

NYCVA
A visual analysis of New York City AirBnBs

Leonardo Balzoni - 1870364
Alberto Contorno - 1873252

February 21, 2020

1 Introduction to the analysis

Housing websites like *AirBnB* [1] are among the websites that generate the most traffic worldwide[2]. Even with an immense amount of unseen work that ensures a pleasurable user experience, it can still be difficult for visitors to find the information they are looking for, given the sheer amount of data available. This project thereby tries to give some otherwise inaccessible insights about the announcements, with the help of visual analytics. Conducting this research on a global scale would be too computationally heavy for our resources, therefore our analysis is limited *New York City*, one of the cities that has the most announcements on the website.

2 Related work

Before starting to develop our tool, we wanted to understand which is the state-of-the-art for housing websites' visualizations. What we found out is that in almost all the cases the visualizations were quite limited and only offered few insights while focusing on mainly one topic in each case.

- *AirBnBShare* [3]: A trend-exploration tool aimed to help people to efficiently construct their travel experience and encourage the participation into the sharing economy. The statement on which their analysis is built is that Airbnb is known for its core philosophy of connecting people and culture by providing a residential-space sharing platform. However, through data exploration, they noticed a drastic gap between the ratio of the Entire Home listings and Shared Room listings.

The issue is that the tool focuses mainly on shared rooms, which is for most tourists not the ideal accommodation.

- *InsideAirBnB* [4]: The question at the base of the visualization is how is Airbnb really being used in and affecting the neighbourhoods of your city? The focus of this analysis is that data shows that the majority of Airbnb listings in most cities are entire homes, many of which are rented all year round - disrupting housing and communities.
Even in this case the tool is used more as a way to prove their statement than to actually help users that want to use AirBnB.

3 Description of the dataset

The dataset that is going to be used was provided from *Kaggle* [5], and it is made of around 50000 tuples with 16 attributes each [6]. We are perfectly aware of the fact that this means that our “AS index” is quite higher than the suggested one, which means that we are faced with 2 options: reduce the amount of attributes or reduce the amount of tuples. What we ended up doing is a mix of the 2, as we dropped useless attributes such as the name of the host or the date of the last review, and we reduced the amount of tuples to 10000. The AS index is still higher than the suggested one (circa 140000 versus the suggested 50000), but we were able to manage it.

Each tuple of the set represents an announcement on the AirBnB website for a house in New York City, some of the most useful attributes are: the location of the house (both its neighbourhood and its coordinates), its cost, the average availability, and the amount of reviews. These information will be used to identify patterns and gather insights about the distribution of rental housing in NYC in order to have a visual representation of the best options available.

4 Goals of the Analysis

An analysis of this kind can be used to fulfill multiple objectives; firstly as already stated the multiple visualizations will allow to see otherwise hidden information, such as for example the average price in each neighbourhood, or the estimated income of the various hosts. This means that both someone who is willing to put a announcement on the website and someone who is looking for an accomodation, will be able to find useful information.

Moreover we could apply some dimensionality reduction algorithms such as *PCA* or *t-SNE* to gather informations about similar announcements, which could for example be used to find an announcement similar to one that we like, but that might be unavailable.

As for non-functional requirements, we want to make sure that the tool can be used on a wide range of systems and devices, this means that responsiveness and performance will be always taken into consideration during our development.

5 Dataset preprocessing

The given dataset was immediately almost ready to be used, as there were no duplicates and the attribute's types were coherent and almost always not null.

Even with this said some preprocessing was necessary, we began by dropping some useless columns such as the name of the host or the date of the last review. Moreover there were some announcements with missing data for important fields, so we dropped those (circa 20 out of 50000).

One more thing that we did was using the information available to evaluate the estimated monthly income for each of the announcements, this was done by multiplying the price for the amounts of days in which the house is rented (365 - availability) and by dividing everything by 12.

Moreover since we saw that only a minuscule percentage of announcements had a cost per night greater than 500\$, we decided to drop these ones. We did this so that the graphs would not be extremely skewed just to represent a couple of announcements.

Lastly there was the “Reviews per month” attribute that was null in the case of 0 reviews on the announcement. Instead of dropping these, what we did was to fill the null values with 0 in the case of no reviews for the house.

