

# AirBnB Hot Spots

## Project 1

1/2/2019

Amanda Wishnie  
Eric Lieu  
Maria J. Villacreses  
Francis Imperial

### **Important to note:**

- *Our groups completed work can be found here in the [Final Code folder!](#)*

### **Primary Question:**

- What affects AirBnB prices?
  - Motivation to understand what external factors might affect AirBnB nightly prices
  - With respect to four factors:
    - Temperature
    - Amount of Crime
    - Household Income
    - Demographics (Age and Race)

### **How & Where:**

- Python, Excel, API's
- AirBnB Open Source Data
- US Census Bureau Data (American Community Survey)
- Accuweather
- NYC Open Data

### **Getting Started:**

- Each team member took on a year (2015-2018) and extracted 2 CSV files from [AirBnB's open data source](#) which broke out each year by month. We then manipulated the data in Excel in order to organize by zip code and date.
  - Downloaded large csv file and extracted zip codes
  - Transferred those zipcodes to csv file with less specific information/information more pertinent to our project (like location, price, etc)
  - Refer to Base code for following procedures ([Base Code 2018 Data](#))
  - Took zip code data and filtered the top 30 zip codes for each year (2015-2018) with respect to volume of listings to alleviate influence of any outlier prices
  - Found the mean and median prices of all 30 zip codes with respect to the month of the year and graphed it

- After cleaning the data in Excel, we created CSV files and worked in Python to merge the 4 years datasets to find top 30 zip codes among all years
- Finding the final 30 zip codes ([Jupyter Notebook for Top 30 Zip Codes](#)):
  - CSV files for the top 30 zip codes for each year (2015 - 2018) generated by each group member were concatenated
  - CSV generated using groupby “Zip code” count (‘consolidated.csv’)
  - ‘consolidated3.csv’ is ‘consolidated.csv’ with externally added for columns ‘Borough’, ‘Neighborhood’, and ‘Precinct’
    - Neighborhood/Borough: <https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoods.htm>
    - NYC Precinct Lookup: <https://www1.nyc.gov/site/nypd/bureaus/patrol/find-your-precinct.page>
- By leveraging the final DataFrame, each team member took a deep dive into their question and broke down the data into different granularities.

### **Question 1:**

**Does temperature affect prices?**

### **Average Monthly Weather from (2015-2018):**

#### **I) Raw Data**

- A. Obtained from closed paid subscription source: ACCUWEATHER

#### **II) Data Cleansing**

- A. Limitations: (paid sources available only. Also, Accuweather only provides tables with temperature data, no csv or API’s available to extract data with Premium subscription. This part took the most work).
- B. Manually extracted temperature data tables and imported them in excel, transposed them in column under each month and used vlookup to separate High and Low daily temp per month individually.

#### **III) Data Manipulation**

```

localhost:8892/notebooks/AvgWeather.ipynb
Jupyter AvgWeather Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [105]: read_fileL = pd.read_csv(file_low)
          mean_low = read_fileL.mean()

          read_fileH = pd.read_csv(file_high)
          mean_high = read_fileH.mean()

          low_data = pd.DataFrame({"Low Temp":mean_low})
          high_data = pd.DataFrame({"High Temp":mean_high})

In [111]: x_axis = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40]
          hd = high_data["High Temp"]
          ld = low_data["Low Temp"]

          plt.figure(figsize=(10,5))
          plt.plot(x_axis, hd)
          plt.plot(x_axis, ld)

          #titles, etc
          plt.title("Average Monthly Temperature for 2015-2018")
          plt.ylabel("Temperature in °F")
          plt.text(1, -3, '2015', fontsize=10, color='blue')
          plt.text(10, -3, '2016', fontsize=10, color='blue')
          plt.text(21, -3, '2017', fontsize=10, color='blue')
          plt.text(34, -3, '2018', fontsize=10, color='blue')
          plt.text(15, -10, 'Months', fontsize=10)

          plt.ylim((min(ld)-5), (max(hd)+30))

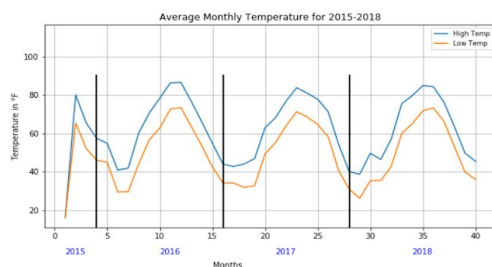
          #legend
          lgnd = plt.legend(loc="best", fontsize="small", frameon=True, markerscale=0.65)

          plt.grid(True, alpha=1, clip_on=True)

          #lines
          plt.plot([4, 4], [-1, 90], color='k', linestyle='-', linewidth=2)
          plt.plot([16, 16], [-1, 90], color='k', linestyle='-', linewidth=2)
          plt.plot([28, 28], [-1, 90], color='k', linestyle='-', linewidth=2)

```

## IV) Result



## Average Price per Month for “Top 30 Zipcodes:

### I) Raw data used to leverage

Contains all data for all zip codes per year. (From main folder “PyBnB\_project1”)

2015: “./Master Data 2015.csv”

2016: “./Listings\_2016/2016\_Listings.csv”

2017: “./Listings\_2017/Merged\_2017.csv”

2018: “./2018\_Listings.csv”

### II) Data Cleansing

Steps:

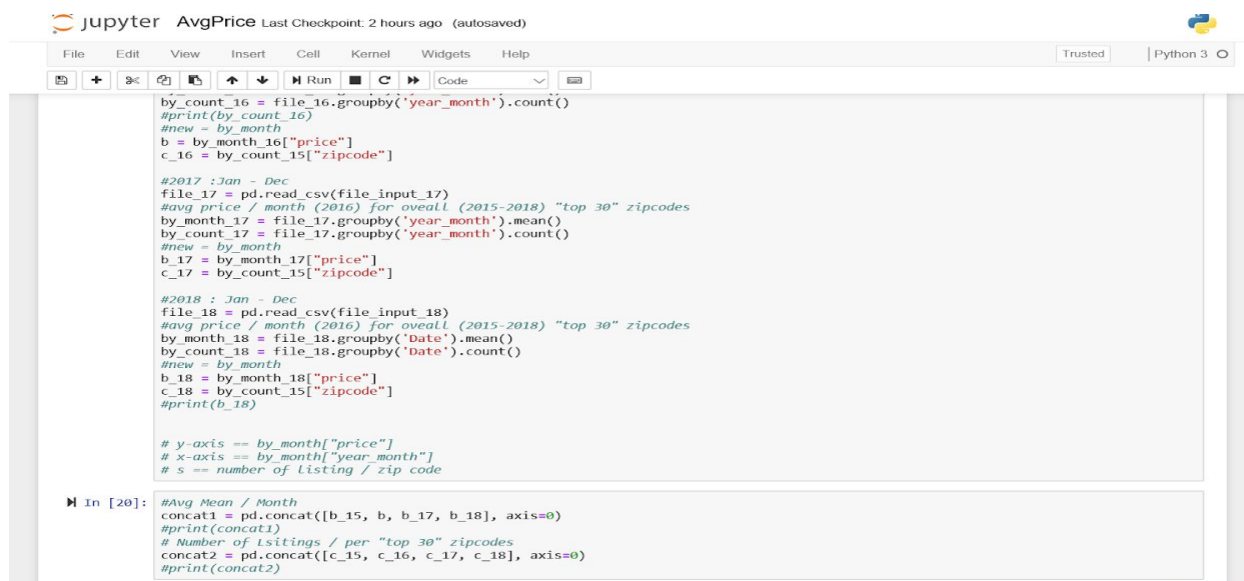
1. Cleansed the data manually using excel:

