In [192…

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
```

In [111…

```python
!pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.8.2.2-cp39-cp39-macosx_10_9_x86_64.whl (160 kB)
     |████████████████████████████████| 160 kB 5.2 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.6.1 in /Users/michaellee/opt/anaconda3/lib/python
3.9/site-packages (from wordcloud) (1.21.3)
Requirement already satisfied: pillow in /Users/michaellee/opt/anaconda3/lib/python3.9/sit
e-packages (from wordcloud) (8.4.0)
Requirement already satisfied: matplotlib in /Users/michaellee/opt/anaconda3/lib/python3.
9/site-packages (from wordcloud) (3.4.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/michaellee/opt/anaconda3/lib/py
thon3.9/site-packages (from matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: python-dateutil>=2.7 in /Users/michaellee/opt/anaconda3/li
b/python3.9/site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/michaellee/opt/anaconda3/lib/pyt
hon3.9/site-packages (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: cycler>=0.10 in /Users/michaellee/opt/anaconda3/lib/python
3.9/site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: six in /Users/michaellee/opt/anaconda3/lib/python3.9/site-p
ackages (from cycler>=0.10->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.2.2
```

In [112…

```python
from wordcloud import WordCloud
```

In [2]:

```python
## Q0 Data Exploration
airbnb = pd.read_csv('./airbnb data/AB_NYC_2019.csv')
airbnb
```

Out[2]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851
...	...	...	...	...	...	...	...

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404

48895 rows × 16 columns

In [3]:

```
airbnb.columns
```

Out[3]:

```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365'],
      dtype='object')
```

In [4]:

```
# can select which columns to remove if unnecessary
# airbnb_subset = airbnb[['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
#         'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
#         'minimum_nights', 'number_of_reviews', 'last_review',
#         'reviews_per_month', 'calculated_host_listings_count',
#         'availability_365']]
```

In [5]:

```
# List the counts of missing values for each column
airbnb.isna().sum()
```

Out[5]:

```
id                                    0
name                                 16
host_id                               0
host_name                            21
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                       10052
reviews_per_month                 10052
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

In [6]:

```python
# List all the rows with name = NaN
airbnb[airbnb['name'].isna()]
```

Out[6]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude
2854	1615764	NaN	6676776	Peter	Manhattan	Battery Park City	40.71239	-74.01620
3703	2232600	NaN	11395220	Anna	Manhattan	East Village	40.73215	-73.98821
5775	4209595	NaN	20700823	Jesse	Manhattan	Greenwich Village	40.73473	-73.99244
5975	4370230	NaN	22686810	Michaël	Manhattan	Nolita	40.72046	-73.99550
6269	4581788	NaN	21600904	Lucie	Brooklyn	Williamsburg	40.71370	-73.94378
6567	4756856	NaN	1832442	Carolina	Brooklyn	Bushwick	40.70046	-73.92825
6605	4774658	NaN	24625694	Josh	Manhattan	Washington Heights	40.85198	-73.93108
8841	6782407	NaN	31147528	Huei-Yin	Brooklyn	Williamsburg	40.71354	-73.93882
11963	9325951	NaN	33377685	Jonathan	Manhattan	Hell's Kitchen	40.76436	-73.98573
12824	9787590	NaN	50448556	Miguel	Manhattan	Harlem	40.80316	-73.95189
13059	9885866	NaN	37306329	Juliette	Manhattan	Chinatown	40.71632	-73.99328
13401	10052289	NaN	49522403	Vanessa	Brooklyn	Brownsville	40.66409	-73.92314
15819	12797684	NaN	69715276	Yan	Manhattan	Upper West Side	40.79843	-73.96404
16071	12988898	NaN	71552588	Andrea	Bronx	Fordham	40.86032	-73.88493
18047	14135050	NaN	85288337	Jeff	Brooklyn	Bedford-Stuyvesant	40.69421	-73.93234
28889	22275821	NaN	49662398	Kathleen	Brooklyn	Bushwick	40.69546	-73.92741

In [7]:

```python
# Summary of numerical values while dropping useless numerical values(id, host_id, etc.)
num_airbnb = airbnb.drop(['host_id', 'latitude', 'longitude'], axis=1)
num_airbnb.describe()
```

Out[7]:

	id	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_hc
count	4.889500e+04	48895.000000	48895.000000	48895.000000	38843.000000	
mean	1.901714e+07	152.720687	7.029962	23.274466	1.373221	
std	1.098311e+07	240.154170	20.510550	44.550582	1.680442	
min	2.539000e+03	0.000000	1.000000	0.000000	0.010000	
25%	9.471945e+06	69.000000	1.000000	1.000000	0.190000	

	id	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_hc
50%	1.967728e+07	106.000000	3.000000	5.000000	0.720000	
75%	2.915218e+07	175.000000	5.000000	24.000000	2.020000	
max	3.648724e+07	10000.000000	1250.000000	629.000000	58.500000	

In [8]:

```
# Checking which row had the max price
max_price_row = airbnb.loc[airbnb['price'].idxmax()]
max_price_row
```

Out[8]:

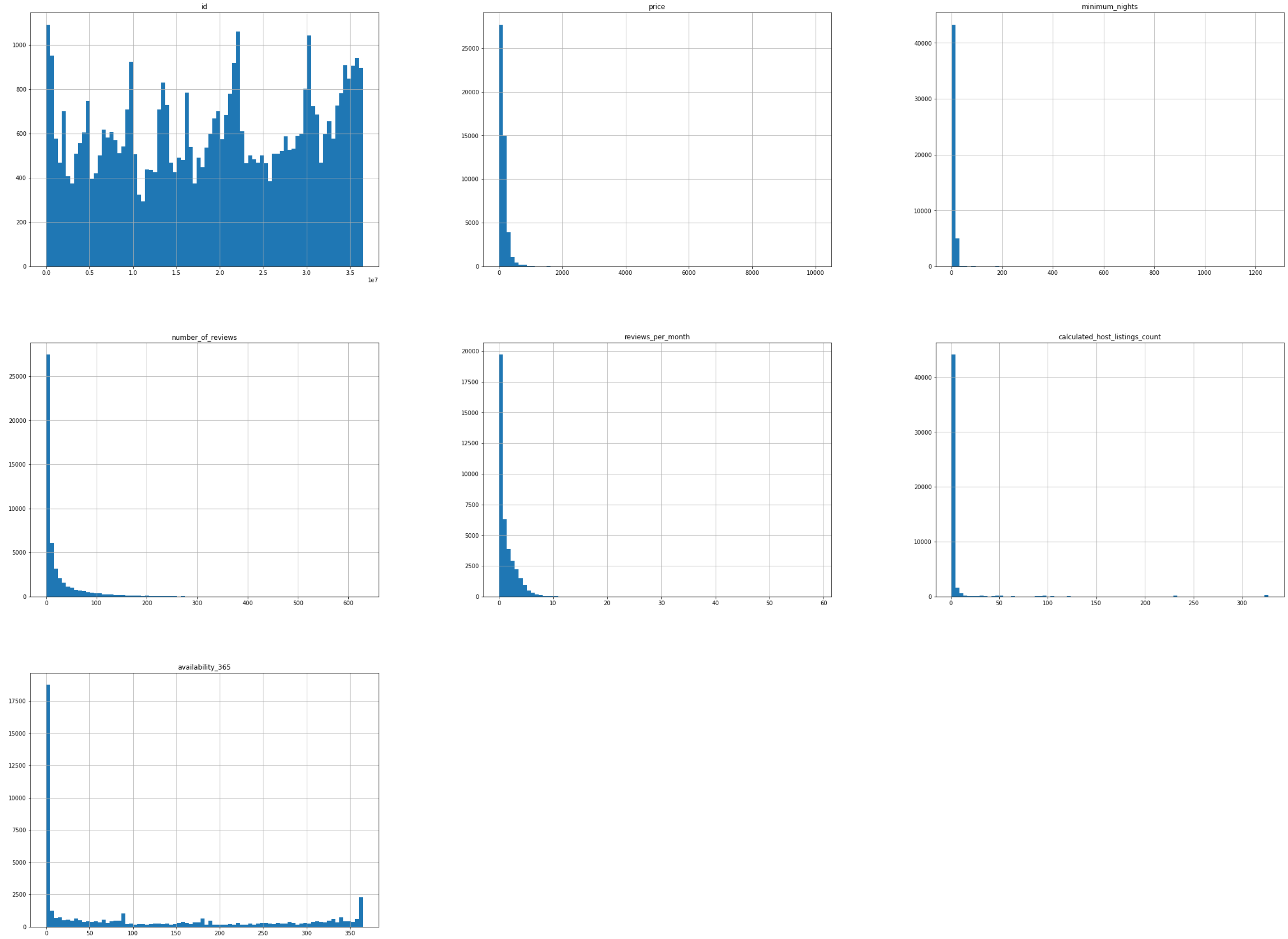
```
id                                                          7003697
name                              Furnished room in Astoria apartment
host_id                                                    20582832
host_name                                                  Kathrine
neighbourhood_group                                          Queens
neighbourhood                                               Astoria
latitude                                                    40.7681
longitude                                                 -73.91651
room_type                                              Private room
price                                                         10000
minimum_nights                                                  100
number_of_reviews                                                 2
last_review                                              2016-02-13
reviews_per_month                                              0.04
calculated_host_listings_count                                    1
availability_365                                                  0
Name: 9151, dtype: object
```