5.1 PCA preprocessing

While what we said is enough for all the analyses that we ended up doing, evaluating PCA requires more attention. This is due to the fact that the algorithm only accepts numerical values for the various fields, and our dataset instead has some categorical ones (i.e. “neighbourhood” or “room_type”). What we did was creating some sort of one-hot encoding for those fields; for example “room_type” could be: an entire apartment, a private room or a shared room, so we replaced the “room_type” field with 3 new ones: “room_type_apt”, “room_type_pvtroom” and “room_type_shroom”, and the values are all 0 except for one which is 1. Doing the same for the rest of the categorical fields we finished the preprocessing for the dataset for PCA.

6 Tools used

The code is mainly written in *Javascript* and *HTML*, although at the core of our analysis is *D3.js* [7], a small, free javascript library for manipulating

documents based on data. Instead of using the basic D3 functions we also utilised *PlotlyJS* [8], a high-level, declarative charting library built on top of D3, which ships with lots of chart types.

It also has to be stated that the preprocessing of the dataset was done in *Python* [9] with the aid of *Pandas* [10], a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Lastly to make things like HTML document traversal, manipulation and event handling much simpler we also used *jQuery* [11].

7 Visualizations

We will now see more in detail each of the graphs that we created, for each we will explain its function and the available interactions that it offers.

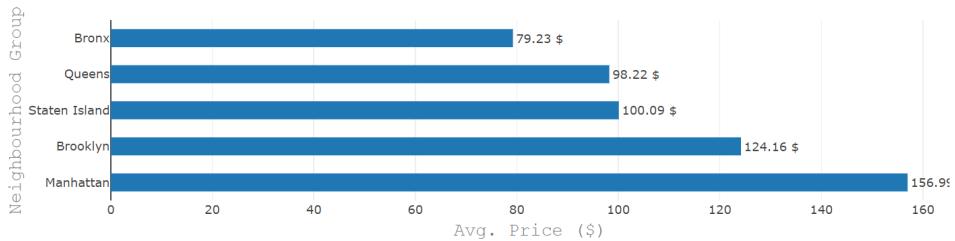
7.1 Average neighbourhood price and income barcharts

We began by creating some barcharts to visualize some pricing information across the city:

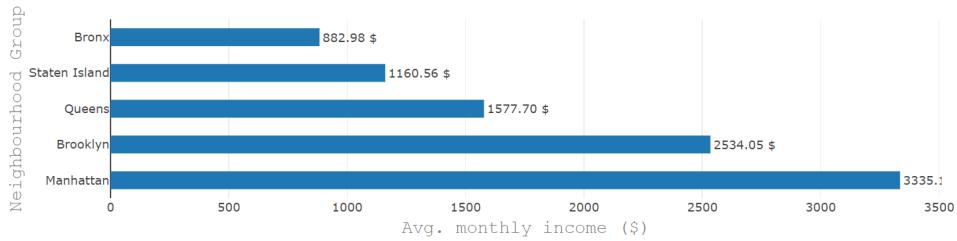
- The first one shows the average price of announcements in each of the NYC neighbourhood groups (Bronx, Staten Island, Queens, Brooklyn, Manhattan).
- The second one shows the average monthly income of announcements in each of the NYC neighbourhood groups.
- The third one is more complex, it allows to see the average price in each single neighbourhood of the city, offering to the user the option to manually select the neighbourhoods he is willing to compare. Moreover we created some buttons that allow to quickly compare all the neighbourhoods inside each of the neighbourhood groups.

Each of these charts allows to pan zoom and select neighbourhoods. By selecting one or more hoods, all the announcements belonging to them are selected across all the other graphs.

