New York City Airbnb Open Data Analysis

Airbnb, Inc is an American company that operates an online marketplace for lodging, primarily homestays for vacation rentals and tourism activities. Based in San Francisco, California, the platform is accessible via the website and mobile app. Airbnb does not own any listed properties; instead, it profits by receiving a commission from each booking. Since 2008, guests and hosts have used Airbnb to travel in a more unique, personalized way.

NYC is the most populous city in the United States, and one of the most popular tourism and business places globally. Questions to be answered during the Analysis of this dataset:

Main Question:

How can we identify the perfect Airbnb listing in New York City by exploring the pricing with neighborhoods, room type, and hosts review rating?

Other Questions to be answered:

```
1. Which are the top neighborhoods, their average prices, and the number of
listings?
2. What are the percent share of different room types?
3. How does the pricing vary with location, property type, and reviews?
4. What are the correlations between the type of hosts and factors like-
reviews & price?
```

This notebook will consist of the following processes:
Data Cleaning
Data Transformation
Data Visualization
Data Analysis of the questions to be answered

Data Cleaning:

In this section we will remove duplicate records and drop unnecessary columns.

To start, we first have to import the necessary libraries:

In [1]:
```
#Importing libraries :
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
import plotly.figure_factory as ff
import plotly.graph_objects as go
import statsmodels.api as sm
sns.set_style("darkgrid")
mpl.rcParams['figure.figsize'] = (20,5)
import warnings
warnings.filterwarnings('ignore')
from scipy import stats
```

Using Pandas Libraries, now it's time to load the Airbnb dataset.

In [2]:
```
# Reading the csv file in pandas dataframe

airbnb = pd.read_csv('C:/Data Analytics/Projects/Capstones/Project Two/Airbnb_Open_Data.
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   NAME                            102349 non-null  object
 2   host id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host name                       102193 non-null  object
 5   neighbourhood group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room type                       102599 non-null  object
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object
 16  service fee                     102326 non-null  object
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   object
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
dtypes: float64(9), int64(2), object(15)
memory usage: 20.4+ MB
```

In [3]:
```
# Now let's check that how many duplicate records we have
airbnb[airbnb.duplicated()].shape[0]
```

Out[3]:
```
541
```

In [4]:
```
# dropping the duplicates
airbnb.drop_duplicates()
```

Out[4]:

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhoo
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensingto
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtov
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harle
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn	Clinton H
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harle
...	...	...	...	...	...	...	
102053	57365208	Cozy bright room near Prospect Park	77326652202	unconfirmed	Mariam	Brooklyn	Flatbu
102054	57365760	Private Bedroom with Amazing Rooftop View	45936254757	verified	Trey	Brooklyn	Bushwi
102055	57366313	Pretty Brooklyn One-Bedroom for 2 to 4 people	23801060917	verified	Michael	Brooklyn	Bedfor Stuyvesa
102056	57366865	Room & private bathroom in historic Harlem	15593031571	unconfirmed	Shireen	Manhattan	Harle
102057	57367417	Rosalee Stewart	93578954226	verified	Stanley	Manhattan	Harle

102058 rows × 26 columns

In [5]:
```
#Now checking for Null values in the dataset
airbnb.isnull().sum()
```

Out[5]:
```
id                                    0
NAME                                250
host id                               0
host_identity_verified              289
host name                           406
neighbourhood group                  29
neighbourhood                        16
lat                                   8
long                                  8
country                             532
country code                        131
instant_bookable                    105
cancellation_policy                  76
room type                             0
Construction year                   214
price                               247
service fee                         273
minimum nights                      409
number of reviews                   183
last review                       15893
reviews per month                 15879
review rate number                  326
calculated host listings count      319
availability 365                    448
house_rules                       52131
license                          102597
dtype: int64
```

In [6]:
```
#Now checking for Unique values in the dataset
airbnb.nunique()
```

Out[6]:
```
id                               102058
NAME                              61281
host id                          102057
host_identity_verified                2
host name                         13190
neighbourhood group                   7
neighbourhood                       224
lat                               21991
long                              17774
country                               1
country code                          1
instant_bookable                      2
cancellation_policy                   3
room type                             4
Construction year                    20
price                              1151
service fee                         231
minimum nights                      153
number of reviews                   476
last review                        2477
reviews per month                  1016
review rate number                    5
calculated host listings count       78
availability 365                    438
house_rules                        1976
license                               1
dtype: int64
```

In [7]:
```
#Here few columns like "id", "NAME", "host id", "host name", "country", "country code",
#"license" are irrelevant and insignificant to our data analysis.
#Therefore, let's proceed with removing columns that are not important and handling of m

airbnb.drop(columns=["id","NAME","host id","host name","country","country code","last re
             axis=1, inplace=True)
airbnb
```

Out[7]:

	host_identity_verified	neighbourhood group	neighbourhood	lat	long	instant_bookable	cancellatic
0	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	False	
1	verified	Manhattan	Midtown	40.75362	-73.98377	False	n
2	NaN	Manhattan	Harlem	40.80902	-73.94190	True	
3	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	True	n
4	verified	Manhattan	East Harlem	40.79851	-73.94399	False	n
...	...	...	...	...	...	...	
102594	verified	Brooklyn	Williamsburg	40.70862	-73.94651	False	
102595	unconfirmed	Manhattan	Morningside Heights	40.80460	-73.96545	True	n
102596	unconfirmed	Brooklyn	Park Slope	40.67505	-73.98045	True	n
102597	unconfirmed	Queens	Long Island City	40.74989	-73.93777	True	
102598	unconfirmed	Manhattan	Upper West Side	40.76807	-73.98342	False	

102599 rows × 17 columns

As we have removed all unwanted columns from the dataset!

Then, it's time to do

Data Transformation

In [8]:
```
#In the first column we need to replace null values with unconfirmed
airbnb['host_identity_verified'] = airbnb['host_identity_verified'].fillna('unconfirmed'

#Replacing null values in the column "reviews per month" with 0 in the dataset
airbnb['reviews per month'].fillna(0,inplace = True)

airbnb
```

Out[8]:

	host_identity_verified	neighbourhood group	neighbourhood	lat	long	instant_bookable	cancellatic

0	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	False	
1	verified	Manhattan	Midtown	40.75362	-73.98377	False	n
2	unconfirmed	Manhattan	Harlem	40.80902	-73.94190	True	
3	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	True	n
4	verified	Manhattan	East Harlem	40.79851	-73.94399	False	n
...	...	...	...	...	...	...	
102594	verified	Brooklyn	Williamsburg	40.70862	-73.94651	False	
102595	unconfirmed	Manhattan	Morningside Heights	40.80460	-73.96545	True	n
102596	unconfirmed	Brooklyn	Park Slope	40.67505	-73.98045	True	n
102597	unconfirmed	Queens	Long Island City	40.74989	-73.93777	True	
102598	unconfirmed	Manhattan	Upper West Side	40.76807	-73.98342	False	

102599 rows × 17 columns

In [9]:
```
#Now checking count of Neighbourhood group
airbnb['neighbourhood group'].value_counts()
```

Out[9]:
```
Manhattan        43792
Brooklyn         41842
Queens           13267
Bronx             2712
Staten Island      955
brookln              1
manhatan             1
Name: neighbourhood group, dtype: int64
```

In [10]:
```
# Let's fix 2 miswritten neighbourhood groups
airbnb['neighbourhood group'].replace({'manhatan':'Manhattan', 'brookln':'Brooklyn'}, in
airbnb['neighbourhood group'].value_counts()
```

Out[10]:
```
Manhattan        43793
Brooklyn         41843
Queens           13267
Bronx             2712
Staten Island      955
Name: neighbourhood group, dtype: int64
```

Now it's time to convert "price" and "service_fee" columns from object type to float64. Also need to remove the $ sign and unwanted characters from the records.

In [11]:
```
# This function will remove dollar sign and unwanted characters from the column records
def remove_dollar_sign(value):
    if pd.isna(value):
        return np.NaN
```

```
    else:
        return float(value.replace("$","").replace(",","").replace(" ",""))
```

In [12]:
```
# Applying function on "price" and "service fee" columns
airbnb["price"]=airbnb["price"].apply(lambda x: remove_dollar_sign(x))
airbnb["service fee"]=airbnb["service fee"].apply(lambda x: remove_dollar_sign(x))
airbnb
```

Out[12]:

	host_identity_verified	neighbourhood group	neighbourhood	lat	long	instant_bookable	cancellatic
0	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	False	
1	verified	Manhattan	Midtown	40.75362	-73.98377	False	n
2	unconfirmed	Manhattan	Harlem	40.80902	-73.94190	True	
3	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	True	n
4	verified	Manhattan	East Harlem	40.79851	-73.94399	False	n
...	...	...	...	...	...	...	
102594	verified	Brooklyn	Williamsburg	40.70862	-73.94651	False	
102595	unconfirmed	Manhattan	Morningside Heights	40.80460	-73.96545	True	n
102596	unconfirmed	Brooklyn	Park Slope	40.67505	-73.98045	True	n
102597	unconfirmed	Queens	Long Island City	40.74989	-73.93777	True	
102598	unconfirmed	Manhattan	Upper West Side	40.76807	-73.98342	False	

102599 rows × 17 columns

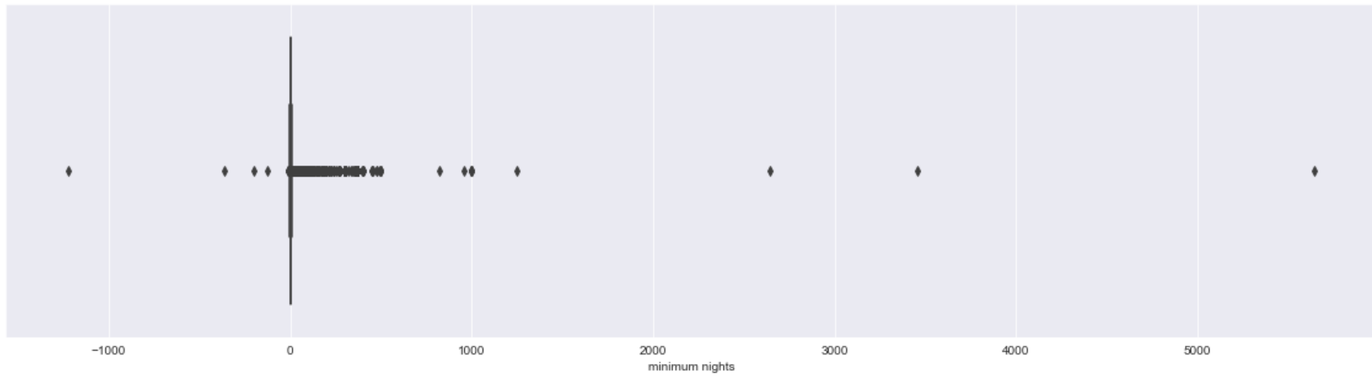
Now let's check minimum nights column values

In [13]:
```
airbnb['minimum nights'].describe()
```

Out[13]:
```
count    102190.000000
mean          8.135845
std          30.553781
min       -1223.000000
25%           2.000000
50%           3.000000
75%           5.000000
max        5645.000000
Name: minimum nights, dtype: float64
```

In [14]:
```
# some absurd values, so lets build a boxplot to understand it better
sns.boxplot(x='minimum nights', data=airbnb)
```

Out[14]:
```
<AxesSubplot:xlabel='minimum nights'>
```



In [15]:
```python
#As we can see negative values for "minimum nights" column. So let's check how many reco
airbnb[airbnb['minimum nights']<1]
```