2. Using the file “consolidated30zips.csv”, containing the over all “top 30 zips for years “2015-2018,” manipulated the data using vlookup and obtain only data from top “30 zips” per month in file.
3. Saved in new file below:
  - a) 2015: file\_input\_15 = './topzips\_raw\_2015.csv'  
From May-Dec, missing July data from Airb&b source.
  - b) 2016: file\_input\_16 = './topzips\_raw\_2016.csv'  
From Jan - Dec, missing March from Airb&b source.
  - c) 2017: file\_input\_17 = './topzips\_raw\_2017.csv'  
From Jan - Dec
  - d) 2018: file\_input\_18 = './topzips\_raw\_2018.csv'  
From Jan - Dec

### III) Data Manupulation saved as “./Listings\_2016/Project\_part\_Weather/MJ/Avg Price.ipynb”

#### Steps:

1. Grouped each csv file by “year\_month” and pulled the mean in order to plot avg month means per month per top zipcodes and count in order to determine the amount of listings per month. Then, merged all data in a series format for plotting.



```

jupyter AvgPrice Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

by_count_16 = file_16.groupby('year_month').count()
#print(by_count_16)
#new = by_month
b = by_month_16["price"]
c_16 = by_count_15["zipcode"]

#2017 :Jan - Dec
file_17 = pd.read_csv(file_input_17)
#avg price / month (2016) for overall (2015-2018) "top 30" zipcodes
by_month_17 = file_17.groupby('year_month').mean()
by_count_17 = file_17.groupby('year_month').count()
#new = by_month
b_17 = by_month_17["price"]
c_17 = by_count_15["zipcode"]

#2018 : Jan - Dec
file_18 = pd.read_csv(file_input_18)
#avg price / month (2016) for overall (2015-2018) "top 30" zipcodes
by_month_18 = file_18.groupby('Date').mean()
by_count_18 = file_18.groupby('Date').count()
#new = by_month
b_18 = by_month_18["price"]
c_18 = by_count_15["zipcode"]
#print(b_18)

# y-axis == by_month["price"]
# x-axis == by_month["year_month"]
# s == number of listing / zip code

In [20]: #Avg Mean / Month
concat1 = pd.concat([b_15, b, b_17, b_18], axis=0)
#print(concat1)
# Number of listings / per "top 30" zipcodes
concat2 = pd.concat([c_15, c_16, c_17, c_18], axis=0)
#print(concat2)

```

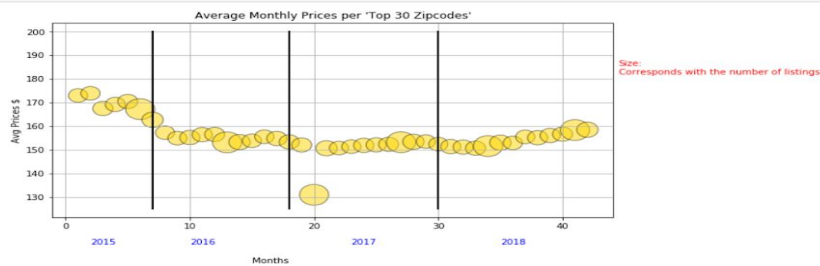
### IV) Final Output

#### Steps:

1. Plotted the Average Price / Night over the 41 Months (2015-2018), circle size was pulled using the count for listings.

- I also used coding to format the text of years and separate them with vertical lines for better visualization.

```
localhost:8892/notebooks/MJ/AvgPrice.ipynb
jupyter AvgPrice Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python
In [21]: x_axis = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40, 41, 42]
plt.figure(figsize=(10,5))
plt.scatter(x_axis, concat, s=concat**0.02, alpha=.5, label = "urban", edgecolor='black', color='gold')
#plt.plot(x_axis,)
plt.title("2016")
plt.title("Average Monthly Prices per 'Top 30 Zipcodes'")
plt.ylabel("Avg Prices $")
#plt.legend(loc="best", title="City Types", fontsize="small", frameon=True, markerscale=0.65)
plt.text(44.5, 182, 'Size: %corresponds with the number of listings', fontsize=10, color='red')
plt.text(2, 110, '2015', fontsize=10, color='blue')
plt.text(10, 110, '2016', fontsize=10, color='blue')
plt.text(23, 110, '2017', fontsize=10, color='blue')
plt.text(35, 110, '2018', fontsize=10, color='blue')
plt.text(15, 102, 'Months', fontsize=10)
#legend
#plt.legend(title="a",loc="best", fontsize="small", frameon=True, markerscale=0.65)
plt.grid(True, alpha=1, clip_on=True)
#lines
plt.plot([7, 7], [125, 200], color='k', linestyle='-', linewidth=2)
plt.plot([18, 18], [125, 200], color='k', linestyle='-', linewidth=2)
plt.plot([30, 30], [125, 200], color='k', linestyle='-', linewidth=2)
fig_size = plt.rcParams["figure.figsize"]
abc = plt.show()
```



In [ ]:

## Question 2:

Does the amount of crime affect prices?

