

AirBnB in Amsterdam, 2019: A Case Study

Deepa Sury

CareerFoundry: Data Analytics Immersion

Starting Date: 21 December 2024

Completion Date: 14 January 2025

Tutor: Obinna Iheanachor

Mentor: Evans Otalor

Project Timeline

1. Data Sourcing, Forming Initial Project Idea, Hypotheses and Funnelling Questions (21.12.2024)

Downloaded dataset, formed project concept, hypotheses and funneling questions.

2. Data Cleaning and Wrangling (28.12.2024)

Chose which data to use and to discard, performed data-cleaning and wrangling operations.

3. Data Exploration and Descriptive Analysis (28.12.2024)

Created a data-dictionary to describe the data before cleaning and wrangling, as well as after these steps.

4. Further Data Analysis (29.12.2024 – 04.01.2025)

Performed further exploratory analysis on the data, ran supervised and unsupervised learning algorithms.

5. Repeating Data Cleaning, Wrangling and Exploration (04.01.2025)

Redid all cleaning and wrangling of raw data, updated data dictionary after finding errors in previously cleaned and wrangled data.

6. Repeating Further Data Analysis (05.01.2025 – 09.01.2025)

Redid all analysis using new cleaned and merged datasets.

7. Creating Final Deliverable (10.01.2025 – 14.01.2025)

Created a dashboard and interactive visuals on Tableau; added presentation, documents and Python scripts to GitHub.



Overview

This project takes place in December 2018 and analyses data scraped from AirBnB's listings for the city of Amsterdam for the upcoming year. Using insights from this data, this project aims to describe the current state of the short-term rental market in Amsterdam and to inform visitors to the city on their choice of accommodation, as well as local residents intending to rent out their own properties on the platform.

Objective

Looking at the AirBnB listing data from the upcoming year can provide us with helpful insights. For travellers planning a trip to Amsterdam, this data can help them decide on when and where to stay, based on criteria such as value, accommodation type or distance from the city centre. For hosts, this data can inform them on which parts of the city are in highest demand, which neighbourhoods are the most or least expensive to stay in and how they can best promote their own properties to visitors.

Questions to Explore

- What is the average price per night in Amsterdam and how does this change during the year?
- Which areas in Amsterdam are the most popular with visitors? Are listings in these areas more expensive than in the rest of the city?
- Do any neighbourhoods appear to be decreasing in popularity? Are there any areas that can be considered “up-and-coming”?
- What kind of accommodation is most commonly offered by AirBnB hosts in Amsterdam?
- What do AirBnB users expect when booking accommodation? What causes them to leave positive or negative reviews?
- In the user reviews, can we find any kind of pattern or trend based on commonly-used words or phrases? Which words or phrases are associated with different neighbourhoods, price ranges, review rating score, accommodation type, etc.?

Data Used

The data used in this project consists of daily AirBnB listings for the city of Amsterdam between the 1st of January and 31st of December, 2019 and was retrieved from [Kaggle](#) .

The complete data-set consisted of 6 tables and a GEOJSON file:

- **calendar.csv** – Contains a year's worth of daily information on the availability and price for each listing.
- **listings.csv** – Contains advertisements for individual rentals, including information about the type of accommodation, number of beds, neighbourhood, hosts and availability throughout the year. This table also contains information on the number of reviews, as well as the date of the first and latest reviews.
- **listings_details.csv** – Contains the same information as listings.csv, along with several additional columns of supplementary information on the property, amenities, host, location and additional fees.
- **neighbourhoods.geojson** – Contains geospatial information on the different neighbourhoods within and around Amsterdam, to be used in visualizations and conducting geospatial analysis.
- **neighbourhoods.csv** – Contains a list of neighbourhoods within and around Amsterdam where the AirBnB listings are located.
- **reviews.csv** – Contains the date and listing ID information for reviews on the site.
- **reviews_details.csv** – Contains additional information about reviews, such as the name and user ID of the reviewer, as well as the comments left in the review.

Tools Used

- Python for data-cleaning, wrangling and analysis. Libraries used include pandas, NumPy and SciKitLearn.
- Tableau Public for visualization and dashboard creation.

Steps Taken

- Data Sourcing, Cleaning and Wrangling
 - Downloaded data, extracted relevant columns; checked for completeness, duplicates, spelling, format, etc. and corrected these when needed.
 - Created new columns out of existing ones to help with analysis.
- Descriptive and Exploratory Analysis
 - Found descriptive statistics for numerical variables.
 - Assessed relationships between these variables.
 - Created a data dictionary describing the raw data and the changes made to it over the course of the project.
- Geospatial Visualization
 - Created choropleth maps using selected columns along with the GEOJSON data
- Linear Regression
 - Ran a supervised learning model (linear regression analysis) on the data to establish a relationship between pairs of variables.
- K-means Clustering Analysis
 - Ran an unsupervised learning model (k-means clustering) to sort the data points into clusters based on similarities.
- Dashboard Creation
 - Displayed the results and insights gained during this project in a Tableau dashboard.
 - Recreated many of the charts made on Python during analysis and added interactive elements.

