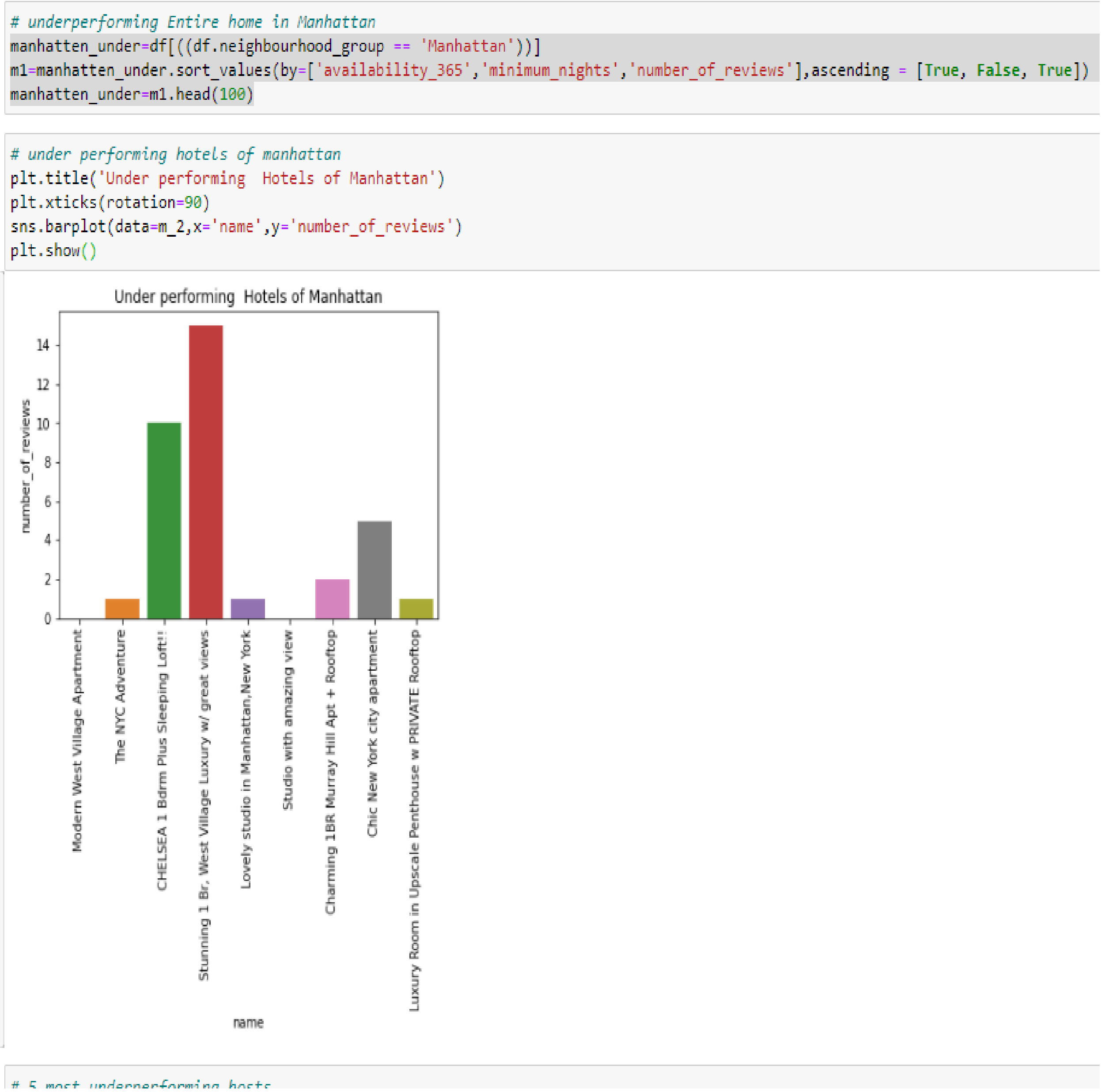


```
plt.title('Best performing  Hotels of Staten_Island')
plt.xticks(rotation=90)
sns.barplot(data=s2,x='name',y='number_of_reviews')
plt.show()
```





4) Bottom most Rooms of various neighborhood :



```
# underperforming Entire home in Manhattan
manhatten_under=df[((df.neighbourhood_group == 'Manhattan'))]
m1=manhatten_under.sort_values(by=['availability_365','minimum_nights','number_of_reviews'],ascending = [True, False, True])
manhatten_under=m1.head(100)
```

```
# under performing hotels of manhattan
plt.title('Under performing  Hotels of Manhattan')
plt.xticks(rotation=90)
sns.barplot(data=m_2,x='name',y='number_of_reviews')
plt.show()
```