Please refer to [Git Repo: Crime vs Price](#)

## DATA ANALYSIS PROCESS

- I. Getting the number of crime reports (violations, felonies, misdemeanors) by precinct and year:
  - A. Jupyter Notebook for [Crime Count by Precinct and Year](#)
  - B. Data Source:
   
<https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i>
    1. API Documentation:
   
<https://dev.socrata.com/foundry/data.cityofnewyork.us/9s4h-37hy>
  - C. Precincts were chosen based on the top 30 zip codes
    1. Issue: lacking csv data breakdown by zip code for each precinct
    2. For the sake of demonstrating the code, the following assumption was taken: each zip code is representative of the crime level in that precinct
  - D. Due to the inordinate amount of time that would have been needed to look up crime count for every precinct, ten randomly chosen precincts were selected for querying
  - E. Lists of crime counts per year generated for further analysis
- II. Analyzing crime count by precinct-zip code vs mean AirBnB mean prices
  - A. Jupyter Notebook for [Crime vs Price \(+ visualizations\)](#)
  - B. Create dataframe with string formatted columns for precinct and corresponding zip codes
  - C. Concatenate precinct-zip code identified (e.g. 83-11221) to serve as list for x-axis in visualizations

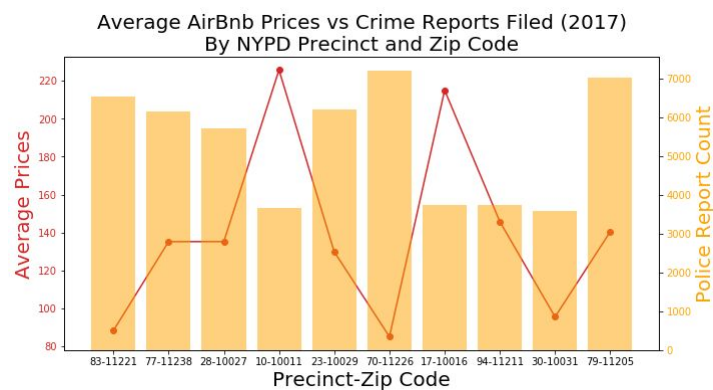
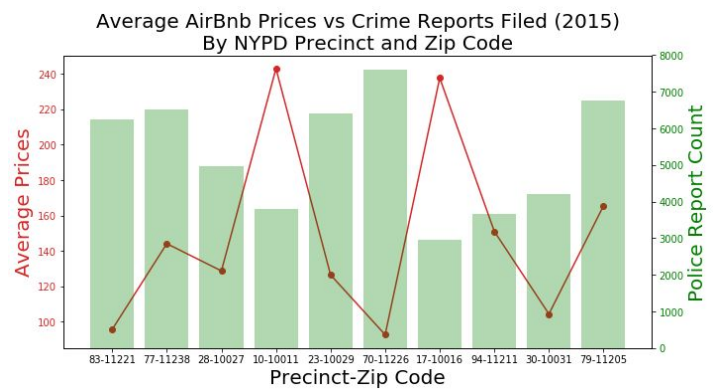
Zip Codes	Mean_2016	Median_2016	Mean_2018	Median_2018	Mean_2017	Median_2017	Mean_2018	Median_2018	Precinots	Precinot-Zip	
83	11221	95.713076	75	89.587325	70	88.334267	69	87.629931	68	83	83-11221
77	11238	144.082211	120	138.259867	106	135.210127	100	137.844509	105	77	77-11238
28	10027	128.709048	110	121.07835	100	135.225297	100	135.70751	100	28	28-10027
10	10011	242.854346	200	225.796276	199	225.691729	195	235.382632	190	10	10-10011
23	10029	126.46144	100	127.42655	100	129.885366	98	140.572068	95	23	23-10029
70	11226	92.680582	75	84.75419	70	85.103783	69	85.033461	69	70	70-11226
17	10016	237.572914	185	214.245995	175	215.009782	170	219.999161	175	17	17-10016
94	11211	150.991682	129	143.768193	119	145.759296	113	146.840053	110	94	94-11211
30	10031	104.395633	80	99.883374	75	95.558082	75	98.003266	75	30	30-10031
79	11205	165.375414	115	154.145494	100	140.440341	99	140.647674	99	79	79-11205

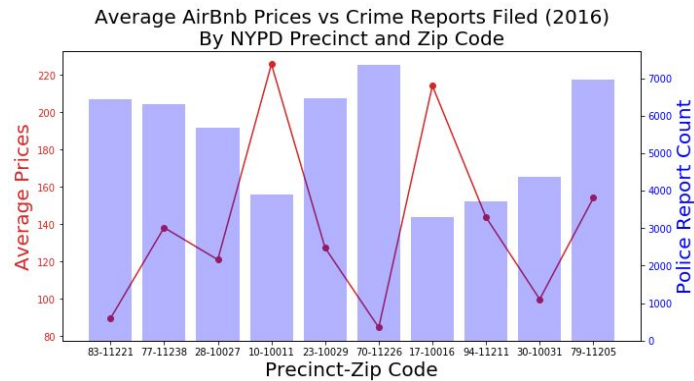
- D.
- E. Create dataframe for year over year total crime count by precinct (below)

	Precinct	Report Count_2015	Report Count_2016	Report Count_2017	Zip Codes	Precinct-Zip
0	83	6236	6442	6538	11221	83-11221
1	77	6513	6310	6157	11238	77-11238
2	28	4956	5674	5710	10027	28-10027

<b>3</b>	10	3786	3907	3670	10011	10-10011
<b>4</b>	23	6403	6469	6207	10029	23-10029
<b>5</b>	70	7615	7356	7198	11226	70-11226
<b>6</b>	17	2961	3290	3729	10016	17-10016
<b>7</b>	94	3658	3719	3740	11211	94-11211
<b>8</b>	30	4204	4381	3580	10031	30-10031
<b>9</b>	79	6768	6950	7021	11205	79-11205

F. See below visualizations for combined data.





### FINDINGS:

All three visualizations show nearly identical trends in that zip codes belonging to precincts with higher counts of police reports filed corresponded with lower average Airbnb prices for the respective zip code. However, given that this analysis does not capture Airbnb prices for all zip codes belonging to a given precinct, the findings are not conclusive. Nevertheless, there is a clear relationship between number of crime reports filed and Airbnb prices. Another interesting findings for precinct-zip code pairs 30-10031 (Washington Heights) and 79-11205 (Northwest Brooklyn) which did not follow the expected trend, which suggests the contrary to the crime versus average price expected trend or that crime is a determining factor in these two areas:

- 30-10031 (*Washington Heights*): Crime reports filed are comparable with 10-10011 (Chelsea and Clinton) yet corresponding average prices not comparable
- 79-11205 (Northwest Brooklyn): Crime reports filed are comparably high yet average prices are not as low as other precinct-zipcodes with comparable number of crime reports filed

### Question 3:

Does median household income affect prices?

