fourBusiness Environments of Berlin Neighborhoods – Cluster Analysis

Project by Eric Becker as Capstone Project for the IBM Data Science Professional Certification

Full URL to the Jupyter Notebook for this project uploaded to Author's GitHub Repository:
https://github.com/enbecker5/Coursera_Capstone/blob/main/notebooks/berlin_business_env.ipynb

1. Introduction to the Project

2020 is a challenging year for starting a business, as the Covid-19 pandemic has reduced demand for most free-time activities as public restrictions limit allowed activities. These new factors, combined with the preexisting challenges associated with opening a business, make it necessary that a business's product or service is appropriately matched to the demographics and consumer preferences in its operational location. This project aims to support entrepreneurs by giving them a clearer understanding of the business environments for each district and neighborhood through clustering at the district and neighborhood levels according to the most popular business categories in each district. Cluster analysis of available data (described below) will allow entrepreneurs to select the appropriate location for their business to ensure it is suited to local tastes to increase chances of success.

Question: Which types of businesses/venues appear to be most common at the district and neighborhood levels throughout the city of Berlin?

2. Introduction to the Data

2.1 Overview of Datasets

- Foursquare Places API Venue Data (https://developer.foursquare.com/places): This API grants developers free access to a massive set of crowdsourced business listing and locational data. Foursquare powers location data for popular Apps such as Uber, Snapchat, Twitter, AppleMaps, and many more. Through gathering data on listed businesses in each Berlin district and neighborhood through calls to the Foursquare API, business data can be analyzed to provide insights into the district- and neighborhood-level business environments throughout the city of Berlin.
- Berlin Open Data Portal Datasets (https://daten.berlin.de/datensaetze): The Berlin government publishes and updates open source datasets covering a wide range of aspects to aid public and private analysis on trends and developments in the city.
    - Registered Residents in Berlin Districts on 31/12/2019 Dataset (https://www.statistik-berlin-brandenburg.de/opendata/EWR_Ortsteile_2019-12-31.csv) – the most recent update to this dataset which contains the number of residents broken down by age groups and at the district and neighborhood levels.
- The Open Data Information Agency of the Technology Foundation Berlin's Foundational Geodatasets of Berlin Portal (https://daten.odis-berlin.de): seeks to enable Berlin's up-

and-coming tech sector through providing a number of open source foundational Geodatasets to enable more consistent geospatial analysis and visualization.

- Berlin Districts Geospatial Dataset (https://daten.odis-berlin.de/de/dataset/bezirksgrenzen/): provides geospatial boundaries of Berlin's 12 administrative districts.
- Berlin Neighborhoods Geospatial Dataset(https://daten.odis-berlin.de/de/dataset/ortsteile/): provides geospatial boundaries of Berlin's 96 neighborhoods.

2.2 Using Datasets to Solve the Problem

1. The geospatial datasets can be used in visualizing districts and neighborhoods over maps of Berlin, as well as in calculating representative central points of each district or neighborhood in order to make calls to the Foursquare API to retrieve venue data for each district or neighborhood. Important data to be used is number and categories of venues.
2. The dataset on registered residents in each district can then be used to gain some insight into the demographics of each district or neighborhood according to the characteristics of population age.
3. After processing venue and demographic data, both regression and cluster analysis can be used to understand trends in the business environments of each district and neighborhood, and ultimately to cluster districts and neighborhoods according to the most popular venue categories. The data analysis and visualizations produced could then be used by entrepreneurs in exploring which districts or neighborhoods have the most appropriate demographics and consumer preferences for their planned business.

3. Methodology

3.1 Data Wrangling

Before analysis can be done, the datasets must be extracted and read into dataframes, cleaned for duplicates and errors, and merged into unified datasets when necessary.

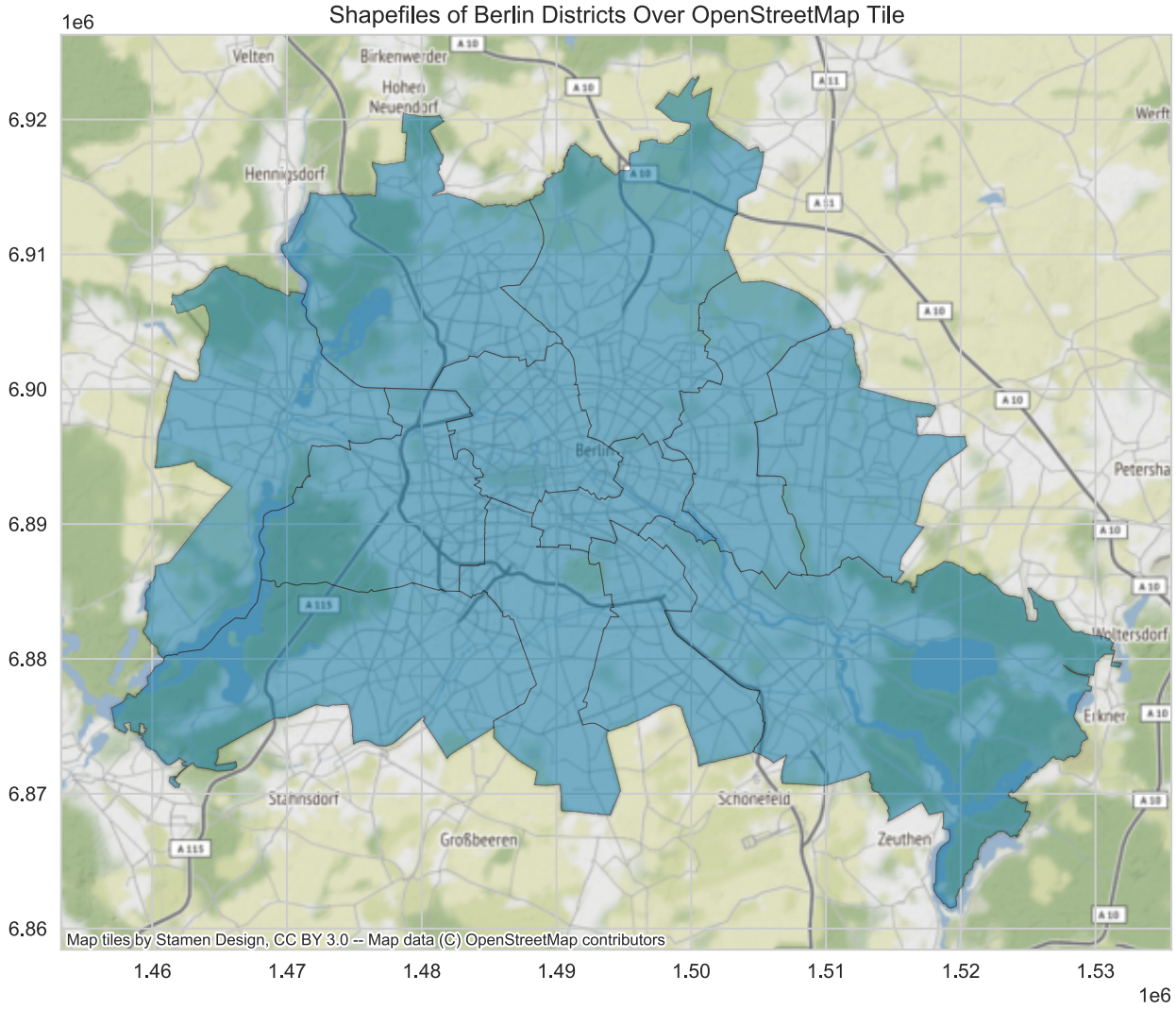
3.1.1: Geospatial Datasets

To make sure our calls to the *Foursquare API* return venue data that is representative of the geographical areas, central representative points are needed for each district and neighborhood. Luckily, the Python library *PyProj* simplifies the process of projecting & transforming geospatial data. The Shapely library allows us to calculate polygon area and their representative points, which were chosen over centroids since representative points are central points guaranteed to be within the boundaries of each polygon to account for irregular shapes. Area is calculated through transforming coordinates from the latitude/longitude coordinate reference system (CRS) to an Lambert Azimuthal Equal-Area Projection, which is useful in minimizing area distortions that come with navigation-oriented CRSs such as lat/long and UTM. Area results are then converted from m^2 to km^2 for improved readability. Representative points are in the standard lat/long CRS.