Out[15]:

	host_identity_verified	neighbourhood group	neighbourhood	lat	long	instant_bookable	cancellatior
176	unconfirmed	Brooklyn	Fort Greene	40.69098	-73.97113	False	
352	unconfirmed	Brooklyn	Crown Heights	40.67174	-73.95663	NaN	
398	verified	Brooklyn	Kensington	40.64302	-73.97255	False	
421	verified	Manhattan	Nolita	40.72094	-73.99706	False	
441	verified	Manhattan	Harlem	40.80497	-73.95016	False	m
478	unconfirmed	Manhattan	Upper West Side	40.77886	-73.98042	True	
525	verified	Brooklyn	Bedford-Stuyvesant	40.68967	-73.95445	False	m
42446	verified	Bronx	Hunts Point	40.81731	-73.89052	False	m
42500	unconfirmed	Manhattan	Hell's Kitchen	40.76694	-73.98773	True	
42538	verified	Brooklyn	Bedford-Stuyvesant	40.68470	-73.94350	True	m
69749	verified	Brooklyn	Williamsburg	40.71534	-73.94906	False	m
91271	unconfirmed	Manhattan	Midtown	40.74433	-73.98318	False	m
91357	verified	Brooklyn	Gowanus	40.67070	-73.99118	True	m

In [16]:
```python
#Replacing less than 1 value records in the column "minimum nights"
airbnb.loc[airbnb['minimum nights']<1, 'minimum nights'] = np.nan
```

In [17]:
```python
#Now checking records greater than 365 value.
airbnb[airbnb['minimum nights']>365]
```

Out[17]:

	host_identity_verified	neighbourhood	neighbourhood	lat	long	instant_bookable	cancellatior

		group					
167	unconfirmed	Manhattan	Harlem	40.82704	-73.94907	False	
186	verified	Brooklyn	Bedford-Stuyvesant	40.67992	-73.94750	False	
263	verified	Brooklyn	Bedford-Stuyvesant	40.68236	-73.94314	False	
299	unconfirmed	Brooklyn	Greenpoint	40.73119	-73.95578	False	
350	verified	Brooklyn	Crown Heights	40.67473	-73.94494	False	
473	unconfirmed	Manhattan	Washington Heights	40.84468	-73.94303	True	
1306	unconfirmed	Brooklyn	Bushwick	40.70202	-73.92402	False	
2855	verified	Manhattan	Battery Park City	40.71239	-74.01620	True	m
5768	verified	Manhattan	Greenwich Village	40.73293	-73.99782	False	m
7356	verified	Queens	Long Island City	40.75104	-73.93863	True	m
8015	verified	Manhattan	Harlem	40.82135	-73.95521	False	m
10830	verified	Queens	Long Island City	40.74654	-73.95778	False	
11194	unconfirmed	Brooklyn	Crown Heights	40.67255	-73.94914	False	m
13405	verified	Manhattan	Harlem	40.82915	-73.94034	False	m
14286	unconfirmed	Brooklyn	Kensington	40.64779	-73.97956	True	m
15947	unconfirmed	Manhattan	Midtown	40.74513	-73.98475	False	m
26342	unconfirmed	Brooklyn	Williamsburg	40.71772	-73.95059	False	m
34488	verified	Brooklyn	Bedford-Stuyvesant	40.69974	-73.94658	True	m
38665	unconfirmed	Manhattan	Greenwich Village	40.73094	-73.99900	True	
42354	unconfirmed	Brooklyn	Prospect-Lefferts Gardens	40.66220	-73.96208	True	m
42369	unconfirmed	Manhattan	Upper East Side	40.76174	-73.96625	False	m
42398	verified	Brooklyn	Bushwick	40.70235	-73.92892	True	

42407	unconfirmed	Brooklyn	Bay Ridge	40.63189	-74.02322	False	m
47621	unconfirmed	Brooklyn	Williamsburg	40.70898	-73.94885	False	m
67608	unconfirmed	Manhattan	Upper East Side	40.77030	-73.96115	False	
69482	unconfirmed	Brooklyn	Bushwick	40.70202	-73.92402	False	
71031	unconfirmed	Manhattan	Battery Park City	40.71239	-74.01620	True	
72663	unconfirmed	Queens	Long Island City	40.74654	-73.95778	True	
73027	verified	Brooklyn	Crown Heights	40.67255	-73.94914	True	
83506	unconfirmed	Brooklyn	Williamsburg	40.71772	-73.95059	True	
84108	verified	Brooklyn	Kensington	40.64779	-73.97956	True	
85769	verified	Manhattan	Midtown	40.74513	-73.98475	False	m
91347	unconfirmed	Queens	Rego Park	40.73048	-73.85331	False	m
97421	verified	Brooklyn	Bedford-Stuyvesant	40.69974	-73.94658	True	
99691	unconfirmed	Manhattan	Harlem	40.81102	-73.94712	False	m

In [18]:
```
#Replacing greater than 365 value records in the column "minimum nights"
airbnb.loc[airbnb['minimum nights']>365, 'minimum nights'] = np.nan
```

In [19]:
```
airbnb['minimum nights'].describe()
```

Out[19]:
```
count    102142.000000
mean          7.856513
std          17.051180
min           1.000000
25%           2.000000
50%           3.000000
75%           5.000000
max         365.000000
Name: minimum nights, dtype: float64
```

In [20]:
```
airbnb['availability 365'].describe()
```

Out[20]:
```
count    102151.000000
mean        141.133254
std         135.435024
min         -10.000000
25%           3.000000
50%          96.000000
75%         269.000000
max        3677.000000
Name: availability 365, dtype: float64
```

In [21]:
```
airbnb['availability 365'] = np.where(airbnb['availability 365']<0, airbnb['availability
```

In [22]:
```
airbnb['availability 365'] = np.where(airbnb['availability 365']>365, 365, airbnb['avail
```

In [23]:
```
airbnb['availability 365'].describe()
```

Out[23]:
```
count    102151.000000
mean        140.313947
std         133.417297
min           0.000000
25%           4.000000
50%          96.000000
75%         269.000000
max         365.000000
Name: availability 365, dtype: float64
```

In [24]:
```
#Let's look at how many null values we have left
(airbnb.isnull().sum()).sum()
```

Out[24]:
```
2709
```

In [25]:
```
#This is not a lot at all. Therefore we will simply delete the rows with null values.

airbnb.dropna(inplace=True)
```

Now we can convert the columns to integer type. We will simply create a dictionary of our column names, and assign their new type.

In [26]:
```
convert_dict = {'Construction year': int, 'price': int,
       'service fee': int, 'minimum nights': int, 'review rate number': int,
       'availability 365': int}

airbnb = airbnb.astype(convert_dict)
```

In [27]:
```
#Now checking last time variables inside out dataset.
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100293 entries, 0 to 102598
Data columns (total 17 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   host_identity_verified          100293 non-null  object
 1   neighbourhood group             100293 non-null  object
 2   neighbourhood                   100293 non-null  object
 3   lat                             100293 non-null  float64
 4   long                            100293 non-null  float64
 5   instant_bookable                100293 non-null  object
 6   cancellation_policy             100293 non-null  object
 7   room type                       100293 non-null  object
 8   Construction year               100293 non-null  int32
 9   price                           100293 non-null  int32
 10  service fee                     100293 non-null  int32
 11  minimum nights                  100293 non-null  int32
 12  number of reviews               100293 non-null  float64
 13  reviews per month               100293 non-null  float64
 14  review rate number              100293 non-null  int32
 15  calculated host listings count  100293 non-null  float64
 16  availability 365                100293 non-null  int32
dtypes: float64(5), int32(6), object(6)
memory usage: 11.5+ MB
```

Final Dataset

In [28]:
```
airbnb
```

Out[28]:

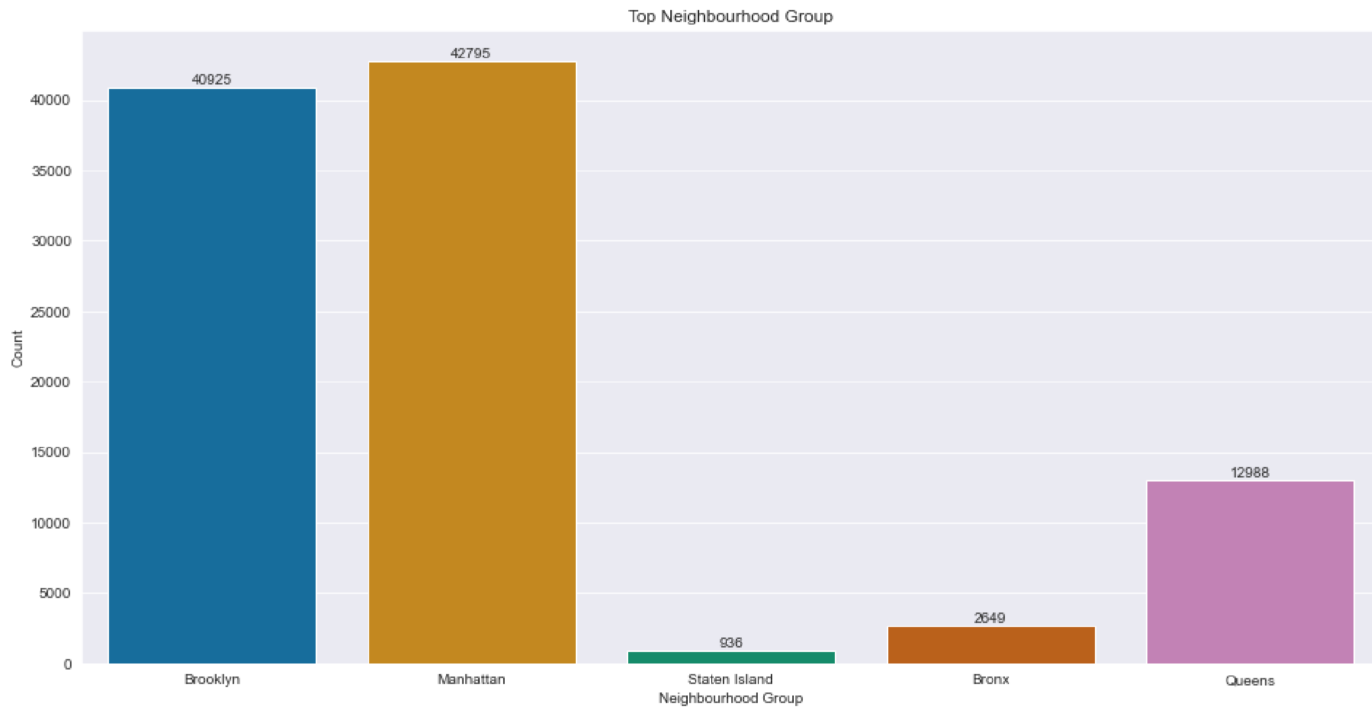
	host_identity_verified	neighbourhood group	neighbourhood	lat	long	instant_bookable	cancellatic
0	unconfirmed	Brooklyn	Kensington	40.64749	-73.97237	False	
1	verified	Manhattan	Midtown	40.75362	-73.98377	False	n
2	unconfirmed	Manhattan	Harlem	40.80902	-73.94190	True	
3	unconfirmed	Brooklyn	Clinton Hill	40.68514	-73.95976	True	n
4	verified	Manhattan	East Harlem	40.79851	-73.94399	False	n
...	...	...	...	...	...	...	
102594	verified	Brooklyn	Williamsburg	40.70862	-73.94651	False	
102595	unconfirmed	Manhattan	Morningside Heights	40.80460	-73.96545	True	n
102596	unconfirmed	Brooklyn	Park Slope	40.67505	-73.98045	True	n
102597	unconfirmed	Queens	Long Island City	40.74989	-73.93777	True	
102598	unconfirmed	Manhattan	Upper West Side	40.76807	-73.98342	False	