Please refer to [Git Repo: Household Income](#)

1. Import the Census API/Python Module to extract the acs5 Household Income metrics:



```

# Import Packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import requests
from census import Census

api_key = "9c43c55c9067920bde793b5c4d99ae0aba10281"
c = Census(api_key, year=2017)

# Run Census Search to retrieve data on selected zip codes
census_data = c.acs5.get(("NAME", "B19013_001E", "B01003_001E", "B01002_001E", "B19301_001E"),
                        {'for': 'zip code tabulation area:*'})

# Convert to DataFrame
census_pd = pd.DataFrame(census_data)

# Column Naming
census_pd = census_pd.rename(columns={"B01003_001E": "Population",
                                     "B01002_001E": "Median Age",
                                     "B19013_001E": "Household Income",
                                     "B19301_001E": "Per Capita Income",
                                     "NAME": "Name", "zip code tabulation area": "Zipcode"})

# Final DataFrame
census_pd = census_pd[["Zipcode", "Population", "Median Age", "Household Income",
                      "Per Capita Income"]]

print(len(census_pd))
census_pd.head()

```

2. Leverage previously built dataframe of top 30 zip codes then find those selected zip codes in the Census data and create a new dataframe
  - Issues: Not being able to find zip codes in census data at first because needed to change datatype to string
  - Once issues were resolved, merged together the AirBnB dataframe and Census Data on zip codes

```

# Import Final zip codes data
zipcode_30 = pd.read_csv("../Final_Trending_Zipcodes.csv")

# Convert Zip Codes to str in order to extract from Census Data
zipcode_30['Zip Codes'] = zipcode_30['Zip Codes'].astype(str)
zipcode_30.dtypes

...

# Create List of 30 codes that need to be searched
zips = zipcode_30.groupby('Zip Codes').count()
zips = zips.sort_values('Borough', ascending=False)

zip_list = list(zips.index.values)
zipcodes = zip_list[0:30]

# Extract zip codes data from Census dataframe
census_pd2 = census_pd[census_pd["Zipcode"].isin(zipcodes)]

# Merge Census data with trended data
zipcode_30 = zipcode_30.rename(columns={'Zip Codes': 'Zipcode'})
census_pd2['Zipcode'] = census_pd2['Zipcode'].astype(str)
zipsmerged = pd.merge(census_pd2, zipcode_30, on="Zipcode", how="outer")

# Add column totaling Medians and Means for all 4 years
zipsmerged['4year_Mean'] = (zipsmerged['Mean_2015'] + zipsmerged['Mean_2016'] + zipsmerged['Mean_2017'] + zipsmerged['Mean_2018'])
zipsmerged['4year_Median'] = (zipsmerged['Median_2015'] + zipsmerged['Median_2016'] + zipsmerged['Median_2017'] + zipsmerged['Median_2018'])

zipsmerged

```

3. Converted data to be grouped-by neighborhood and converted all data to be integers in order to graph properly
  - Calculated the 4 year mean of AirBnB prices in order to have a larger data pool

```
zipsmerged.to_csv('ZipsCensus_Merged.csv', index=False)
```

```
# Build DF of all zip codes & total mean
totals = zipsmerged[['Zipcode', 'Neighborhood', 'Household Income', '4year_Mean']]

# Convert values to int
totals['Zipcode'] = totals['Zipcode'].astype(int)
totals['4year_Mean'] = totals['4year_Mean'].astype(int)
totals['Household Income'] = totals['Household Income'].astype(int)

# Sort Zipcodes
means = totals.sort_values('4year_Mean', ascending=True)
means
```

Neighborhood	Household Income	4year_Mean
Flatbush	51926	72
Central Brooklyn	36009	76
Bushwick and Williamsburg	51269	84
Flatbush	48495	86
Bushwick and Williamsburg	47170	90
Bushwick and Williamsburg	34122	97
Inwood and Washington Heights	44040	99
Central Brooklyn	56025	109
Northwest Brooklyn	116446	130

4. Built a bar graph and scatter plot to analyze if there was a correlation between Airbnb prices and Household Income:

```

# Create analysis by Neighborhoods
neighborhoods = pd.read_csv('Neighborhood_ZipMeans.csv')

# Plot Neighborhood Night Prices & Add in Household Income as a text box
plt.figure(figsize = (15,10));
x_axis = np.arange(len(neighborhoods['Neighborhood']));
tick_locations = [x for x in x_axis];
plt.xticks(tick_locations, neighborhoods['Neighborhood'], rotation=60, fontsize=10, horizontalalignment='right');
plt.bar(x_axis, neighborhoods['4year_Mean'], label = 'Airbnb Nightly Price_4 Year Mean',width = 0.5, color='lightcoral');
plt.legend(loc='upper left', fontsize=11);
plt.xlabel('Neighborhoods', fontsize=12, fontweight='bold');
plt.ylabel('4-Year Mean', fontsize=12, fontweight='bold');
plt.title('AirBnB Nightly Prices by Neighborhood', fontsize=15, fontweight='bold');

# Add unique text label for Household Income
plt.text(-.4, 88, '$50,211', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(0.65, 98, '$44,187', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(1.65, 105, '$44,040', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(2.65, 115, '$57,016', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(3.65, 136, '$45,536', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(4.65, 137, '$34,015', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(5.65, 145, '$72,200', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(6.65, 154, '$96,000', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(7.58, 187, '$110,803', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(8.65, 194, '$66,804', fontsize=10, bbox={'facecolor': 'lightyellow'});
plt.text(9.55, 228, '$109,250', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(10.58, 229, '$105,326', fontsize=10, bbox={'facecolor': 'lightgreen'});
plt.text(11.58, 245, '$116,267', fontsize=10, bbox={'facecolor': 'lightgreen'});

# Add key for text boxes
plt.text(12.97, 209, 'Key:', fontsize=13.5);
plt.text(13.05, 200, 'xx', color='lightgreen', fontsize=11, bbox={'facecolor': 'lightgreen'});
plt.text(13.5, 200, 'Household Income Avrg', fontsize=11.5, bbox={'facecolor': 'w'});
plt.text(13.05, 188, 'xx', color='lightyellow', fontsize=11, bbox={'facecolor': 'lightyellow'});
plt.text(13.5, 188, 'Outlier Trend', fontsize=11.5, bbox={'facecolor': 'w'});

# Save fig
plt.savefig("PriceNeighborhood_BarGraph.png")