Average price by neighbourhood group

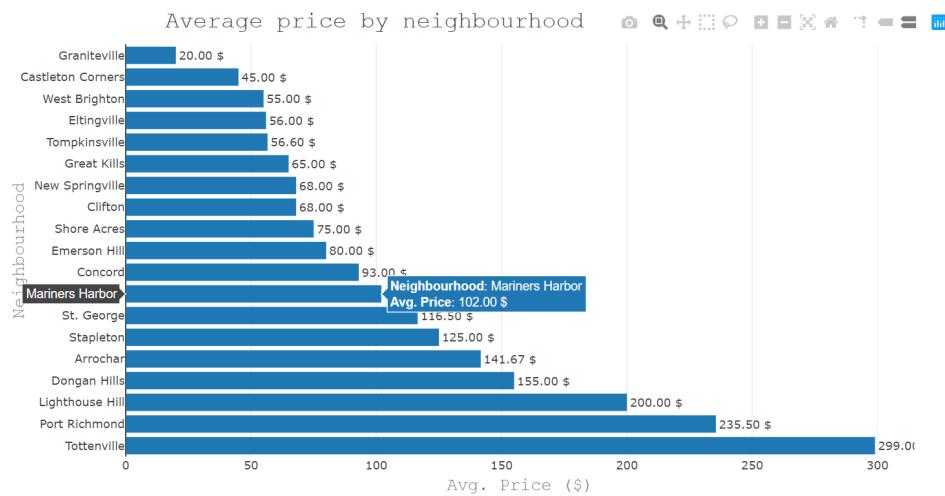


Average monthly income by neighbourhood group



Select neighbourhoods:

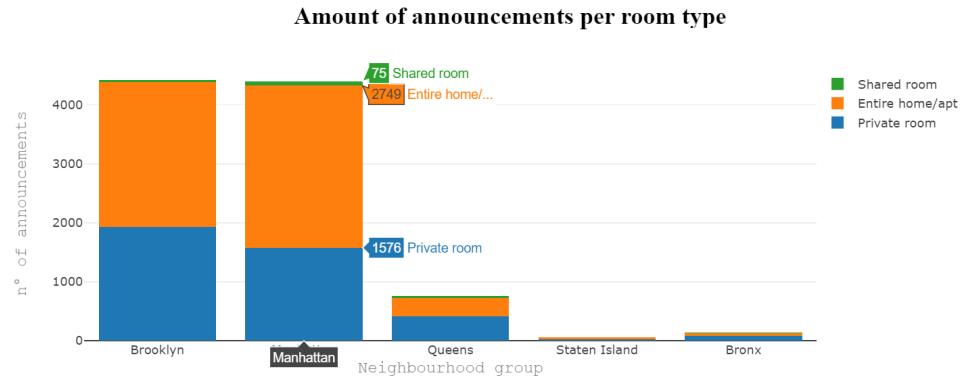
Bronx Staten Island Queens Brooklyn Manhattan Select top 5
 St. George Tompkinsville Emerson Hill Shore Acres Arrochar Clifton Graniteville Stapleton New Springville
 Tottenville Mariners Harbor Concord Port Richmond Eltingville Lighthouse Hill West Brighton Great Kills
 Dongan Hills Castleton Corners



7.2 Amount of announcements per neighbourhood barchart

Following the price barcharts, we noticed a huge disparity not only in the amounts of announcements in the different neighbourhood groups but also in the kinds of room types available for rental, so we created this stacked barchart to visualize it.

The available housing types are: entire apartment, private room or shared room. We divided the barchart per neighbourhood group, hovering over one of these shows the exact housing type distribution in that neighbourhood.



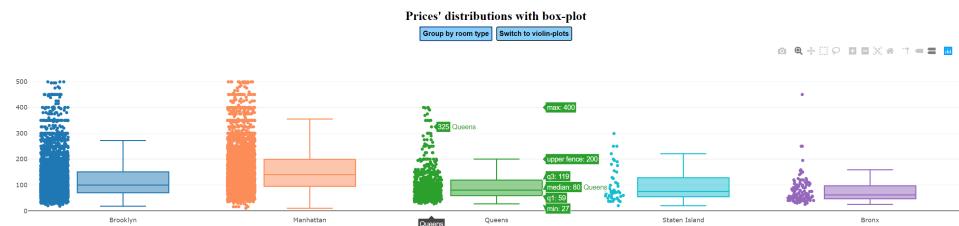
7.3 Box-plot for pricing distribution across neighbourhoods

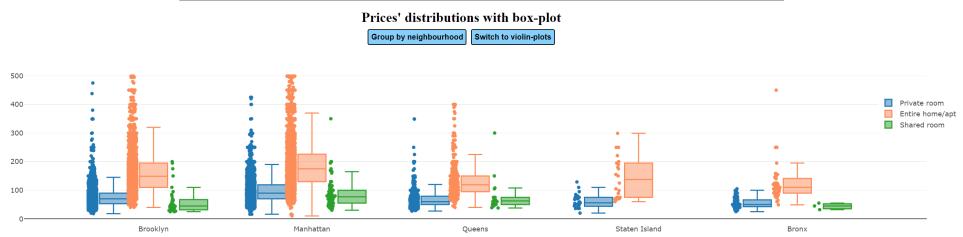
Here we created boxplots showing the price distribution across neighbourhood groups, hovering on each one shows informations about the distribution (max, min, upper and lower quartiles and median)

Next to each boxplot we also put the points that generated it, allowing users to make selections in whichever price section they want.

