NYC Airbnb Data Analysis

Jose Cao-Alvira, Mohammad Shafique, Melis Ocal, Victoria Vayner

FP Group 11

Part B

The following notebook relates to the analysis of the dataset "New York City Airbnb Open Data", available from Kaggle.

Source of the data is: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data (https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data)

The analysis uncovers insighful aspects of the data. Some of these include the geographical location of Airbnb accomodotions in NYC, their physical attributes, the reviewal process by users and the pricing.

Step #1

Import necessary libraries

In [1]:
```
# Import traditional libraries relevant to handle and visualize the data

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

%matplotlib inline
```

In [2]:
```
# Import Bokeh libraries relevant to visulazing the data using diagrams

import pandas_bokeh

from bokeh.resources import INLINE
import bokeh.io
bokeh.io.output_notebook(INLINE)

# import figure and gridplot objects
from bokeh.plotting import figure
from bokeh.layouts import gridplot

from bokeh.io import output_file, show

from bokeh.models import FactorRange

from bokeh.transform import dodge

# ColumnDataSource is bokeh's native data structure similar to pandas dataframe.
from bokeh.models import ColumnDataSource

# To change the color and shape of the markers
from bokeh.transform import factor_cmap, factor_mark

import panel as pn

pn.extension()
```

(https://Bokeh 2.2.3 successfully loaded.

In [3]:
```
# Import Bokeh libraries relevant to visulazing the data using maps

from bokeh.tile_providers import get_provider, WIKIMEDIA, CARTODBPOSITRON, STAMEN_TERRAIN, STAMEN_TONER, ESRI_IMAG
ERY, OSM
from bokeh.io import output_notebook

import warnings

warnings.filterwarnings("ignore")
pd.set_option("max_columns", 30)

output_notebook()
```

(https://bokeh.org) BokehJS 2.2.3 successfully loaded.

Step #2

Initial Data Processing

First upload the data.

"AB_NYC_2019.csv" contains the Kaggle dataset

In [4]:
```
DATA_DFo = pd.read_csv('AB_NYC_2019.csv')
```

We print the column names, the shape of the data, and the data types of the columns.

Null values can negatively impact the quaility of our analysis. Most null values are associated to the attributes related to reviews.

In [5]:
```
print(DATA_DFo.columns)
print(DATA_DFo.shape)
print(DATA_DFo.dtypes)

#Look for Null Values
for col in DATA_DFo.columns:
        na_df = DATA_DFo[col].isna()
        print('Number of Nulls in',col,':',na_df[na_df==True].count())

DATA_DF = DATA_DFo.copy()
```

```
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')
(48895, 16)
id                                int64
name                             object
host_id                           int64
host_name                        object
neighbourhood_group              object
neighbourhood                    object
latitude                        float64
longitude                       float64
room_type                        object
price                             int64
minimum_nights                    int64
number_of_reviews                 int64
last_review                      object
reviews_per_month               float64
calculated_host_listings_count    int64
availability_365                  int64
dtype: object
Number of Nulls in id : 0
Number of Nulls in name : 16
Number of Nulls in host_id : 0
Number of Nulls in host_name : 21
Number of Nulls in neighbourhood_group : 0
Number of Nulls in neighbourhood : 0
Number of Nulls in latitude : 0
Number of Nulls in longitude : 0
Number of Nulls in room_type : 0
Number of Nulls in price : 0
Number of Nulls in minimum_nights : 0
Number of Nulls in number_of_reviews : 0
Number of Nulls in last_review : 10052
Number of Nulls in reviews_per_month : 10052
Number of Nulls in calculated_host_listings_count : 0
Number of Nulls in availability_365 : 0
```

We chose to make null values equal to zero

In [6]:
```
DATA_DF.fillna(0)
print(DATA_DF.shape)
for col in DATA_DF.columns:
        na_df = DATA_DF[col].isna()
        print('Number of Nulls in',col,':',na_df[na_df==True].count())

print(DATA_DFo.shape)
```

```
(48895, 16)
Number of Nulls in id : 0
Number of Nulls in name : 16
Number of Nulls in host_id : 0
Number of Nulls in host_name : 21
Number of Nulls in neighbourhood_group : 0
Number of Nulls in neighbourhood : 0
Number of Nulls in latitude : 0
Number of Nulls in longitude : 0
Number of Nulls in room_type : 0
Number of Nulls in price : 0
Number of Nulls in minimum_nights : 0
Number of Nulls in number_of_reviews : 0
Number of Nulls in last_review : 10052
Number of Nulls in reviews_per_month : 10052
Number of Nulls in calculated_host_listings_count : 0
Number of Nulls in availability_365 : 0
(48895, 16)
```