Data Sourcing, Cleaning and Wrangling

Tools used: Python (pandas, NumPy)

- Checked all fields for completeness, format and cleanliness.
- Dropped columns that were empty or irrelevant to the purpose of the project (e.g. scrape information, host profile picture URL)
- Corrected the format, spelling and consistency of the remaining columns.
- Created new variables using existing columns:
 - Added a column showing the distance of each property from the city centre in kilometres. This value was calculated by applying the Haversine formula using the latitude and longitude values, plus the coordinates of Dam Square in the centre of Amsterdam.

```
def haversine(lat1, lon1, lat2, lon2):  
    lat1, lon1, lat2, lon2 = map(np.radians, [lat1, lon1, lat2, lon2])  
    dlon = lon2 - lon1  
    dlat = lat2 - lat1  
    a = np.sin(dlat/2)**2 + np.cos(lat1) * np.cos(lat2) * np.sin(dlon/2)**2  
    c = 2 * np.arcsin(np.sqrt(a))  
    r = 6371  
    return c * r  
  
lon1 = df_listings_sub["longitude"].iloc[0]  
lat1 = df_listings_sub["latitude"].iloc[0]  
lon2 = 4.892351  
lat2 = 52.373100  
  
df_listings_sub['distance_from_center'] = haversine(df_listings_sub['latitude'].shift(), df_listings_sub['longitude'].shift(), 52.373100, 4.892351)
```

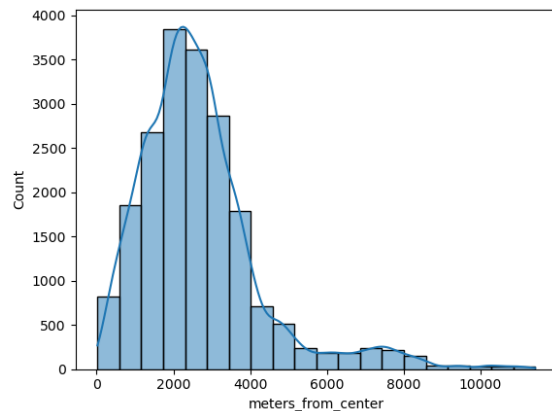
- Aggregated rows with daily information by month and created a new column for each month of the year.
- Aggregated the number of days available per month and per year and created columns showing the average percentage of days a property was available for these periods of time.
- Aggregated the daily price values and created columns showing the average daily price for each month and for the year.

Discarded the reviews.csv and reviews_details.csv tables. I had initially intended to keep them for the purpose of text-mining; however this ended up being too complicated of a step to undertake.

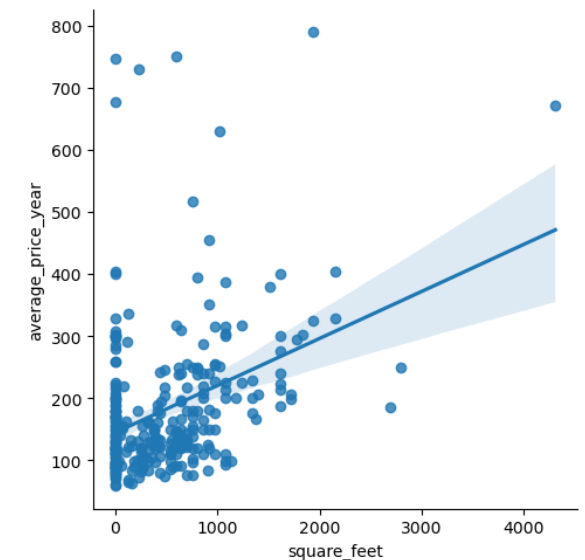
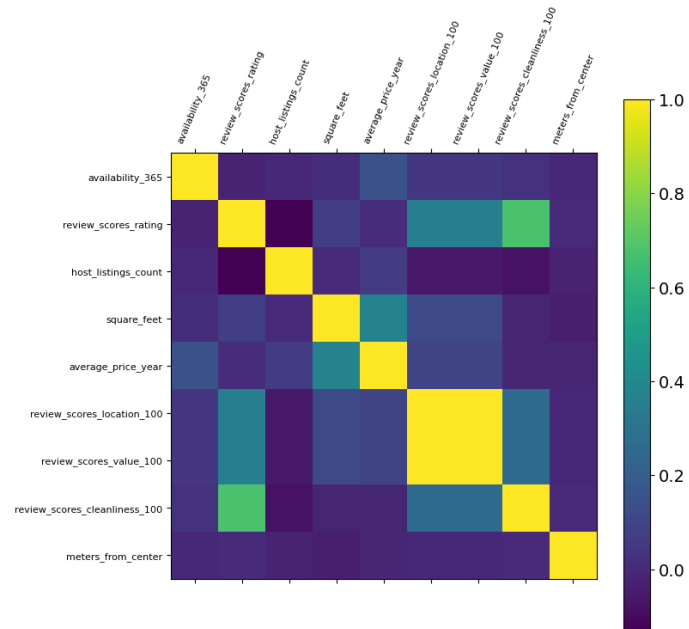
Descriptive and Exploratory Analysis

