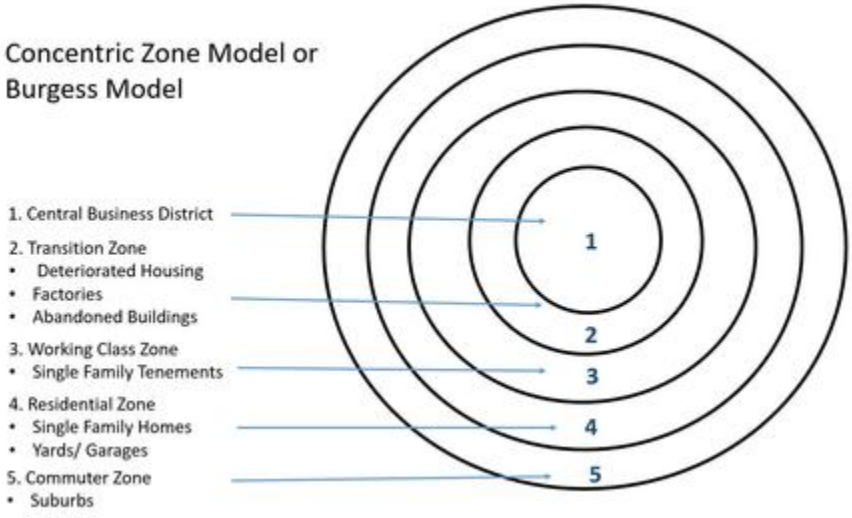
Cagan Abney, Fernanda Valdez, Joshua Hale, Jonathan Unger

# Project 3 - NYC House Prices

## Inspiration

The Burgess concentric city model was one of the earliest models created to explain urban structures. It states that every city has a center, denominated the Central Business District (CBD) and that land usage and therefore social groups are distributed in concentric rings based on the amount people are willing to pay for land.

The model was created in 1925 and was based on the city of Chicago, and generalized as a rule for cities in developed countries, where the CBD is the most accessible area of the city and comprises mainly shops, restaurants and offices. The CBD is surrounded by a transition or industrial zone with factories and some deteriorated housing. The Working class is mainly composed of workers that have moved outside from the degradation zone but still wish to live close to their workplace. Finally the 4th and 5th zone are residential areas with houses getting bigger the furthest one gets from the center.



In modern times, the CBD model has been challenged by contemporary geographers who've proposed alternative models like the sector model or the multi nuclei model, based on more contemporary living allocations and different factors like affinity, commuter villages, gentrification, geographical restrictions, among others. With this project, we want to explore if the city of New York follows the Concentric City Model or if, on the contrary, it appears to have a different layout that would apply to a different city model.

## Data Set

For this project we chose the New York Housing Market dataset on Kaggle: https://www.kaggle.com/datasets/nelgiriyewithana/new-york-housing-market. This dataset included data on homes for sale in the New York City area as of November 2023. For our dataset, we wanted to utilize data we could tell a data driven story with and also included spatial coordinates (latitude and longitude). The spatial coordinates were essential data points to achieve our goal of creating a map visualization. Additional fields included: the home addresses, price, number of beds, number of baths, and the size of each home (square feet).

## Data Cleansing/Manipulation

We began cleaning and manipulating our data by importing our modules and reading in our CSV dataset. Once this was done, we checked for null values and dropped all duplicate records. We found that some homes were listed multiple times and the total dropped ended at 214. After dropping the duplicates, we wanted to view all of the types of homes listed in the set. This is how we started:

```
7]:  1 # check types of homes in data set
     2 df.TYPE.unique()
```

```
7]: array(['Condo for sale', 'House for sale', 'Townhouse for sale',
           'Co-op for sale', 'Multi-family home for sale', 'For sale',
           'Contingent', 'Land for sale', 'Foreclosure', 'Pending',
           'Coming Soon', 'Mobile house for sale', 'Condop for sale'],
          dtype=object)
```