```



```

from matplotlib.lines import Line2D

# Deep dive into Lower East Side
gvs = neighborhoods.loc[neighborhoods["Neighborhood"] == "Greenwich Village and Soho"]
cc = neighborhoods.loc[neighborhoods["Neighborhood"] == "Chelsea and Clinton"]
gpmh = neighborhoods.loc[neighborhoods["Neighborhood"] == "Gramercy Park and Murray Hill"]
les = neighborhoods.loc[neighborhoods["Neighborhood"] == "Lower East Side"]
uws = neighborhoods.loc[neighborhoods["Neighborhood"] == "Upper West Side"]

# Build the scatter plots for top 5 nightly prices
plt.scatter(gvs["4year_Mean"], gvs["Income_Mean"],
            s=gvs["Zipcode_Count"] * 100.5, alpha=.7, color = 'lightcoral', edgecolors='black');

plt.scatter(cc["4year_Mean"], cc["Income_Mean"],
            s=cc["Zipcode_Count"] * 100.5, alpha=.7, color = 'lightblue', edgecolors='black');

plt.scatter(gpmh["4year_Mean"], gpmh["Income_Mean"],
            s=gpmh["Zipcode_Count"] * 100.5, alpha=.7, color = 'lightgreen', edgecolors='black');

plt.scatter(les["4year_Mean"], les["Income_Mean"],
            s=les["Zipcode_Count"] * 100.5, alpha=.7, color = 'lightyellow', edgecolors='black');

plt.scatter(uws["4year_Mean"], uws["Income_Mean"],
            s=uws["Zipcode_Count"] * 100.5, alpha=.7, color = 'lightgrey', edgecolors='black');

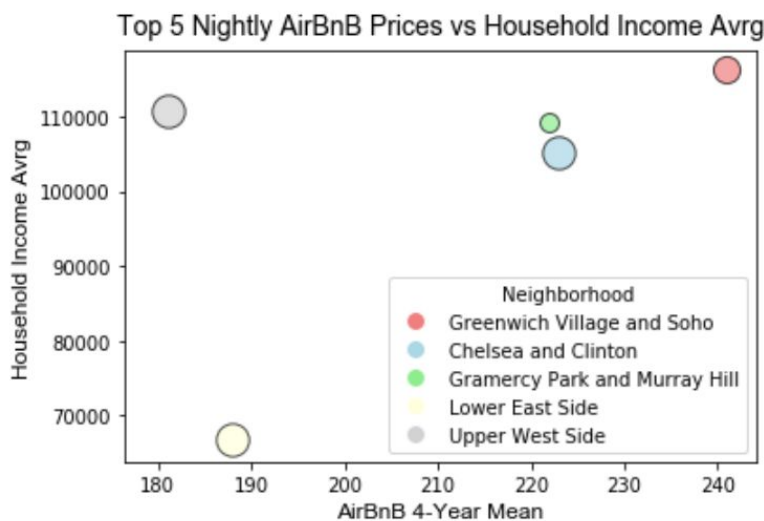
# Incorporate the other graph properties
title_font = {'fontname':'Arial', 'size':14, 'color':'black', 'weight':'normal',
              'verticalalignment':'bottom'}
axis_font = {'fontname':'Arial', 'size':12}

plt.xlabel('AirBnB 4-Year Mean', **axis_font);
plt.ylabel('Household Income Avrg', **axis_font);
plt.title('Top 5 Nightly AirBnB Prices vs Household Income Avrg', **title_font);

# Create a Legend
line1 = Line2D(range(1), range(1), color="white", marker='o', markersize=10, markerfacecolor="lightcoral")
line2 = Line2D(range(1), range(1), color="white", marker='o', markersize=10, markerfacecolor="lightblue")
line3 = Line2D(range(1), range(1), color="white", marker='o', markersize=10, markerfacecolor="lightgreen")
line4 = Line2D(range(1), range(1), color="white", marker='o', markersize=10, markerfacecolor="lightyellow")
line5 = Line2D(range(1), range(1), color="white", marker='o', markersize=10, markerfacecolor="lightgrey")
plt.legend((line1,line2,line3,line4,line5),('Greenwich Village and Soho',
                                           'Chelsea and Clinton', 'Gramercy Park and Murray Hill',
                                           'Lower East Side', 'Upper West Side'),
           numpoints=1, loc='best', title='Neighborhood', frameon=True);

# Add analysis to the right
plt.text(247, 100000, 'Outlier Trend:\nAlthough LES is among the Top 5 highest nightly rates, \nthe Househol
         fontsize=10, bbox={'facecolor': 'w'});

```



### **Final Conclusions:**

1. Household Income and AirBnB prices have a pattern:
  - a. Lower Household Income = Lower AirBnB Prices
2. The Lower East Side was one outlier in the data:
  - a. Although LES is among the Top 5 highest nightly rates, the Household Income average was \$34k below the Top 5 Neighborhoods Avg
    - i. LES is a trending neighborhood with a younger demographic which reflects why the income is so low. LES is an extremely popular neighborhood in NYC with mass amounts of bars and restaurants. Based on the neighborhoods popularity, it makes sense that the AirBnB nightly rate is so high while the HH Income is lower.

### **Question 4:**

Does demographics (i.e. ethnicity & age) affect prices?

Please refer to my [Demographics Visualizations](#) jupyter notebook

Data sources:

- US Census Bureau
  - [American Community Survey](#)
    - Yearly survey conducted by US Census to appropriately allocate funding based on demographic information
    - Chose data from Neighborhood Tabulation Areas (NTA's) to find information on specific zip codes/neighborhoods



Italiano

Translate

Text-Size

Home

About

Zoning

Applications

Projects

Communities

Data/Maps

Search

NYC Population

Economy & Housing

Open Data

Maps & Geography

Decennial Census

American Community Survey

Population FactFinder

Current & Projected Populations

Visualizing NYC

Reports & Presentations

Tools and Geographic Reference

Data Background and Archive

American Community Survey (ACS)

The American Community Survey (ACS) is the most extensive nationwide survey currently available. From its annual releases we are able to examine the city's detailed demographic, socioeconomic, and housing characteristics. [View About the ACS](#). [View How to Use the ACS](#).

Maps

Profiles

Year ending

2016

- Please select from dropdown list for other years.