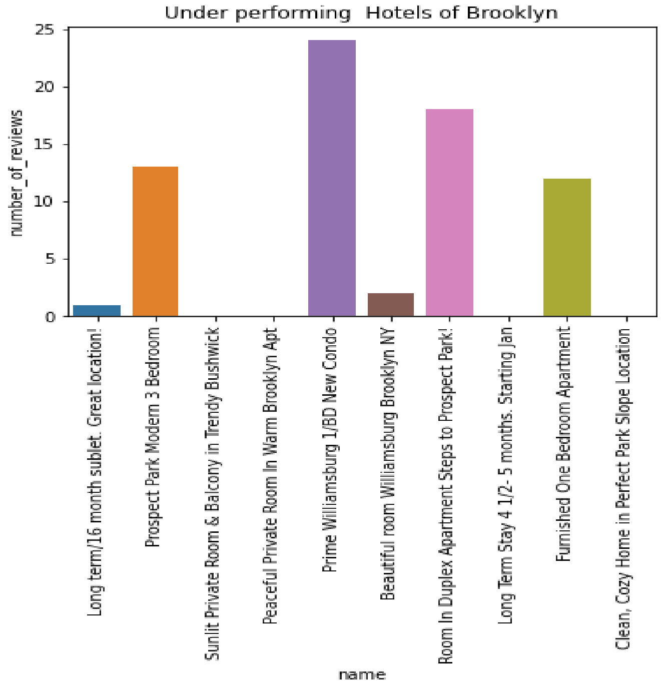
Under performing  Hotels of Manhattan
number_of_reviews
0
2
4
6
8
10
12
14
Modern West Village Apartment
The NYC Adventure
CHELSEA 1 Bdrm Plus Sleeping Loft!!
Stunning 1 Br, West Village Luxury w/ great views
Lovely studio in Manhattan,New York
Studio with amazing view
Charming 1BR Murray Hill Apt + Rooftop
Chic New York city apartment
Luxury Room in Upscale Penthouse w PRIVATE Rooftop
name

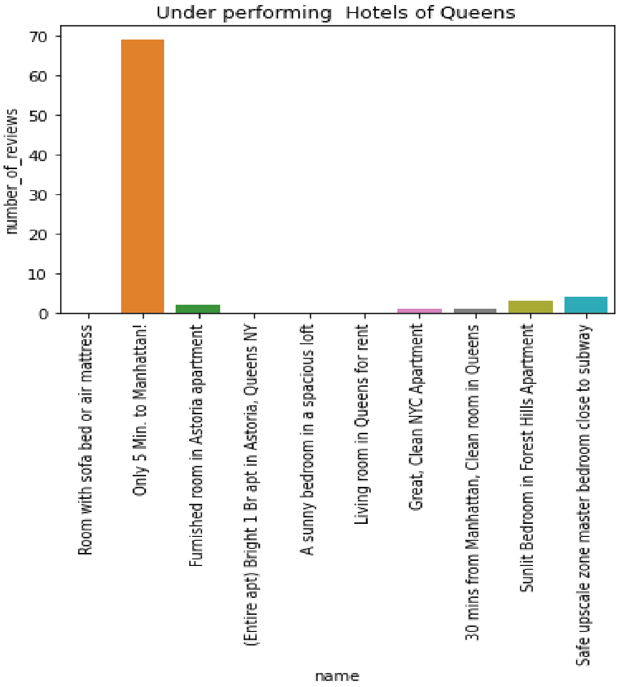
```
# 5 most underperforming hosts
```

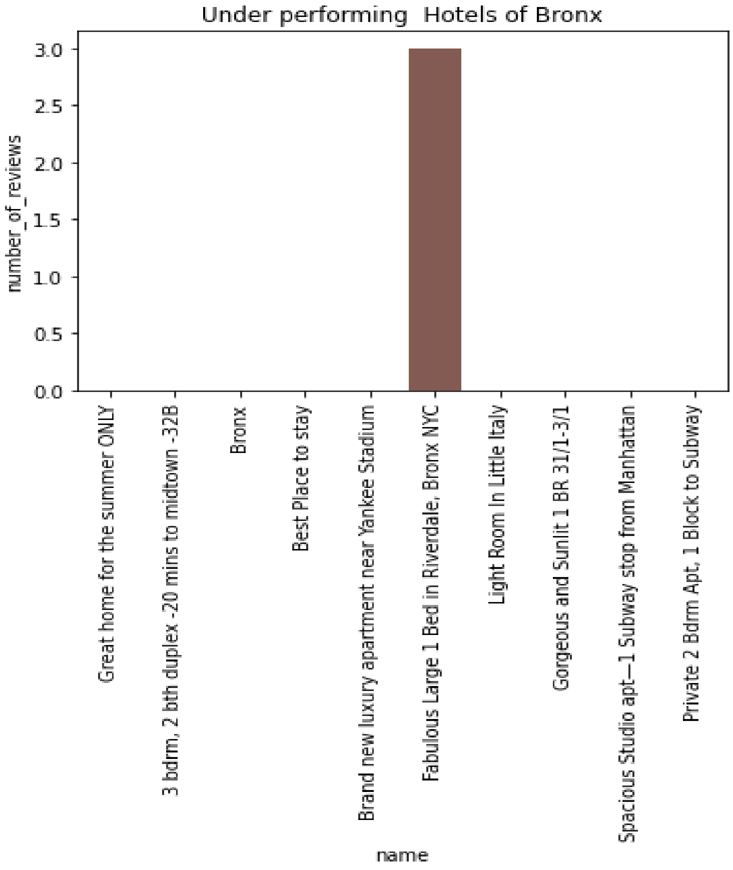
Explanation:

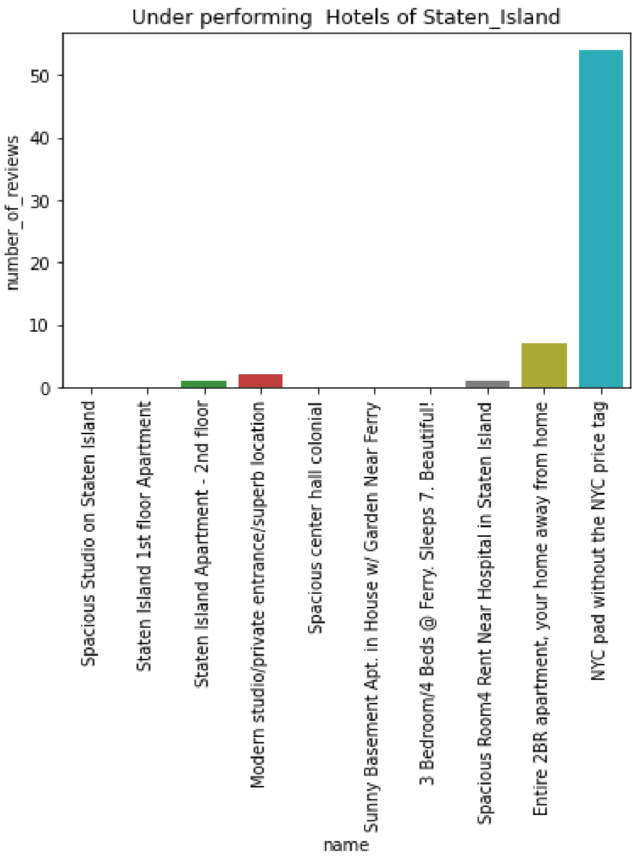
- *df* dataframe – contains Airbnb .csv file
- First selected all the rows from Manhattan neighborhood, sorted it such that rooms with least availability, highest minimum nights and lowest number of reviews are in the top and selected top 100 rows.
- Then made bar chart of bottom 10 rooms vs number of reviews.

Similarly bottom most rooms of other neighborhood is also found and plotted as bar plot :


Under performing  Hotels of Brooklyn
number_of_reviews
Long term/16 month sublet. Great location!
Prospect Park Modern 3 Bedroom
Sunlit Private Room & Balcony in Trendy Bushwick
Peaceful Private Room In Warm Brooklyn Apt
Prime Williamsburg 1/BD New Condo
Beautiful room Williamsburg Brooklyn NY
Room In Duplex Apartment Steps to Prospect Park!
Long Term Stay 4 1/2- 5 months. Starting Jan
Furnished One Bedroom Apartment
Clean, Cozy Home in Perfect Park Slope Location
name


Under performing  Hotels of Queens
number_of_reviews
Room with sofa bed or air mattress
Only 5 Min. to Manhattan!
Furnished room in Astoria apartment
(Entire apt) Bright 1 Br apt in Astoria, Queens NY
A sunny bedroom in a spacious loft
Living room in Queens for rent
Great, Clean NYC Apartment
30 mins from Manhattan, Clean room in Queens
Sunlit Bedroom in Forest Hills Apartment
Safe upscale zone master bedroom close to subway
name


Under performing  Hotels of Bronx
number_of_reviews
Great home for the summer ONLY
3 bdrm, 2 bth duplex -20 mins to midtown -32B
Bronx
Best Place to stay
Brand new luxury apartment near Yankee Stadium
Fabulous Large 1 Bed in Riverdale, Bronx NYC
Light Room In Little Italy
Gorgeous and Sunlit 1 BR 31/1-3/1
Spacious Studio apt—1 Subway stop from Manhattan
Private 2 Bdrm Apt, 1 Block to Subway
name


Under performing  Hotels of Staten_Island
number_of_reviews
Spacious Studio on Staten Island
Staten Island 1st floor Apartment
Staten Island Apartment - 2nd floor
Modern studio/private entrance/superb location
Spacious center hall colonial
Sunny Basement Apt. in House w/ Garden Near Ferry
3 Bedroom/4 Beds @ Ferry. Sleeps 7. Beautiful!
Spacious Room4 Rent Near Hospital in Staten Island
Entire 2BR apartment, your home away from home
NYC pad without the NYC price tag
name