Tools used: Python (pandas, NumPy, matplotlib, Seaborn)

- Performed descriptive analysis on all numerical columns.
 - This provided general statistics, such as minimum, mean and maximum values, as well as percentile distribution values.
- Found the distribution of values within certain variables
 - This allowed me to understand the nature of most of the listings I would be working with.
- Found the correlation strength between all variables by creating a correlation matrix.
 - This provided some insights that immediately stood out: for example, the correlation between the review scores for cleanliness and the overall review score was particularly strong, while distance from the city centre did not appear to have any significant relation to the other variables.
- Calculated the correlation strength between pairs of individual variables and depicted these using scatterplots (numerical values) and catplots (non-numerical values).
 - This provided a more detailed look at the relationships between pairs of individual variables.



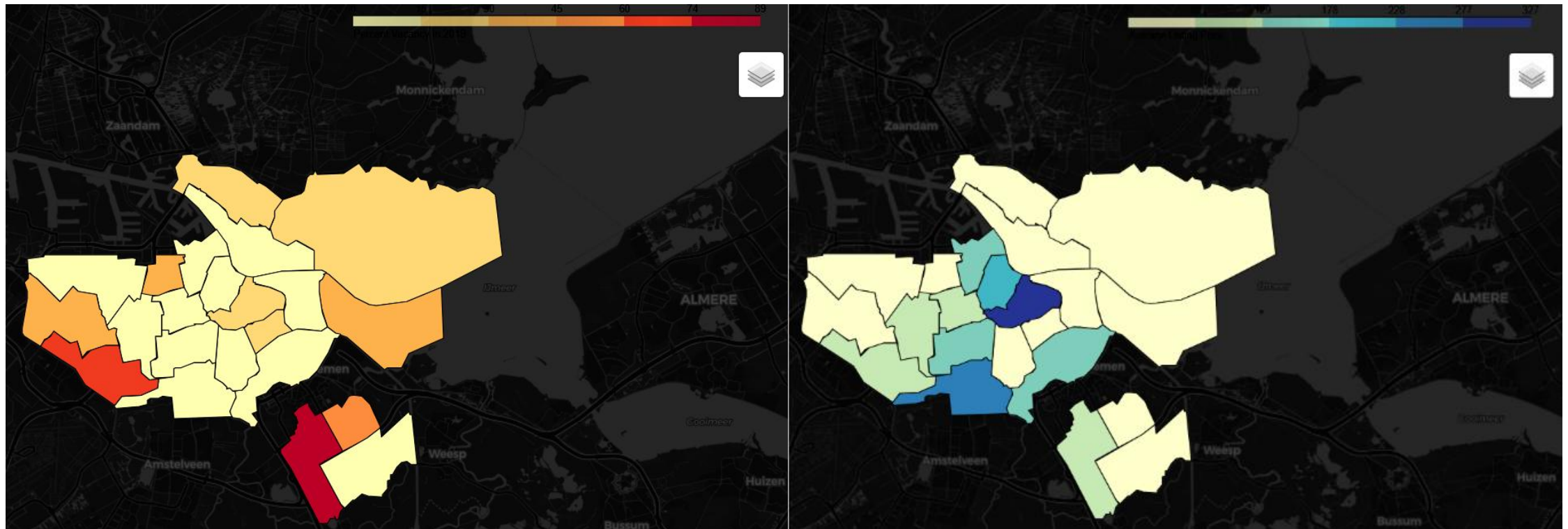
This histogram shows that the majority of listings are located between 1 and 3 kilometres from the city centre.



Geospatial Visualization

Tools used: Python (pandas, JSON, folium)