In [9]:

```
# Making a histogram of num_airbnb with the bin size and figure size
num_airbnb.hist(bins = 80, figsize=(40,30))
```

Out[9]:

```
array([[<AxesSubplot:title={'center':'id'}>,
        <AxesSubplot:title={'center':'price'}>,
        <AxesSubplot:title={'center':'minimum_nights'}>],
       [<AxesSubplot:title={'center':'number_of_reviews'}>,
        <AxesSubplot:title={'center':'reviews_per_month'}>,
        <AxesSubplot:title={'center':'calculated_host_listings_count'}>],
       [<AxesSubplot:title={'center':'availability_365'}>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```



In [10]:

```python
# Filtering out the rows of price where the price is greather than 2000
high_price = num_airbnb[num_airbnb['price'] > 2000]
high_price
```

Out[10]:

	id	name	host_name	neighbourhood_group	neighbourhood	room_type	price
946	363673	Beautiful 3 bedroom in Manhattan	Tracey	Manhattan	Upper West Side	Private room	3000
1862	826690	Sunny, Family-Friendly 2 Bedroom	Lucy	Brooklyn	Prospect Heights	Entire home/apt	4000
2018	893413	Architecturally Stunning Former Synagogue!	Martin	Manhattan	East Village	Entire home/apt	2500
2698	1448703	Beautiful 1 Bedroom in Nolita/Soho	Jessica	Manhattan	Nolita	Entire home/apt	5000
3537	2110145	UWS 1BR w/backyard + block from CP	Jay And Liz	Manhattan	Upper West Side	Entire home/apt	6000
...	...	...	...	...	...	...	...
45867	34981637	bay ridge & sunset park furnished apartment	Nony	Brooklyn	Bay Ridge	Entire home/apt	4200
46533	35297214	Amazing Chelsea 4BR Loft!	Viberlyn	Manhattan	Chelsea	Entire home/apt	2995

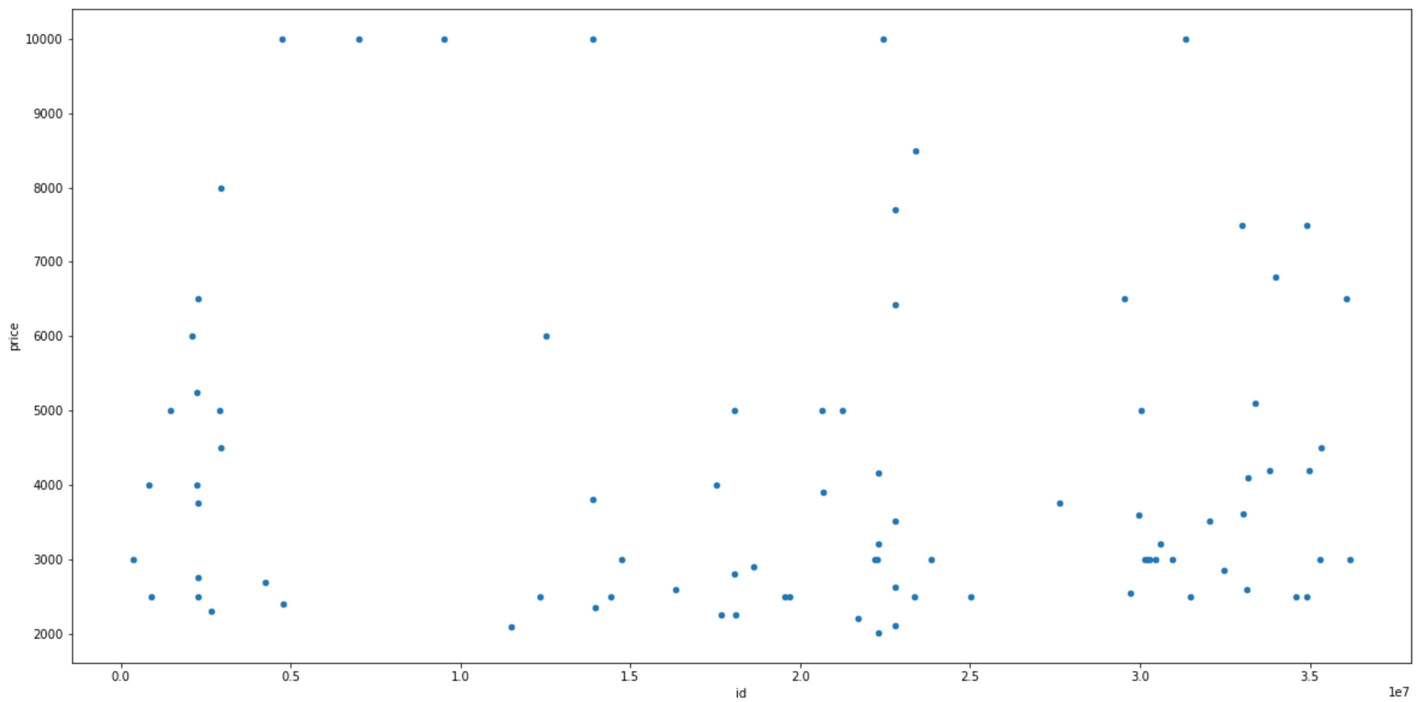
	id	name	host_name	neighbourhood_group	neighbourhood	room_type	price
46614	35345358	Northside Williamsburg Stunner	Alex	Brooklyn	Williamsburg	Entire home/apt	4500
48043	36056808	Luxury TriBeCa Apartment at an amazing price	Jenny	Manhattan	Tribeca	Entire home/apt	6500
48304	36189195	Next to Times Square/Javits/MSG! Amazing 1BR!	Rogelio	Manhattan	Hell's Kitchen	Entire home/apt	2999

86 rows × 13 columns

In [11]:

```
# Scatter plot of airbnb with high prices
high_price[['id', 'price']].plot(kind="scatter", x="id", y="price", figsize=(20,10))
```

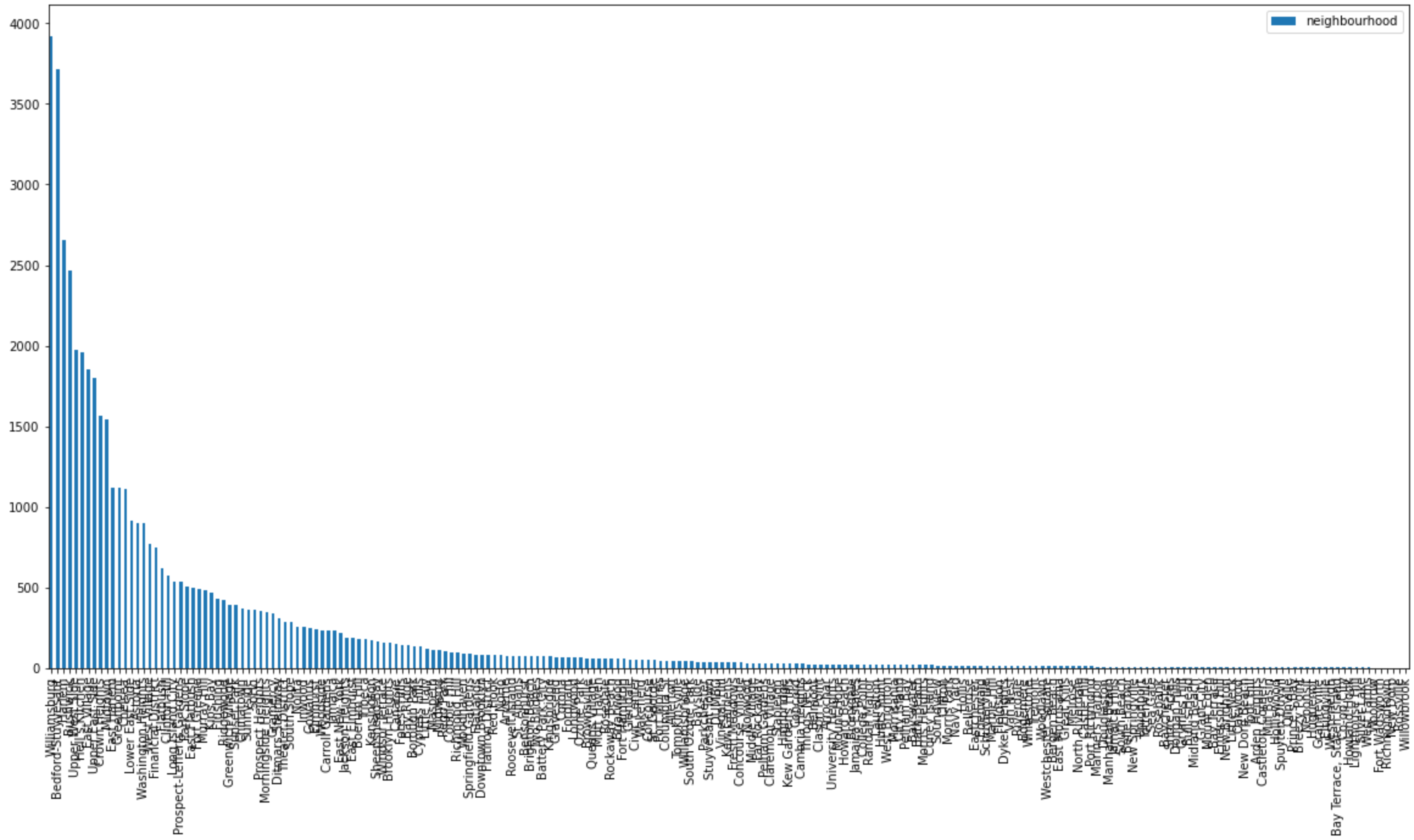
Out[11]: <AxesSubplot:xlabel='id', ylabel='price'>



In [12]:

```
# Bar graph of count of neighbourhood names
pd.DataFrame(airbnb['neighbourhood'].value_counts()).plot(kind='bar', figsize=(20, 10))
```

Out[12]: <AxesSubplot:>



In [13]:

```python
# Count of cases where number of reviews = 0
(airbnb['number_of_reviews']==0).sum()
```

Out[13]:

```
10052
```

In [14]:

```python
# Count of each value in each column
for column_name in airbnb.columns:
    print("Value counts for column: ", column_name)
    print(airbnb[column_name].value_counts())
    print('\n')
```

```
Value counts for column:  id
2539        1
25583366    1
25551687    1
25552076    1
25554120    1
           ..
13121809    1
13122135    1
13122318    1
13122932    1
36487245    1
Name: id, Length: 48895, dtype: int64


Value counts for column:  name
Hillside Hotel                                    18
Home away from home                               17
New york Multi-unit building                      16
Brooklyn Apartment                                12
Loft Suite @ The Box House Hotel                  11
                                                  ..
Large 1BR Apt. in Williamsburg                     1
Feel at Home                                       1
Spacious Modern Alcove Studio in a Luxury Building 1
```

```
Artist's Room in Large Apartment                        1
Trendy duplex in the very heart of Hell's Kitchen      1
Name: name, Length: 47905, dtype: int64


Value counts for column:  host_id
219517861    327
107434423    232
30283594     121
137358866    103
16098958      96
            ...
23727216       1
89211125       1
19928013       1
1017772        1
68119814       1
Name: host_id, Length: 37457, dtype: int64


Value counts for column:  host_name
Michael               417
David                 403
Sonder (NYC)          327
John                  294
Alex                  279
                     ...
Rhonycs                 1
Brandy-Courtney         1
Shanthony               1
Aurore And Jamila       1
Ilgar & Aysel           1
Name: host_name, Length: 11452, dtype: int64


Value counts for column:  neighbourhood_group
Manhattan        21661
Brooklyn         20104
Queens            5666
Bronx             1091
Staten Island      373
Name: neighbourhood_group, dtype: int64


Value counts for column:  neighbourhood
Williamsburg          3920
Bedford-Stuyvesant    3714
Harlem                2658
Bushwick              2465
Upper West Side       1971
                      ...
Fort Wadsworth           1
Richmondtown             1
New Dorp                 1
Rossville                1
Willowbrook              1
Name: neighbourhood, Length: 221, dtype: int64


Value counts for column:  latitude
40.71813    18
40.68444    13
40.69414    13
40.68634    13
40.76125    12
            ..
```

```
40.78084     1
40.66767     1
40.77473     1
40.79343     1
40.81475     1
Name: latitude, Length: 19048, dtype: int64


Value counts for column:  longitude
-73.95677    18
-73.95427    18
-73.95405    17
-73.95060    16
-73.94791    16
             ..
-73.85155     1
-73.83167     1
-73.85058     1
-73.79232     1
-73.80844     1
Name: longitude, Length: 14718, dtype: int64


Value counts for column:  room_type
Entire home/apt    25409
Private room       22326
Shared room         1160
Name: room_type, dtype: int64


Value counts for column:  price
100    2051
150    2047
50     1534
60     1458
200    1401
       ...
780       1
386       1
888       1
483       1
338       1
Name: price, Length: 674, dtype: int64