100293 rows × 17 columns

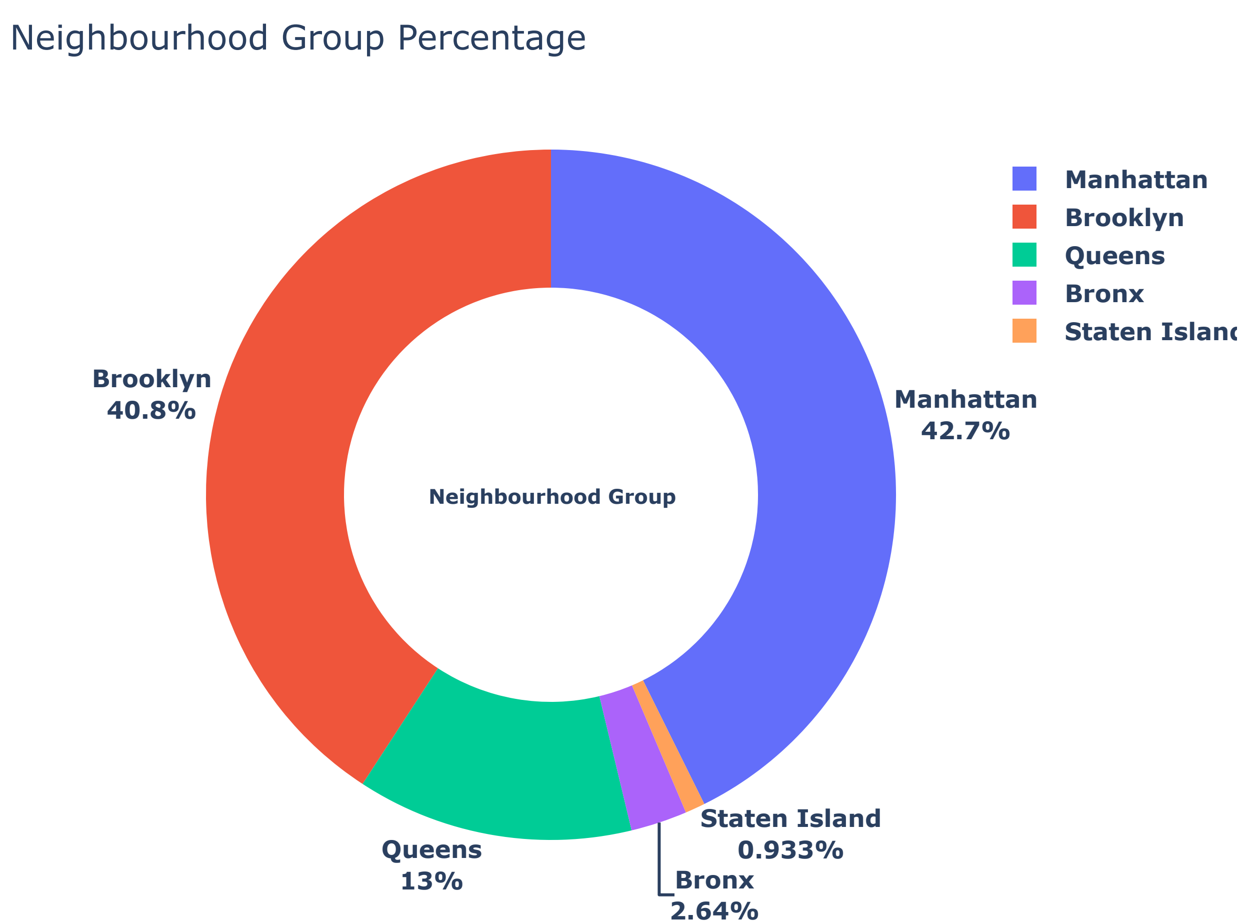
Data Visualization:

Now it's time to create visualizations to explore the New Your city Airbnb Open Data Analysis based on a defined issue tree.

In [29]:
```
#Plot bar chart on the room type:
plt.figure(figsize=(16,8))
plt.title("Top Neighbourhood Group")
ax = sns.countplot(airbnb['neighbourhood group'], palette="colorblind")
plt.xlabel("Neighbourhood Group")
plt.ylabel("Count")
ax.bar_label(ax.containers[0])
plt.show()
```

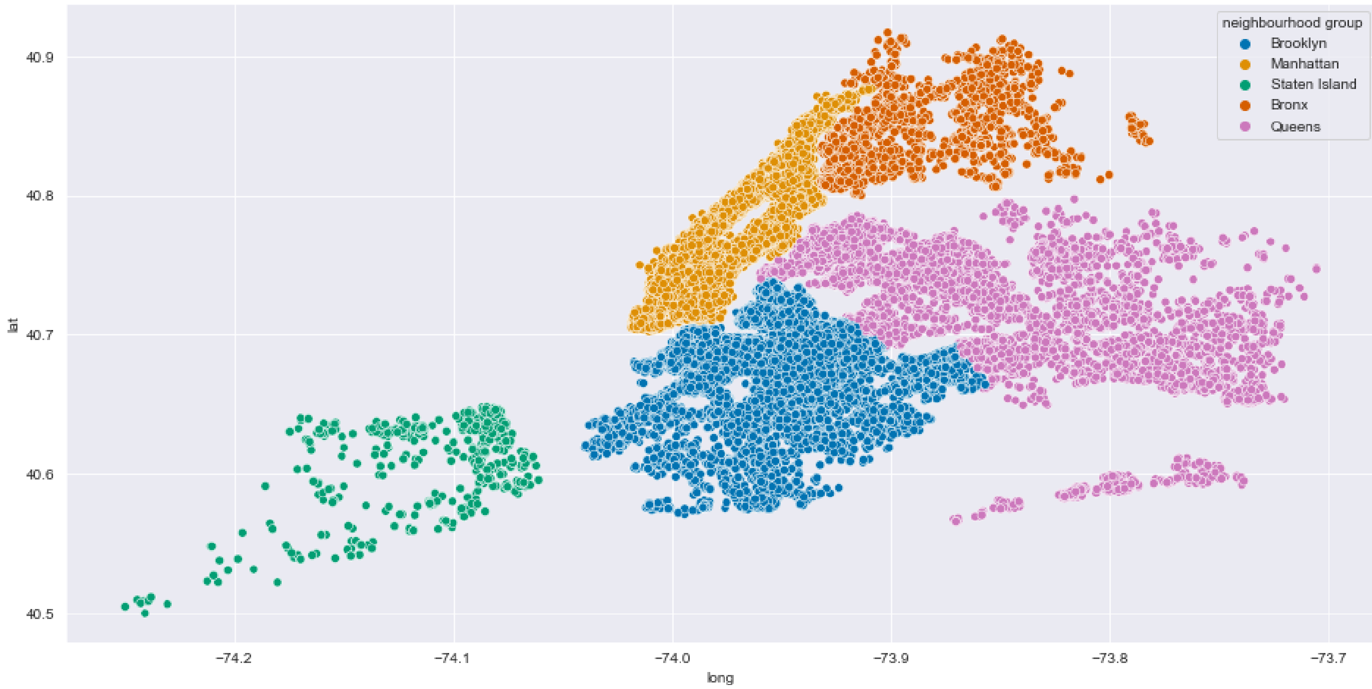


In [30]:
```
#Plot pie chart on the neighbourhood group percentage distribution:
airbnb['tmp']=1
fig=px.pie(airbnb, names="neighbourhood group", values='tmp',hole=0.6, title="Neighbourh
fig.update_traces(textposition='outside', textinfo='percent+label')
fig.update_layout(title_text="Neighbourhood Group Percentage",
                  annotations=[dict(text='Neighbourhood Group', x=0.5, y=0.5, font_size=
```

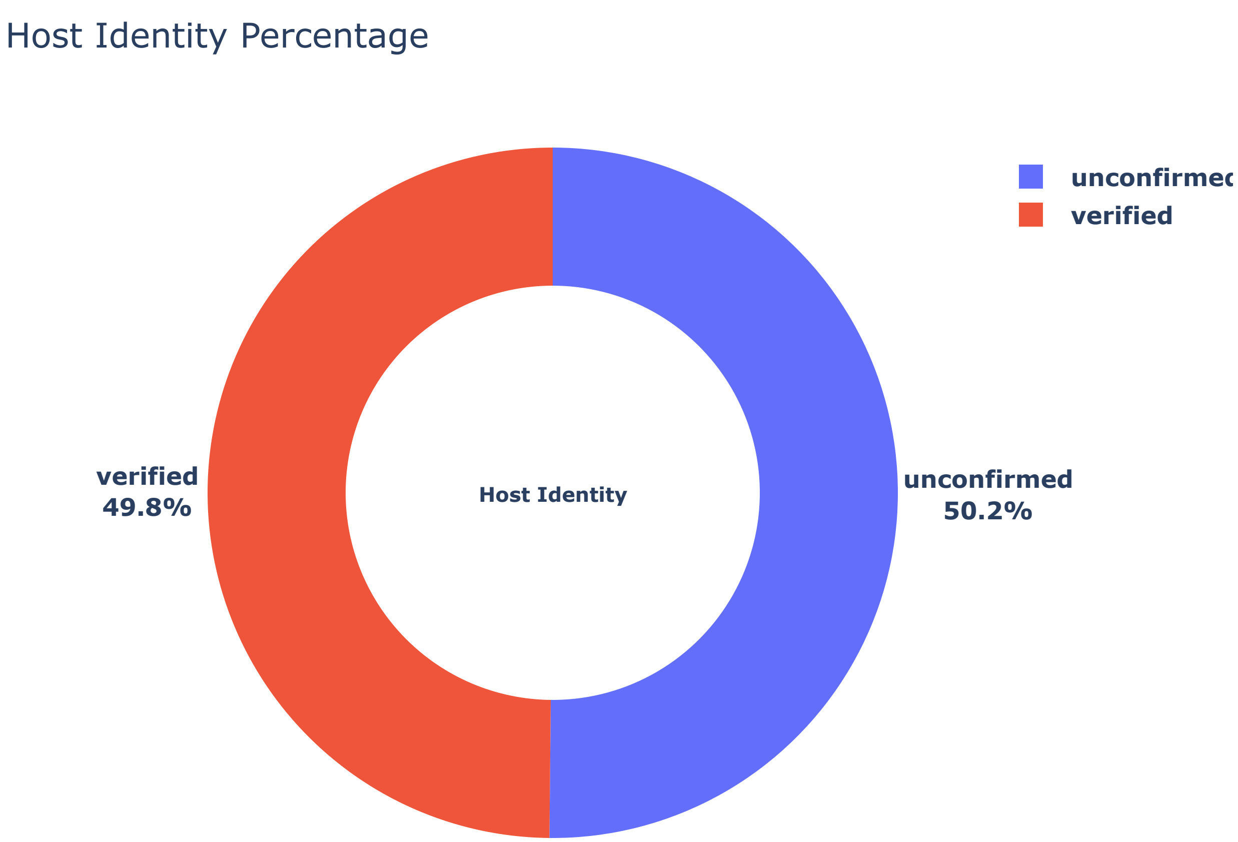


Most of the neighbourhood are in Brooklyn and Manhattan, over 40%

In [31]:
```
#Plot geographical graph for different neighbourhood groups
f,ax = plt.subplots(figsize=(16,8))
ax = sns.scatterplot(x=airbnb['long'],y=airbnb['lat'],hue=airbnb['neighbourhood group'],
plt.show()
```