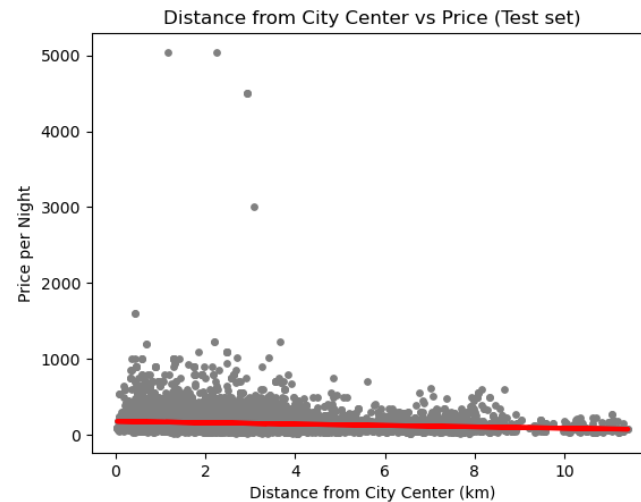
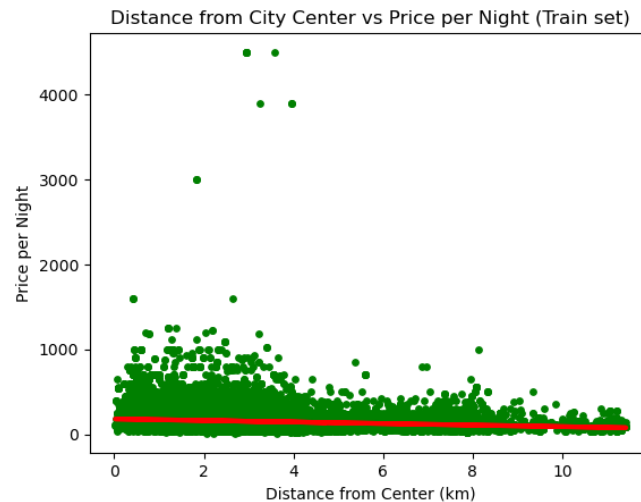
- Loaded the GEOJSON data into Python and created Folium choropleth maps showing the geospatial distribution of different variables (e.g. count of listings, availability rate, average price).
- The maps below show the distribution of availability (left) and average price (right) across the city by neighbourhood.



Linear Regression Analysis

Tools used: Python (pandas, numPy, matplotlib pyplot, Seaborn; SciKitLearn train-test-split, linear regression, metrics)

- Further explored the relationships between variables through supervised linear regression.
- The results of this method appeared to echo those of the initial statistical analysis: mostly weak correlations between variables and high levels of variance in the data.
- Overall, this method did not prove to be ideal for analyzing this particular set of data.



```
rmse = mean_squared_error(y_test, y_predicted)
r2 = r2_score(y_test, y_predicted)
```

```
print('Slope: ', regression.coef_)
print('Mean squared error: ', rmse)
print('R2 score: ', r2)
```

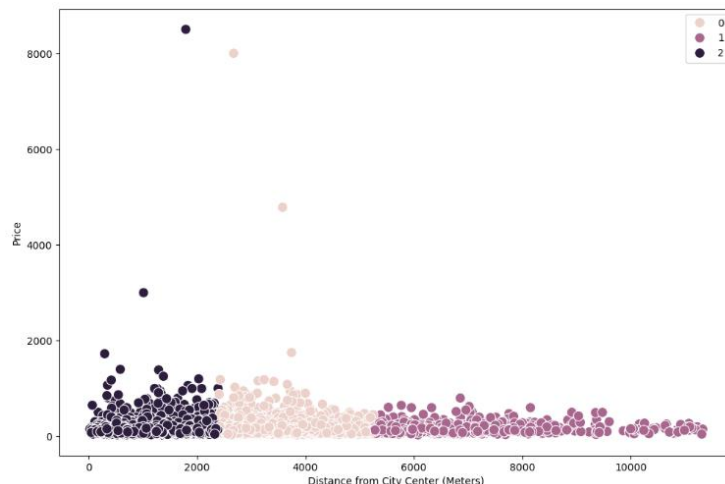
```
Slope: [[-9.07350788]]
Mean squared error: 14802.81802876858
R2 score: 0.017895992154792872
```

The high RMSE (root mean squared error) value and low R2 score indicate that there is too much variance in the data for a linear regression model to accurately explain the relationship between these variables.

K-Means Clustering Analysis

Tools used: Python (pandas, numPy, matplotlib pyplot, pylab, SciKitLearn KMeans clustering)

- Performed k-means clustering analysis on a selection of numerical columns. With k-means clustering, I attempted to sort the data into clusters based on their similarities.
 - The columns I used consisted of the average daily price, availability during the year, distance from the city centre, host listings counts and the review scores for cleanliness, location, value and overall.
 - Using scikitlearn's k-means clustering tool, I found that the data would be best split into 3 clusters.
- Identified the characteristics of each cluster with descriptive statistics for each variable and with scatterplots.

