# Correlation Analysis Between Airbnb Prices & Subway Distance

Gangyi Li
12/04/2025

# Issues and Objectives

- ***Core issue:***.Are Airbnb prices really higher when you are closer to the subway?

- ***Code:*** Multi-library collaboration (pandas/geopandas/sklearn/folium)

- ***Process:*** data → modeling → visualization

- ***Outcome:*** Python implementation for spatial data processing and regression analysis.

# Data Source

## Airbnb Listing Data:

***Content:*** 20,000+ NYC records in CSV/GZ format

***Python Implementation:*** **pd.read_csv()** with native compression support

## Subway Station Data:

***Content:*** Latitude, Longitude, Station Name

***Python Implementation:*** Custom **read_subway()** function for automatic column name detection

## Neighborhood Income Data:

***Content:*** Sourced from Census API (Population, Median Income)

***Python Implementation:*** Data fetched via **requests** and parsed into a **pd.DataFrame()**

# Analysis and Models

## Data Cleaning:

- *Remove Missing Values*
- *Standardize Parameters*
- *Spatial Format Conversion*

## Spatial Analysis:

*Use KMeans to cluster properties with similar geographic locations into 5 categories*

## Regression Modeling:

- *Why np.log1p(y)? Reduce impact of $1000+ luxury listings outliers*
- *Outputs: Coefficients (impact strength) + $R^2$ (model fit)*