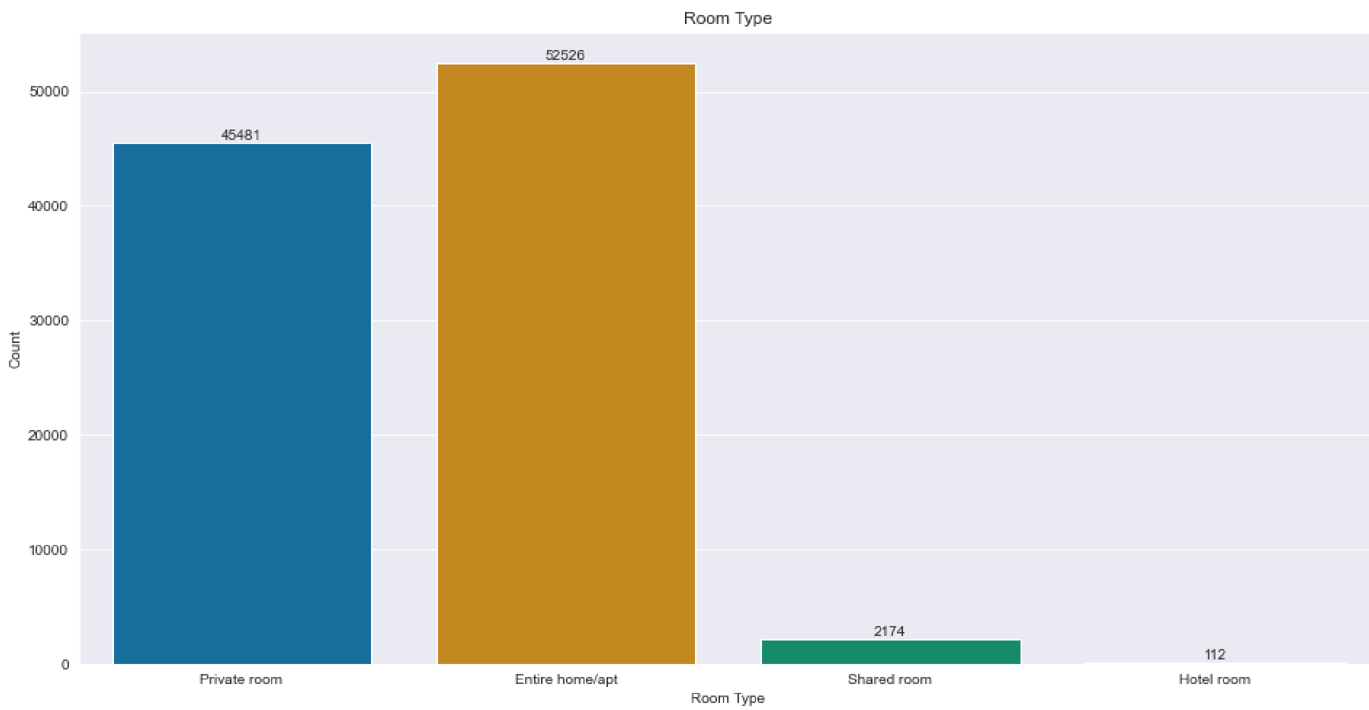


In [32]:
```
#Plot pie chart on the host identity percentage distribution:
airbnb['tmp']=1
fig=px.pie(airbnb, names="host_identity_verified", values='tmp',hole=0.6, title="Host Id
fig.update_traces(textposition='outside', textinfo='percent+label')
fig.update_layout(title_text="Host Identity Percentage",
                  annotations=[dict(text='Host Identity', x=0.5, y=0.5, font_size=10, sh
```



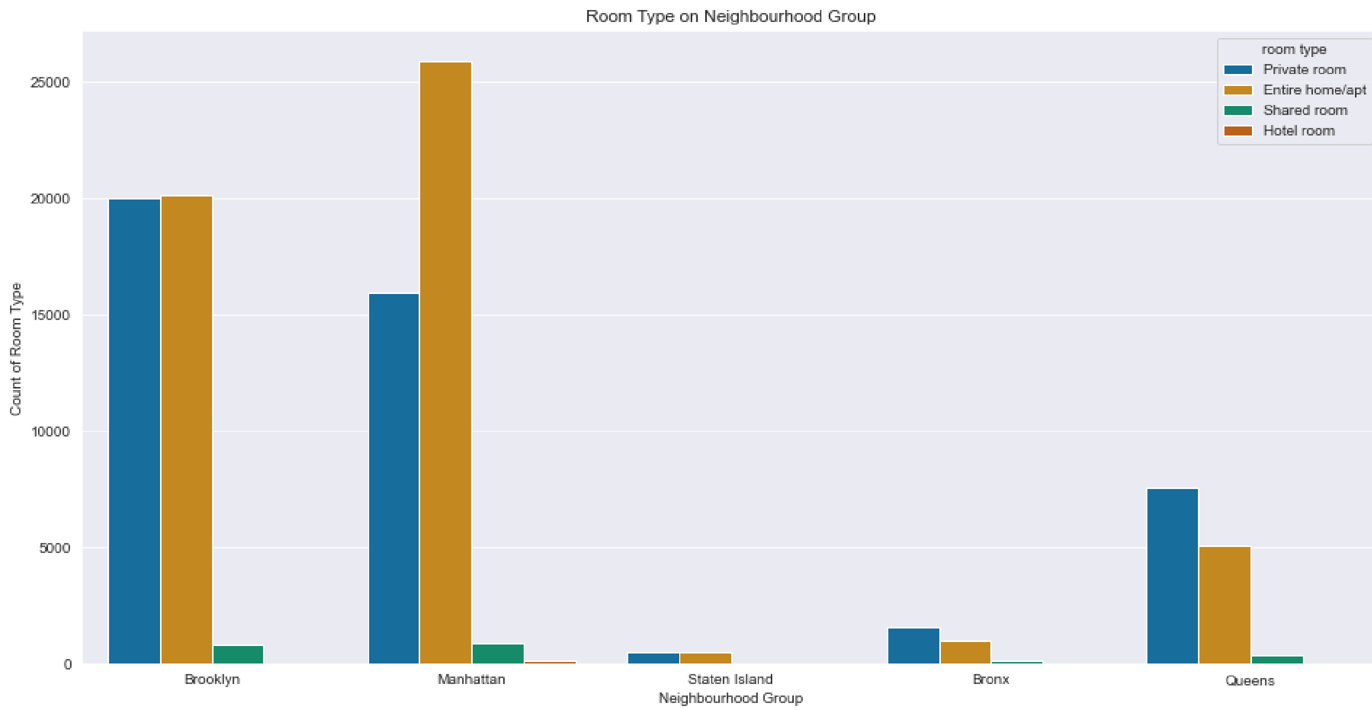
The above pie chart shows that almost 50% of hosts identities have been verified.

In [33]:
```
#Plot bar chart on the room type:
plt.figure(figsize=(16,8))
plt.title("Room Type")
ax = sns.countplot(airbnb['room type'], palette="colorblind")
plt.xlabel("Room Type")
plt.ylabel("Count")
ax.bar_label(ax.containers[0])
plt.show()
```



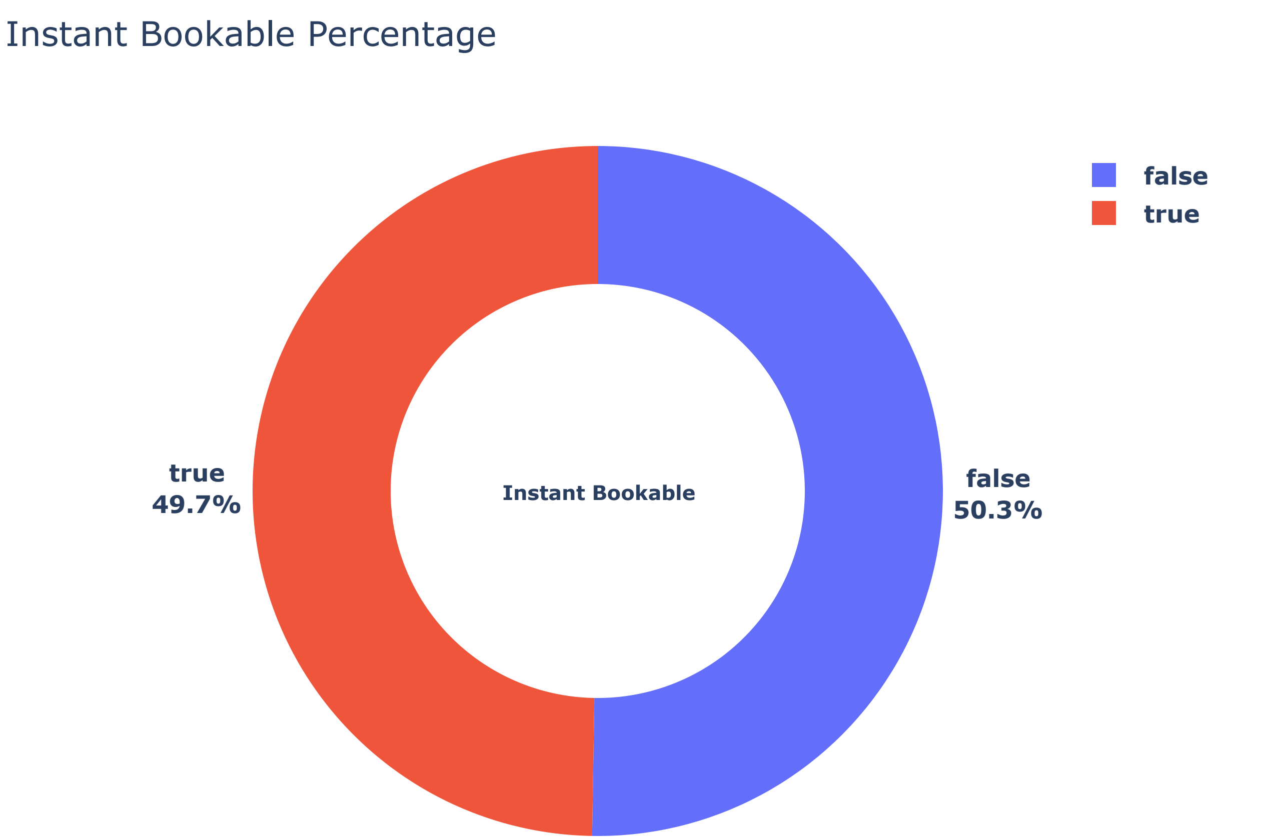
From the above graph, we can conclude the maximum number of Airbnb in New York City is the "Entire home/apt" type followed by "Private room". Very few options for "Shared room" and "Hotel room" types.