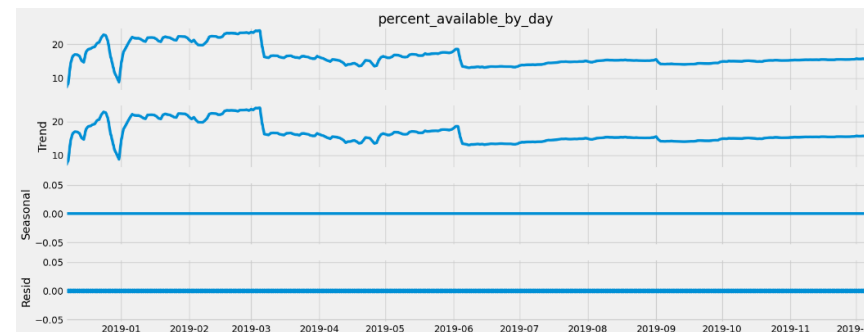
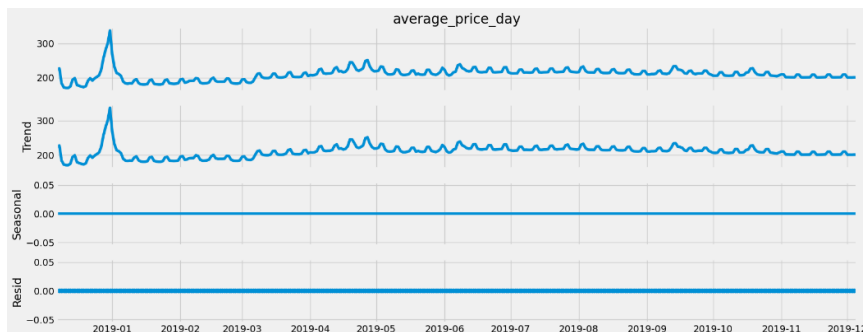
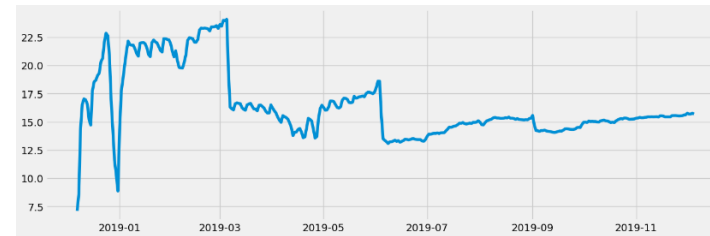
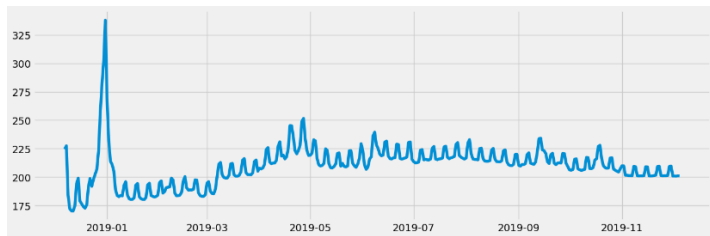


	cluster	cluster0	cluster1	cluster2
listing_avg_price_year	mean	175.069412	169.462147	175.127602
	median	145.623955	140.684211	146.899428
	min	26.514469	30.047619	25.333333
	max	8000.000000	799.000000	8500.000000
review_scores_rating	mean	95.313241	94.820300	95.166810
	median	97.000000	97.000000	97.000000
	min	20.000000	50.000000	20.000000
	max	100.000000	100.000000	100.000000
availability_percent	mean	30.014823	29.434048	30.375400
	median	15.890411	15.616438	16.986301
	min	0.273973	0.273973	0.273973
	max	100.000000	100.000000	100.000000
meters_from_center	mean	3261.361779	7277.666352	1534.195530
	median	3088.013709	7153.202037	1605.890884
	min	2396.979978	5267.184572	22.258728
	max	5266.973813	11326.606980	2396.801035

Time-Series Analysis

Tools used: Python (pandas, NumPy, matplotlib pyplot, pylab, statsmodels.api)

- This type of analysis looks at data points collected over time and attempts to identify patterns and trends that can be used to forecast future values.
- I decided to perform time-series analysis on the average price per night and the average availability rate per day across the city. This required me to create new variables by aggregating the daily prices and the average availability of all listings.
- Plotted the time-series data for both variables and noted the changes in their value over the year.
- Decomposition analysis results show the same trends throughout the year, including the sharp fluctuations during the first two months.