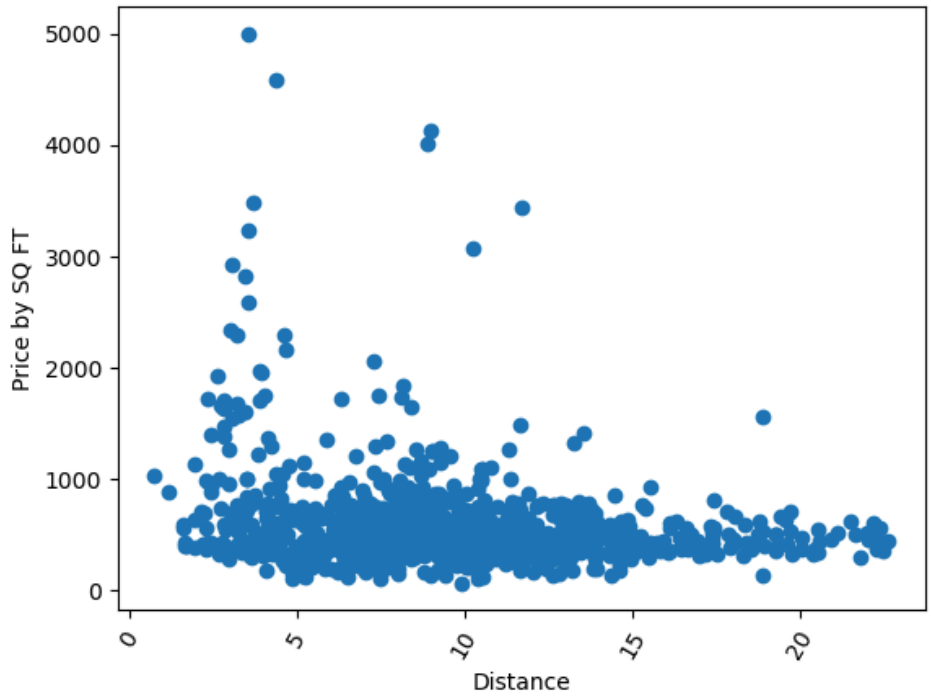
We then removed all of the nonspecific house types like: 'Coming Soon', 'Contingent', 'For sale', 'Mobile house for sale', 'Condop for sale', 'Pending', 'Foreclosure', 'Land for sale'. This left us with:

```
1 df.House_Type.unique()
```

```
array(['Condo', 'House', 'Townhouse', 'Co-op', 'Multi-family'],
      dtype=object)
```

After cleaning our data, we began brainstorming possibilities for our visualizations, we conducted a group-by to find the average price per home type and created a bar graph, however we ultimately decided to go in a different direction for our application. We decided to leave this code included due to the information being interesting and it could be used for future work.

Next, we added in the latitude and longitude from the center of New York City (40.730610, -73.935242) to compare the distance of each house from this point. After this was added, we then added a “Price by SQ FT” column to our data set by dividing the price of each home by the respective total square feet and created a scatter plot to compare this new column to the distance from the center of NYC.




We initially added a new column ‘Boroughs’ to label the city borough where each home was located, which ultimately was not used for our application due to the complexity of the New York area and our lack of familiarity with the city. Lastly, we checked our data for outliers and removed the top 1% priced houses to prevent our data from being skewed.

## Queries

After cleansing our data and planning the visualizations for our application, we then created our SQLite database from our cleaned dataset. We created an engine to connect to our SQLite database, and then created queries to support our filters for our interactive dashboards. We started with one field and used a uniform query across all of our desired fields that could be

filtered by the user. The query would call all housing data where the field was less than or equal to the user inputted value for all numeric fields. Additionally, the user would be able to filter for housing data for specified house types. The uniform query was the following:

```
price = 500000

# allow the user to choose price
if price == "All":
    where_clause = "1=1"
else:
    where_clause = f"PRICE <= '{price}'"

query = f"""
        SELECT
            *
        FROM
            NYC_Houses
        WHERE
            {where_clause};
"""

print(query)
```