Also in this case when the user makes a selection, the same announcements are selected across all the other graphs.

Via a button the user can also switch the visualization to one that shows detailed boxplots for each of the housing types (entire apartment, private room, shared room).

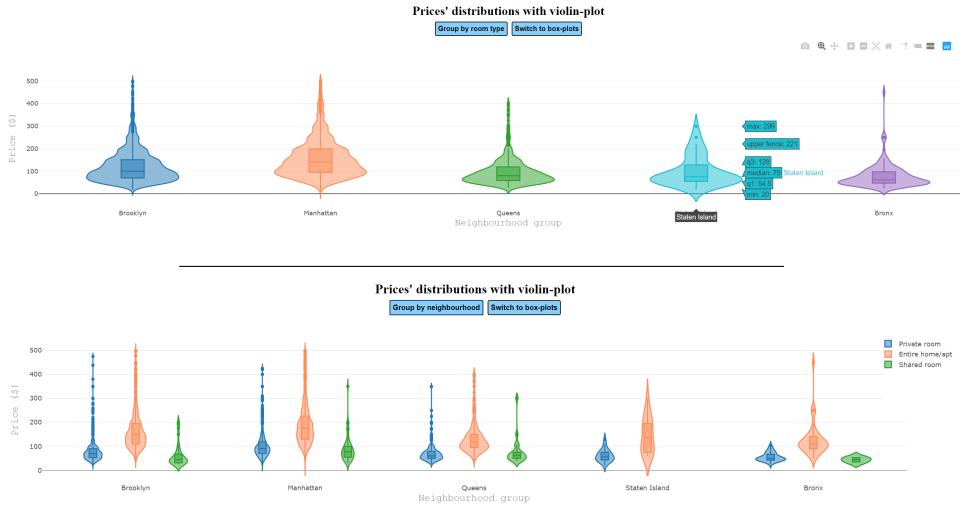




7.4 Violin-plot for pricing distribution across neighbourhoods

From the box-plot section the user is able to click a button that switch the graph to a violin plot one. Since the information gainable from these 2 graphs is the same we thought that having both viewable at the same would be useless, so we allow the user to use the one he prefers.

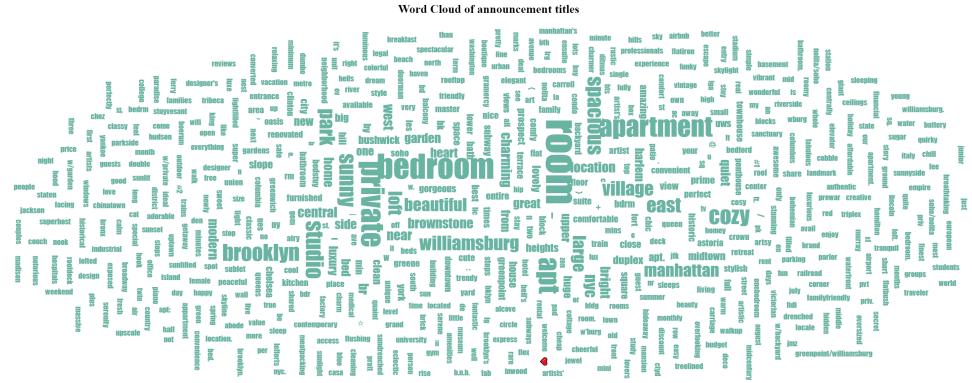
Also in this graph via a button the user can switch the visualization to one that shows detailed violin-plots for each of the housing types (entire apartment, private room, shared room).



