Final Proposal: Airbnb New York City Pricing Dynamics

Principal Investigators : Samantha Warsop & Brittany Miu

Email: sw3469@nyu.edu bm2352@nyu.edu

This project will explore how pricing differs across Airbnbs and the influences it has on demand. We will also explore the seasonal pattern of airbnb prices in New York and the effects it has on travel. As an example, in New York City, Airbnb prices across different neighbourhood groups such as Manhattan, Brooklyn, Queens, Staten Island, and the Bronx might differ. We can look at how pricing is varies across these different neighbourhood groups in terms of the number of the listings available and property type available.

The key element of the project is the use of Airbnb's data, providing access to measures such as prices, number of listings, property type, etc. in New York. Detailed of this dataset are described below in the data report.

There will be three different sections in this project:

1. Basic Data Analysis

This section will have different summary statistics describing the number of listings and property type in each neighbourhood group.

2. Pricing Effect on Demand for New York City Airbnbs

This section will explore how prices differ across different neighbourhood groups and discover what factors prices are dependent on. We will have visualizations such as a map to indicate where entire apartments/homes are most prevalent. There will be a bar chart illustrating the average prices in each neighbourhood. By analyzing the number of listings and prices per neighbourhood, we can find out which neighbourhood is the most optimal.

3. Seasonal Pattern of Prices

The last will explore how prices vary across different seasons. We plan to have visualizations showing how prices change over the year and provide explanations as to why. For example, airbnb prices during the holidays might be more expensive than during non-holidays.

Data Report

Overview:

The data behind our project comes from [insideairbnb (http://insideairbnb.com/get-the-data.html)](http://insideairbnb.com/get-the-data.html) . Their [New York city data (http://insideairbnb.com/new-york-city/)](http://insideairbnb.com/new-york-city/) provides access to information on room types, availability, activity, as well as listings per host.

Important Variables:

The key series that we must retrive is within insideairbnb's data on [New York city data (http://insideairbnb.com/new-york-city/)](http://insideairbnb.com/new-york-city/). This data provides the airbnb locations, as well as pricing, which will allow us to determine answers to both questions 1 and 2.
This data combined with utalizing datetime and holiday functions will allow us to analyze Airbnb's seasonal pattern of prices.

Access:

We will use insideairbnb to download and acess the data. Below we will demonstrate that we have the ability to access the data.

Requisite Packages:

Below we will bring in the packages we need:

In [1]:

```
import pandas as pd
import numpy as np #numerical analysis
import matplotlib.pyplot as plt #plotting
import geopandas as gpd
import os
import time
from datetime import date
import datetime
import holidays
import calendar
from mpl_toolkits.axes_grid1.inset_locator import zoomed_inset_axes
from mpl_toolkits.axes_grid1.inset_locator import mark_inset
```

In [2]:

```
file= "/Users/SamanthaWarsop 1/Airbnb New York/listings.csv"
```

In [3]:

```
listings = pd.read_csv(file)
```

```
/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshe
ll.py:3020: DtypeWarning: Columns (43,95) have mixed types. Specif
y dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [4]:

```
listings.head()
```

Out[4]:

	id	listing_url	scrape_id	last_scraped	nam
0	1742654	https://www.airbnb.com/rooms/1742654	20190503153024	2019-05-04	Hig Floo apt.nea Columbu Circ
1	23502842	https://www.airbnb.com/rooms/23502842	20190503153024	2019-05-04	Cozy Ea Villag stud
2	15984984	https://www.airbnb.com/rooms/15984984	20190503153024	2019-05-04	Grea Locatio b Subwa
3	13820083	https://www.airbnb.com/rooms/13820083	20190503153024	2019-05-04	Beautif Coz Garde Ap Histor Clinto H
4	6170979	https://www.airbnb.com/rooms/6170979	20190503153024	2019-05-04	Cozy Bedroo apartme fitting

5 rows × 106 columns

Then we will clean up the data a bit by replacing all the NaN with 0, converting the price type to a floating number, and excluding the listing with 0 for price, bedrooms, accommodations, etc.

In [5]:

```
#replacing Nan with 0
listings.fillna(0, inplace = True)

#Getting rid of $ signs and converting price to float
listings['price'] = listings['price'].str.replace('[^\d\.]', '').astype(float)

#Excluding listings with 0 for price, bedrooms, accomodations, etc.
listings = listings[listings.bathrooms >0]
listings = listings[listings.bedrooms > 0]
listings = listings[listings.beds > 0]
listings = listings[listings.price  > 0]
listings = listings[listings.review_scores_rating  > 0]
listings = listings[listings.reviews_per_month > 0]
listings = listings[listings.accommodates  > 0]

listings.head()
```

Out[5]:

	id	listing_url	scrape_id	last_scraped	nam
0	1742654	https://www.airbnb.com/rooms/1742654	20190503153024	2019-05-04	Hig Floo apt.nea Columbu Circ
2	15984984	https://www.airbnb.com/rooms/15984984	20190503153024	2019-05-04	Grea Locatio b Subwa
3	13820083	https://www.airbnb.com/rooms/13820083	20190503153024	2019-05-04	Beautif Coz Garde Ap Histor Clinto H
4	6170979	https://www.airbnb.com/rooms/6170979	20190503153024	2019-05-04	Cozy Bedroo apartme fitting
5	27283214	https://www.airbnb.com/rooms/27283214	20190503153024	2019-05-04	Room Luxu Building Midtow

5 rows × 106 columns

Now we are going to delete the unecessary columns

In [6]:

```
listings.drop(['listing_url', 'scrape_id', 'last_scraped', 'name', 'summary',
'space', 'description', 'neighborhood_overview', 'cancellation_policy', 'notes
', 'transit', 'access', 'interaction', 'house_rules', 'thumbnail_url', 'medium
_url', 'picture_url', 'xl_picture_url', 'host_url', 'host_name', 'host_since',
'host_location', 'host_about', 'host_response_time', 'host_response_rate', 'ho
st_thumbnail_url', 'host_picture_url', 'host_neighbourhood', 'host_listings_co
unt', 'host_total_listings_count', 'host_verifications', 'host_has_profile_pic
', 'host_identity_verified', 'amenities', 'square_feet', 'minimum_minimum_nigh
ts', 'maximum_minimum_nights', 'minimum_maximum_nights', 'maximum_maximum_nigh
ts', 'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'calendar_updated', '
calendar_last_scraped', 'first_review', 'last_review', 'license', 'jurisdictio
n_names'], axis=1, inplace = True)
```

In [7]:

```
listings.drop(['calculated_host_listings_count_shared_rooms', 'calculated_host
_listings_count_private_rooms', 'calculated_host_listings_count_entire_homes',
'calculated_host_listings_count', 'security_deposit', 'cleaning_fee', 'bed_typ
e'], axis=1, inplace=True)
```

In [8]:

```
listings.head()
```

Out[8]:

	id	experiences_offered	host_id	host_acceptance_rate	host_is_superhost	s
0	1742654	none	9173924	0.0	t	York U S
2	15984984	none	9737900	0.0	t	Broo U S
3	13820083	none	31829334	0.0	f	Broo U S
4	6170979	none	31104121	0.0	f	Broo U S
5	27283214	none	3508466	0.0	f	York U S

5 rows × 52 columns

In [9]:

```
listings.columns
```

Out[9]:

```
Index(['id', 'experiences_offered', 'host_id', 'host_acceptance_ra
te',
       'host_is_superhost', 'street', 'neighbourhood',
       'neighbourhood_cleansed', 'neighbourhood_group_cleansed', '
city',
       'state', 'zipcode', 'market', 'smart_location', 'country_co
de',
       'country', 'latitude', 'longitude', 'is_location_exact',
       'property_type', 'room_type', 'accommodates', 'bathrooms',
'bedrooms',
       'beds', 'price', 'weekly_price', 'monthly_price', 'guests_i
ncluded',
       'extra_people', 'minimum_nights', 'maximum_nights', 'has_av
ailability',
       'availability_30', 'availability_60', 'availability_90',
       'availability_365', 'number_of_reviews', 'number_of_reviews
_ltm',
       'review_scores_rating', 'review_scores_accuracy',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'requires_license', 'instant_bookabl
e',
       'is_business_travel_ready', 'require_guest_profile_picture'
,
       'require_guest_phone_verification', 'reviews_per_month'],
      dtype='object')
```

Now we will grab the time series data from calendar.csv to evaluate how price changes based on season

In [10]:

```
calendar_file = "/Users/SamanthaWarsop 1/Airbnb New York/calendar.csv"
```

In [11]:

```
calendar = pd.read_csv(calendar_file)
calendar.head()
```

Out[11]:

	listing_id	date	available	price	adjusted_price	minimum_nights	maximum_nights
0	36647	2019-03-07	f	$69.00	$69.00	2.0	730.0
1	36647	2019-03-08	f	$69.00	$69.00	2.0	730.0
2	36647	2019-03-09	f	$69.00	$69.00	2.0	730.0
3	36647	2019-03-10	f	$69.00	$69.00	2.0	730.0
4	36647	2019-03-11	f	$69.00	$69.00	2.0	730.0

Then I am going to clean up the data a bit by replacing all the NaN with 0, converting the price to a floating number, and separating the date column into day, month, and year.

In [12]:

```
#replacing NaN with 0
calendar.fillna(0, inplace = True)

#converting price to float
calendar['price'] = calendar['price'].str.replace('[^\d\.]', '').astype(float)

#Excluding listing with 0 for price
calendar = calendar[calendar['price'] >= 0]

#Separating date column into day, month, and year
calendar['Year'],calendar['Month'],calendar['Day']=calendar['date'].str.split(
'-',2).str

#Deleting column for adjusted price
calendar.drop(['adjusted_price'], axis=1, inplace=True)

calendar.head()
```

Out[12]:

	listing_id	date	available	price	minimum_nights	maximum_nights	Year	Month	Day
0	36647	2019-03-07	f	69.0	2.0	730.0	2019	03	07
1	36647	2019-03-08	f	69.0	2.0	730.0	2019	03	08
2	36647	2019-03-09	f	69.0	2.0	730.0	2019	03	09
3	36647	2019-03-10	f	69.0	2.0	730.0	2019	03	10
4	36647	2019-03-11	f	69.0	2.0	730.0	2019	03	11

Here are some summary statistics.

In [13]:

```
room_type = listings.groupby('room_type').id.count()
```

In [14]:

```
room_type
```

Out[14]:

```
room_type
Entire home/apt    16467
Private room       16139
Shared room          737
Name: id, dtype: int64
```

In [15]:

```
neighborhood_group = listings.groupby('neighbourhood_group_cleansed').id.count
()
```

In [16]:

```
neighborhood_group
```

Out[16]:

```
neighbourhood_group_cleansed
Bronx              745
Brooklyn         14555
Manhattan        13817
Queens            3969
Staten Island      257
Name: id, dtype: int64
```

In [17]:

```
listings[listings.neighbourhood_group_cleansed == "Manhattan"].head(10)
```

Out[17]:

	id	experiences_offered	host_id	host_acceptance_rate	host_is_superhost	st
0	1742654	none	9173924	0.0	t	Ur St
5	27283214	none	3508466	0.0	f	Ur St
7	33014	none	143048	0.0	f	Ur St
11	150804	none	726333	0.0	f	Ur St
16	32783365	none	25312503	0.0	f	Ur St
17	1182844	none	6470443	0.0	f	Ur St
18	1151782	none	1002618	0.0	f	Ur St
19	19219624	none	134521683	0.0	f	Ur St
21	19830008	none	139942077	0.0	f	Ur St
24	26518779	none	6072790	0.0	f	Ur St

10 rows × 52 columns

Summary

With the listings.csv, we can answer the first and second question that will evaluate the differences in number of listings, property type, etc. to explain the price differences across different neighbourhood groups. With the calendar.csv, we can evaluate the seasonal patterns of prices. With the combined data frame, we can go more indepth of how prices changes over the years based on different neighborhood groups.

We look forward to finding the answers to our questions, and seeing where the data takes us!

Exporting the csv below

In [18]:

```
listings.rename(columns={'id':'listing_id'}, inplace=True)
```

In [19]:

```
listings.columns
```

Out[19]:

```
Index(['listing_id', 'experiences_offered', 'host_id', 'host_accep
tance_rate',
       'host_is_superhost', 'street', 'neighbourhood',
       'neighbourhood_cleansed', 'neighbourhood_group_cleansed', '
city',
       'state', 'zipcode', 'market', 'smart_location', 'country_co
de',
       'country', 'latitude', 'longitude', 'is_location_exact',
       'property_type', 'room_type', 'accommodates', 'bathrooms',
'bedrooms',
       'beds', 'price', 'weekly_price', 'monthly_price', 'guests_i
ncluded',
       'extra_people', 'minimum_nights', 'maximum_nights', 'has_av
ailability',
       'availability_30', 'availability_60', 'availability_90',
       'availability_365', 'number_of_reviews', 'number_of_reviews
_ltm',
       'review_scores_rating', 'review_scores_accuracy',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'requires_license', 'instant_bookabl
e',
       'is_business_travel_ready', 'require_guest_profile_picture'
,
       'require_guest_phone_verification', 'reviews_per_month'],
      dtype='object')
```

In [20]:

```
listings_calendar = pd.merge(listings, calendar, on='listing_id', how='outer')
```

In [21]:

```
listings_calendar
```

Out[21]:

	listing_id	experiences_offered	host_id	host_acceptance_rate	host_is_superhc
0	1742654	none	9173924.0	0.0	
1	1742654	none	9173924.0	0.0	
2	1742654	none	9173924.0	0.0	
3	1742654	none	9173924.0	0.0	
4	1742654	none	9173924.0	0.0	
5	1742654	none	9173924.0	0.0	
6	1742654	none	9173924.0	0.0	
7	1742654	none	9173924.0	0.0	
8	1742654	none	9173924.0	0.0	
9	1742654	none	9173924.0	0.0	

10	1742654	none	9173924.0	0.0
11	1742654	none	9173924.0	0.0
12	1742654	none	9173924.0	0.0
13	1742654	none	9173924.0	0.0
14	1742654	none	9173924.0	0.0
15	1742654	none	9173924.0	0.0
16	1742654	none	9173924.0	0.0
17	1742654	none	9173924.0	0.0
18	1742654	none	9173924.0	0.0
19	1742654	none	9173924.0	0.0
20	1742654	none	9173924.0	0.0
21	1742654	none	9173924.0	0.0

22	1742654	none	9173924.0	0.0	
23	1742654	none	9173924.0	0.0	
24	1742654	none	9173924.0	0.0	
25	1742654	none	9173924.0	0.0	
26	1742654	none	9173924.0	0.0	
27	1742654	none	9173924.0	0.0	
28	1742654	none	9173924.0	0.0	
29	1742654	none	9173924.0	0.0	
...	...	...	...	...	
18160235	32765748	NaN	NaN	NaN	Na
18160236	32765748	NaN	NaN	NaN	Na
18160237	32765748	NaN	NaN	NaN	Na
18160238	32765748	NaN	NaN	NaN	Na
18160239	32765748	NaN	NaN	NaN	Na
18160240	32765748	NaN	NaN	NaN	Na
18160241	32765748	NaN	NaN	NaN	Na

18160242	32765748	NaN	NaN	NaN	Na
18160243	32765748	NaN	NaN	NaN	Na
18160244	32765748	NaN	NaN	NaN	Na
18160245	32765748	NaN	NaN	NaN	Na
18160246	32765748	NaN	NaN	NaN	Na
18160247	32765748	NaN	NaN	NaN	Na
18160248	32765748	NaN	NaN	NaN	Na
18160249	32765748	NaN	NaN	NaN	Na
18160250	32765748	NaN	NaN	NaN	Na
18160251	32765748	NaN	NaN	NaN	Na
18160252	32765748	NaN	NaN	NaN	Na
18160253	32765748	NaN	NaN	NaN	Na
18160254	32765748	NaN	NaN	NaN	Na
18160255	32765748	NaN	NaN	NaN	Na
18160256	32765748	NaN	NaN	NaN	Na
18160257	32765748	NaN	NaN	NaN	Na
18160258	32765748	NaN	NaN	NaN	Na
18160259	32765748	NaN	NaN	NaN	Na
18160260	32765748	NaN	NaN	NaN	Na
18160261	32765748	NaN	NaN	NaN	Na
18160262	32765748	NaN	NaN	NaN	Na
18160263	32765748	NaN	NaN	NaN	Na
18160264	32765748	NaN	NaN	NaN	Na

18160265 rows × 60 columns

In [22]:

```
listings_calendar = listings_calendar.dropna()
```

In [23]:

```
listings_calendar
```

Out[23]:

	listing_id	experiences_offered	host_id	host_acceptance_rate	host_is_superho
0	1742654	none	9173924.0	0.0	
1	1742654	none	9173924.0	0.0	
2	1742654	none	9173924.0	0.0	
3	1742654	none	9173924.0	0.0	
4	1742654	none	9173924.0	0.0	
5	1742654	none	9173924.0	0.0	
6	1742654	none	9173924.0	0.0	
7	1742654	none	9173924.0	0.0	
8	1742654	none	9173924.0	0.0	

9	1742654	none	9173924.0	0.0
10	1742654	none	9173924.0	0.0
11	1742654	none	9173924.0	0.0
12	1742654	none	9173924.0	0.0
13	1742654	none	9173924.0	0.0
14	1742654	none	9173924.0	0.0
15	1742654	none	9173924.0	0.0
16	1742654	none	9173924.0	0.0
17	1742654	none	9173924.0	0.0
18	1742654	none	9173924.0	0.0
19	1742654	none	9173924.0	0.0
20	1742654	none	9173924.0	0.0

21	1742654	none	9173924.0	0.0
22	1742654	none	9173924.0	0.0
23	1742654	none	9173924.0	0.0
24	1742654	none	9173924.0	0.0
25	1742654	none	9173924.0	0.0
26	1742654	none	9173924.0	0.0
27	1742654	none	9173924.0	0.0
28	1742654	none	9173924.0	0.0
29	1742654	none	9173924.0	0.0
...	...	...	...	...
11376266	1672860	none	2119276.0	0.0
11376267	1672860	none	2119276.0	0.0
11376268	1672860	none	2119276.0	0.0

11376269	1672860	none	2119276.0	0.0
11376270	1672860	none	2119276.0	0.0
11376271	1672860	none	2119276.0	0.0
11376272	1672860	none	2119276.0	0.0
11376273	1672860	none	2119276.0	0.0
11376274	1672860	none	2119276.0	0.0
11376275	1672860	none	2119276.0	0.0
11376276	1672860	none	2119276.0	0.0
11376277	1672860	none	2119276.0	0.0
11376278	1672860	none	2119276.0	0.0
11376279	1672860	none	2119276.0	0.0

11376280	1672860	none	2119276.0	0.0
11376281	1672860	none	2119276.0	0.0
11376282	1672860	none	2119276.0	0.0
11376283	1672860	none	2119276.0	0.0
11376284	1672860	none	2119276.0	0.0
11376285	1672860	none	2119276.0	0.0
11376286	1672860	none	2119276.0	0.0
11376287	1672860	none	2119276.0	0.0
11376288	1672860	none	2119276.0	0.0
11376289	1672860	none	2119276.0	0.0
11376290	1672860	none	2119276.0	0.0
11376291	1672860	none	2119276.0	0.0

11376292	1672860	none	2119276.0	0.0
11376293	1672860	none	2119276.0	0.0
11376294	1672860	none	2119276.0	0.0
11376295	1672860	none	2119276.0	0.0

11374115 rows × 60 columns

In [24]:

```
listings.to_csv("clean_listings.csv")
```

In [25]:

```
calendar.to_csv("clean_calendar.csv")
```

In [26]:

```
listings_calendar.to_csv('listings_calendar.csv')
```