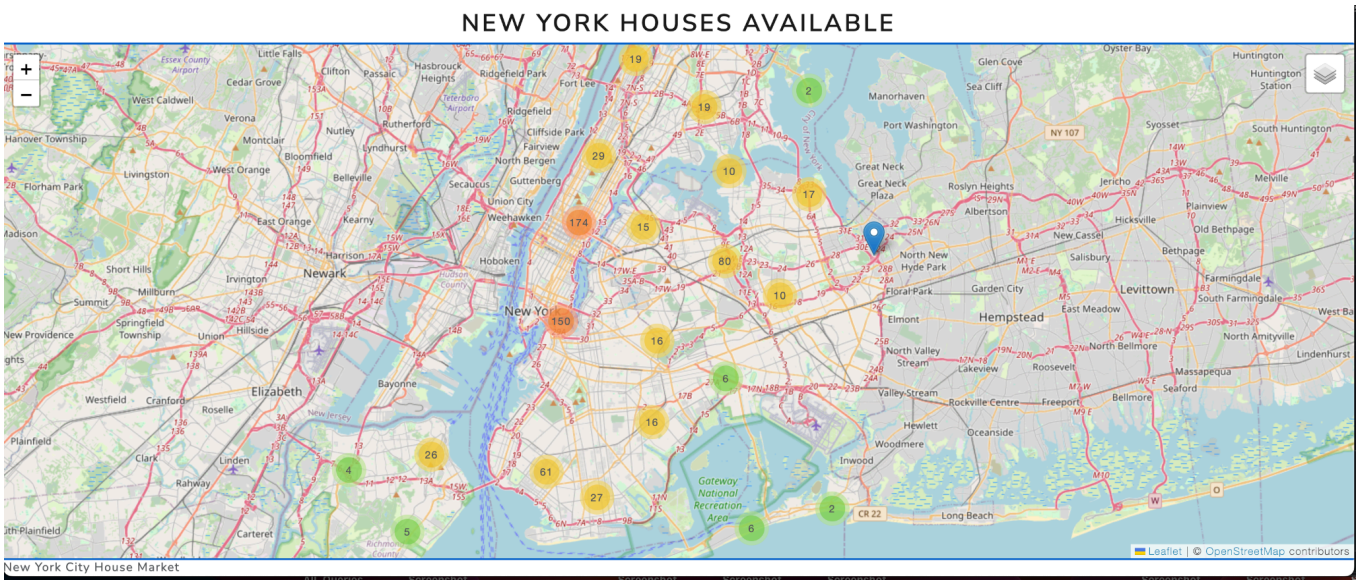
```
SELECT
```

The goal of this query was to bring back all the home data for all homes where the price of the home was less than or equal to a user input. We wanted our visualization to be similar to typical housing sites and accept multiple user inputs to filter through the homes that appear. So we conducted a similar query for each of the following filters a stakeholder would want to input: house type, price, number of bedrooms, number of baths, and distance from the center of New York City. For all the filters to work together we had to create one large query that accepted the where clause of each user inputted value. In order to write efficient code that doesn't repeat itself, we defined a function that would get the if-else where clause for all of our filter fields. As such, when creating a query for each individual visualization we could call on the function rather than re-writing the same lines of code. These queries for each visualization would supply our API and, subsequently, the dashboard with the housing data needed.

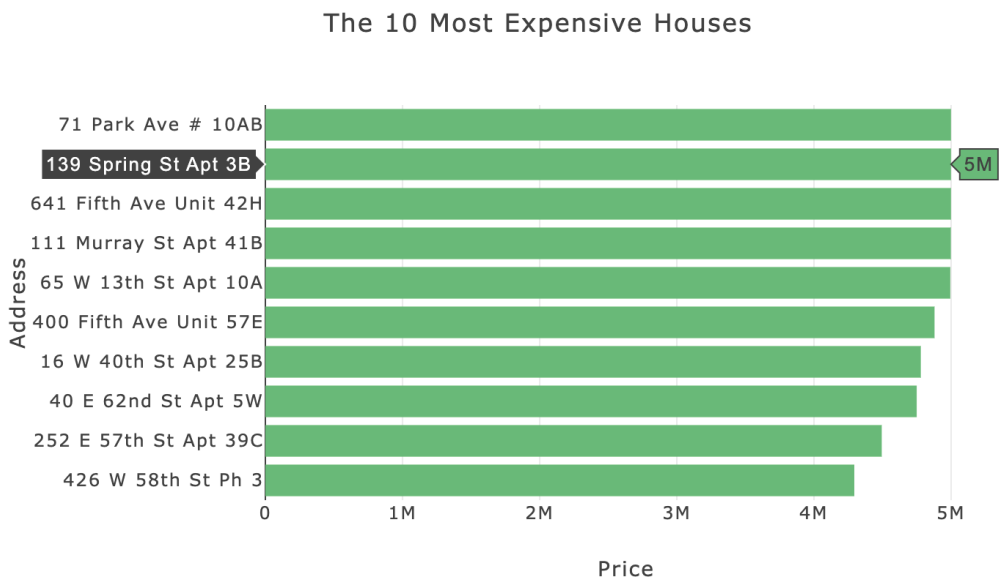
## Our Application

The final product was a full stack application containing an interactive dashboard with visualizations that are dynamic with user inputted filters. The first step in creating this application was creating a Flask API that would feed our dashboard the housing data.To create this Flask API we created a SQL Helper file that included our combined query for all our house data filters for each visualization. We then created our Flask API to make a request from our SQLite database depending on the inputs for our designated "filters". The API would return data as JSON.