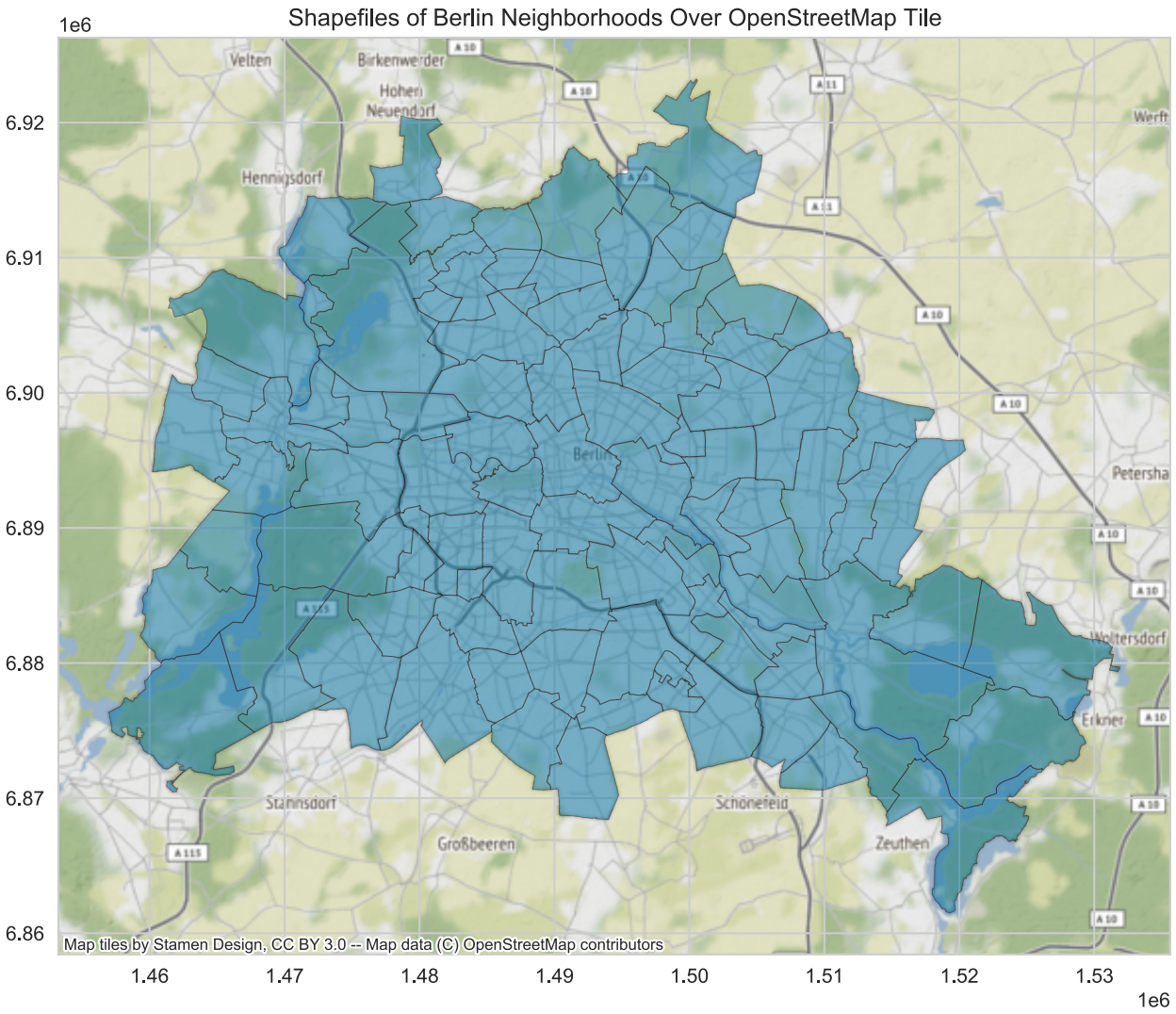
The 12 Districts of Berlin

Visualization: For visualization this project uses two libraries, *GeoPandas* and *Folium*. *Folium* is useful for interactive geospatial visualizations on open source map tiles such as *OpenStreetMap*, which gives us interactive visualizations. *GeoPandas*' visualization tools are well-suited for static images, such as the one generated below where the same GeoJSON file to generate a geodataframe, rather than a *Pandas* dataframe, that is used to visualize

Adding Layers: If we want to incorporate layers, such as a *OpenStreetMap* tiles under our static images of the Berlin area, this can be done in *GeoPandas* as well. Incorporating an additional library, *Contextily*, simplifies the process as it all us to easily generate a background map that is aligned with our geospatial data. The geospatial data is first converted to the Web Mercator CRS, so that our polygon and the map tiles are using the same CRS, after which we plot our *GeoPandas* polygon plot over the basemap layer using *Contextily*.



The 96 Neighborhoods of Berlin



3.1.2: Demographic Data

To give us additional insights into the business environments of each neighborhood or district, the registered residents in Berlin dataset is extracted and processed into a *Pandas* dataframe to find the demographics in the districts and neighborhoods along the characteristic of age. The original dataset is quite fine-grained, so it is aggregated into generation bins (Ages 0-15, 15-30, 30-50, 50-65, and 65+) in order to give us insights into generational differences in geographical areas. After compiling the dataframe, a new one is created where the count of each generational bin is divided by the total to understand the relative proportions of each age group for each district and neighborhood in order to understand the relative proportions of each bin for each district.

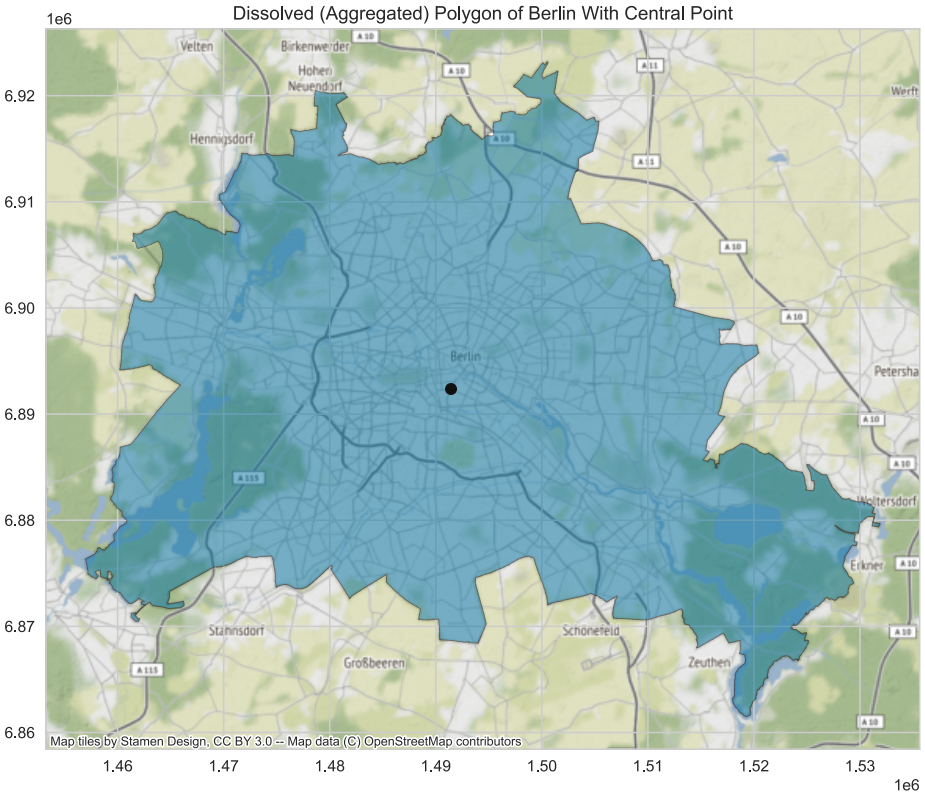
3.1.3 Foursquare API Venue Data for Districts and Neighborhoods

Conversions between Coordinate Reference Systems (CRSs) The peculiarities of each CRSs makes different CRSs useful in different situations. The standard latitude/longitude CRS is a geographic CRS, meaning it is referencing points on a globe. In practice, we deal with geospatial data more often in flat maps, meaning we need a Cartesian CRS to reference points projected onto a flat plane. UTM is the most widely used Cartesian CRS, making it the more appropriate CRS for our purposes. Luckily, both PyProj and GeoPandas (which is built on PyProj) makes creating conversion objects from the lat/long CRS to cartesian CRSs simple, only requiring a few lines of code, as shown throughout this section.