Value counts for column:  minimum_nights
1      12720
2      11696
3       7999
30      3760
4       3303
       ...
186        1
366        1
68         1
87         1
36         1
Name: minimum_nights, Length: 109, dtype: int64


Value counts for column:  number_of_reviews
0      10052
1       5244
2       3465
3       2520
4       1994
```

```
       ...
313        1
540        1
480        1
326        1
341        1
Name: number_of_reviews, Length: 394, dtype: int64


Value counts for column:  last_review
2019-06-23    1413
2019-07-01    1359
2019-06-30    1341
2019-06-24     875
2019-07-07     718
              ...
2012-12-25       1
2013-10-01       1
2014-05-29       1
2014-04-19       1
2018-03-29       1
Name: last_review, Length: 1764, dtype: int64


Value counts for column:  reviews_per_month
0.02     919
0.05     893
1.00     893
0.03     804
0.16     667
        ...
9.53       1
9.74       1
6.06       1
8.25       1
10.54      1
Name: reviews_per_month, Length: 937, dtype: int64


Value counts for column:  calculated_host_listings_count
1      32303
2       6658
3       2853
4       1440
5        845
6        570
8        416
7        399
327      327
9        234
232      232
10       210
96       192
12       180
13       130
121      121
11       110
52       104
103      103
33        99
49        98
91        91
87        87
15        75
14        70
23        69
```

```
34        68
17        68
65        65
31        62
28        56
18        54
25        50
50        50
47        47
43        43
20        40
39        39
37        37
32        32
30        30
29        29
27        27
26        26
21        21
19        19
16        16
Name: calculated_host_listings_count, dtype: int64


Value counts for column:  availability_365
0      17533
365     1295
364      491
1        408
89       361
       ...
195       26
183       24
196       24
181       23
202       20
Name: availability_365, Length: 366, dtype: int64
```

In [15]:

```python
# Q1 Data Cleaning
# Check for duplicates
airbnb.duplicated().sum()
```

Out[15]:

```
0
```

In [16]:

```python
# Check for the missing values
airbnb.isna().sum()
```

Out[16]:

```
id                                    0
name                                 16
host_id                               0
host_name                            21
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                       10052
reviews_per_month                 10052
```

```
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

In [17]:

```
# Replace all missing values
airbnb['name'].fillna(airbnb['id'], inplace=True)
airbnb['host_name'].fillna(airbnb['host_id'], inplace=True)
airbnb['last_review'].fillna("Not Applicable", inplace=True)
airbnb['reviews_per_month'].fillna(0, inplace=True)
```

In [18]:

```
# Q2 Price vs Neighbourhood
# Filter airbnb so that it has only rows where neighbourhood is listed at least 6 times
airbnb['neighbourhood'].value_counts()
neighbourhood_airbnb = airbnb.groupby('neighbourhood').filter(lambda x: len(x) > 5)
neighbourhood_airbnb
```

Out[18]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851
...	...	...	...	...	...	...	...
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404

48803 rows × 16 columns

In [29]:
```
# Part A
# The top 5 neighbourhoods with the highest prices
# Calculate the means of each neighbourhoods
neighbourhood_prices = neighbourhood_airbnb.groupby('neighbourhood')['price'].mean().reset
neighbourhood_prices
```

Out[29]:

	neighbourhood	price
0	Allerton	87.595238
1	Arrochar	115.000000
2	Arverne	171.779221
3	Astoria	117.187778
4	Bath Beach	81.764706
...	...	...
185	Williamsburg	143.802806
186	Windsor Terrace	138.993631
187	Woodhaven	67.170455
188	Woodlawn	60.090909
189	Woodside	85.097872

190 rows × 2 columns

In [36]:
```
# Sort neighbourhood_prices
neighbourhood_prices_sorted = neighbourhood_prices.sort_values(by='price', ascending=False
neighbourhood_prices_sorted
```

Out[36]:

	neighbourhood	price
170	Tribeca	490.638418
150	Sea Gate	487.857143
144	Riverdale	442.090909
5	Battery Park City	367.557143
68	Flatiron District	341.925000
...	...	...
21	Bronxdale	57.105263
154	Soundview	53.466667
169	Tremont	51.545455
90	Hunts Point	50.500000
24	Bull's Head	47.333333

190 rows × 2 columns

In [42]:
```
# Top 5 neighbourhoods with highest prices
neighbourhood_prices_sorted.head(5)
```

Out[42]:

	neighbourhood	price

	neighbourhood	price
170	Tribeca	490.638418
150	Sea Gate	487.857143
144	Riverdale	442.090909
5	Battery Park City	367.557143
68	Flatiron District	341.925000

In [45]:

```
# Bottom 5 neighbourhoods with the lowest prices
neighbourhood_prices_sorted.tail(5)
```

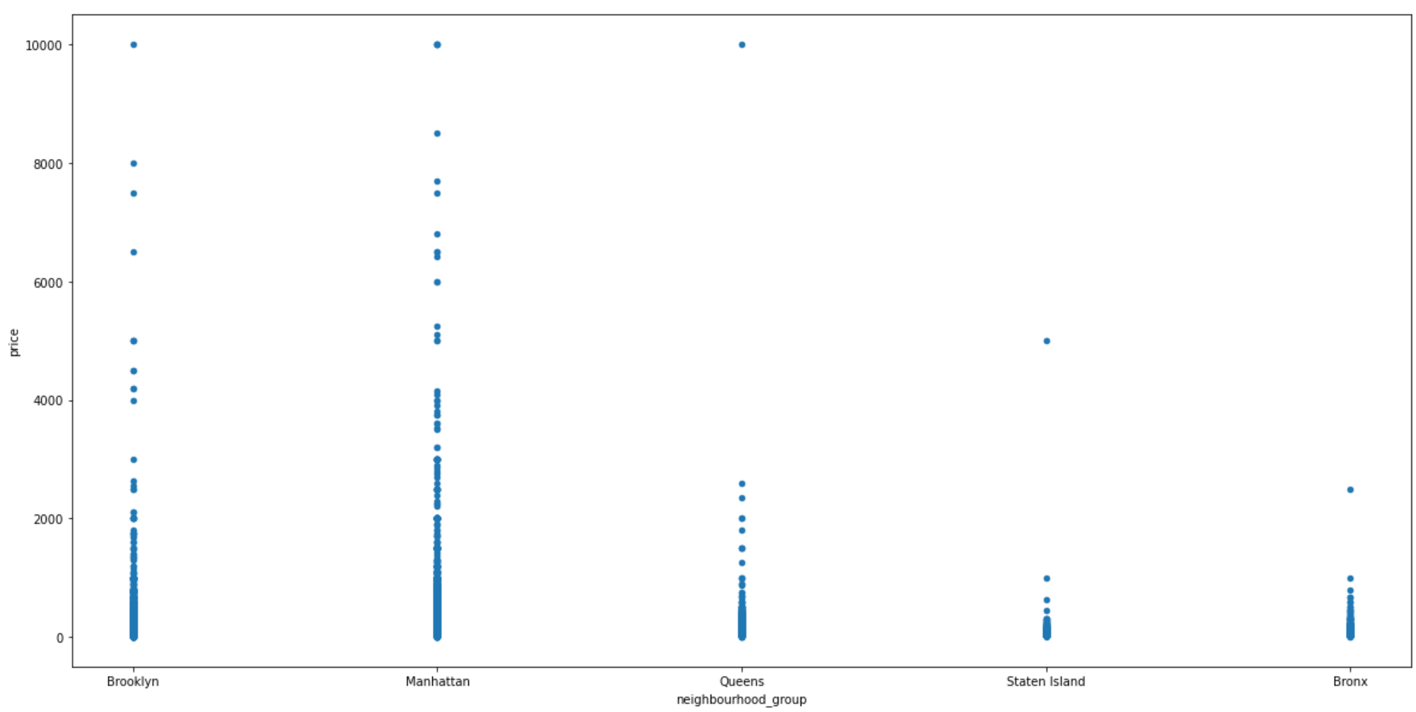
Out[45]:

	neighbourhood	price
21	Bronxdale	57.105263
154	Soundview	53.466667
169	Tremont	51.545455
90	Hunts Point	50.500000
24	Bull's Head	47.333333