In [34]:
```
#Plot bar chart for room type on neighbourhood group
plt.figure(figsize=(16,8))
plt.title("Room Type on Neighbourhood Group")
ax = sns.countplot(airbnb['neighbourhood group'],hue=airbnb['room type'], palette="colori
plt.xlabel("Neighbourhood Group")
plt.ylabel("Count of Room Type")
plt.show()
```



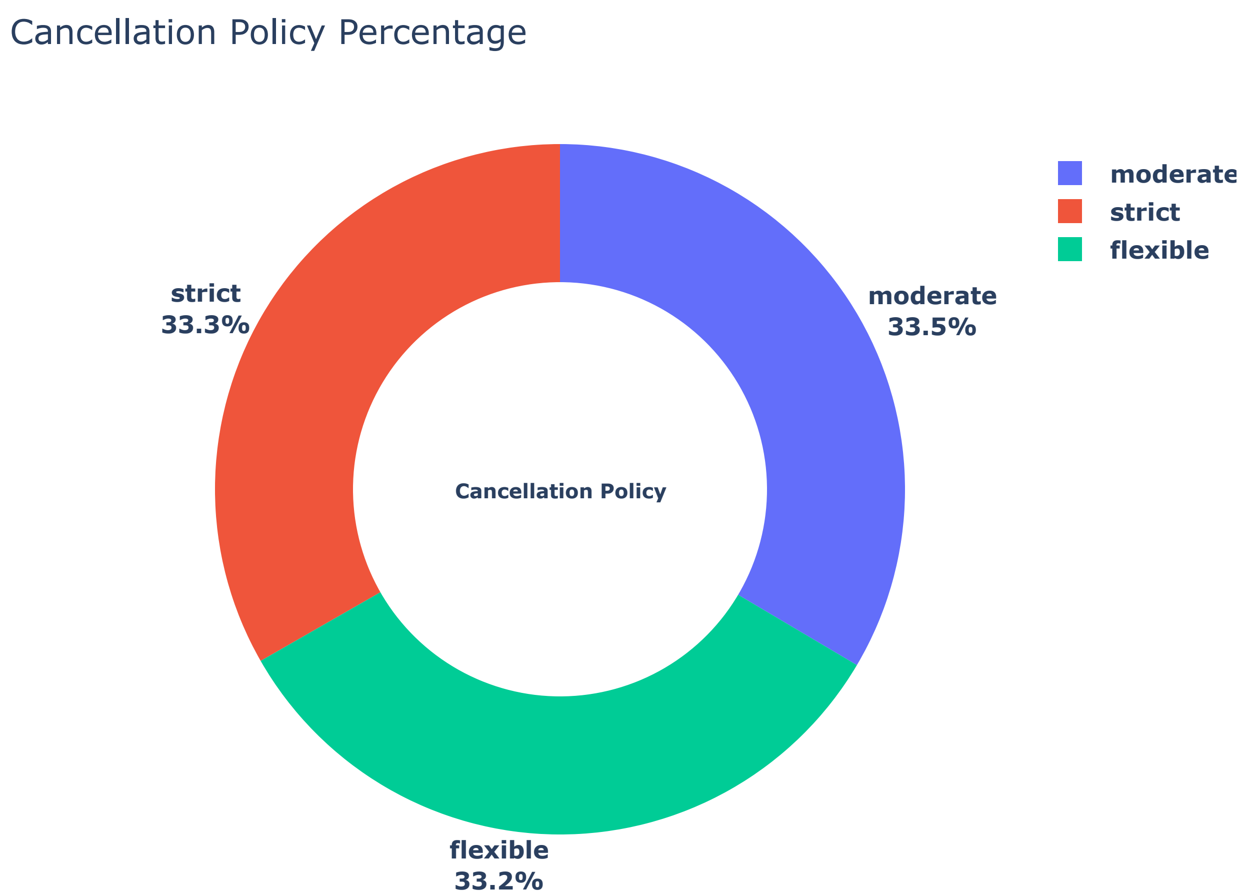
The above graph shows that the Entire Home/Apartment is listed most near Manhattan, The number of Private airbnbs in Brooklyn is way more than in Manhattan. Also, the total number of Hotel rooms are comparetively very less than anyother type.

In [35]:
```
#Plot pie chart on the instant booking option:
airbnb['tmp']=1
fig=px.pie(airbnb, names="instant_bookable", values='tmp',hole=0.6, title="Instant Booka
fig.update_traces(textposition='outside', textinfo='percent+label')
fig.update_layout(title_text="Instant Bookable Percentage",
                  annotations=[dict(text='Instant Bookable', x=0.5, y=0.5, font_size=10,
```

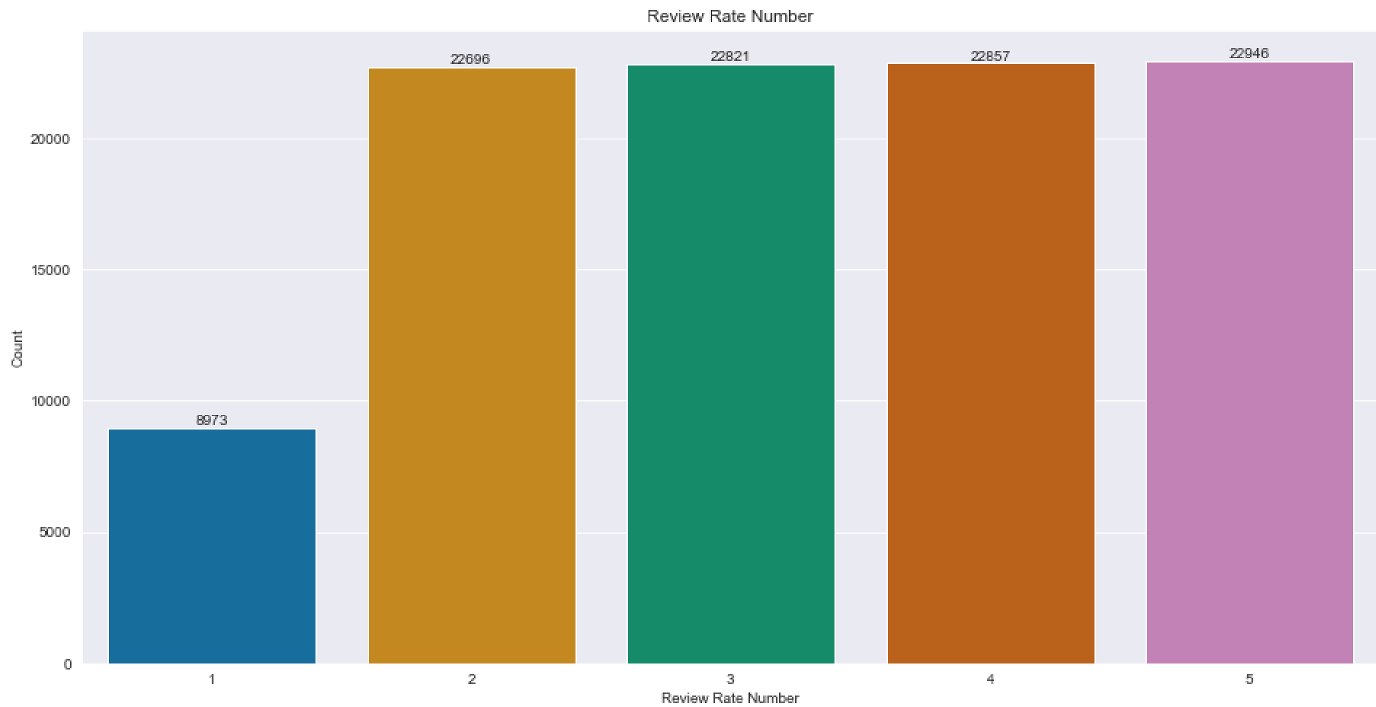


The above pie chart indicates that almost 50% bookings will be available instantly.

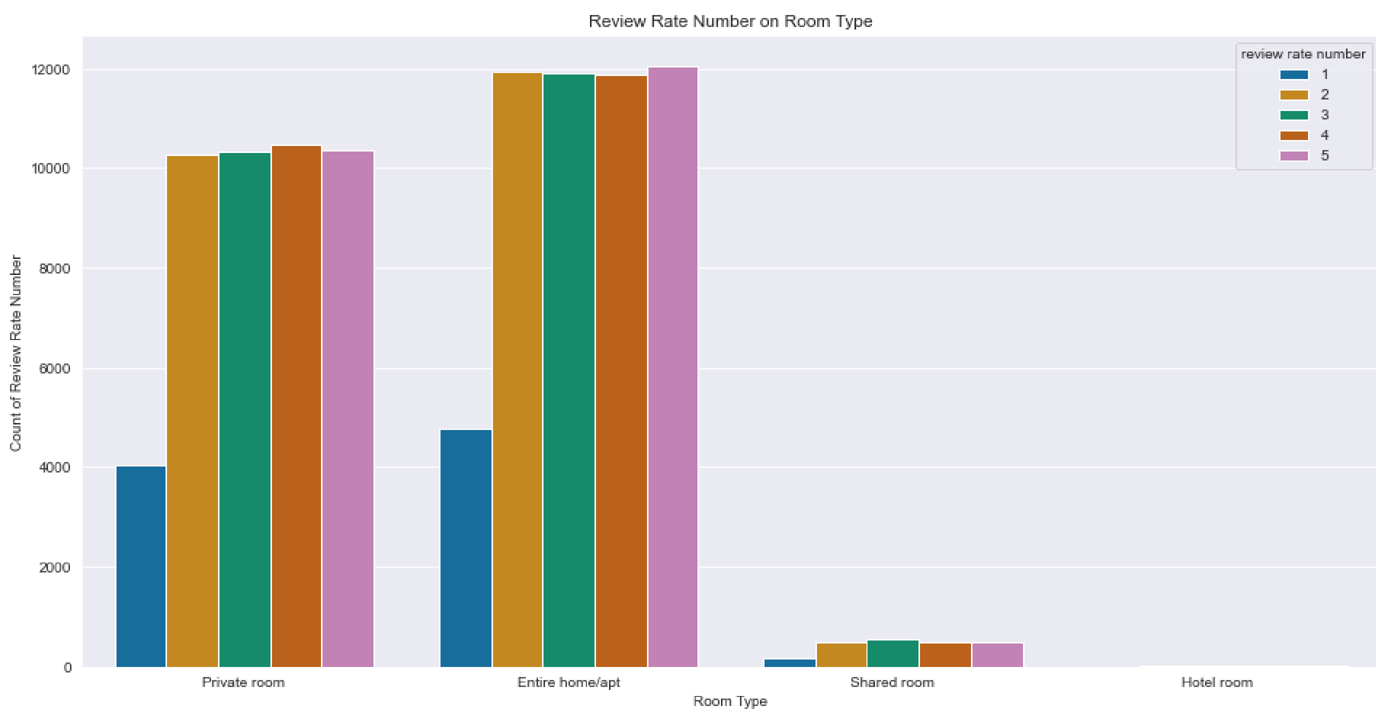
In [36]:
```
#Plot pie chart on the Cancellation Policy Percentage Distribution:
airbnb['tmp']=1
fig=px.pie(airbnb, names="cancellation_policy", values='tmp',hole=0.6, title="Cancellati
fig.update_traces(textposition='outside', textinfo='percent+label')
fig.update_layout(title_text="Cancellation Policy Percentage",
                  annotations=[dict(text='Cancellation Policy', x=0.5, y=0.5, font_size=
```



In [37]:
```
#Plot bar chart on the Review Rate Number:
plt.figure(figsize=(16,8))
plt.title("Review Rate Number")
ax = sns.countplot(airbnb['review rate number'], palette="colorblind")
plt.xlabel("Review Rate Number")
plt.ylabel("Count")
ax.bar_label(ax.containers[0])
plt.show()
```