3.1.3.a Setting up a Sampling Grid

This is a grid of evenly spaced points that allows us to make calls to the Foursquare API throughout the entire city limits of Berlin, this is useful for detailed analysis since the FourSquare API has a limit of returning only 50 venues per point.

Finding the Center of Berlin First, we will need to aggregate (or "dissolve") our geodataframe of Berlin's districts to give us a polygon object that represents the outer borders of the city-state of Berlin. *Geopandas*, a high-level geospatial library built on pyproj, shapely, and others, makes the process of converting from long/lat to UTM xy coordinates quite simple, as demonstrated in section 3.1.3a of the published notebook.



Determining Appropriate Size of Sampling Grid: Our sampling grid should cover the entire area of Berlin, so to do this we will look at the bounds of our Berlin polygon that has been transformed into UTM Zone 33 North from before. Through dividing the length of Berlin's range along each axis and dividing it in half, we get the appropriate distances from our center point along each axis to define in the sampling grid. Calculated values for the sampling grid are listed below.

- Min X = 370000.83
- Max X = 415786.55
- Min Y = 5799520.66
- Max Y = 5837259.27
- The length of Berlin's X range is 45785.73 meters and Y range is 37738.61 meters.
- The sampling grid should be 22892.86 meters from the center along the X axis and 1886 9.30 meters from the center along the Y axis.

3.1.3b Making Calls to the Foursquare API to Extract Venue Data

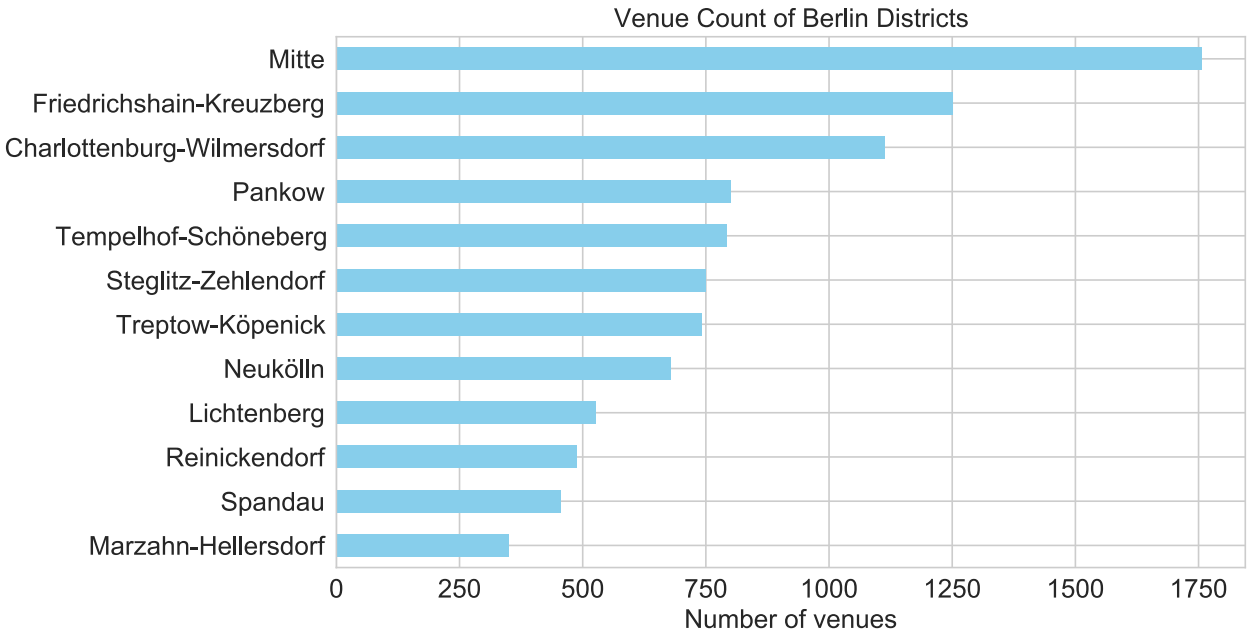
Through reading in personal Foursquare API credentials and defining a function that that uses the points on the sampling grid in a series of get requests to the Foursquare Places API while compiling the relevant venue data into a single dataframe. This function is first applied function to Berlin districts with a radius of 710m is used to ensure entire area is covered, after which duplicates are removed to ensure the returned dataset will be useful for this analysis. Due to the long execution time, the returned dataset from the initial extraction on 31/12/2020 at 13:34 was filtered for duplicates with a reset index to create the central dataframe used in the following analysis, which was saved to the local device. Code for both the initial extraction and loading the csv are shown in section 3.1.3b of the published notebook.

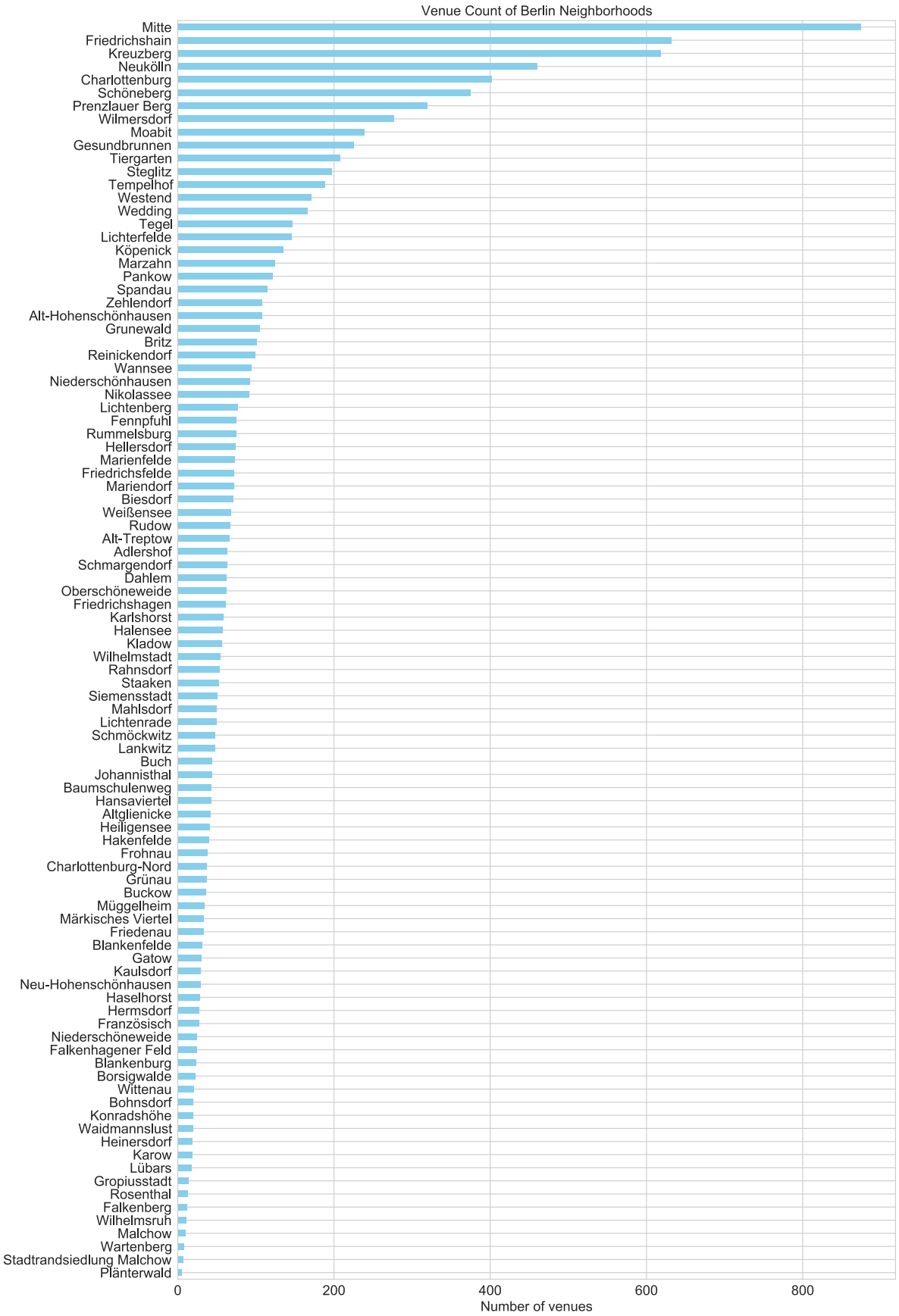
Heatmaps are a great visualization tool, so now that the venue dataset has been extracted a heatmap can be generated quite easily using Folium. Refer to section 3.1.3b of the published notebook to see the interactve heatmap generated by *Folium.*