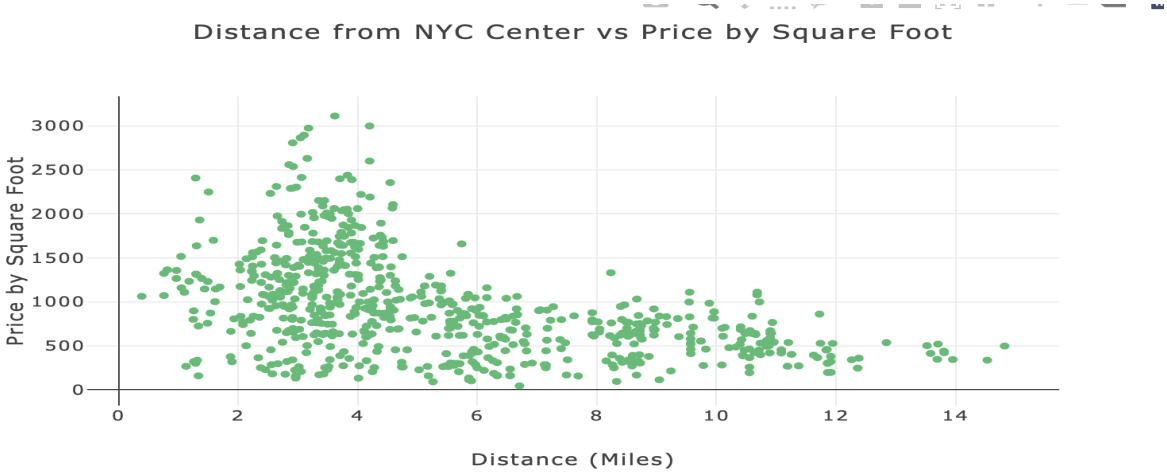
After our Flask API was created, we began the front end development of our application. We utilized Leaflet to embed an interactive map of New York in our dashboard and created markers from the spatial data received from our API with pop up boxes that contained the address and price of each respective house. This Leaflet map included cluster markers to display the amount of houses for sale in close proximity to each other. Additionally, the map background could be toggled from street view to topography (refer to the embedded Leaflet map below).




We then created a horizontal bar chart using Plotly. The bar chart consisted of the top 10 priced homes based on the user inputted filters. To achieve this visualization, the X-axis would receive the price of each home and the Y-axis would be the address of each home. Our last visualization to add was a Plotly scatter plot. This would give insight on how expensive New York can be (refer to the Plotly bar chart below).

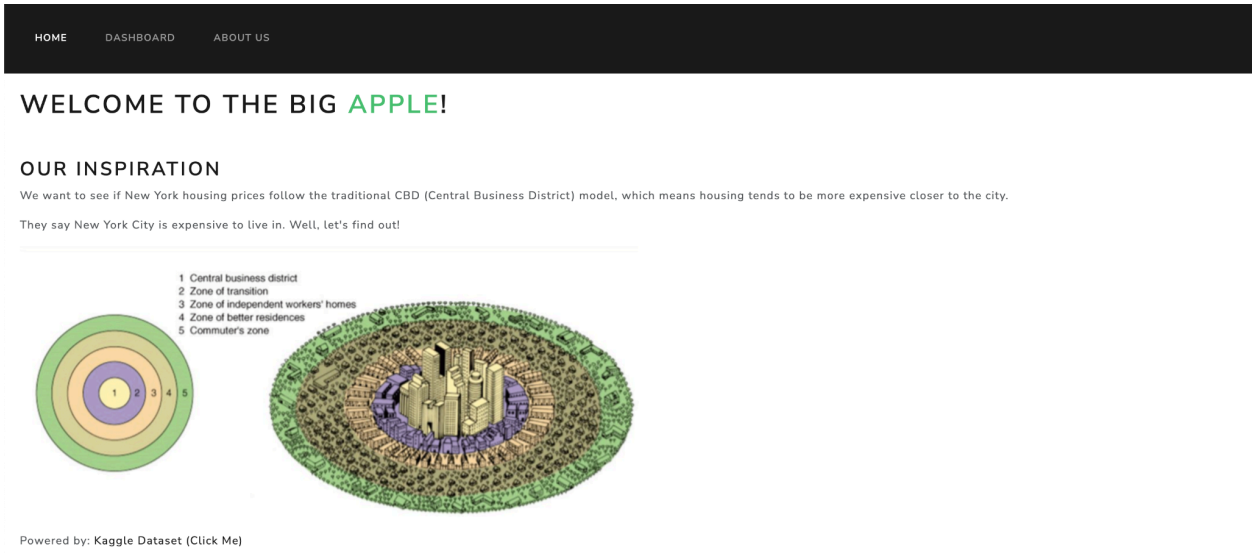



Similarly to the bar chart, we used Plotly documentation to build out a scatter plot, however, this visualization was to display the distance from New York City center against the price per square foot of each home depending on the corresponding user inputted filters. The X-axis contained the distance from New York City center and the Y-axis consisted of the price by square foot of each home. If there is a CCM trend with the pricing of homes and the distance from the city, the scatter plot would show a negative correlation (refer to the Plotly scatter plot below).