7.5 Word Cloud

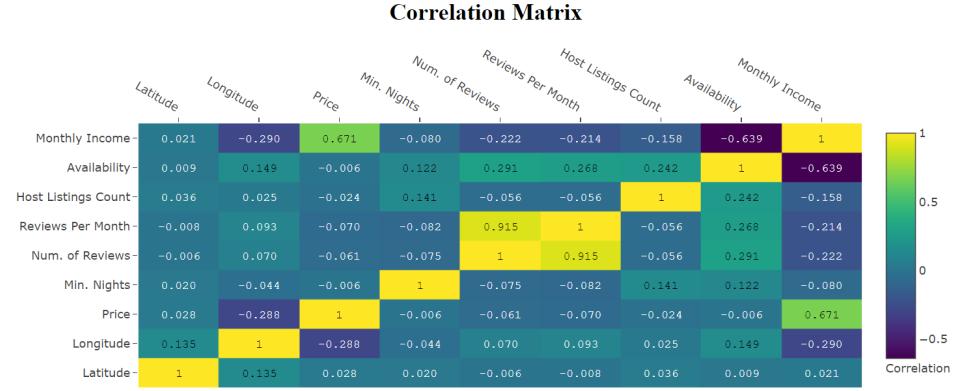
By exploring the dataset it was immediately clear that hosts tend to put catchy words in their announcements to make the house sound more appealing. So by counting each word's occurrences in all the titles of the announcements in the dataset we were able to find the most used ones (after eliminating stopwords) and to plot them in a word cloud view.

The dimension of the words in the word cloud is linearly dependant on their occurrence in the announcements' titles.



7.6 Correlation Matrix

While searching for ideas of graphs to visualize, we wanted to see if some of the attributes of the announcements were tied to each other. The best way to do so is to create a correlation matrix between attributes able to highlight the attributes that are related to each other.



As one can see, its easy to tell that there are no statistically relevant correlations except for the obvious ones like between price and income or between total number of reviews and reviews per month.

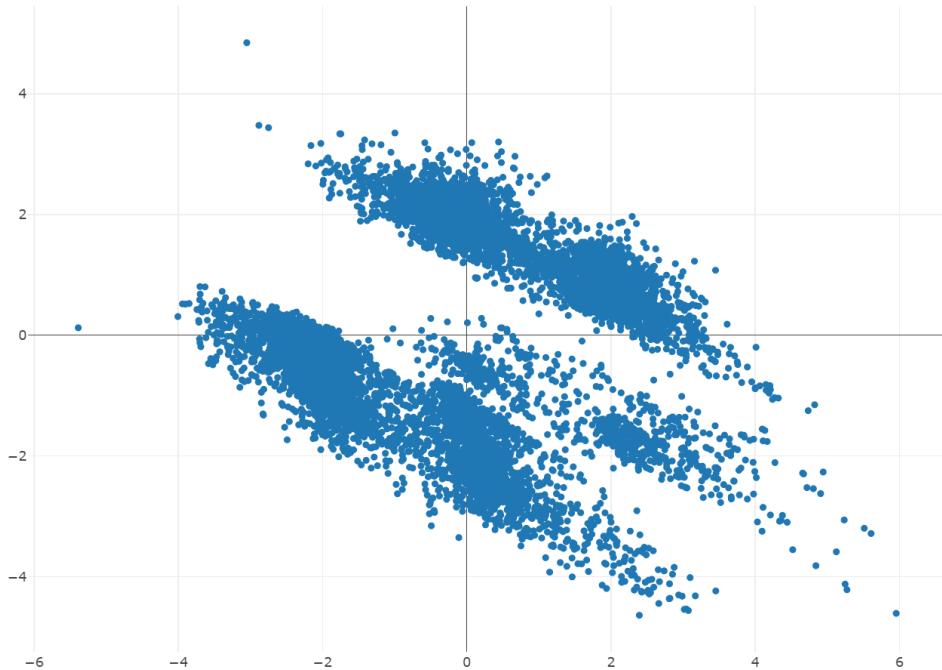
7.7 PCA

Something that can be especially useful in sites like AirBnB is a way to find similar announcements. One of the ways to do so is to use a dimensionality

reduction algorithm, we chose PCA. As a normalization algorithm we used *z-norm*, in this way the closeness between the announcements is preserved. Since is extremely computationally heavy what we did was evaluating it once and then saving in a file its results. In this way the loading of the browser remains snappy.

Of course we made it possible to select announcements from the scatterplot to others graphs, in this way we are able to see how the announcements in the clusters created from PCA are related to eachother.