# Output files

## airbnb_processed.csv:

***Content:*** Cleaned data (including distance, clustering, and community attributes)

***Python Implementation:*** Save using **df.to_csv()** with new added features.

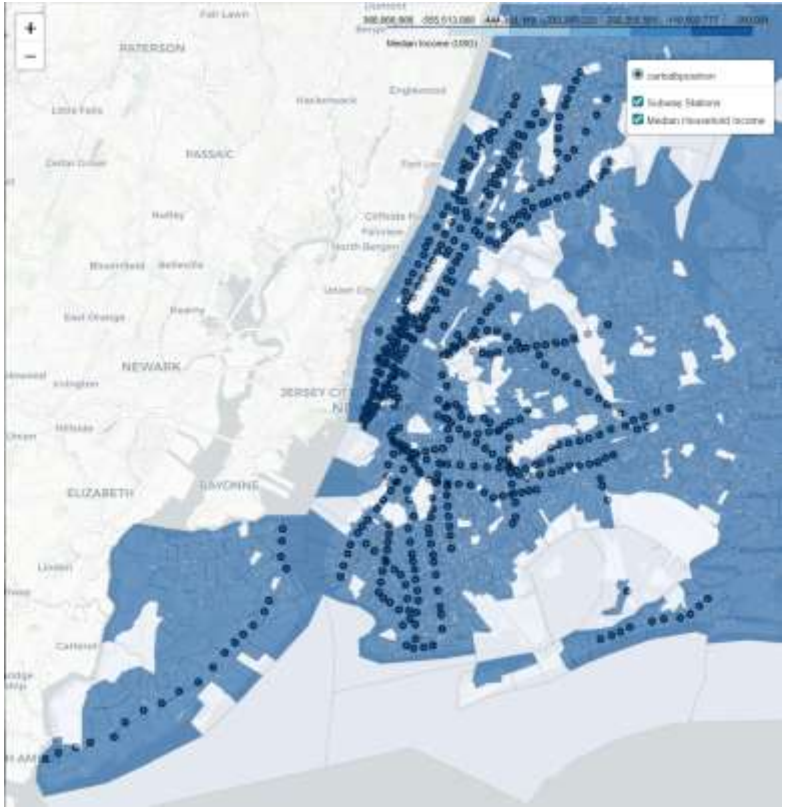
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | id | name | host_id | neighbour | latitude | longitude | price | room_type | number_of | index_righ | GEOID | median_h | population | dist_to_sta | nearest_st | cluster |
| 2 | 40824219 | Room clo | 3.18E+08 | Neighborh | 40.74698 | -73.9176 | 66 | Private roo | 16 | 274 | 3.61E+10 | 93356 | 5387 | 572.4552 | 46 St-Bliss | 0 |
| 3 | 40843080 | Cozy 2 Be | 2.95E+08 | Neighborh | 40.6823 | -73.8455 | 97 | Entire hom | 93 | 11 | 3.61E+10 | 84630 | 2152 | 327.1645 | Rockaway | 3 |
| 4 | 40824301 | Cozy roor | 14890430 | Neighborh | 40.71316 | -73.9431 | 60 | Private roo | 26 | 2594 | 3.6E+10 | 96875 | 2939 | 232.7909 | Graham Av | 1 |
| 5 | 40825740 | House of | 7728754 | Neighborh | 40.67412 | -73.9412 | 425 | Entire hom | 1 | 2387 | 3.6E+10 | 50417 | 4489 | 700.6149 | Kingston A | 1 |
| 6 | 2595 | Skylit Stud | 2845 | Neighborh | 40.75356 | -73.9856 | 240 | Entire hom | 47 | 1160 | 3.61E+10 | 91250 | 122 | 149.6056 | 42 St-Brya | 0 |
| 7 | 6848 | Only 2 stc | 15991 | | 40.70935 | -73.9534 | 96 | Entire hom | 195 | 2624 | 3.6E+10 | 63608 | 5921 | 364.1934 | Hewes St | 1 |
| 8 | 6872 | Uptown S | 16104 | Neighborh | 40.80107 | -73.9426 | 59 | Private roo | 1 | 4473 | 3.61E+10 | 39365 | 7810 | 373.6827 | 116 St | 2 |
| 9 | 6990 | UES Beaut | 16800 | Neighborh | 40.78778 | -73.9476 | 73 | Private roo | 249 | 4460 | 3.61E+10 | 60778 | 6011 | 414.8134 | 103 St | 2 |
| 10 | 7097 | Perfect for | 17571 | Neighborh | 40.69194 | -73.9739 | 216 | Private roo | 423 | 1694 | 3.6E+10 | -6.7E+08 | 0 | 726.8216 | Fulton St | 1 |
| 11 | 40846106 | Comfortal | 3.02E+08 | Neighborh | 40.70652 | -73.9029 | 83 | Private roo | 25 | 384 | 3.61E+10 | 67801 | 5136 | 308.8811 | Forest Av | 1 |
| 12 | 40861640 | Essex Hou | 36793116 | | 40.76561 | -73.9773 | 420 | Entire hom | 12 | 4414 | 3.61E+10 | 165600 | 6352 | 241.4055 | 57 St | 0 |
| 13 | 40869871 | Sonder at | 2.2E+08 | Neighborh | 40.70797 | -74.0068 | 448 | Entire hom | 34 | 1059 | 3.61E+10 | 182348 | 8485 | 214.2973 | Fulton St | 0 |
| 14 | 40873866 | Home awi | 1.43E+08 | Neighborh | 40.65853 | -73.8937 | 99 | Private roo | 54 | 3370 | 3.6E+10 | 58547 | 4303 | 630.1081 | New Lots A | 1 |
| 15 | 40874667 | Full size b | 1.43E+08 | Neighborh | 40.65797 | -73.8929 | 90 | Private roo | 76 | 4951 | 3.6E+10 | 75833 | 4571 | 709.3146 | New Lots A | 1 |
| 16 | 8490 | Maison dk | 25183 | | 40.68456 | -73.9296 | 170 | Entire hom | 189 | 2319 | 3.6E+10 | 104537 | 5303 | 693.875 | Kingston-T | 1 |
| 17 | 9357 | Midtown I | 30193 | Neighborh | 40.76724 | -73.9886 | 175 | Entire hom | 58 | 1060 | 3.61E+10 | 106915 | 9316 | 544.9131 | 59 St-Colu | 0 |
| 18 | 10452 | Radiant O | 35935 | Neighborh | 40.68294 | -73.9568 | 90 | Private roo | 82 | 2247 | 3.6E+10 | 121250 | 4425 | 229.0201 | Franklin Av | 1 |
| 19 | 12937 | 1 Stop to | 50124 | Neighborh | 40.74757 | -73.9457 | 232 | Private roo | 456 | 574 | 3.61E+10 | 122125 | 4131 | 51.83518 | Court Sq-2 | 0 |
| 20 | 12940 | Charming | 50148 | Neighborh | 40.67946 | -73.9642 | 151 | Entire hom | 80 | 3478 | 3.6E+10 | 61250 | 2499 | 248.6577 | Franklin Av | 1 |
| 21 | 14314 | Greenpoir | 56246 | | 40.73535 | -73.9558 | 115 | Entire hom | 176 | 1688 | 3.6E+10 | 146458 | 4778 | 607.4033 | Greenpoin | 0 |