3.2 Exploratory Data Analysis

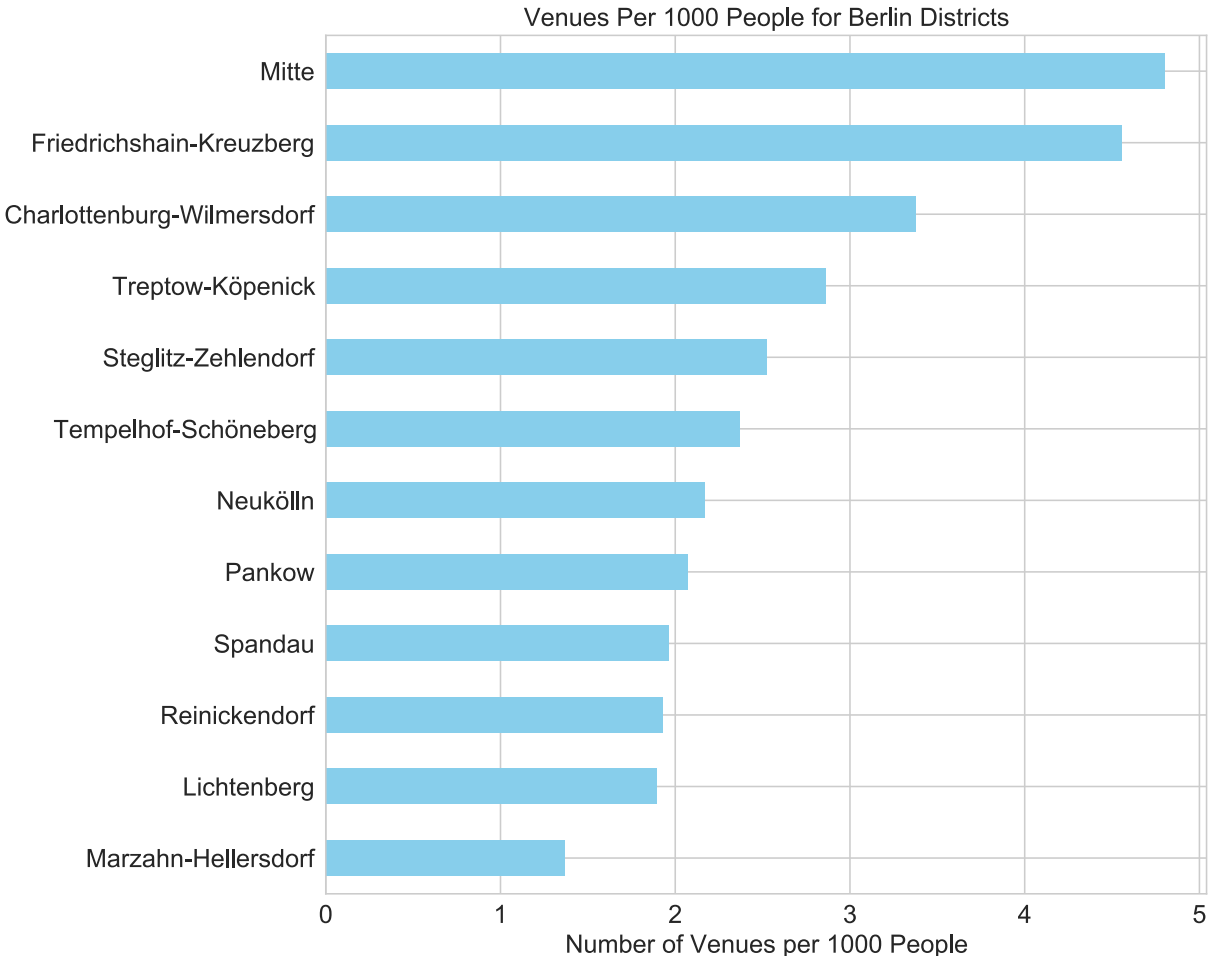
3.2.1 Bar Charts

These plots great for comparative visualization, providing an overview of the number of venues found within each district and neighborhood. The bar charts below show venue counts in the extracted data at the district and neighborhood level.