Utilizing JavaScript, we were able to create a dynamic dashboard of these 3 visualizations with 5 filters (house type, price, number of bedrooms, number of bathrooms, and distance from the center of New York city) and a button that allows the user to apply the filters to housing data that would feed the dashboard. This would allow the stakeholder to adapt the dashboard to their liking to find different insights they may be curious about (refer to filter feature below).


In our dashboard, we decided to use a bright green color for our visualizations to mimic the theme of New York's infamous nickname, "The Big Apple". Additionally, we added a landing page to our application to better the user experience. This home page included a visualization of the CBD model and text to summarize the inspiration of the project. If you're curious about who created this great full stack application, we also included an 'About Us' page with the bios of our group members and a link to each of our Githubs. Lastly, we utilized Python Anywhere to deploy our interactive dashboard for public use! Check it out at halej2.pythonanywhere.com.




## Conclusions

NYC does not necessarily follow the Concentric City Model, mainly because the area evaluated was limited to areas closest to the city center. This is probably because of historical development and expansion over nearly 400 years and changing standards of city planning. The NY area analyzed is delimited by sea, therefore prices vary more by boroughs than by distance from the center. Also, many large entities like the prison system, various utilities, transport hubs, parks and preserves occupy extensive regions, limiting residential development.

What was very interesting though was that we found a larger amount of condos, townhouses and multi-family constructions closest to the city center, which aligns with the CCM as price per square foot is higher the closer one gets to the center.

## Limitations/Bias

The most impactful limitation of this project was us being newly introduced to HTML/CSS and JavaScript. There are a lot of additional features and design that could add to the user experience. Some limitations in our data include the data being static. If possible, having dynamic data that updates daily with new homes for sale or what is sold would allow for this dashboard to be autonomous instead of manually inputting a new SQLite file every month. Additionally, our group is not familiar with New York, so there was limited knowledge of the boroughs of New York, and lack of information to make some conclusions about why homes were priced how they were.

A large limitation/bias to the dataset ist that the data only includes houses relatively close to New York City, and doesn't include houses within the suburbs. This would also allow us to truly see if there is a pattern of the distance from New York homes vs price by square foot.

## Future Work

In the future, the dataset can be expanded to cover more cities and compare which cities may follow the CCM and which do not. There could be a dashboard for each city (Los Angeles, Chicago, Washington D.C., etc.) A comparison can be made with East coast city development patterns against other regions of the US- California, Pacific Northwest, Alaska, Florida, and central US. Houston sprawl stands out as a particularly potent market for analysis, as well as Dallas/Ft. Worth with census estimated growth in excess of 30% year on year in Frisco.

This dataset should be expanded to include more of the suburbs and therefore understand if analyzed at a greater scale, the CCM does apply to the greater NY. Range feature functionality can be brought to the filters. For example, the user could select a price range $500,000-$1,000,000 and/or 3-6 bedrooms. Additional drop-down for square footage could be implemented- and thus compare co-ops for sale in NYC for $600,000 to multi-family, condos, townhouses, and houses in their respective price ranges.

# Works Cited

Kaggle Dataset:

"New York Housing Market." Kaggle, Nelgiri Yewithana, www.kaggle.com/datasets/nelgiriyewithana/new-york-housing-market.

Archived Webpage:

Burgess, Ernest W. "The Growth of the City: An Introduction to a Research Project." Geographical Review, vol. 26, no. 3, 1936, pp. 383-396. Web Archive, web.archive.org/web/20110629113720/http://people.hofstra.edu/geotrans/eng/ch6en/conc6en/burgess.html.

Transport Geography Website:

Burgess, Ernest W. "The Growth of the City: An Introduction to a Research Project." Transport Geography, transportgeography.org/contents/chapter8/urban-land-use-transportation/burgess-land-use/.

Stack Overflow. "Calculate distance between two latitude-longitude points? Haversine formula." Stack Overflow, Stack Exchange, 2008, https://stackoverflow.com/questions/27928/calculate-distance-between-two-latitude-longitude-points-haversine-formula.