Topic/ Year(s)	NYC & Boroughs	PUMA (Community Districts)	Neighborhood Tabulation Areas
Demographic			
2016			
2012-13-14-15-16			
Social			
2016			
2012-13-14-15-16			
Economic			
2016			

Facebook

Twitter

Google+

LinkedIn

Email

Share

Print

## Data Cleansing and Analysis:

1. Find top three most expensive neighborhoods vs bottom three least expensive neighborhoods
  - a. Imported neighborhood data
    - i. Eliminated median prices, only want to look at mean prices
    - ii. Eliminated rows with \$0 in their prices for the year to avoid any results from being skewed

```

In [3]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

price_data_file = "Demographic Data/Neighborhood_Data.csv"
price_df = pd.read_csv(price_data_file)
price_df

#remove Median columns and rows that have 0 in any of the Means
price_df = price_df.drop([25,26,27,28], axis=0)
del price_df['Median 2015'], price_df['Median 2016'], price_df['Median 2017'], price_df['Median 2018']
price_df

Out[3]:
  Zip Codes  Borough  Neighborhood  Mean 2015  Mean 2016  Mean 2017  Mean 2018
0    11206  Brooklyn  Bushwick and Williamsburg  102.391970  94.994238  94.821426  96.825835

```

- b. Then grouped and set index by neighborhood
  - i. Found mean for each neighborhood
  - ii. Transposed the columns and rows for future plotting

```
In [7]: #Find Mean price for each year and put into new dataframe
mean_2015 = price_df.groupby("Neighborhood")['Mean 2015'].mean()
mean_2016 = price_df.groupby("Neighborhood")['Mean 2016'].mean()
mean_2017 = price_df.groupby("Neighborhood")['Mean 2017'].mean()
mean_2018 = price_df.groupby("Neighborhood")['Mean 2018'].mean()

neighborhood_price = pd.DataFrame({
    'Mean 2015': mean_2015,
    'Mean 2016': mean_2016,
    'Mean 2017': mean_2017,
    'Mean 2018': mean_2018,
})

#Transpose columns and rows to prepare data for plotting
neighborhood_trans = neighborhood_price.T
neighborhood_trans
```

Out[7]:

Neighborhood	Bushwick and Williamsburg	Central Brooklyn	Central Harlem	Chelsea and Clinton	East Harlem	Flatbush	Gramercy Park and Murray Hill	Greenpoint	Greenwich Village and Soho	Inwood and Washington Heights	Lower East Side	Northw Brook
Mean 2015	96.673829	127.991209	128.709048	230.272173	126.461440	92.680582	237.572914	145.712109	253.860292	104.395633	193.502639	164.761
Mean 2016	90.232181	124.141194	121.078350	217.793999	127.426550	84.754190	214.245595	136.195679	234.945388	99.883374	184.260796	156.174
Mean 2017	87.961485	121.284081	135.229297	218.227701	129.885366	85.103783	215.009782	134.947242	234.414585	95.558082	187.059027	149.772
Mean 2018	87.891787	124.240472	135.707509	226.652251	140.572068	85.033461	219.999161	141.255819	241.926452	98.003266	188.286289	149.977

- c. Narrowed data down to top and bottom three neighborhoods:

- i. Top Three:
  1. Greenwich Village and Soho
  2. Gramercy Park and Murray Hill
  3. Chelsea and Clinton
- ii. Bottom Three:
  1. Bushwick and Williamsburg
  2. Flatbush
  3. Inwood and Washington Heights

- d. Plotted on a line chart:

## 2. Analysis of Age Demographics

- a. Initially the age demographic data was given in age categories of 5 years, giving me almost 20 different potential parts of a pie chart
  - i. Consolidated age categories into four categories (<19, 20-39, 40-64, >65)

```
#separate data into demographic types ie. age and ethnicity and create respective visualizations
age_df = demo_df[[
    'Neighborhood',
    '<5', '5-9', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40-44', '45-49',
    '50-54', '55-59', '60-64', '65-69', '70-74', '75-79', '80-84', '>85'
]]
age_df = age_df.set_index('Neighborhood')

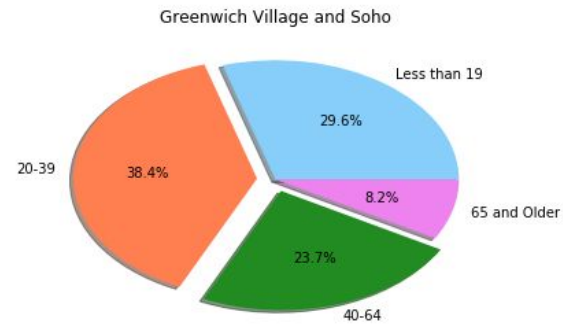
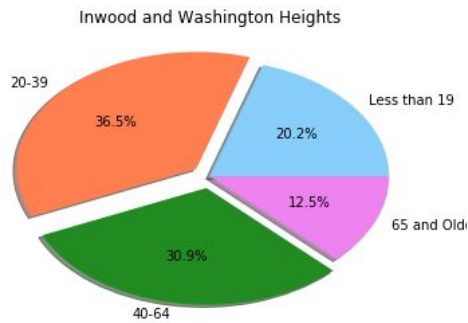
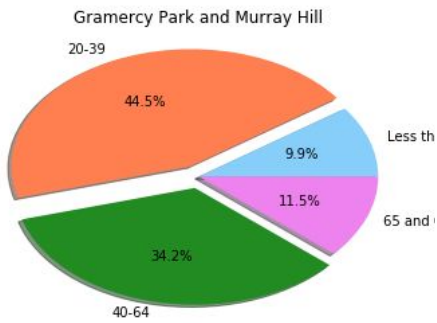
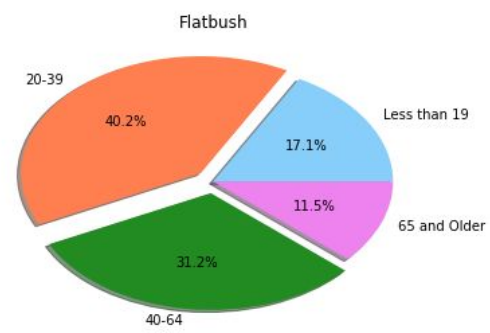
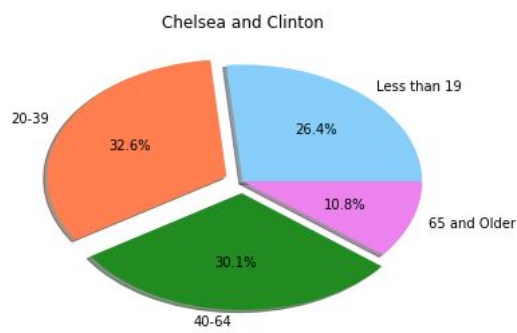
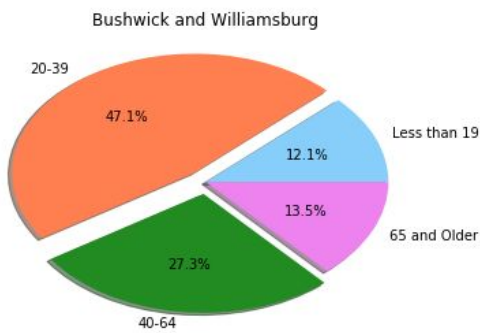
#create more specific age categories
less19 = []
between20_39 = []
between40_64 = []
older65 = []
neighborhoods = ['Greenwich Village and Soho', 'Gramercy Park and Murray Hill', 'Chelsea and Clinton',
    'Bushwick and Williamsburg', 'Flatbush', 'Inwood and Washington Heights']

for index, row in age_df.iterrows():
    less19 = row['<5'] + row['5-9'] + row['10-14'] + row['15-19']
    between20_39 = row['20-24'] + row['25-29'] + row['30-34'] + row['35-39']
    between40_64 = row['40-44'] + row['45-49'] + row['50-54'] + row['55-59'] + row['60-64']
    older65 = row['65-69'] + row['70-74'] + row['75-79'] + row['80-84'] + row['>85']
```