We create a series of dummy variables related to the NYC Boroughs and Types of accomodations, and renamed some columns for easier management

In [7]:
```
Borough = DATA_DF["neighbourhood_group"].copy()
Types   = DATA_DF["room_type"].copy()

DATA_DF["Borough"] = Borough
DATA_DF["Types"]   = Types

DATA_DF = pd.get_dummies(data=DATA_DF, drop_first = False, columns=['neighbourhood_group','room_type'])

DATA_DF.rename(columns={'neighbourhood_group_Bronx': "Bronx", 'neighbourhood_group_Brooklyn': "Brooklyn", 'neighbo
urhood_group_Queens': "Queens", 'neighbourhood_group_Manhattan': "Manhattan", 'neighbourhood_group_Staten Island':
"StatenIsland",}, inplace = True)

DATA_DF.rename(columns={'room_type_Entire home/apt': 'HomeEntire','room_type_Private room': 'RoomPrivate','room_ty
pe_Shared room': 'RoomShared'}, inplace = True)
```

We print the column names to confirm the changes

In [8]:
```
DATA_DF.columns
```

Out[8]:
```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood', 'latitude',
       'longitude', 'price', 'minimum_nights', 'number_of_reviews',
       'last_review', 'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'Borough', 'Types', 'Bronx', 'Brooklyn',
       'Manhattan', 'Queens', 'StatenIsland', 'HomeEntire', 'RoomPrivate',
       'RoomShared'],
      dtype='object')
```

Step #3

Create an Interactive Map-Visualization of the location of Airbnbs in NYC

We referenced methods followed in [https://coderzcolumn.com/tutorials/data-science/plotting-maps-using-bokeh (https://coderzcolumn.com/tutorials/data-science/plotting-maps-using-bokeh)](https://coderzcolumn.com/tutorials/data-science/plotting-maps-using-bokeh)

This FIRST step creates a blank World Map using the Mercatr projection

In [9]:
```
from pyproj import Proj, transform

inProj = Proj(init='epsg:3857')
outProj = Proj(init='epsg:4326')

world_lon1, world_lat1 = transform(outProj,inProj,-180,-85)
world_lon2, world_lat2 = transform(outProj,inProj,180,85)

cartodb = get_provider(CARTODBPOSITRON)

fig = figure(plot_width=800, plot_height=700,
             x_range=(world_lon1, world_lon2),
             y_range=(world_lat1, world_lat2),
             x_axis_type="mercator", y_axis_type="mercator")

fig.add_tile(cartodb)

#show(fig)
```

Out[9]: **TileRenderer**(id = '1043', …)

The SECOND step maps the "actual" Latitude and Longitude for each Aribnb (from Kaggle), and creates a Mercator XY coordinate to include it in the map

This step usually takes a lot of time for a big data set.