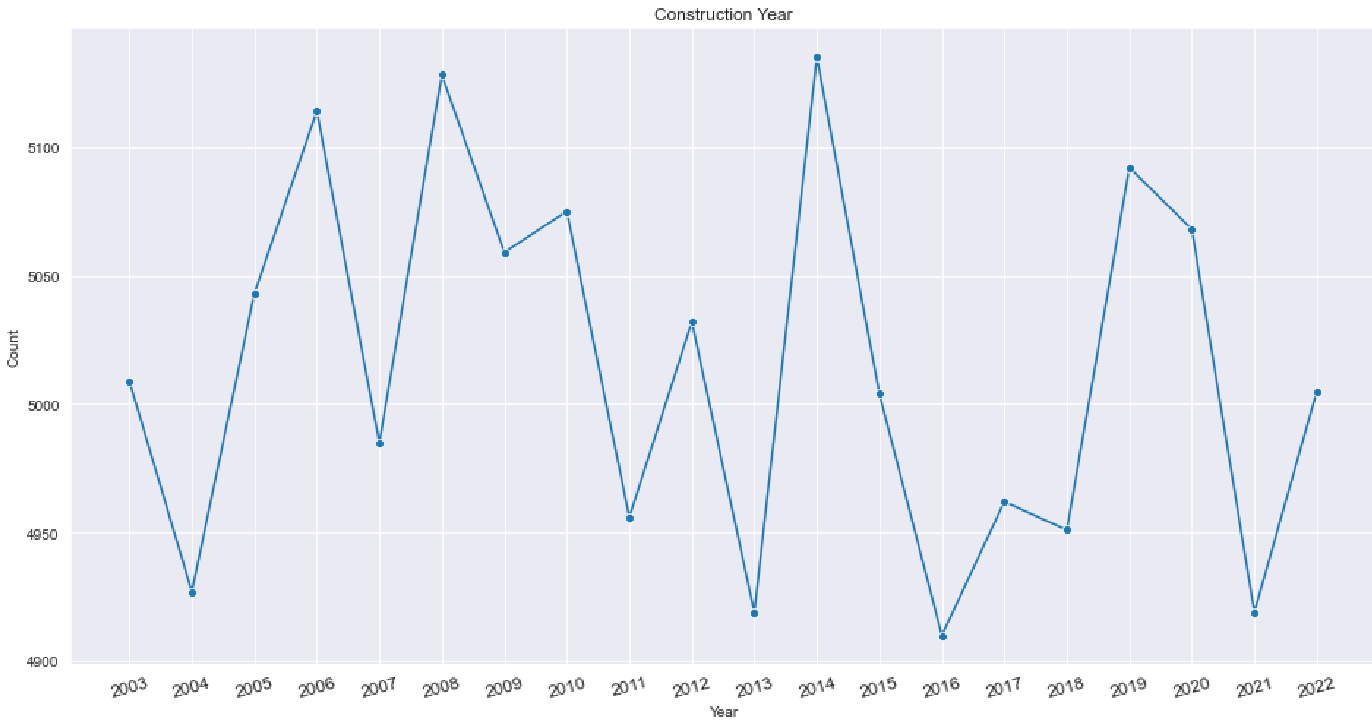
In [38]:
```
#Plot bar chart for Review Rate Number on Room Type
plt.figure(figsize=(16,8))
plt.title("Review Rate Number on Room Type")
ax = sns.countplot(airbnb['room type'],hue=airbnb['review rate number'], palette="colorb
plt.xlabel("Room Type")
plt.ylabel("Count of Review Rate Number")
plt.show()
```



In the room types, the number of 5, 4, 3, and 2 star reviews are distributed in equal numbers. 1 star reviews account for a much lower rate.

In [39]:
```
#Plot Line chart on the Construction Year:
plt.figure(figsize=(16,8))
valcnt=airbnb['Construction year'].value_counts().sort_index()
keys = np.array(valcnt.keys(), dtype = np.int16)
plt.title("Construction Year")
sns.lineplot(data=valcnt, marker='o', palette="colorblind")
plt.xlabel("Year")
```

```
plt.ylabel("Count")
plt.xticks(ticks = keys, fontsize = 12, rotation = 15)
plt.show()
```

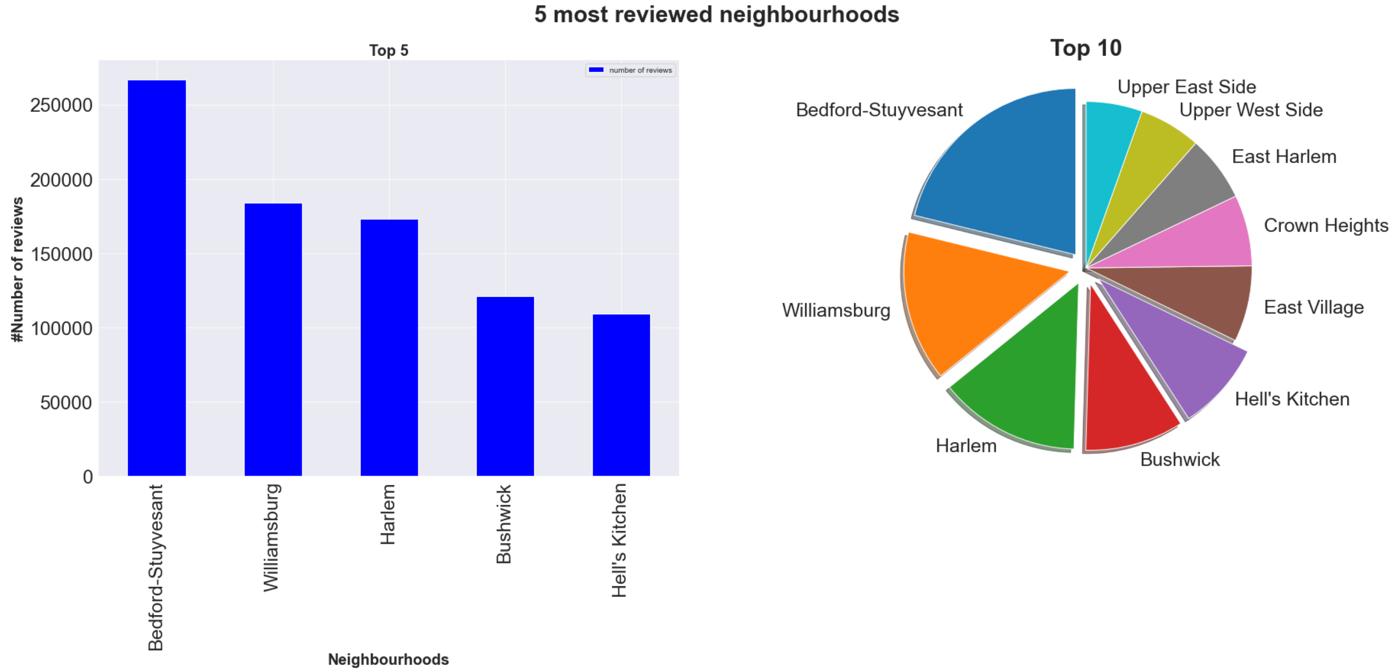


Data Analysis of the questions to be answered

Five most reviewed neighbourhoods

In [40]:
```
temp = airbnb[['neighbourhood','number of reviews']].groupby('neighbourhood',as_index=Fa
temp = temp.sort_values(['number of reviews'],ascending=False)

explode = (0.1,0.1,0.1,0.1,0.1,0,0,0,0,0)
fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('5 most reviewed neighbourhoods',fontweight='bold',fontsize=30)
temp.head(5).plot.bar(x='neighbourhood',color='blue',figsize=(20,15),fontsize=25,ax=ax1)
ax1.set_title('Top 5',fontweight='bold',fontsize=20)
ax1.set_ylabel('#Number of reviews',fontweight='bold',fontsize=20)
ax1.set_xlabel('Neighbourhoods',fontweight='bold',fontsize=20)
temp.head(10).plot(kind='pie',x='neighbourhood',y='number of reviews',figsize=(30,10),st
                   labels=temp['neighbourhood'], legend = False,fontsize=25,explode=expl
ax2.set_title('Top 10',fontweight='bold',fontsize=30)
ax2.set_ylabel('')
ax2.set_xlabel('')
fig.subplots_adjust(hspace=0.5)
```



On the left, bar graph shows top 5 most reviewed neighbourhoods. On the right, pie chart shows the top 10 most popular neighbourhoods in New York City. The maximum number of people love to stay in these neighbourhoods. The reason behind the popularity of neighbourhoods may depend upon the price, number of reviews, review rate number, and many more.

The Average Price of Top 10 Neighbourhoods

In [41]:
```
#Find out the average price in top 10 neighbourhoods
neigbourhoods = airbnb['price'].groupby([airbnb['neighbourhood'],airbnb['neighbourhood g
neigbourhoods.sort_values(ascending=False).head(10)
```

Out[41]:
```
neighbourhood            neighbourhood group
New Dorp                 Staten Island          1045.000000
Chelsea, Staten Island   Staten Island          1042.000000
Fort Wadsworth           Staten Island          1024.000000
Little Neck              Queens                  817.750000
Jamaica Hills            Queens                  812.904762
Arden Heights            Staten Island           804.888889
Shore Acres              Staten Island           797.590909
Midland Beach            Staten Island           796.176471
East Morrisania          Bronx                   786.950000
Mill Basin               Brooklyn                775.142857
Name: price, dtype: float64
```

Third value, 'Chelsea Staten Island' should just be 'Chelsea'. Let's fix that and run the code again.

In [42]:
```
airbnb['neighbourhood'].loc[airbnb['neighbourhood']=='Chelsea, Staten Island']='Chelsea'
neigbourhoods = airbnb['price'].groupby([airbnb['neighbourhood'],airbnb['neighbourhood g
neigbourhoods.sort_values(ascending=False).head(10)
```