b. Created pie charts for each neighborhood

```
#plot age demographic pie charts
#plot Greenwich Village and Soho pie chart
compiled_ages_trans = compiled_ages.set_index('Neighborhood').T
compiled_ages_trans
labels = ['Less than 19', '20-39', '40-64', '65 and Older']
sizes = compiled_ages_trans['Greenwich Village and Soho']
colors = ['lightskyblue', 'coral', 'forestgreen', 'violet']
explode = [0,0.1,0.1,0]

plt.pie(sizes, explode = explode, labels = labels, colors=colors,autopct="%1.1f%%", shadow = True)
plt.title('Greenwich Village and Soho')
plt.savefig("Images/GreenwichSohoAge")
```





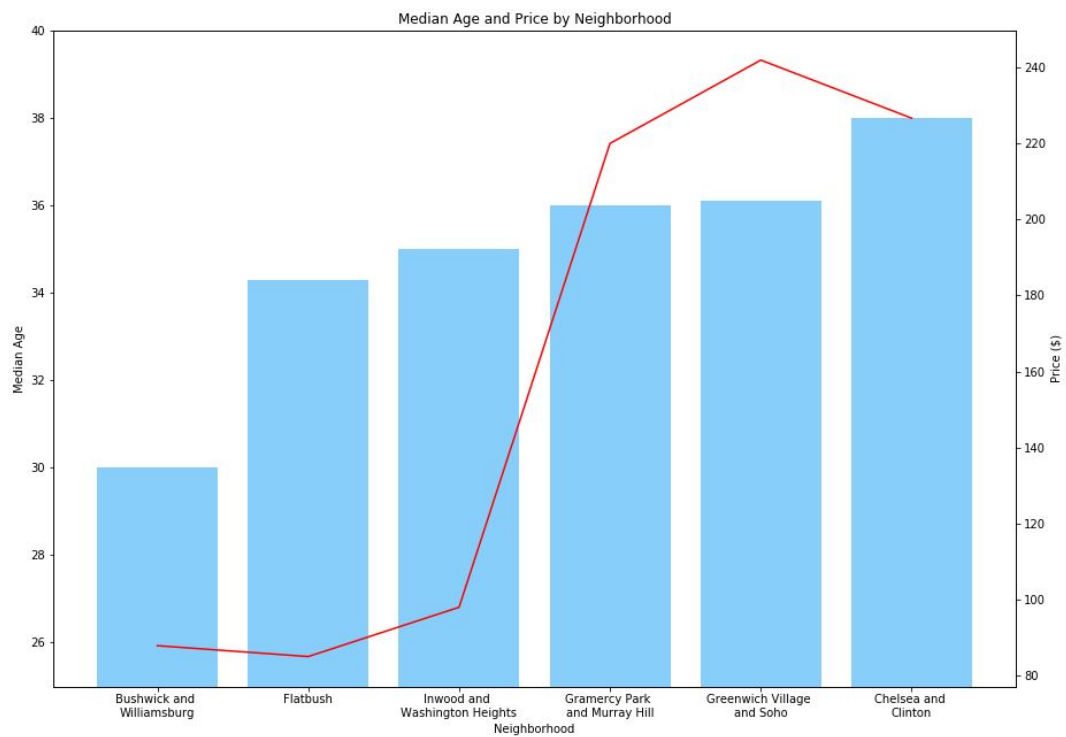
- c. Consolidated median ages into dataframe and plotted ages on a graph along with mean prices of Airbnbs by neighborhood

```
median_age = demo_df[[
    'Neighborhood',
    'Borough',
    'Total Population',
    'Median Age'
]]

#reorder rows to match means
median_age = median_age.reindex([0,5,9,6,8,3])

#reorder original filtered neighborhood dataframe to fit plotting
reordered_means = neighborhood_trans[[
    'Bushwick and Williamsburg',
    'Flatbush',
    'Inwood and Washington Heights',
    'Gramercy Park and Murray Hill',
    'Greenwich Village and Soho',
    'Chelsea and Clinton'
]]
reordered_means
```

Neighborhood	Bushwick and Williamsburg	Flatbush	Inwood and Washington Heights	Gramercy Park and Murray Hill	Greenwich Village and Soho	Chelsea and Clinton
Mean 2015	96.673829	92.680582	104.395633	237.572914	253.860292	230.272173
Mean 2016	90.232181	84.754190	99.883374	214.245595	234.945388	217.793999
Mean 2017	87.961485	85.103783	95.558082	215.009782	234.414585	218.227701
Mean 2018	87.891787	85.033461	98.003266	219.999161	241.926452	226.652251



### 3. Race Demographics

- Consolidated demographic data and created pie charts
- Then graphed percentage of white and black/hispanics inhabitants in relation to prices of Airbnbs by neighborhoods

```
ethnic_df = demo_df[[
    'Neighborhood',
    'Hispanic', 'White', 'Black',
    'Native American', 'Asian', 'Hawaiian/PI',
    'Other', 'Two or More'
]]
ethnic_df = ethnic_df.set_index('Neighborhood').T
ethnic_df
```