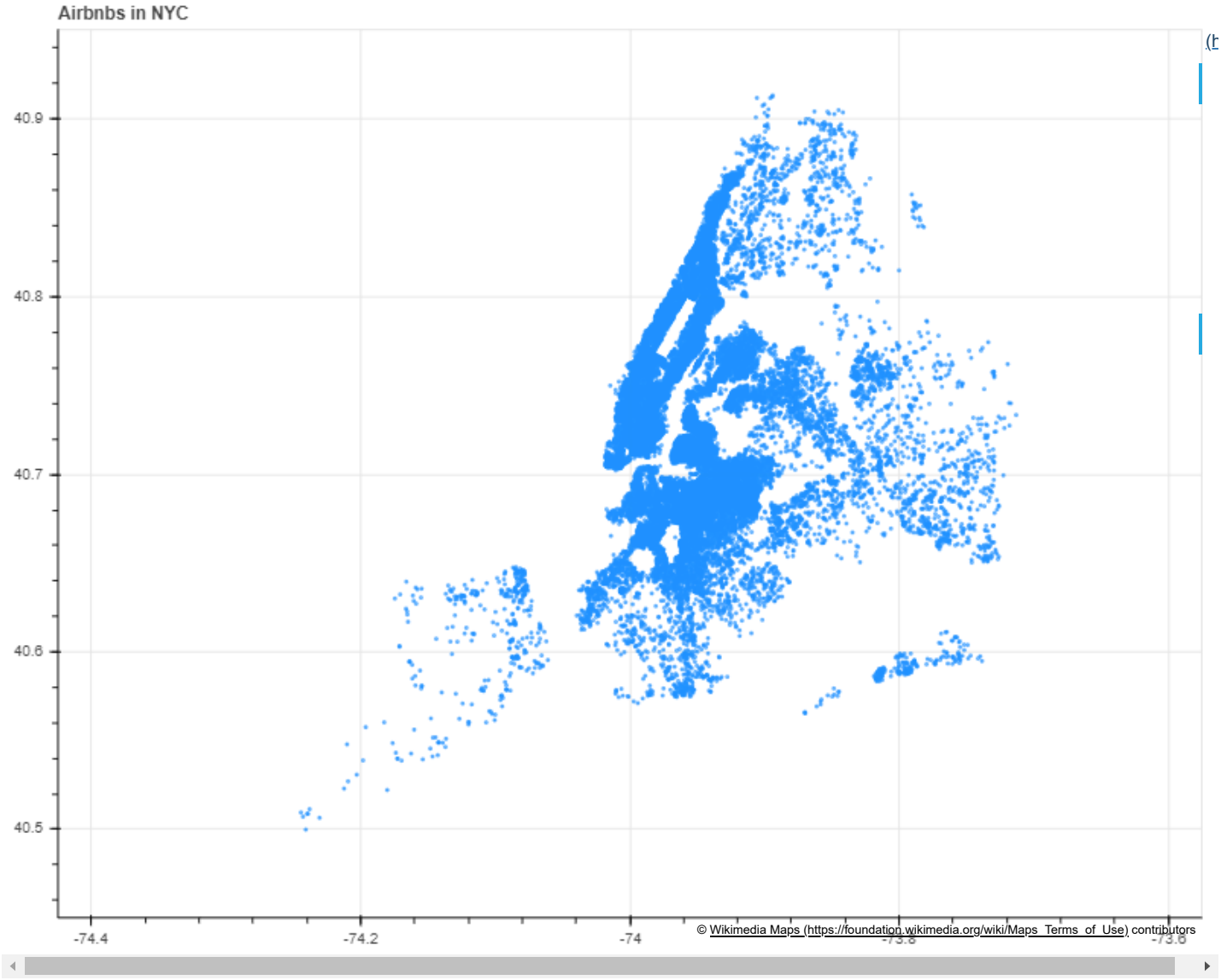
In our case, it took 33mins

In [10]:
```
%%time

lons, lats = [], []
for lon, lat in list(zip(DATA_DF['longitude'], DATA_DF['latitude'])):
    x, y = transform(outProj,inProj,lon,lat)
    lons.append(x)
    lats.append(y)

DATA_DF["MercatorX"] = lons
DATA_DF["MercatorY"] = lats
```

```
Wall time: 42min 39s
```

The LAST step creates the map using the MercatorXY coordinates

In [11]:
```
wikimedia = get_provider(WIKIMEDIA)

#ny_lon1, ny_lat1 = transform(outProj,inProj,-140,10)
#ny_lon2, ny_lat2 = transform(outProj,inProj,-50,55)

ny_lon1, ny_lat1 = transform(outProj,inProj,-74.0,40.45)
ny_lon2, ny_lat2 = transform(outProj,inProj,-74.0,40.95)

p = figure(plot_width=900, plot_height=700,
           x_range=(ny_lon1, ny_lon2), y_range=(ny_lat1, ny_lat2),
           x_axis_type="mercator", y_axis_type="mercator",
           tooltips=[
                    ("Neighbourhood", "@neighbourhood"), ("name", "@name"), ("(Long, Lat)", "(@longitude, @latitud
e)")
                    ],
           title="Airbnbs in NYC")

p.add_tile(wikimedia)

p.circle(x="MercatorX", y="MercatorY",
         size=2,
         fill_color="dodgerblue", line_color="dodgerblue",
         fill_alpha=0.3,
         source=DATA_DF)

show(p)
```



As a nudge to Part A of the Group Project We illustrate a map containing the Airbnb in the Murray Hill neighborhood of NYC

In [26]:

```
wikimedia = get_provider(WIKIMEDIA)

#ny_lon1, ny_lat1 = transform(outProj,inProj,-140,10)
#ny_lon2, ny_lat2 = transform(outProj,inProj,-50,55)

ny_lon1, ny_lat1 = transform(outProj,inProj,-73.98,40.74)
ny_lon2, ny_lat2 = transform(outProj,inProj,-73.97,40.75)

p_MH = figure(plot_width=900, plot_height=700,
           x_range=(ny_lon1, ny_lon2), y_range=(ny_lat1, ny_lat2),
           x_axis_type="mercator", y_axis_type="mercator",
           tooltips=[
                    ("Neighbourhood", "@neighbourhood"), ("name", "@name"), ("(Long, Lat)", "(@longitude, @latitud
e)")
                    ],
           title="Airbnbs in Murray Hill (nudge to Part A of the Group Project)")

p_MH.add_tile(wikimedia)

p_MH.circle(x="MercatorX", y="MercatorY",
         size=2,
         fill_color="dodgerblue", line_color="dodgerblue",
         fill_alpha=0.3,
         source=DATA_DF)

show(p_MH)
```