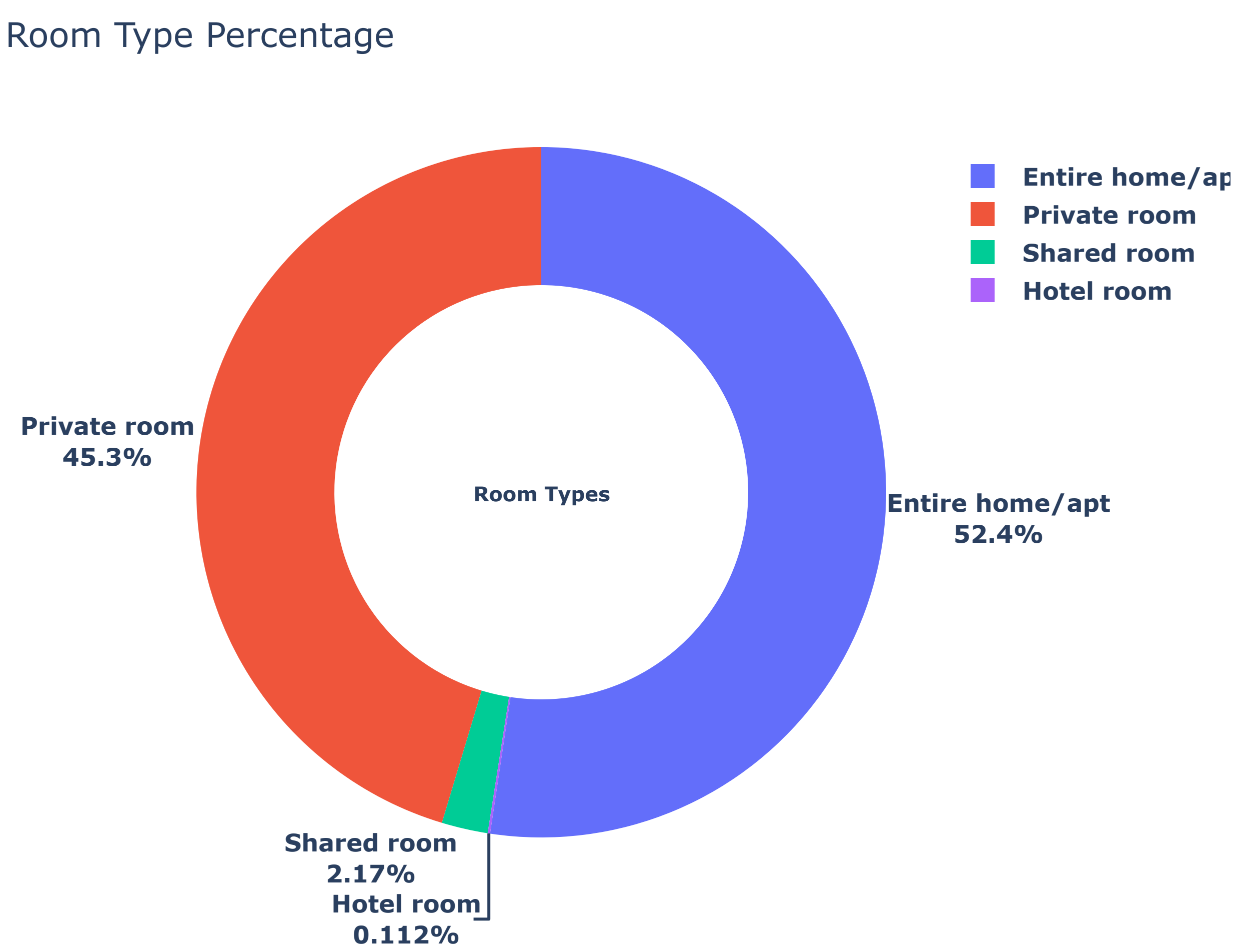
Out[42]:
```
neighbourhood    neighbourhood group
New Dorp         Staten Island          1045.000000
Chelsea          Staten Island          1042.000000
Fort Wadsworth   Staten Island          1024.000000
Little Neck      Queens                  817.750000
Jamaica Hills    Queens                  812.904762
Arden Heights    Staten Island           804.888889
Shore Acres      Staten Island           797.590909
Midland Beach    Staten Island           796.176471
```

```
East Morrisania  Bronx                  786.950000
Mill Basin       Brooklyn               775.142857
Name: price, dtype: float64
```

Here, we got the average price of top ten neighbourhoods in New York City!

The Percent Share of Different Room Types

In [43]:
```python
#Plot pie chart on different Room Types Percentage Distribution:
airbnb['tmp']=1
fig=px.pie(airbnb, names="room type", values='tmp',hole=0.6, title="Room Types Distribut
fig.update_traces(textposition='outside', textinfo='percent+label')
fig.update_layout(title_text="Room Type Percentage",
                  annotations=[dict(text='Room Types', x=0.5, y=0.5, font_size=10, showa
```



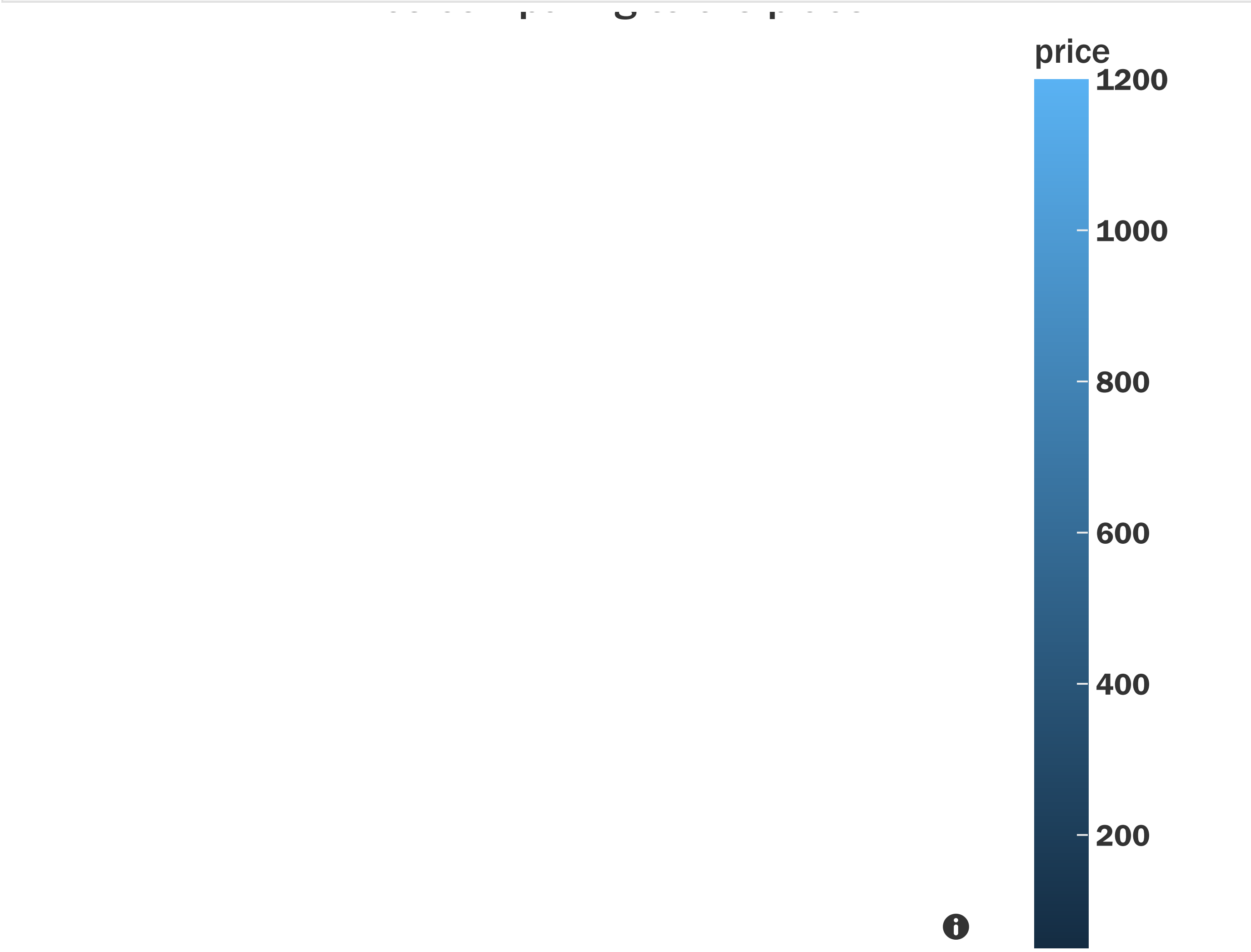
Mostly Entire home/apt and Private room. Very few Shared room and Hotel room types.

The Price variation with location, property type, and Reviews

In [44]:
```python
#Comparing the price and location:

fig = px.scatter_mapbox(airbnb,
          lat="lat",
          lon="long",
          opacity = 0.3,
          hover_name="neighbourhood group",
          hover_data=["neighbourhood group", "price"],
```

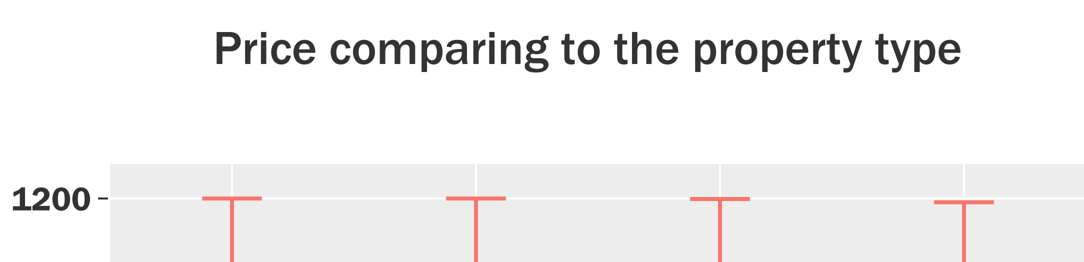
```
            color="price",
            color_discrete_sequence=px.colors.sequential.PuBuGn,
            title = "Price comparing to the place",
            template = "ggplot2",
            zoom=10
            )
fig.update_layout(mapbox_style="open-street-map")
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0},font = dict(size=17,family="Franklin 
fig.show()
```

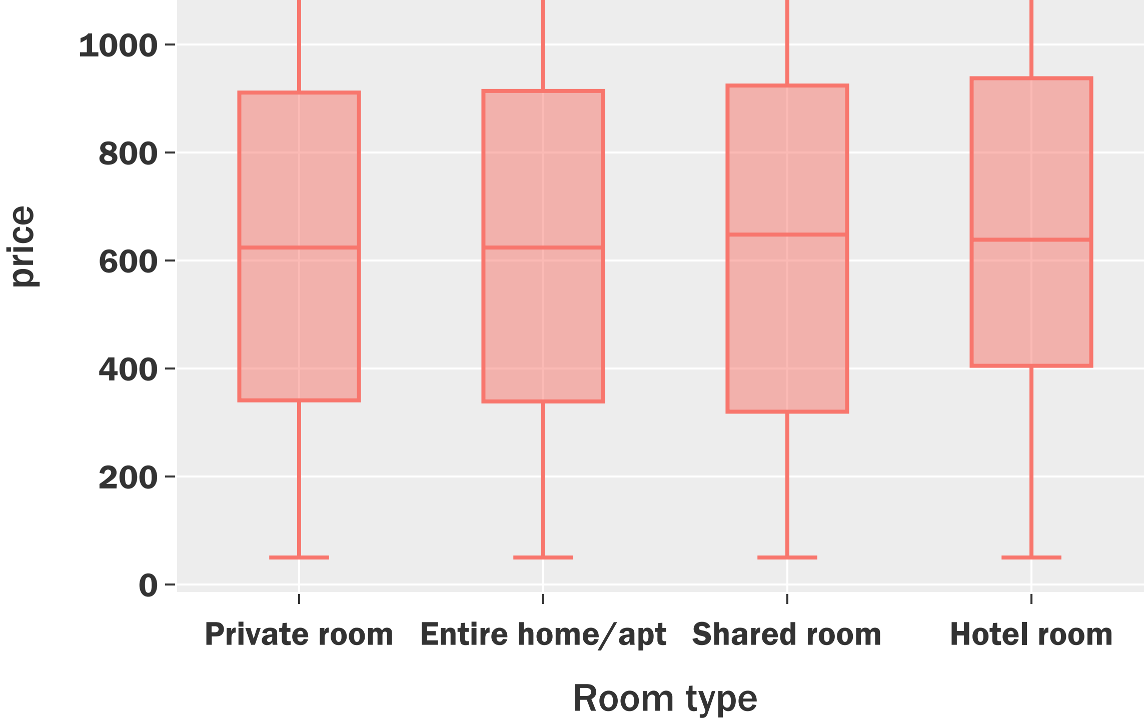


In [45]:

```
#Comparing the price and property type:

fig = px.box(
    x=airbnb["room type"],
    y=airbnb['price'],
    template= 'ggplot2',
    title = 'Price comparing to the property type')
fig.update_layout(
    xaxis_title = "Room type",
    yaxis_title = "price",
    font = dict(size=17, family="Franklin Gothic"))
fig.show()
```





Is the number of reviews more in pricier places or the cheaper places? And are they good or bad reviews in these places?