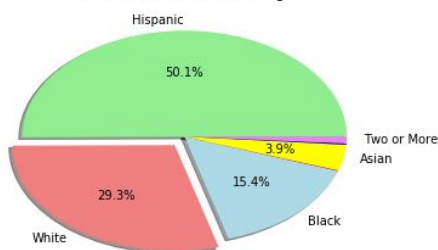
Neighborhood	Bushwick and Williamsburg	Chelsea and Clinton	Flatbush	Gramercy Park and Murray Hill	Greenwich Village and Soho	Inwood and Washington Heights
Hispanic	84947	19426	20589	7819	2839	153384
White	49668	73650	24359	49084	28050	38922
Black	26063	6939	49067	3094	1023	17086
Native American	184	131	61	197	43	63
Asian	6669	18120	11129	13147	9281	5897
Hawaiian/PI	11	12	11	0	0	95
Other	369	709	974	255	125	584
Two or More	1693	2511	2463	2064	1618	2649

```
#plot ethnic demographic pie charts
#Plot Bushwick and Williamsburg
labels = ['Hispanic','White','Black','','Asian','','','Two or More'] #too little data to show every race
sizes = ethnic_df['Bushwick and Williamsburg']
colors = ['lightgreen','lightcoral','lightblue','red','yellow','coral','blue','violet']
explode = [0,0.1,0,0,0,0,0,0]

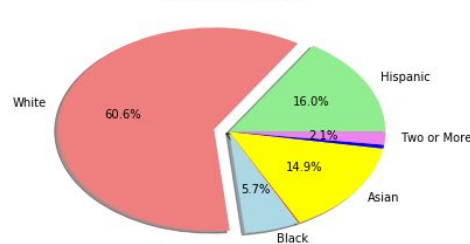
#drop any percentages Less than 2%
def my_autopct(pct):
    return ("%1.1f%%" % pct) if pct > 2 else ''

plt.pie(sizes,colors=colors,labels = labels,explode = explode,autopct=my_autopct, shadow = True)
plt.title('Bushwick and Williamsburg')
plt.savefig("Images/BushWillRace")
```

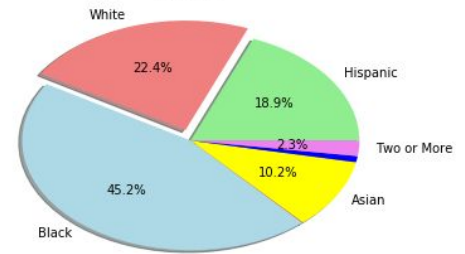
Bushwick and Williamsburg



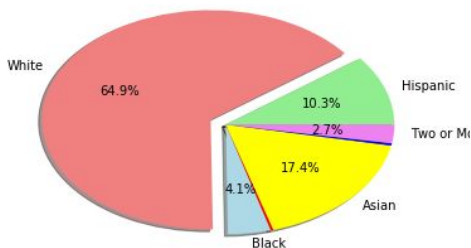
Chelsea and Clinton



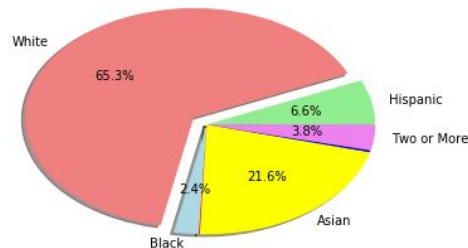
Flatbush



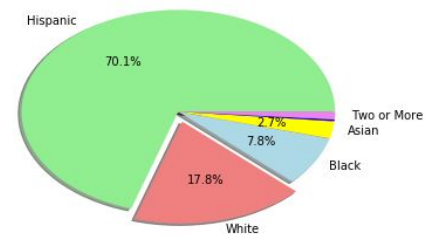
Gramercy Park and Murray Hill

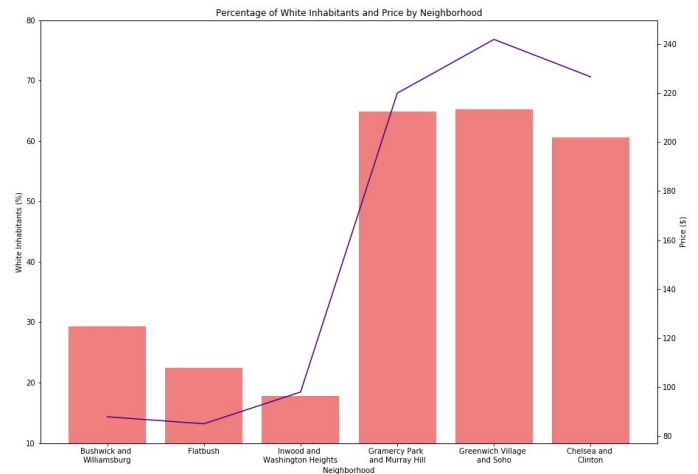
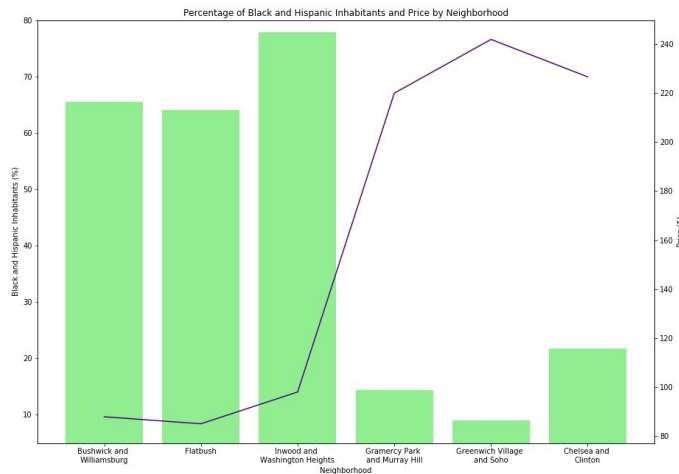


Greenwich Village and Soho



Inwood and Washington Heights





### Findings:

1. The neighborhoods with high Airbnb prices tend to be within close range of tourist attractions and are located in Midtown Manhattan. Those neighborhoods with lower Airbnb prices tend to be much further away either being located in an outer borough like Brooklyn or Queens or in Upper Manhattan.
2. Although age distribution seems to be similar to each other throughout each neighborhood, the median ages by neighborhood slightly correlates with the prices of Airbnbs. This is because those who are older and more well established in their careers are able to afford more expensive real estate and thus can price their Airbnbs higher.
3. Neighborhoods with higher Airbnb prices tend to have higher rates of White Inhabitants while those with lower Airbnb prices tend to have higher rates of Black and Hispanic Inhabitants.