In [46]:

```
# Part B
# Scatter Plot of price vs neighbourhood group
neighbourhood_airbnb[['neighbourhood_group', 'price']].plot(kind="scatter", x="neighbourho
```

Out[46]: <AxesSubplot:xlabel='neighbourhood_group', ylabel='price'>



In [51]:

```
# Analysis
neighbourhood_airbnb['neighbourhood_group'].value_counts()
# Generally, the more populated the neighbourhood group was, the more expensive it was.
```

Out[51]:

```
Manhattan        21661
Brooklyn         20100
Queens            5651
```

```
Bronx              1079
Staten Island       312
Name: neighbourhood_group, dtype: int64
```

In [118…

```
# Q3
# I wanted to look at the correlation between the attributes  price, minimum_nights,
# calculated_host_listings_count, number_of_reviews, and  avability_365.
# Select features for Pearson correlation
pearson_airbnb = airbnb[['price', 'availability_365', 'minimum_nights', 'calculated_host_l
pearson_airbnb
```

Out[118…

	price	availability_365	minimum_nights	calculated_host_listings_count	number_of_reviews
0	149	365	1	6	9
1	225	355	1	2	45
2	150	365	3	1	0
3	89	194	1	1	270
4	80	0	10	1	9
...	...	...	...	...	...
48890	70	9	2	2	0
48891	40	36	4	2	0
48892	115	27	10	1	0
48893	55	2	1	6	0
48894	90	23	7	1	0

48895 rows × 5 columns

In [119…

```
# Pairwise pearson correlations
correlations = pearson_airbnb.corr()
correlations
```

Out[119…

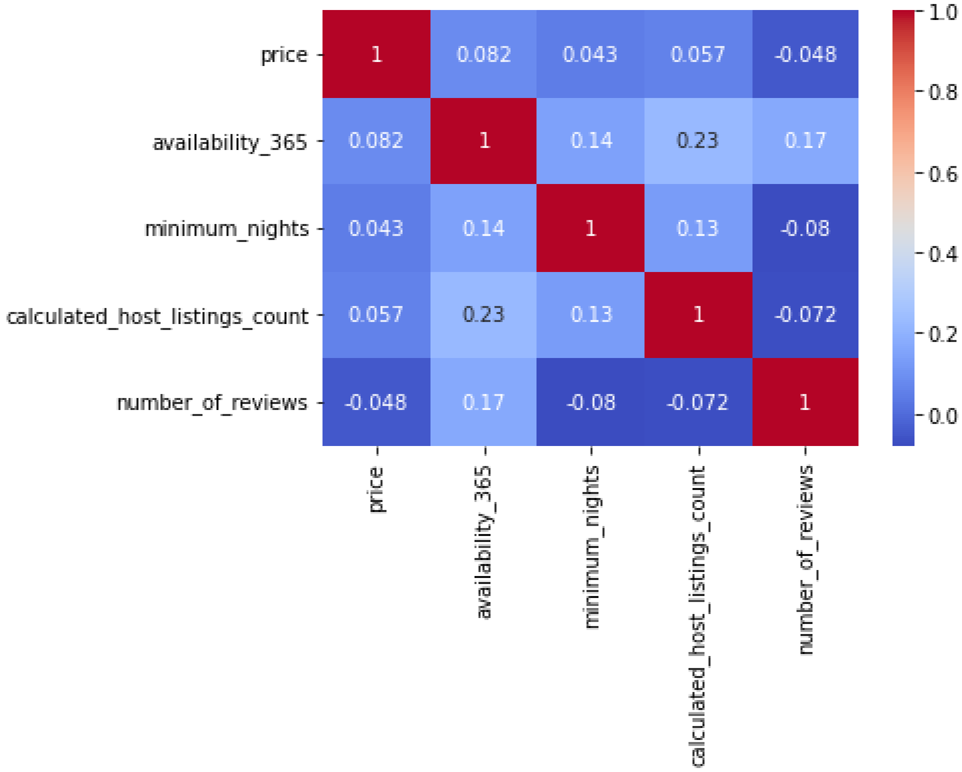
	price	availability_365	minimum_nights	calculated_host_listings_count
price	1.000000	0.081829	0.042799	0.057472
availability_365	0.081829	1.000000	0.144303	0.225701
minimum_nights	0.042799	0.144303	1.000000	0.127960
calculated_host_listings_count	0.057472	0.225701	0.127960	1.000000
number_of_reviews	-0.047954	0.172028	-0.080116	-0.072376

In [178…

```
# Heatmap of correlations
sns.heatmap(correlations, annot=True, cmap='coolwarm')
```