Scatterplot of PCA



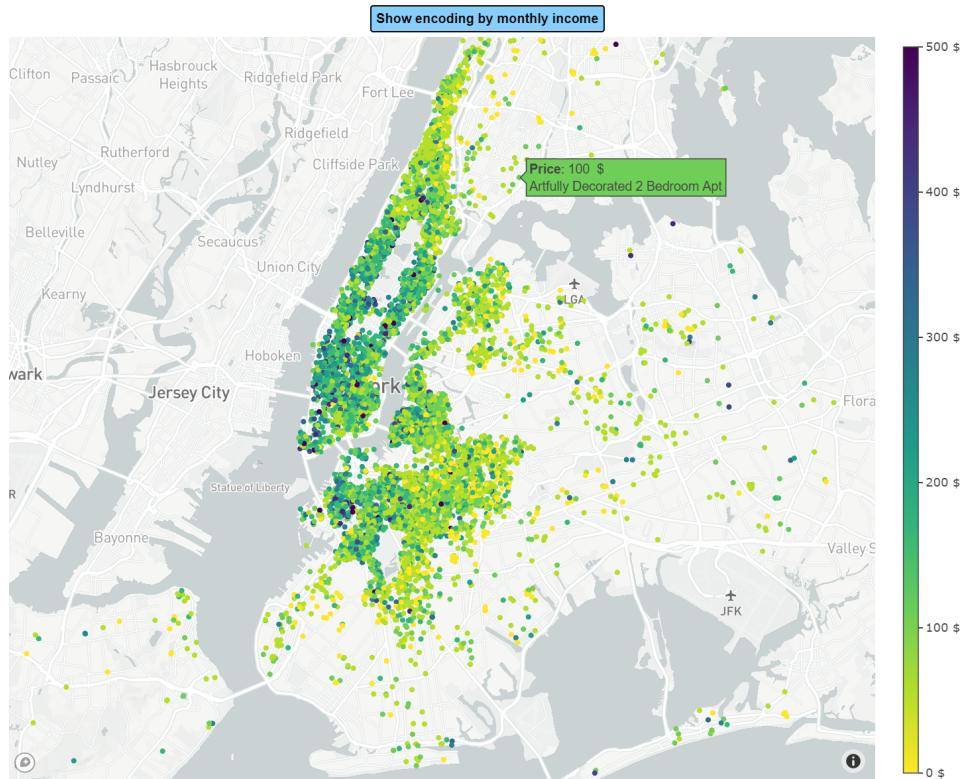
7.8 Map Scatterplot with encoding for price and average income

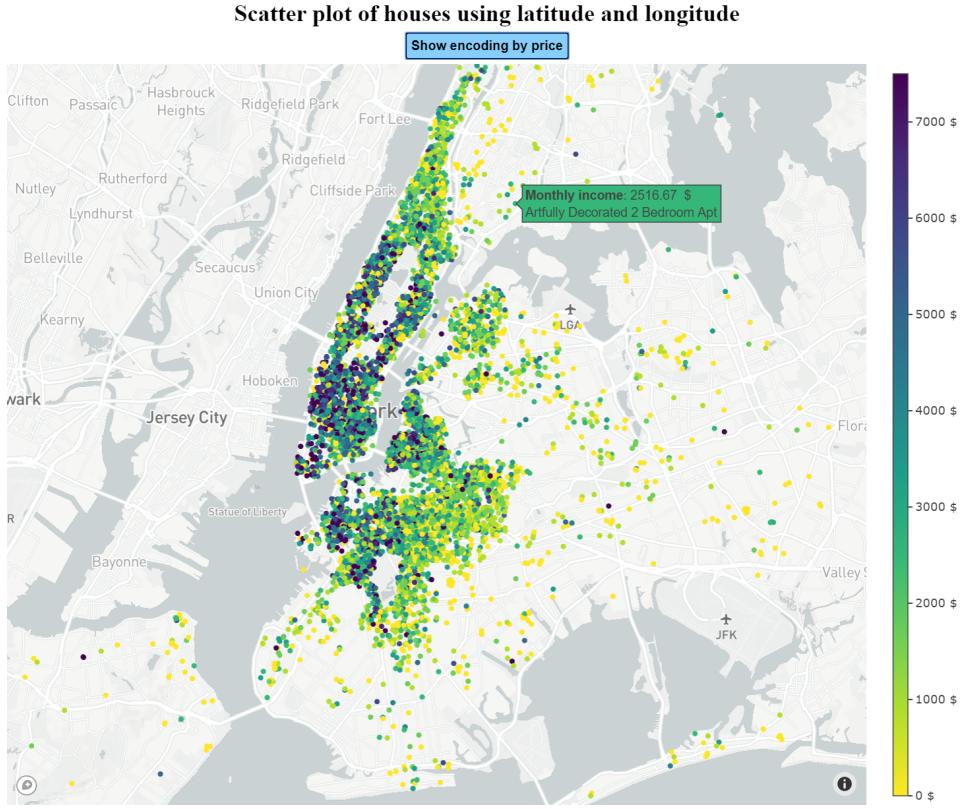
Last but not least we used the announcements' latitude and longitude values to create a scatterplot of the announcements using their location. We overlayed this info to a New York City map obtaining an interesting result. As an added encoding we used the color of the points in the scatterplot to show either the announcements' price per night or their average monthly income.