In [46]:
```
# To find out the relation between price and number of reviews, we will copy the dataset
rpnyc=airbnb.sort_values('price', ascending = False)
rpnyc.head(10)
```

Out[46]:

	host_identity_verified	neighbourhood group	neighbourhood	lat	long	instant_bookable	cancellatior
76937	unconfirmed	Manhattan	Harlem	40.82284	-73.95546	False	
77507	verified	Brooklyn	Greenpoint	40.72253	-73.94350	False	
70765	verified	Manhattan	Harlem	40.80861	-73.94574	True	
50535	verified	Brooklyn	Bedford-Stuyvesant	40.67842	-73.91024	False	m
5207	unconfirmed	Brooklyn	Bushwick	40.70322	-73.92913	True	
90165	unconfirmed	Brooklyn	Greenpoint	40.72253	-73.94350	True	
57509	unconfirmed	Queens	Flushing	40.74582	-73.83153	False	
19773	verified	Manhattan	Harlem	40.82284	-73.95546	True	
24028	verified	Brooklyn	East New York	40.67392	-73.88892	False	m
67377	unconfirmed	Manhattan	East Village	40.72790	-73.98347	False	m

In [47]:
```python
# Now we will create two temp data frame where store one dataset with price > 200USD and
data200plus =airbnb[airbnb['price'] >200]
print(data200plus)
data200cheap =airbnb[airbnb['price'] <=200]
print(data200cheap)
```

```
       host_identity_verified neighbourhood group        neighbourhood  \
0                 unconfirmed            Brooklyn           Kensington
2                 unconfirmed           Manhattan               Harlem
3                 unconfirmed            Brooklyn         Clinton Hill
4                    verified           Manhattan          East Harlem
5                    verified           Manhattan          Murray Hill
...                       ...                 ...                  ...
102594               verified            Brooklyn         Williamsburg
102595            unconfirmed           Manhattan  Morningside Heights
102596            unconfirmed            Brooklyn           Park Slope
102597            unconfirmed              Queens     Long Island City
102598            unconfirmed           Manhattan       Upper West Side

             lat      long instant_bookable cancellation_policy  \
0       40.64749 -73.97237            False              strict
2       40.80902 -73.94190             True            flexible
3       40.68514 -73.95976             True            moderate
4       40.79851 -73.94399            False            moderate
5       40.74767 -73.97500             True            flexible
...          ...       ...              ...                 ...
102594  40.70862 -73.94651            False            flexible
102595  40.80460 -73.96545             True            moderate
102596  40.67505 -73.98045             True            moderate
102597  40.74989 -73.93777             True              strict
102598  40.76807 -73.98342            False            flexible

              room type  Construction year  price  service fee  \
0          Private room               2020    966          193
2          Private room               2005    620          124
3       Entire home/apt               2005    368           74
4       Entire home/apt               2009    204           41
5       Entire home/apt               2013    577          115
...                 ...                ...    ...          ...
102594     Private room               2003    844          169
102595     Private room               2016    837          167
102596     Private room               2009    988          198
102597  Entire home/apt               2015    546          109
102598  Entire home/apt               2010   1032          206

        minimum nights  number of reviews  reviews per month  \
0                   10                9.0               0.21
2                    3                0.0               0.00
3                   30              270.0               4.64
4                   10                9.0               0.10
5                    3               74.0               0.59
...                ...                ...                ...
102594               1                0.0               0.00
102595               1                1.0               0.02
102596               3                0.0               0.00
102597               2                5.0               0.10
102598               1                0.0               0.00

        review rate number  calculated host listings count  availability 365  \
0                        4                             6.0               286
2                        5                             1.0               352
3                        4                             1.0               322
4                        3                             1.0               289
5                        3                             1.0               365
...                    ...                             ...               ...
```

```
102594                  3                            1.0               227
102595                  2                            2.0               365
102596                  5                            1.0               342
102597                  3                            1.0               365
102598                  3                            1.0                69

        tmp
0         1
2         1
3         1
4         1
5         1
...     ...
102594    1
102595    1
102596    1
102597    1
102598    1

[87275 rows x 18 columns]
       host_identity_verified neighbourhood group       neighbourhood  \
1                    verified           Manhattan             Midtown
6                 unconfirmed            Brooklyn  Bedford-Stuyvesant
14                   verified           Manhattan     Upper West Side
111               unconfirmed           Manhattan      Hell's Kitchen
137               unconfirmed            Brooklyn           Flatlands
...                       ...                 ...                 ...
102536            unconfirmed            Brooklyn            Bushwick
102546               verified           Manhattan Morningside Heights
102563            unconfirmed           Manhattan     Lower East Side
102578               verified           Manhattan         East Harlem
102581            unconfirmed            Brooklyn  Bedford-Stuyvesant

             lat      long instant_bookable cancellation_policy  \
1       40.75362 -73.98377            False            moderate
6       40.68688 -73.95596            False            moderate
14      40.79826 -73.96113            False            flexible
111     40.75527 -73.99291             True              strict
137     40.63188 -73.93248             True            moderate
...          ...       ...              ...                 ...
102536  40.69034 -73.91666             True              strict
102546  40.80481 -73.96375            False              strict
102563  40.72115 -73.98308            False              strict
102578  40.79674 -73.94449             True            moderate
102581  40.68742 -73.92825            False            flexible

              room type  Construction year  price  service fee  \
1       Entire home/apt               2007    142           28
6          Private room               2015     71           14
14         Private room               2019    149           30
111        Private room               2014     66           13
137        Private room               2020     96           19
...                 ...                ...    ...          ...
102536     Private room               2021     70           14
102546     Private room               2008    129           26
102563  Entire home/apt               2008    162           32
102578     Private room               2016    177           35
102581     Private room               2022     77           15

        minimum nights  number of reviews  reviews per month  \
1                   30               45.0               0.38
6                   45               49.0               0.40
14                   2              113.0               0.91
111                  2              334.0               3.00
137                  2                2.0               0.02
...                ...                ...                ...
```

```
102536             1            1.0            0.02
102546            21            4.0            0.09
102563             5            1.0            0.02
102578             5            4.0            0.08
102581             2           34.0            0.70

        review rate number  calculated host listings count  availability 365  \
1                        4                             2.0               228
6                        5                             1.0               224
14                       3                             1.0                68
111                      1                             2.0                34
137                      4                             1.0               334
...                    ...                             ...               ...
102536                   1                             1.0                73
102546                   4                             1.0               313
102563                   2                             1.0               133
102578                   1                             1.0               323
102581                   5                             3.0               299

        tmp
1         1
6         1
14        1
111       1
137       1
...     ...
102536    1
102546    1
102563    1
102578    1
102581    1

[13018 rows x 18 columns]
```

In [48]:
```
data200plus['number of reviews'].mean()
```

Out[48]:
```
27.405958178172444
```

In [49]:
```
data200cheap['number of reviews'].mean()
```

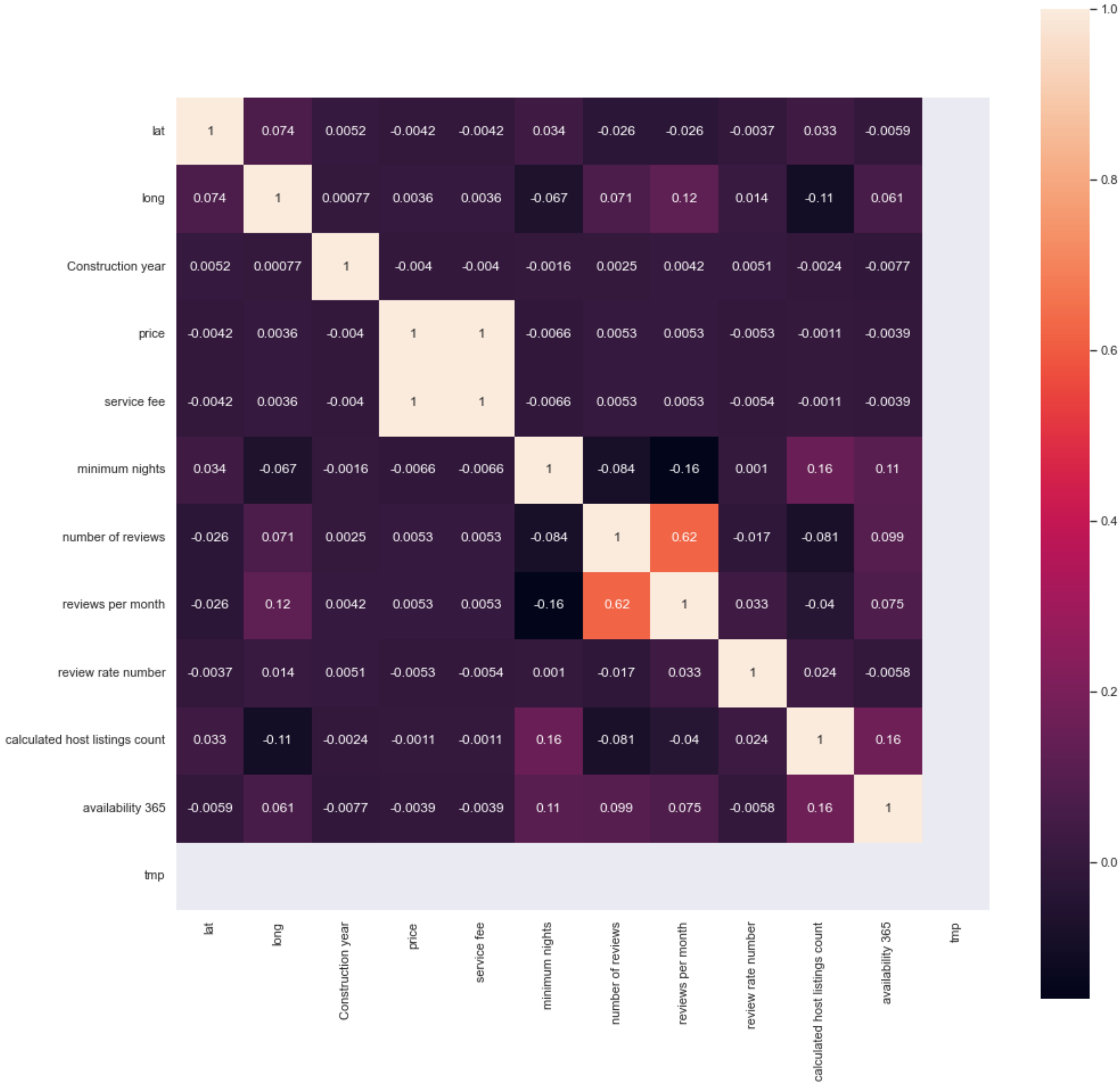
Out[49]:
```
26.490705177446614
```

As there is very little difference between average number of reviews in the cheaper places and pricier places, with the given data, we can be derived that there is no relation between number of reviews and price.

Now below we will check the correlation between all columns

In [50]:
```
# heatmap of correlation between all columns
sns.set(rc={"figure.figsize":(16, 16)})
sns.heatmap(airbnb.corr(), annot=True, square=True)
```

Out[50]:
```
<AxesSubplot:>
```



There is perfect correlation of "service fee" and "price" might be because the fee is computed by airbnb depending on the price.