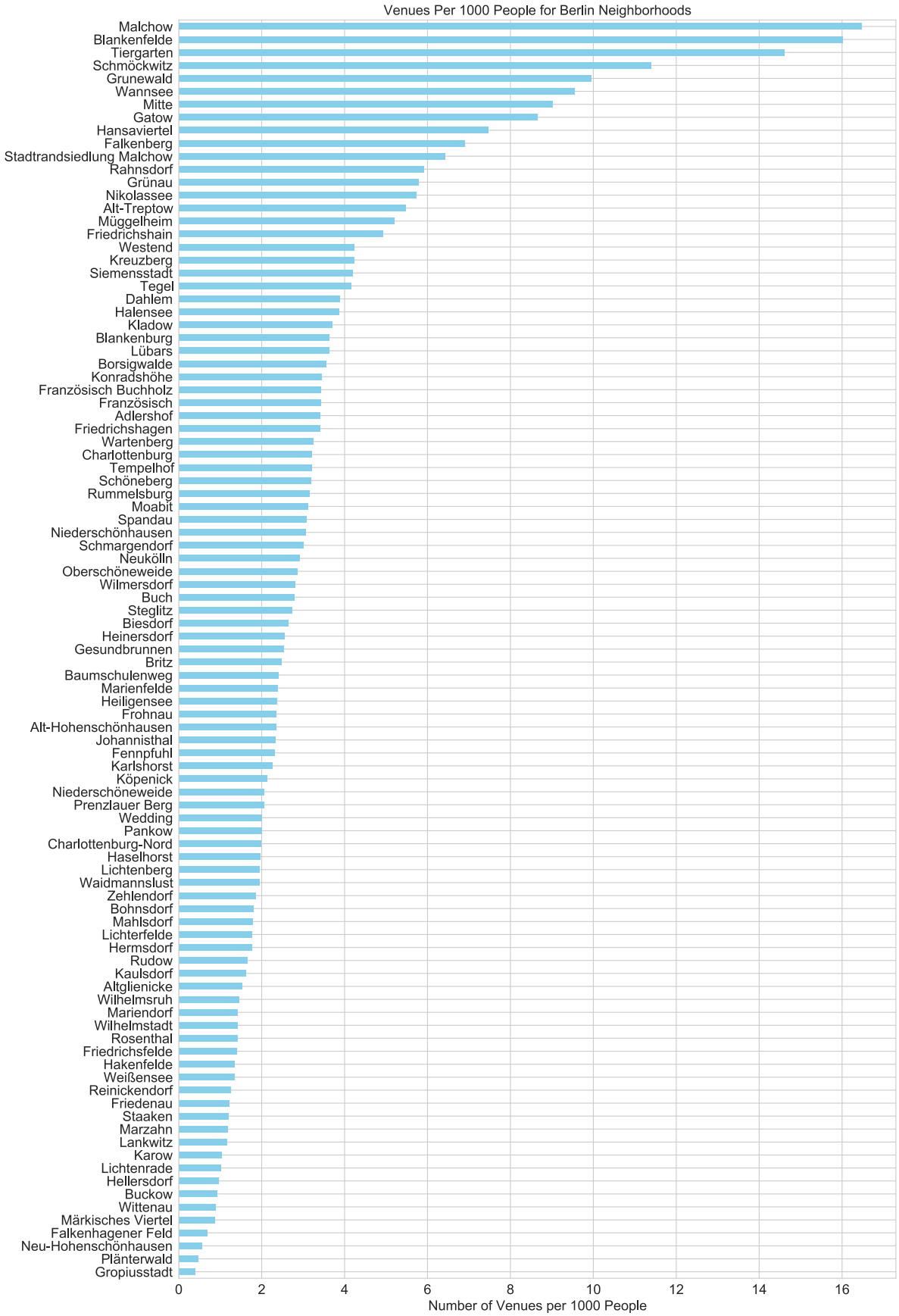

Venue Count of Berlin Districts
Mitte
Friedrichshain-Kreuzberg
Charlottenburg-Wilmersdorf
Pankow
Tempelhof-Schöneberg
Steglitz-Zehlendorf
Treptow-Köpenick
Neukölln
Lichtenberg
Reinickendorf
Spandau
Marzahn-Hellersdorf
0
250
500
750
1000
1250
1500
1750
Number of venues


Venue Count of Berlin Neighborhoods
Mitte
Friedrichshain
Kreuzberg
Neukölln
Charlottenburg
Schöneberg
Prenzlauer Berg
Wilmersdorf
Moabit
Gesundbrunnen
Tiergarten
Steglitz
Tempelhof
Westend
Wedding
Tegel
Lichterfelde
Köpenick
Marzahn
Pankow
Spandau
Zehlendorf
Alt-Hohenschönhausen
Grunewald
Britz
Reinickendorf
Wannsee
Niederschönhausen
Nikolassee
Lichtenberg
Fennpfuhl
Rummelsburg
Hellersdorf
Marienfelde
Friedrichsfelde
Mariendorf
Biesdorf
Weißensee
Rudow
Alt-Treptow
Adlershof
Schmargendorf
Dahlem
Oberschöneweide
Friedrichshagen
Karlshorst
Halensee
Kladow
Wilhelmstadt
Rahnsdorf
Staaken
Siemensstadt
Mahlsdorf
Lichtenrade
Schmöckwitz
Lankwitz
Buch
Johannisthal
Baumschulenweg
Hansaviertel
Altglienicke
Heiligensee
Hakenfelde
Frohnau
Charlottenburg-Nord
Grünau
Buckow
Müggelheim
Märkisches Viertel
Friedenau
Blankenfelde
Gatow
Kaulsdorf
Neu-Hohenschönhausen
Haselhorst
Hermsdorf
Französisch
Niederschöneweide
Falkenhagener Feld
Blankenburg
Borsigwalde
Wittenau
Bohnsdorf
Konradshöhe
Waidmannslust
Heinersdorf
Karow
Lübars
Gropiusstadt
Rosenthal
Falkenberg
Wilhelmsruh
Malchow
Wartenberg
Stadtrandsiedlung Malchow
Plänterwald
0
200
400
600
800
Number of venues

Our understanding can be improved through considering the number of venues relative to population, so we create a column in our dataframe with the values of venues per 1000 people and create a new bar chart.



NOTE: When plotting a horizontal bar chart at the neighborhood level (shown on the next page) we see some strange results, as the peripheral and less populated areas Malchow and Blankenfelde top the list in terms of venues per 1000 people, but upon closer inspection it is clear this is due to the total population of these neighborhood being 607 and 1998 residents, respectively. Areas such as these have a lot to offer Berlin's residents in terms of outdoor activities, thus resulting in the high number of venues relative to their population.