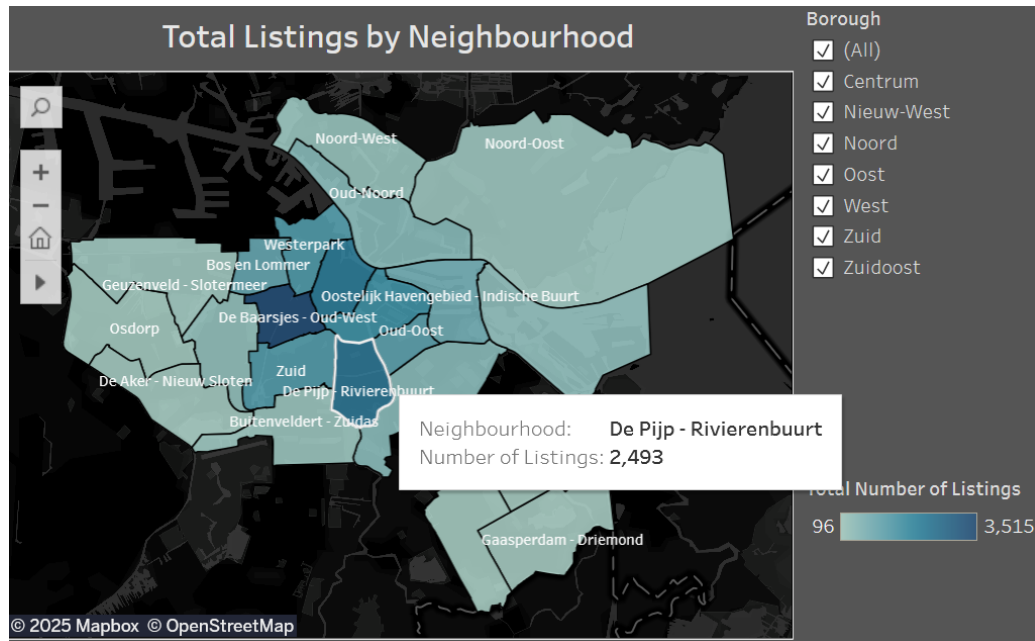
Dashboard Creation

Tools used: Tableau Public

- Created a dashboard on Tableau with the most relevant insights and findings from this project.
- Recreated many of the charts and visualizations and added interactive elements to them. Examples include the following:

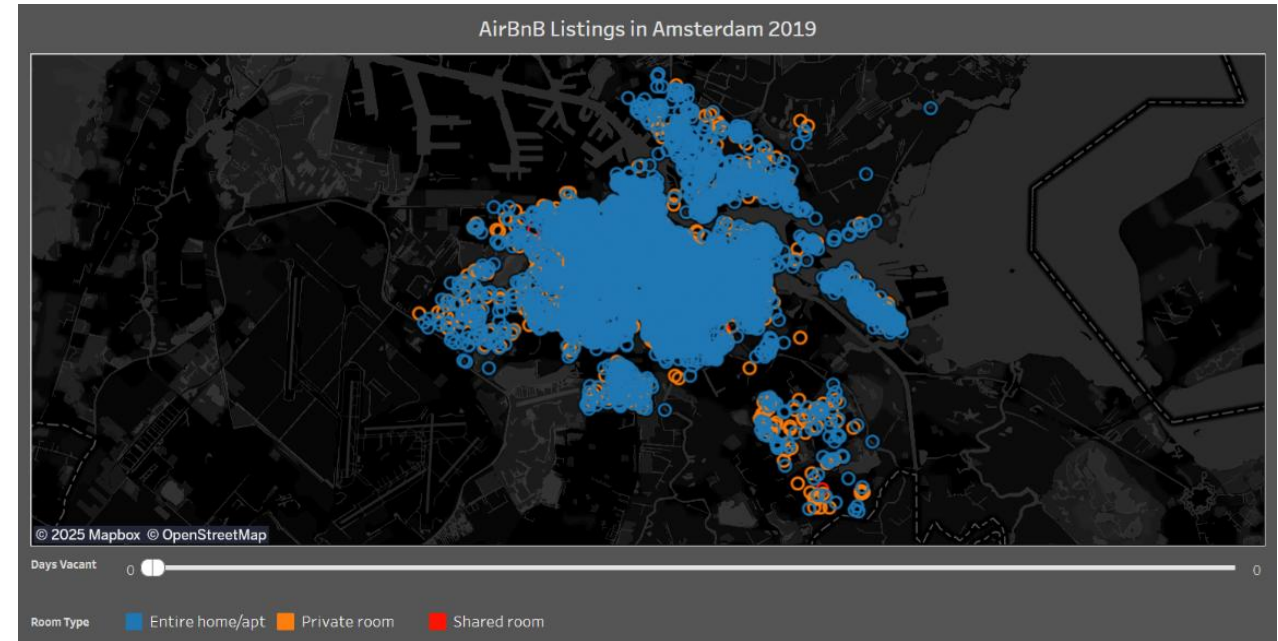
Allowed users to view detailed information in mouse-over popups.

Here we can see a popup highlighting the neighbourhood of De Pijp-Rivierenbuurt and the number of listings located there.