Even in this case when the user selects announcements the selection car-

ries across other graphs. As we will show this allows for some extremely interesting insights.

Scatter plot of houses using latitude and longitude



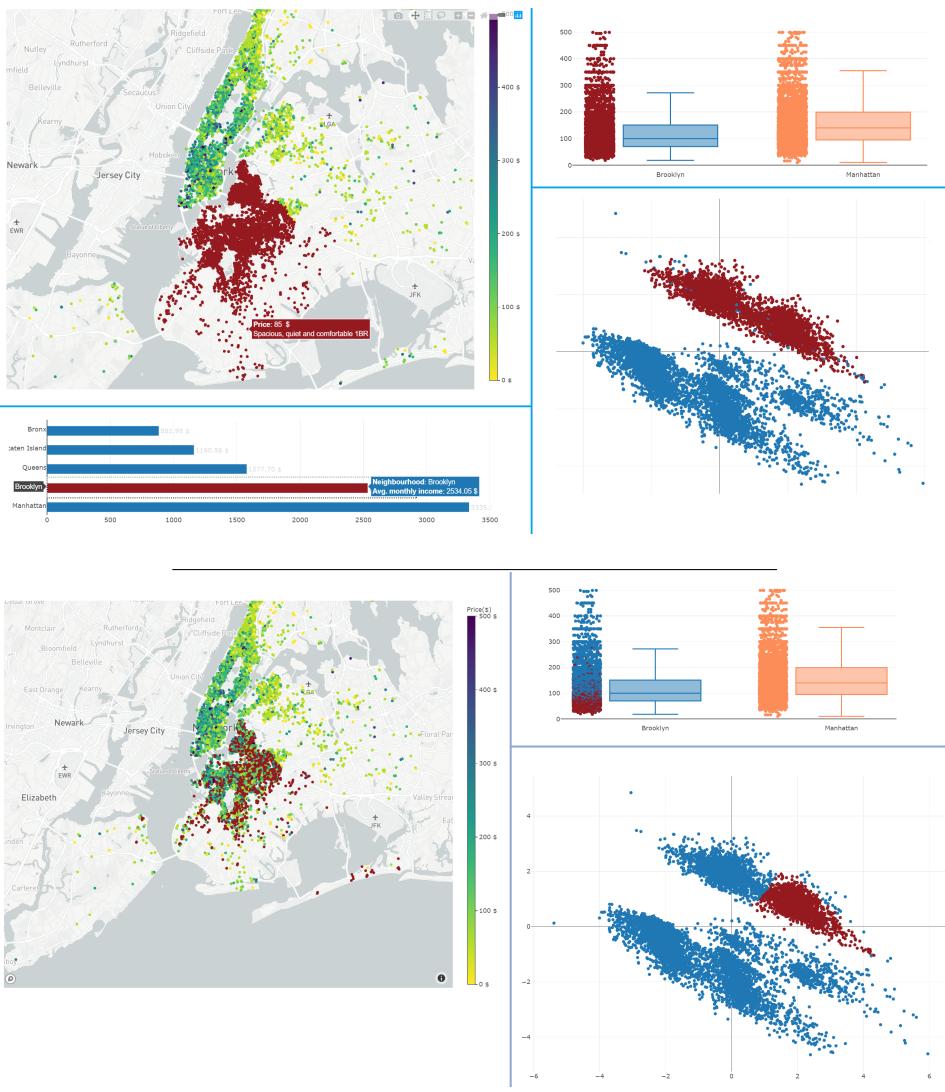


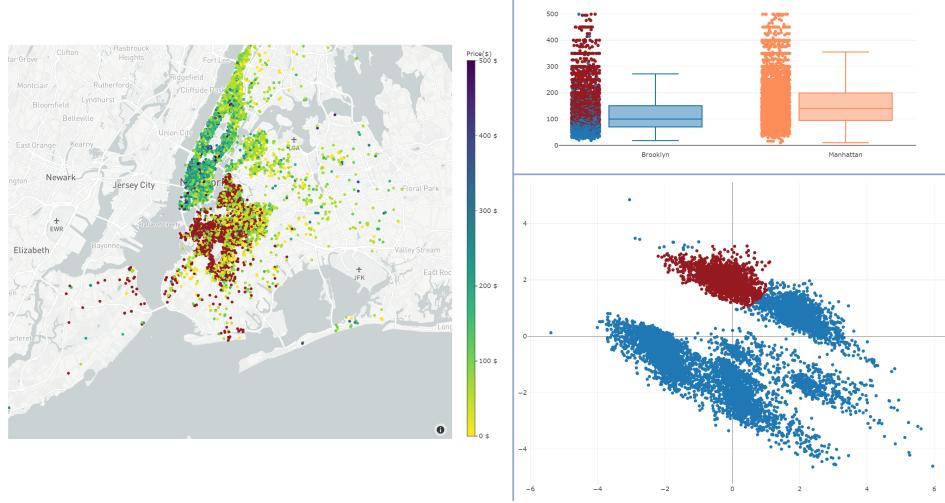
8 Interaction design

Visualizing data is one thing, being able to interact with it is another. We wanted to make it possible for the user to be able to interact with all the graphs in our project, and the choice of using *PlotlyJS* surely helped in this regard. Thanks to it, all the graphs done using the library are: pannable, zoomable and selectable both with a box selection or a lasso selection tool; moreover the user is also able to reset the graph to its original view and download it as a png image.

Even though this already makes a huge difference with respect to static data, we wanted something more. There are multiple scatterplots across the project (in the pricing distribution graph, on the map, and on the PCA representation) and we thought that it would have been interesting to make the selection in a across-project manner. This means that when the user selects some houses in one of these graphs, the selection is carried across to all the others (we also allowed selection from the pricing barcharts; if the

user selects a neighbourhood, the selection transfers to the scatterplots). This is especially useful to see how the clusters created by PCA are reflected in the actual pricing/position of the announcements, the images below show how clearly announcements are clustered taking into consideration both the price and the neighbourhood:





9 Future work

While we are quite proud of the results that we were able to obtain with this project, one main thing that we have in mind and that for sure we will do but that is outside the scope of this exam is to create a “intelligent pricing” algorithm.