# Output files

## nyc_airbnb_map.html:

***Content:*** Interactive Maps (Property Listings, Subway Lines, Income Heat Maps)

***Python Implementation:*** Built with **folium.Map()**, supporting layer control.

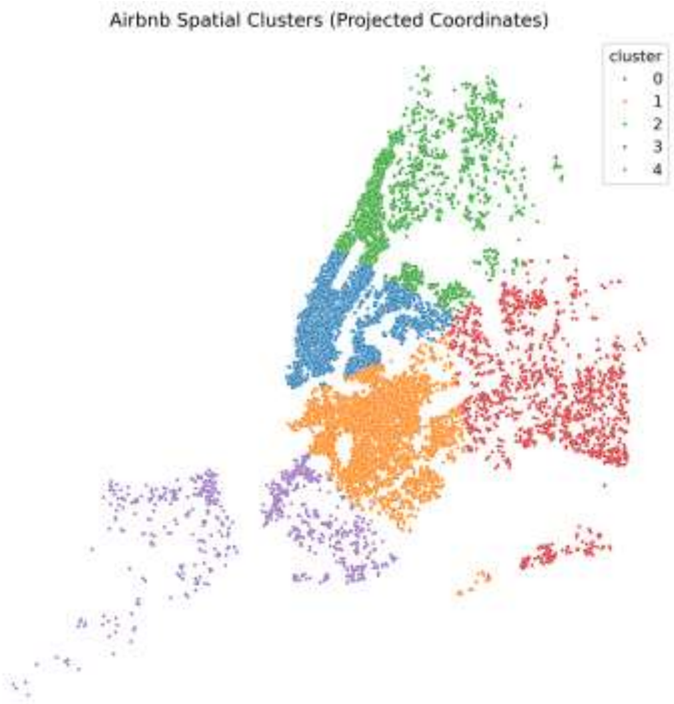


# Output files

## airbnb_clusters.png:

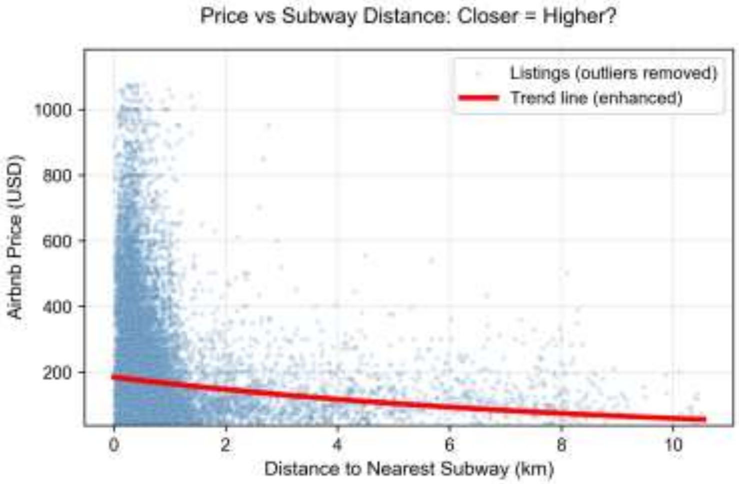
*Content:* Spatial cluster results (5 groups)

*Python Implementation:* Clustered via **sklearn.KMeans**, visualized with seaborn.



# Result

- The spatial autocorrelation index Moran‘s I is 0.035 (p=0.026) : Weak but significant price clustering.
- In the regression model, the coefficient for subway distance is -0.05: 5% price drop per 1km.
- Model fit $R^2 = 0.45$: 45% price variation explained (distance & income)




- The red trendline illustrates the pattern that ‘the farther from the subway, the lower the housing prices,’ controlling for confounding factors such as community income.

# Challenge

## Data Format Inconsistency:

- The data contains numerous mixed string formats (e.g., price information: $100, $200, “350”, etc.).
- def parse_price(x): s = str(x).replace('$', '').replace(',', '').strip()

## Outlier Impact on Regression:

- High-priced properties distort the results.
- y = np.log1p(df_model['price'])