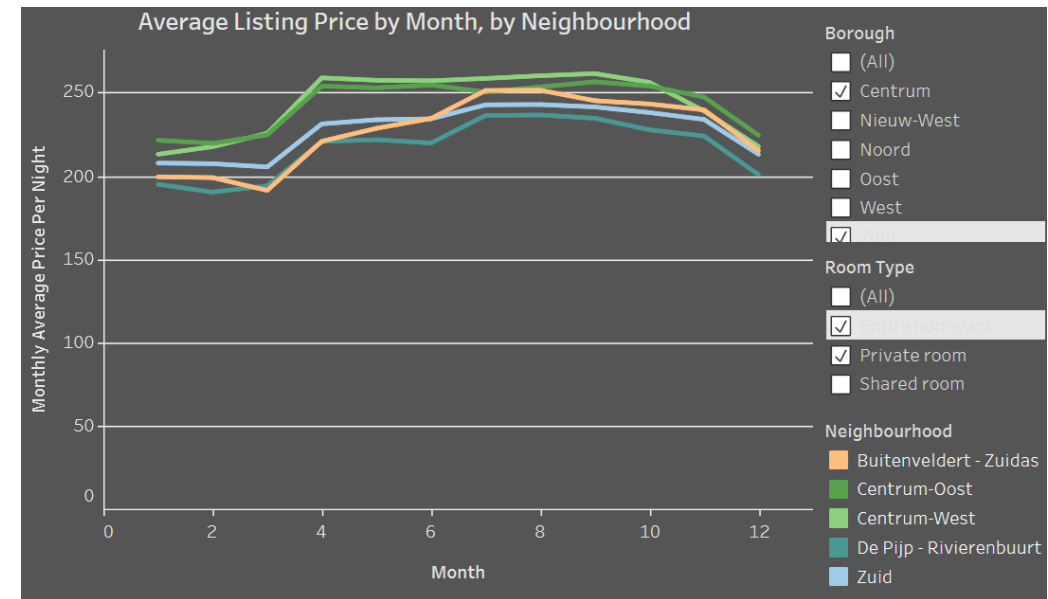
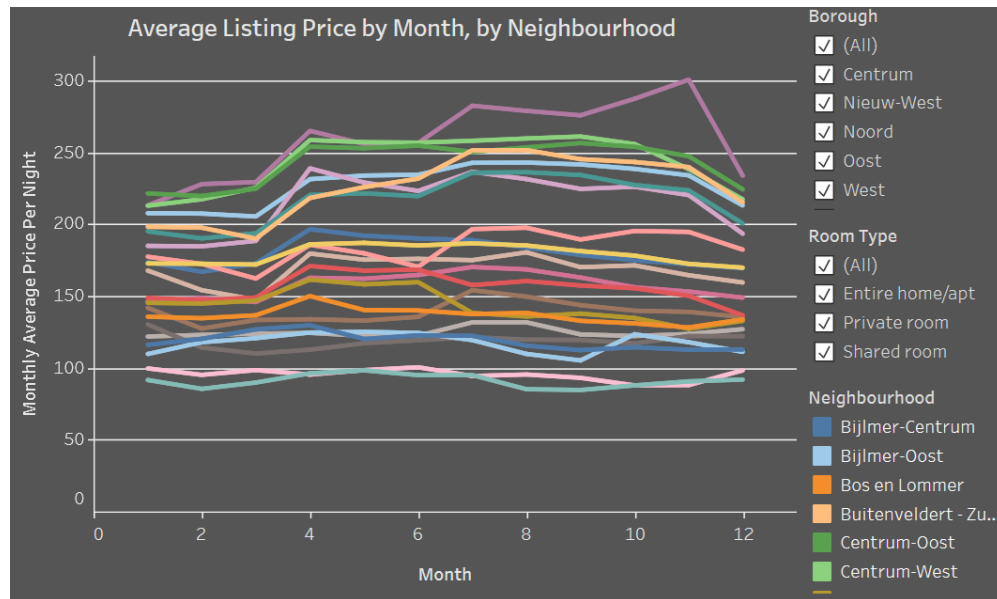
Added a slider to adjust the number of days vacant between 0 and 365, allowing the user to visualize the distribution of the most and least popular accommodations on the map

Here we can see a map of all the accommodations that are fully booked for 2019 with 0 days vacant.



Dashboard Creation

Recreated the time-series plots using aggregated values for each neighbourhood instead of for the whole city. Allowed users to select and view data for individual neighbourhood or borough, as well as for selected room-types.



Showing data for all neighbourhoods and room-types (left); showing data for entire homes and private rooms in neighbourhoods belonging to the boroughs of Centrum and Zuid (right).

Conclusion and Recommendations

- The most popular neighbourhoods in Amsterdam for short-term guests are located in the boroughs of Amsterdam-West and Amsterdam-Zuid.
- The very centre of the city has higher rates of vacancy than surrounding areas likely due to its higher-than-average prices.
- Residents of the most popular neighbourhoods can consider becoming an AirBnB host.
- The highest vacancy rates and lowest prices per night can be found in the outskirts of the city, in the boroughs of Amsterdam-Zuidoost and Nieuw-West.
- Guests consider cleanliness and value to be more important than distance from the city centre when choosing their accommodation. This is likely due to the fact that Amsterdam is a compact, densely-populated city with extensive public transit and bicycle infrastructure, making it easy to reach the city centre even from the outskirts.
- Vacancy peaks in January and February, much like many other European cities.
- The vast majority of Airbnb listings advertise entire homes and apartments. Shared rooms make up less than 1% of total Amsterdam listings.
- Properties offered by hosts with multiple listings are more likely to receive a low review score from guests. Visitors who book properties from these hosts should check the reviews to avoid potential disappointment.
- Hosts listing multiple accommodations for rent should put more effort into maintaining and caring for their properties in order to provide the best experience they can for their guests.