Out[178…

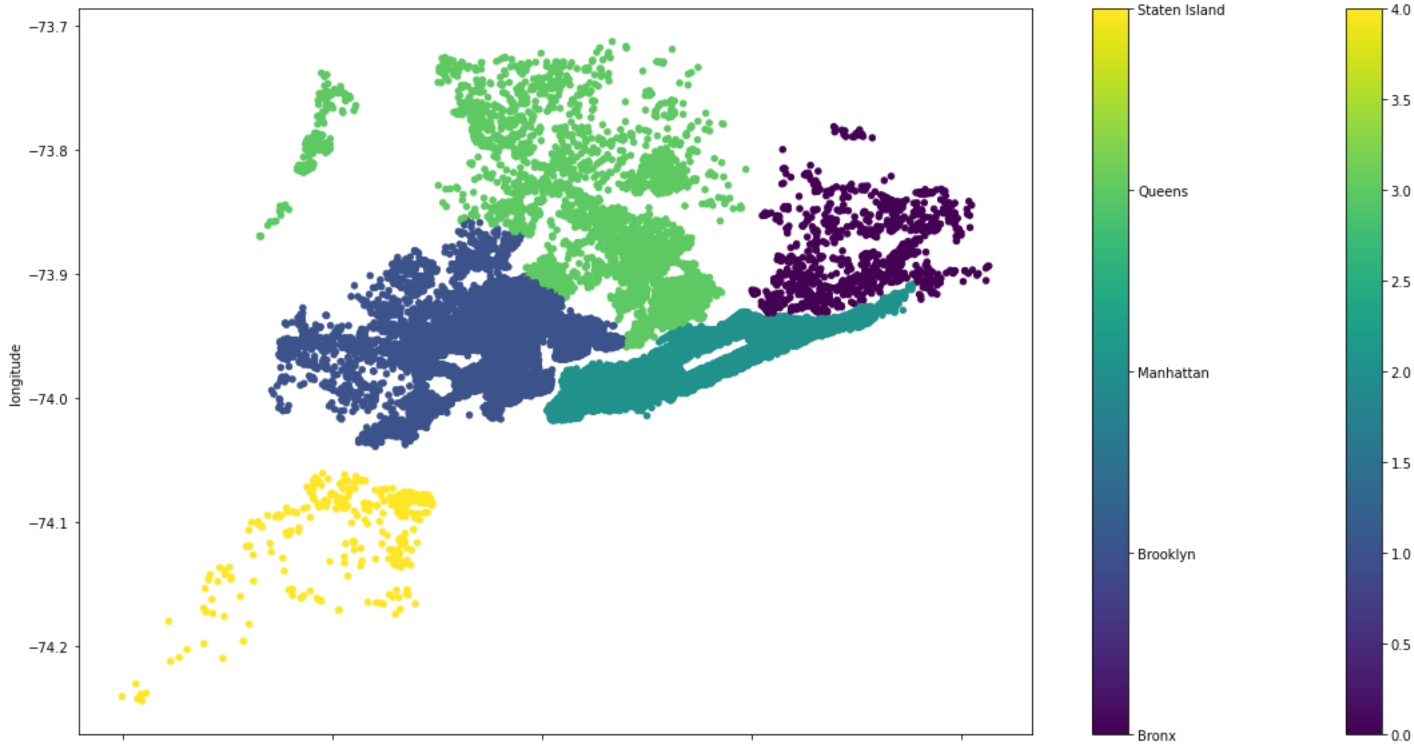
```
<AxesSubplot:>
```



In [66]:
```
# Most positive correlation: calculated_host_listings count and availability_365 with a va
# Most negative correlation: minimum_nights and number_of_reviews with a value of -0.08
```

In [90]:
```
# Q4
# Part A
locations_airbnb = airbnb[['latitude', 'longitude', 'neighbourhood_group']]
locations_airbnb.loc[:,'neighbourhood_group'] = locations_airbnb['neighbourhood_group'].as
```

In [93]:
```
# create a dictionary mapping codes to neighborhood names
neighborhoods = dict(enumerate(locations_airbnb['neighbourhood_group'].cat.categories))

# create the scatter plot with color-coded points
ax = locations_airbnb.plot(kind='scatter', x='latitude', y='longitude', figsize=(20, 10),

# add a color key legend to the plot
colorbar = plt.colorbar(ax.collections[0])
colorbar.set_ticks(range(len(neighborhoods)))
colorbar.set_ticklabels([neighborhoods[key] for key in neighborhoods.keys()])
```

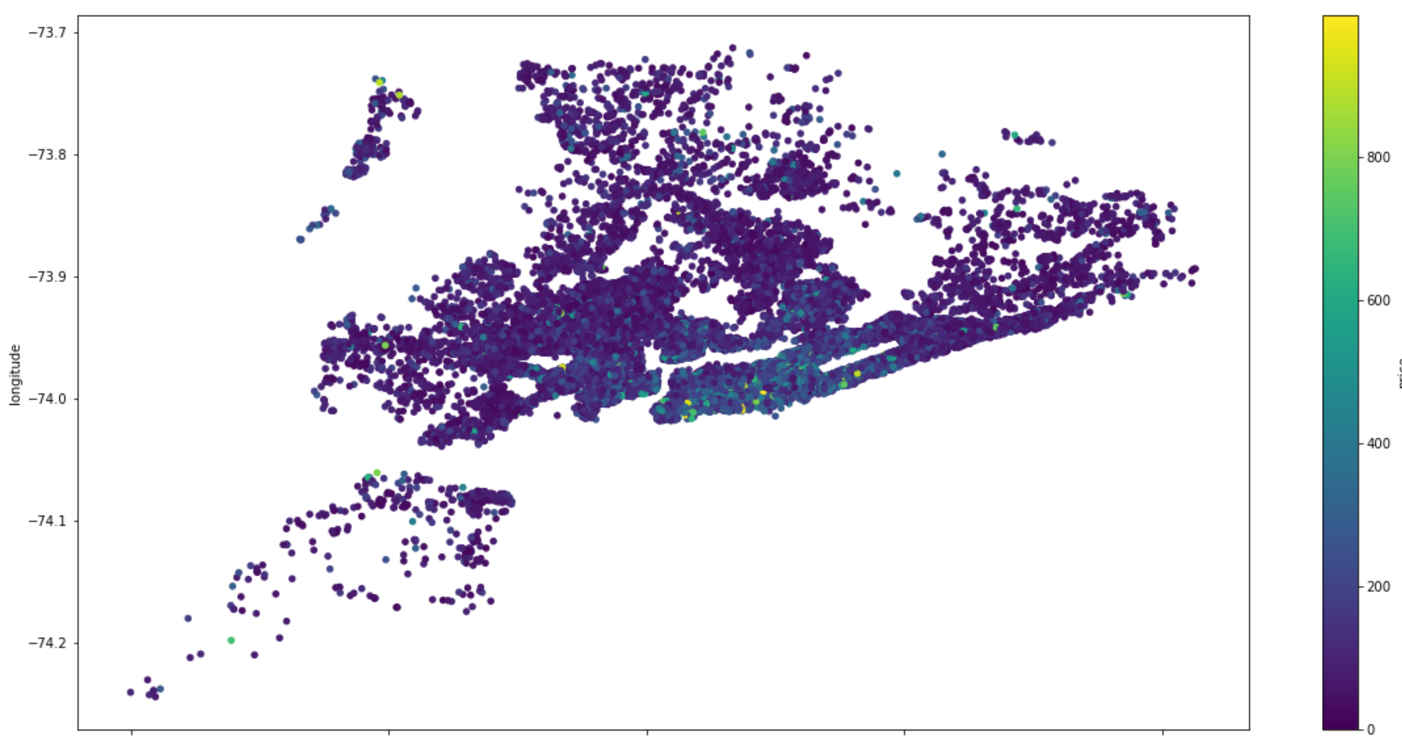


In [106…

```
# Part B
# Scatter plot with price as color-code
airbnb_location_price = airbnb.loc[airbnb['price'] < 1000, ['latitude', 'longitude', 'pric

airbnb_location_price.plot(kind="scatter", x='latitude', y='longitude', c='price', figsize
```