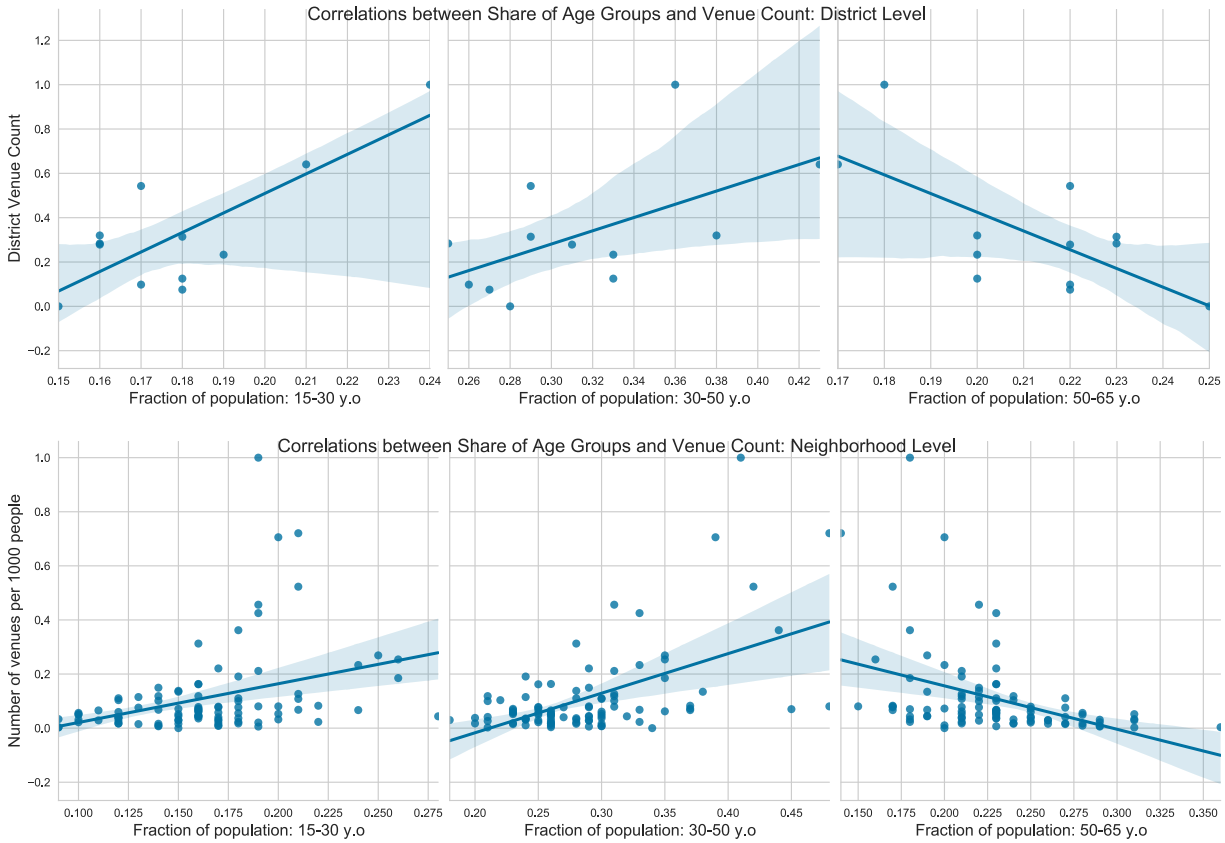

Venues Per 1000 People for Berlin Neighborhoods
Malchow
Blankenfelde
Tiergarten
Schmöckwitz
Grunewald
Wannsee
Mitte
Gatow
Hansaviertel
Falkenberg
Stadtrandsiedlung Malchow
Rahnsdorf
Grünau
Nikolassee
Alt-Treptow
Müggelheim
Friedrichshain
Westend
Kreuzberg
Siemensstadt
Tegel
Dahlem
Halensee
Kladow
Blankenburg
Lübars
Borsigwalde
Konradshöhe
Französisch Buchholz
Französisch
Adlershof
Friedrichshagen
Wartenberg
Charlottenburg
Tempelhof
Schöneberg
Rummelsburg
Moabit
Spandau
Niederschönhausen
Schmargendorf
Neukölln
Oberschöneweide
Wilmersdorf
Buch
Steglitz
Biesdorf
Heinersdorf
Gesundbrunnen
Britz
Baumschulenweg
Marienfelde
Heiligensee
Frohnau
Alt-Hohenschönhausen
Johannisthal
Fennpfuhl
Karlshorst
Köpenick
Niederschöneweide
Prenzlauer Berg
Wedding
Pankow
Charlottenburg-Nord
Haselhorst
Lichtenberg
Waidmannslust
Zehlendorf
Bohnsdorf
Mahlsdorf
Lichterfelde
Hermsdorf
Rudow
Kaulsdorf
Altglienicke
Wilhelmsruh
Mariendorf
Wilhelmstadt
Rosenthal
Friedrichsfelde
Hakenfelde
Weißensee
Reinickendorf
Friedenau
Staaken
Marzahn
Lankwitz
Karow
Lichtenrade
Hellersdorf
Buckow
Wittenau
Märkisches Viertel
Falkenhagener Feld
Neu-Hohenschönhausen
Plänterwald
Gropiusstadt
0
2
4
6
8
10
12
14
16
Number of Venues per 1000 People

Insights from Bar Charts: Through comparing our bar charts for venues per 1000 people at the district and neighborhood levels, the results are mostly in line with our expectations in that central areas have more venues relative to their population, but at the neighborhood level we see that certain neighborhoods with low populations may have more venues due to certain unique characteristics such as natural landscapes in close proximity to the city.

3.2.2 Regression Plots

These plots will help us observe potential correlations between two variables. Using regression plots we can regress VenPerPop against selected age demographic bins, which will give us some insight as to how relative age of local populations may affect the number of venues. This can be done quite simply using seaborn and matplotlib.

Normalizing Data: It is important to note that our age demographics data have been transformed to be their relative share to the total, where the sum of the age bins in each row is 1. Our venues per 1000 people and number of venues data have ranges much larger than 1, so in order to produce useful regressions, we must make sure all data is at similar scales to ensure we are making a meaningful comparison. To do this, we will use *Scikit-Learn*'s MinMaxScaler on both columns, which scales all values on a range of zero (the minimum value) to 1 (the maximum value), a process called *data normalization*, with the result seen when referring to the Y-axes on the figures below.



Insights from Regression Plots: We see some correlation between relative age share and number of venues at both the district and neighborhood levels. The large amount of heteroskedasticity (inconsistent variance) also tells us that while there is a correlation, regression analysis is not suited for estimation or prediction for this relationship without incorporating other explanatory variables. Such models are beyond the scope of this study, but the key insight here is that relatively younger districts and neighborhoods are ideal locations for opening a new business. This will prove useful as we interpret results of our cluster analysis below.

3.3 Modelling – Cluster Analysis

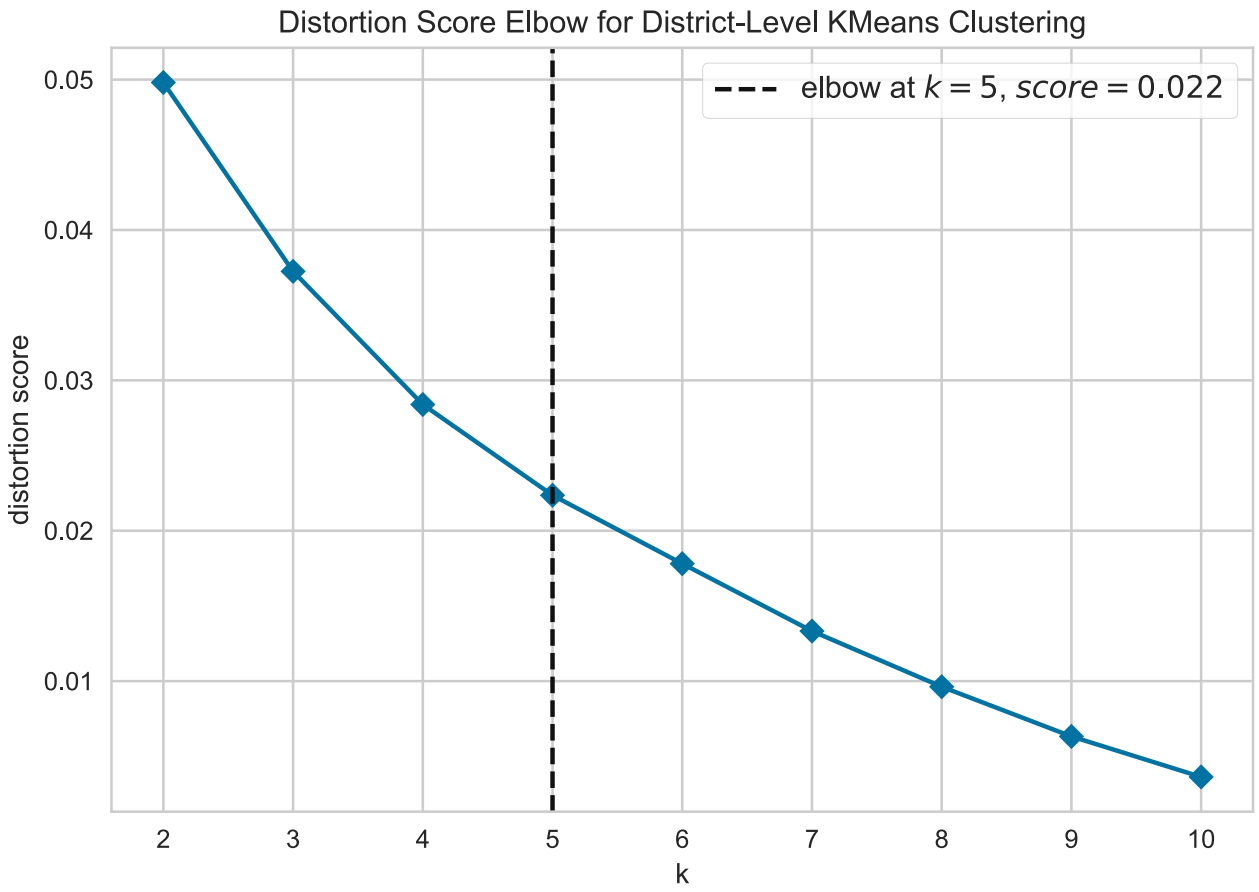
K-Means clustering models are well-suited for our analysis, since they will use the venue data to group districts and neighborhoods into clusters based on their similarity to one another. This will produce insights into the business environments for each district and neighborhood in Berlin through observing the most popular venues in each cluster.