My Experience

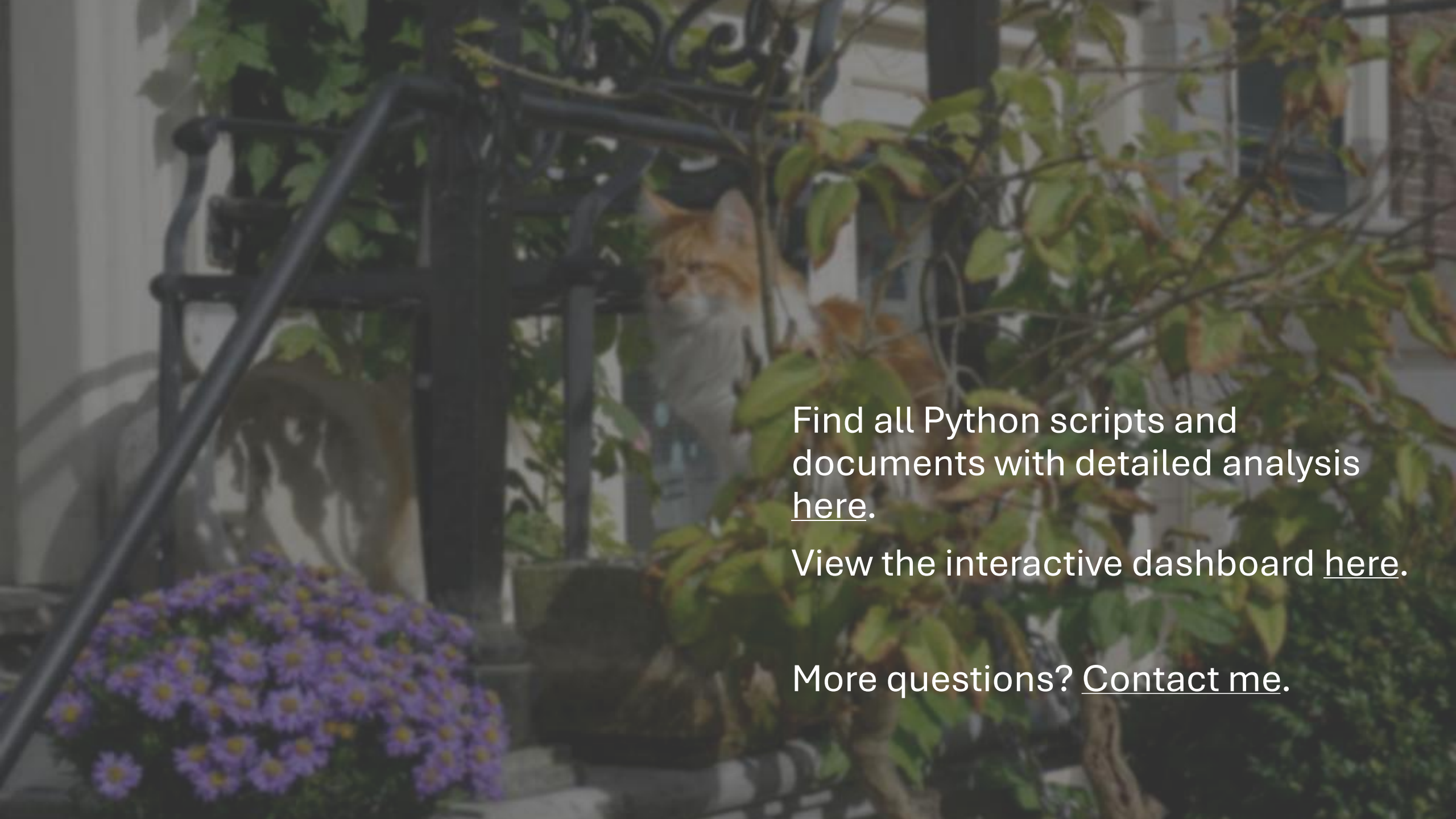
At the start of the project, **I intended to analyze the extensive text data included in the listings and reviews.** However, this proved to be beyond the scope of my skills and would have required significantly more time than I had, as I was not at all familiar with performing text analysis with Python.

Text analysis of the listing descriptions as well as the reviews would have provided me with significantly more information to answer questions such as the following:

- Which words are most commonly used by hosts to describe their listings and are these words echoed by guests in reviews?
- Are there certain words or phrases that can be associated with certain neighbourhoods? Using listing and review text, how can the city of Amsterdam be described by hosts and by visitors?
- What do guests say in reviews with the highest and lowest scores? How does this vary by location and by price?

Halfway through the further analysis, **I found that I had incorrectly joined two tables of cleaned data**, resulting in a table with multiple duplicate rows and incorrect aggregate values. This set me back several days, as I had to start with the raw data and redo all the data cleaning, wrangling and analysis. This experience reminded me to pay attention to details when performing a merge in Pandas and to make sure I have specified the right type of merge and the right column on which to join the two tables.

Ultimately, I was able to deliver a complete report with enough insights to form a few conclusions and provide some generic recommendations. Additionally, I was able to learn and apply new techniques in Python (e.g. using NumPy to calculate distance based on geographical coordinates) and in Tableau (e.g. creating a slider to adjust data displayed on a map).



Find all Python scripts and documents with detailed analysis [here](#).

View the interactive dashboard [here](#).

More questions? [Contact me](#).