Out[106…

```
<AxesSubplot:xlabel='latitude', ylabel='longitude'>
```



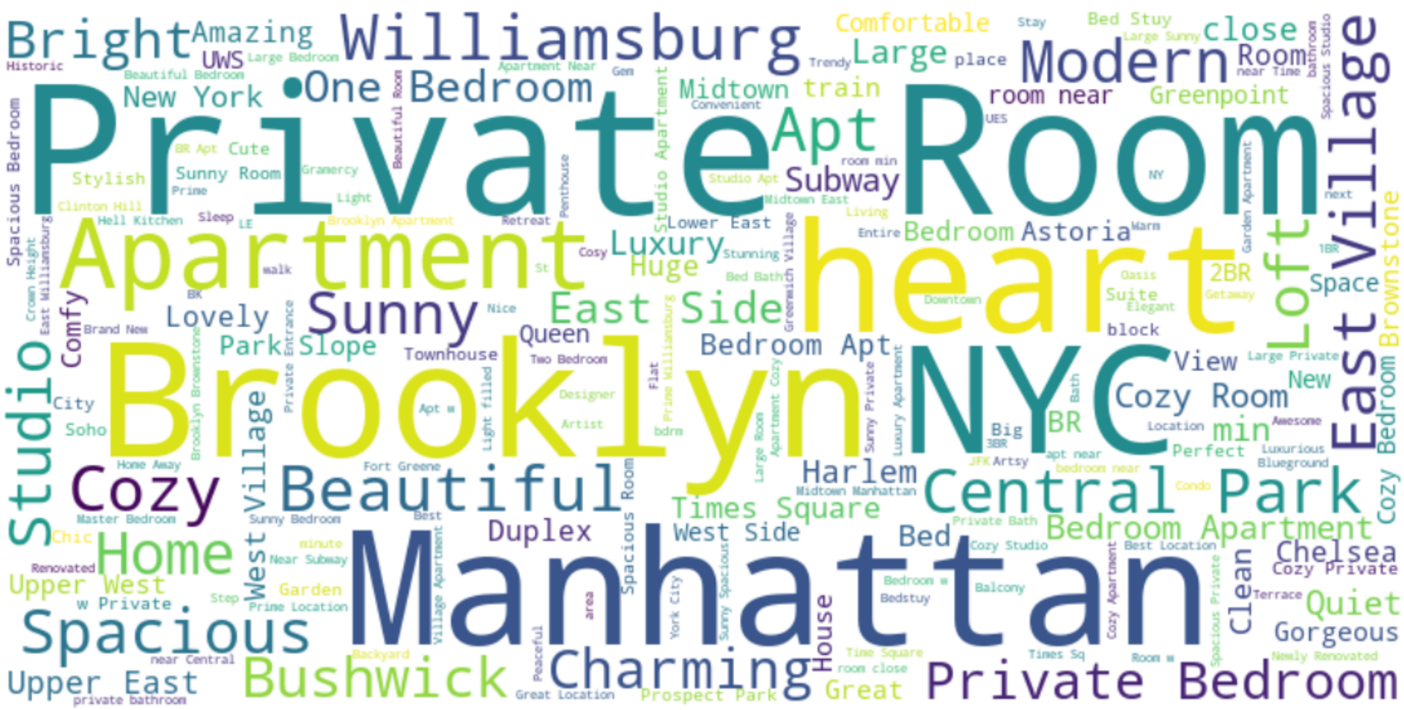
In [107…

```
# We can see the colors of green and dark blue which are more expensive than the majority
# are most prevalent in the Manhattan, Brooklyn, and Queens area. The ones with yellow co
# in the Manhattan area.
```

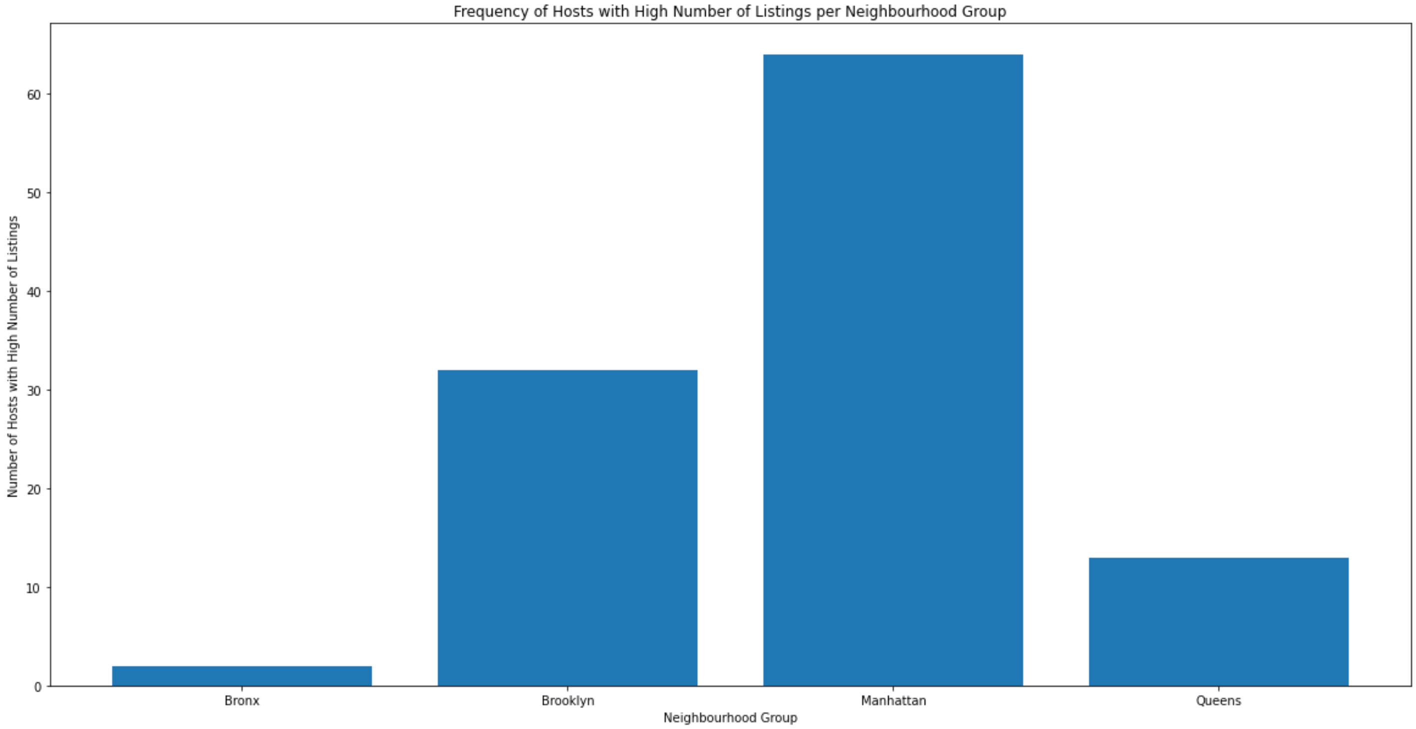
In [114…

```
# Q5
# Extract the names of the listings
```

```
names = airbnb['name'].astype(str)

# Combine all the names into a single string
all_names = ' '.join(names)

# Generate the word cloud
wordcloud = WordCloud(background_color='white', width=800, height=400).generate(all_names)

# Plot the word cloud
plt.figure(figsize=(20, 10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

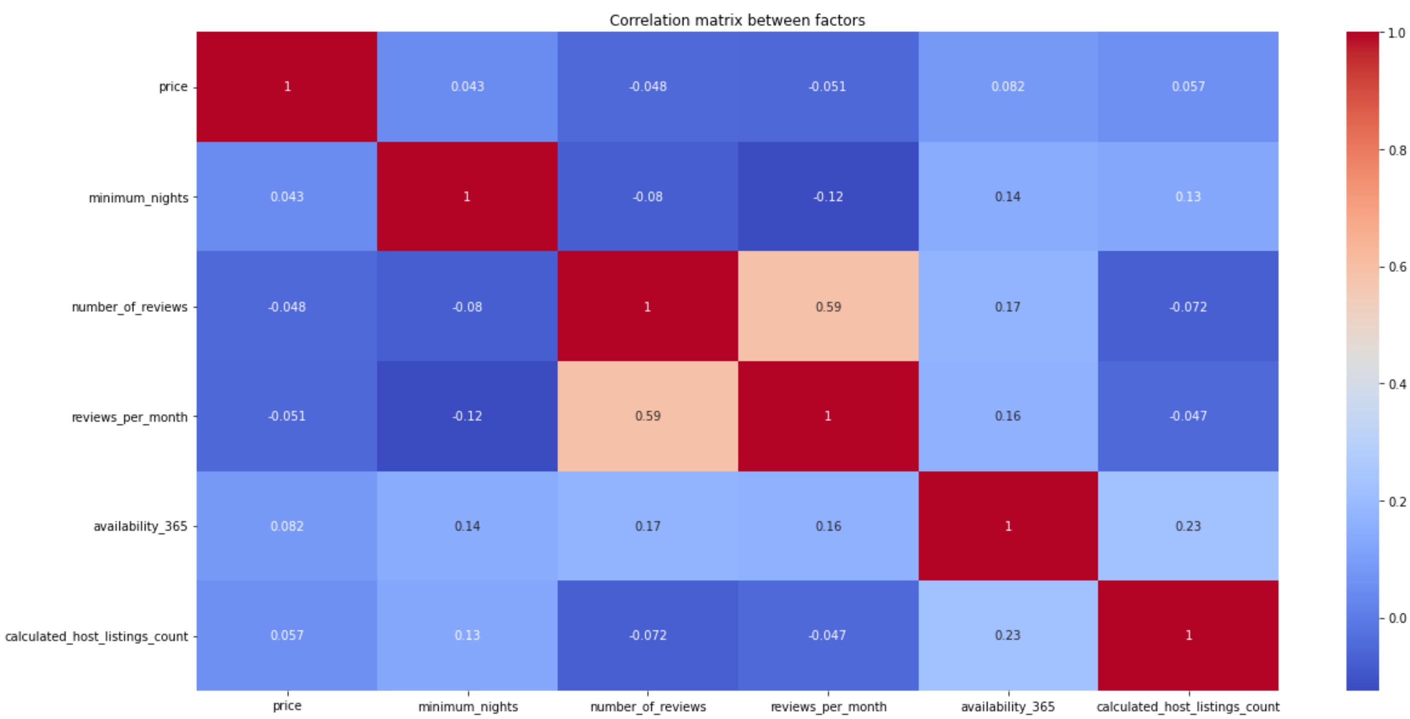