3.3.1 Create Datasets to be Used in the Model

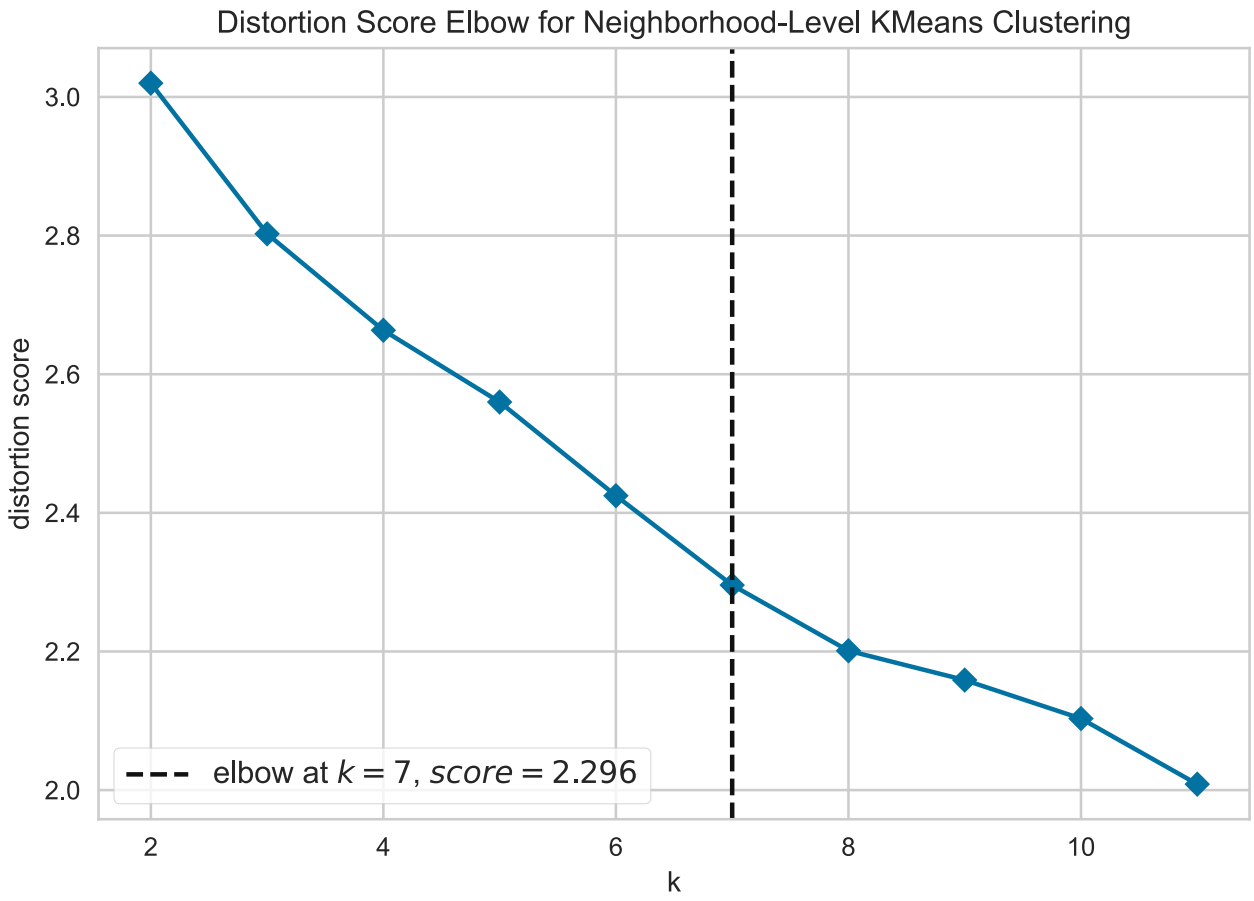
To begin developing our model, we will first need to transform categorical venue data to numerical values through the *one-hot encoding* method. After this, we will need to split our encoded dataset into two datasets, one for district-level analysis, and the other for neighborhood-level analysis.

3.3.2 Selecting Optimal Number of Clusters for Model

In K-Means Clustering, we need to select the most appropriate value for k to produce meaningful clusters. While explanatory value always increases *(through a decrease in distortion/error)* with a higher number of clusters, but we must not use too many clusters as this will result in overfitting, where the model is more tuned to the noise within the data than to the underlying relationships. The **elbow method** is a relatively simple heuristic evaluation method where the model is fitted using a range of values for k, and then plotting each k against its distortion score. The yellowbrick library provides several useful visualization tools for machine learning, and will be used in our Elbow evaluation for our neighborhood and district-level models.



While it is clear that the **elbow point at k = 5** is a relatively weak one, it is still clear that the decrease in distortion observed with each increase in K is relatively higher before 5 than it is after 5, making 5 the value of k we will use in the application of this model at the district level.



This **elbow point at k=7** is also a relatively weak elbow point, but there is still relatively less decrease in distortion beyond it so it will be used in application of this model at the neighborhood level.

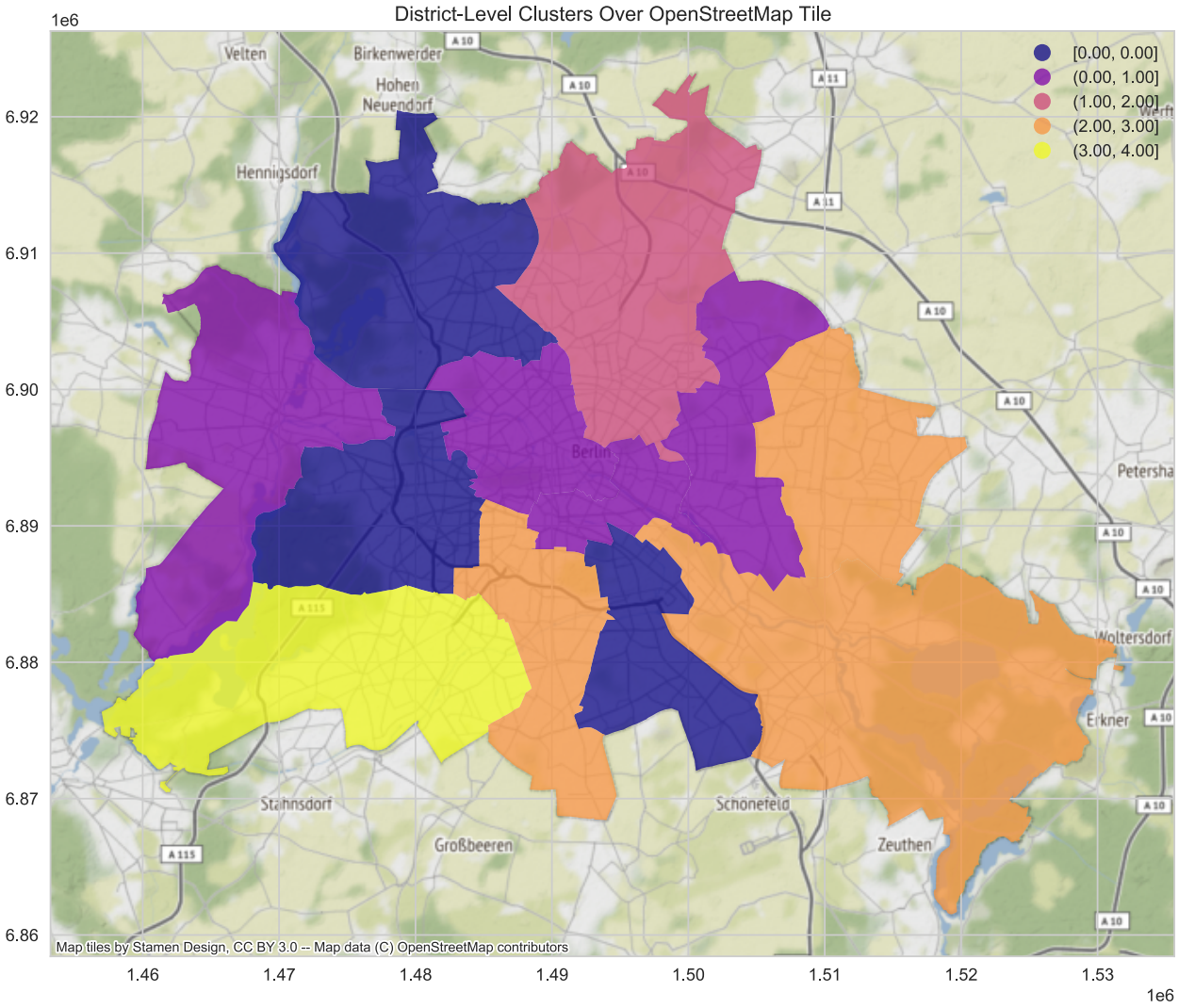
3.3.4 Application of District- and Neighborhood-level Models