By this we mean a machine learning algorithm that is able, given the details of a new house, to estimate its optimal pricing, by taking into account similar announcements. We are quite confident that some powerful regressors like *Lasso* [12] or *Extreme Gradient Boosting* [13] would be able to extract the most important features and assign weights to them, so that a new house’s price could be accurately estimated. This could of course be extremely useful for someone that is willing to add an announcement to the site but is unsure about the best price to put it at.

References

- [1] “AirBnB’s website”
[https://www.airbnb.it/.](https://www.airbnb.it/)
- [2] “Web traffic on AirBnB”
[https://www.similarweb.com/website/airbnb.com.](https://www.similarweb.com/website/airbnb.com)
- [3] “AirBnBShare data exploration”
[http://tangyidesign.com/tableau-airbnb-la-data-visualization-1.](http://tangyidesign.com/tableau-airbnb-la-data-visualization-1)
- [4] “Inside AirBnB’s data exploration results”
[http://insideairbnb.com/.](http://insideairbnb.com/)
- [5] “Kaggle website”
[https://www.kaggle.com/.](https://www.kaggle.com/)
- [6] “New York City AirBnB’s dataset”
[https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data/.](https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data/)
- [7] “Data-Driven Documents website”
[https://d3js.org/.](https://d3js.org/)
- [8] “PlotlyJS website”
[https://plot.ly/javascript/.](https://plot.ly/javascript/)
- [9] “Python website”
[https://www.python.org/.](https://www.python.org/)
- [10] “Pandas website”
[https://pandas.pydata.org/.](https://pandas.pydata.org/)
- [11] “jQuery website”
[https://jquery.com/.](https://jquery.com/)
- [12] “Lasso regression explained”
[https://en.wikipedia.org/wiki/Lasso_\(statistics\).](https://en.wikipedia.org/wiki/Lasso_(statistics))
- [13] “Extreme Gradient Boosting documentation”
[https://xgboost.readthedocs.io/en/latest/.](https://xgboost.readthedocs.io/en/latest/)