```
# Q6
# Identify "high" host listing counts
host_counts = airbnb[airbnb['calculated_host_listings_count'] > 10]

# Group the data by neighbourhood group and count the number of unique host IDs
grouped = host_counts.groupby('neighbourhood_group')['host_id'].nunique()

# Plot the results
fig, ax = plt.subplots(figsize=(20, 10))
ax.bar(grouped.index, grouped.values)
ax.set_xlabel('Neighbourhood Group')
ax.set_ylabel('Number of Hosts with High Number of Listings')
ax.set_title('Frequency of Hosts with High Number of Listings per Neighbourhood Group')
plt.show()
```



In [177…

```
# Calculate the correlation between the number of listings and other factors such as avai
corr_matrix = airbnb[['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month',

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(20, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation matrix between factors')
plt.show()
```



In [179…

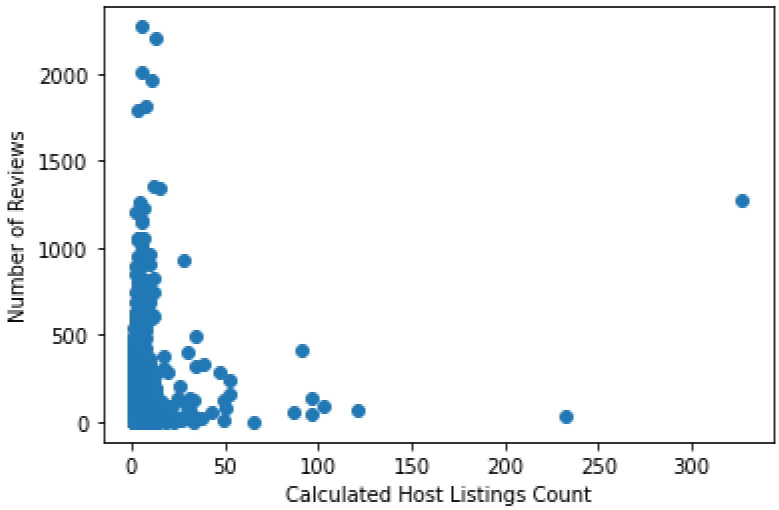
```
# Although it isn't a significant correlation, we can see based on the Pearson correlatio
# the more available the room was, the more host_listings_count it had.
```

In [190…

```
# Q7
# Plot 1: I will plot number of reviews per unique id vs calculated host listing count
```

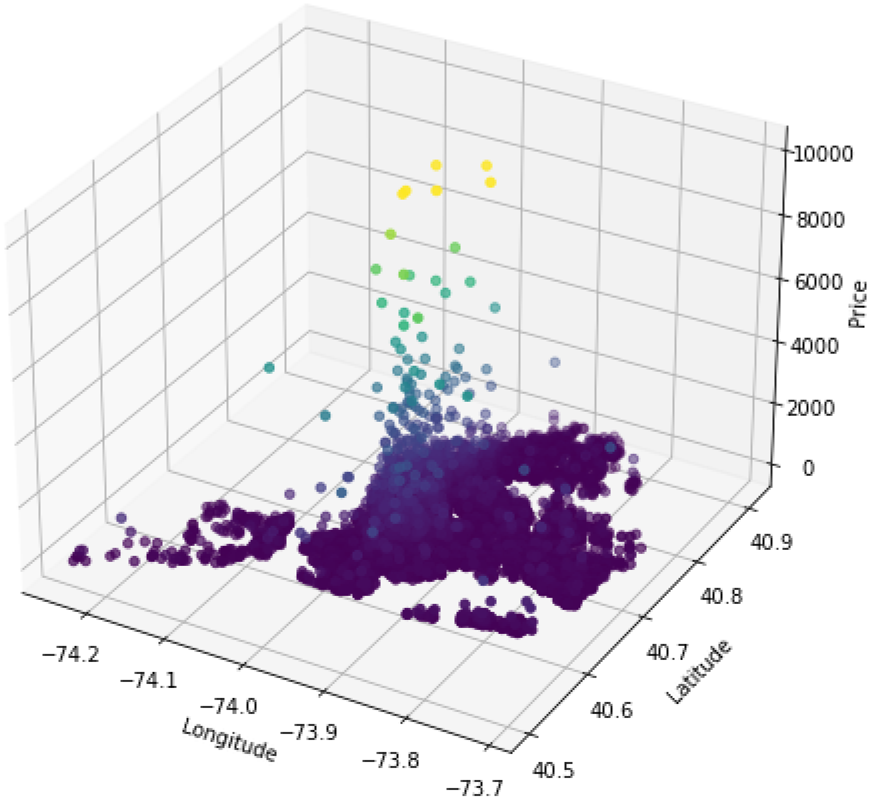
In [189…

```
# group the data by host_id and calculate the number of reviews per host
reviews_per_host = airbnb.groupby('host_id')['number_of_reviews'].sum()

# group the data by host_id and calculate the calculated_host_listings_count per host
listings_per_host = airbnb.groupby('host_id')['calculated_host_listings_count'].max()

# plot a scatter plot of listings per host versus reviews per host
plt.scatter(listings_per_host, reviews_per_host)
plt.xlabel('Calculated Host Listings Count')
plt.ylabel('Number of Reviews')
plt.show()

# This plot an attempt to show something I believed would be true: that if a host has higl
# they would also have high number of reviews. However, it was the opposite. A lot of low
# high number of reviews.
```



In [193…

```
# Plot 2: I will plot a 3d scatter plot of the location of each listings and the price
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

x = airbnb['longitude']
y = airbnb['latitude']
z = airbnb['price']

ax.scatter(x, y, z, c=z, cmap='viridis')

ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Price')

plt.show()

# This plot shows a better visualization of part B of Q4. It shows visually  which areas
# are more expensive and I believe it has more clarity since the previous one had too mucl
# overlap in colors.
```



10000
8000
6000
4000
2000
0
Price
−74.2
−74.1
−74.0
−73.9
−73.8
−73.7
Longitude
40.5
40.6
40.7
40.8
40.9
Latitude