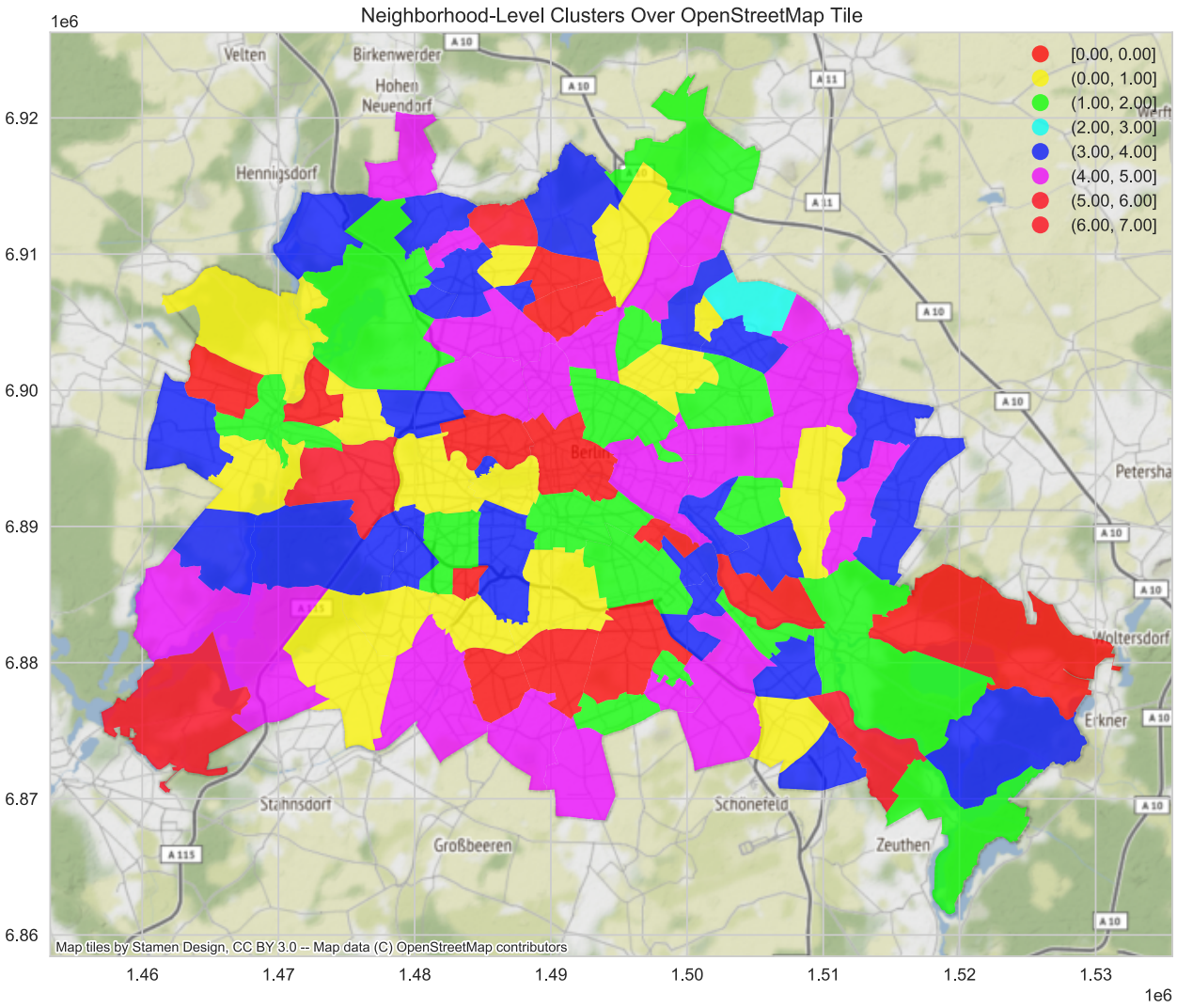
Using the values for k above to set the number of clusters, we apply out district- and neighborhood-level models while observing the most common venues in each model.

4. Exploration and Discussion of KMeans-Cluster Results

4.1 Visualizing Clusters in Choropleth Maps

This is a useful tool to help communicate the results of our analysis, and with GeoPandas visualizing our clusters is quite simple requiring only a few lines of code once we convert our dataframe to a GeoDataFrame.


District-Level Clusters Over OpenStreetMap Tile
1e6
6.92
6.91
6.90
6.89
6.88
6.87
6.86
1.46
1.47
1.48
1.49
1.50
1.51
1.52
1.53
1e6
[0.00, 0.00]
(0.00, 1.00]
(1.00, 2.00]
(2.00, 3.00]
(3.00, 4.00]
Velten
Birkenwerder
Hohen Neuendorf
Hennigsdorf
Berlin
Petersha
Woltersdorf
Erkner
Schönefeld
Zeuthen
Großbeeren
Stahnsdorf
A 10
A 11
A 115
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors



4.2 Selected District-Level Clusters

District Cluster One: Central Affluent Districts

This cluster represents the most central and affluent neighborhoods in Berlin. We can see the most common venue types are hotels, restaurants, cafes and bakeries, reflecting the numerous attractions and higher levels of disposable income in these districts. Refer to this section in the published notebook for a snapshot of the data.

District Cluster Two: Peripheral Districts
For contrast, we see that in the peripheral district supermarkets and tram stations top the ranking, implying that the areas are more residential with residents commuting to the city. Refer to section 4.2 in the published notebook for a snapshot of the data.

Selected Neighborhood-Level Clusters

Issue of Scarce Data While it was still possible to group clusters at the neighborhood level, the dependence of the Foursquare Places API data on user input has caused an issue of data scarcity, with some neighborhoods having less than 10 individual venue categories within them. This caused some clusters to not provide as much insight as was initially hoped for, but some clusters still provide useful insights into their respective business environments.

Cluster 6: Residential Consumer Preferences

We see the added benefits of observation at the neighborhood level, as we see more specific common venues in these residential districts such as pet stores, gardens, drugstores, and shopping malls along with the generally common venues such as supermarkets and bus stops. Refer to section 4.2 in the published notebook for a snapshot of the data.

5. Concluding Remarks

This project allowed for exploration of powerful Python libraries in the areas of geospatial data manipulation and visualization, statistics, and machine learning. Through applying these libraries on available data for cities such as Berlin, it is clear that useful insights into local environments can be generated. Despite issues cause by data scarcity, some generalizations about business environments could be made at the district and neighborhood levels.

Tourism-/Gastronomy-/Nightlife-oriented businesses are best suited in the clusters representing more central and affluent districts, as these sectors are generally more active in these areas.

Retail-/Grocery-/Family Needs-oriented businesses are more suited to peripheral and residential areas, as there is more activity in these areas.

These conclusions are highly generalized due to the general nature of this project's question, but the data and visualizations generated in this assignment can be used to provide tailored advice to entrepreneurs with a specific business idea in mind.