## Peer-graded Assignment: Capstone Project - The Battle of Neighborhoods

# Best Seattle Neighborhood for Drinkers

1. **Introduction where you discuss the business problem and who would be interested in this project.**

2. **Data where you describe the data that will be used to solve the problem and the source of the data.**

3. **Methodology section which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, and what machine learnings were used and why.**

4. **Results section where you discuss the results.**

5. **Discussion section where you discuss any observations you noted and any recommendations you can make based on the results.**

6. **Conclusion section where you conclude the report.**

## 1. Introduction where you discuss the business problem and who would be interested in this project.

Young tech workers love to party. Work hard, play hard is the mantra. And because most tech workers are mobile and affluent enough to choose what neighborhood they live in, one of the criteria for choosing a neighborhood is the availability of drinking establishments within walking distance of the neighborhood.

It is well known that Seattle is well endowed with breweries, wineries, and distilleries. Seattle has sufficient public transportation, car services, etc. so life without a car is preferable. Ideally you can choose to live in the neighborhood where you party and take public transportation, bike, or walk to your office (if you don't just work at home.) And when the party is over, you don't want to drive anyway.

Most apartment search applications like Craig's List and Apartment List categorize available apartments by neighborhood. So once you determine what neighborhood you want to live in, it's easy to find available units.

If drinking is your thing, then how do you choose the best neighborhoods? I propose studying which neighborhoods have the most drinking establishments and exploring the concept of a Drunk Scoresimilar to a Walk Score, Transit Score or Bike Score used by apartment hunting apps. This will be similar to a Walk Score but only take into consideration Bars, Lounges, Restaurants, Clubs, Cafes, where alcohol is served. I will not include package retail. Anyone can drink at home.

I was originally planning on analyzing Seattle and Portland because they are very similar and someone moving to the Pacific Northwest would probably consider both. However, it turned out the data wrangling for this project was way more complex than I expected so I reduced the scope to Seattle only.

## 2. Data where you describe the data that will be used to solve the problem and the source of the data.

For the purposes of this study, Foresquare is not specific enough. For example, there is a Starbucks next to me where they also serve craft beer and wine. Or a Whole Foods will have a beer or wine tasting bar. I will get the information about locations from the respective state agencies:

1. Washington State Liquor and Cannabis Board

```
# WAOnPrem = pd.read_excel("OnPremise.xls")
WAOnPrem.head()
```

[6]:

| | Tradename | License Number | Unnamed: 2 | ID Number | Unnamed: 4 | Loc Address | Loc Room | Loc City | Loc St | Loc Zip | ... | Business Startup Date | Issue Date | Expire Date | Privilege | Application Date | Reason Description | County |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 'AMONOS! MEXICAN KITCHEN | 409730 | NaN | 6033641060010001 | NaN | 15321 MAIN ST NE STE 201 | | DUVALL | WA | 980198574 | ... | 20180327 | 20190207 | 20200131 | BEER/WINE REST - BEER/WINE | 20190207 | RENEWAL | KING |
| 1 | /NA:M/ THAI CUISINE | 76997 | NaN | 6028006010010001 | NaN | 1404 34TH AVE | | SEATTLE | WA | 981223334 | ... | 20090608 | 20190212 | 20200229 | SPIRITS/BR/WN REST LOUNGE + | 20190212 | RENEWAL | KING |
| 2 | 00 PIZZA | 409193 | NaN | 6029630190010002 | NaN | 31217 124TH AVE SE STE 6 | | AUBURN | WA | 980923618 | ... | 20121015 | 20171104 | 20181031 | BEER/WINE REST - BEER/WINE | 20171104 | RENEWAL | KING |
| 3 | 028 BARBER SHOP | 426995 | NaN | 6042693900010001 | NaN | 14031 NE WOODINVILLE DUVALL RD | | WOODINVILLE | WA | 980728504 | ... | 20181115 | 20190306 | 20200331 | SNACK BAR | 20190306 | RENEWAL | KING |
| 4 | 101 BAR & GRILL | 353737 | NaN | 6028914650010001 | NaN | 322 LINCOLN ST | | HOQUIAM | WA | 985501409 | ... | 20090317 | 20190129 | 20200131 | SPIRITS/BR/WN REST LOUNGE - | 20190129 | RENEWAL | GRAYS HARBOR |

It turns out Washington State has a downloadable excel file indicating the license holders. Both these files have addresses but do not have latitude longitude. I will need to convert that information.

Luckily, Washington State also provides neighborhood information as well. Seattle provides latitude longitude.

```
neighborhoods = pd.read_csv('seattle_data.csv')
neighborhoods.head()
```

Out[5]:

| | X | Y | OBJECTID | FEATURE_ID | CODE | NAME | ADDRESS | NEIGH_NUM | XCOORD | YCOORD | LONGITUDE | LATITUDE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -122.355548 | 47.723233 | 1 | 90001 | Community Area/Business Center/Neighborhood | Bitter Lake Neighborhood | N 130th St & Greenwood Ave N | 3 | 1.265776e+06 | 267518.406033 | -122.355550 | 47.723232 |
| 1 | -122.323274 | 47.708595 | 2 | 90002 | Community Area/Business Center/Neighborhood | Northgate Neighborhood | NE Northgate Way & 5th Ave NE | 28 | 1.273615e+06 | 262024.468858 | -122.323276 | 47.708593 |
| 2 | -122.295226 | 47.719280 | 3 | 90003 | Community Area/Business Center/Neighborhood | Lake City Neighborhood | NE 125th & Lake City Way NE | 21 | 1.280593e+06 | 265788.625044 | -122.295228 | 47.719278 |
| 3 | -122.229209 | 47.727482 | 4 | 90004 | Community Area/Business Center/Neighborhood | Finn Hill Neighborhood | NE 141st St & 84th Ave NE | 13 | 1.296895e+06 | 268478.406082 | -122.229210 | 47.727480 |
| 4 | -122.290272 | 47.675785 | 5 | 90005 | Community Area/Business Center/Neighborhood | Wedgewood Neighborhood | NE 65th St & 35th Ave NE | 39 | 1.281512e+06 | 249903.141067 | -122.290273 | 47.675783 |

If I were generating an API similar to traditional Walk Score apps, I would be able to provide the model with the latitude longitude of a specific apartment building and generate a Drunk score. But instead, I will use the location data of the center of the neighborhoods in order to come up with a neighborhood score.

I will also need to make an assumption what is a drunken stumble distance. I will use a half mile for my initial analysis but may change that based on the results.

## 3. Methodology section which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, and what machine learnings were used and why.

### Overview of the Process

As mentioned earlier, I was going to do the analysis for both Seattle and Portland. I had downloaded Portland Neighborhood shapefile and State of Oregon liquor licensing files. But because of time and nuanced differences in the file structures I decided to limit my analysis for Seattle only. I had created a task list and from a high level kept to the sequence. However as I proceeded, I realized that my initial plan for analysis was flawed. I determined that DBSCAN was the best method to cluster drinking establishments.

```
db = DBSCAN(eps=epsilon, min_samples=30, algorithm='ball_tree',
metric='haversine').fit(np.radians(coords))
```

whereas K-Means was not apropos. Also after playing around with epsilon (distance) and min_samples, I got to 13 clusters that made sense given my understanding of the neighborhoods.

> **Of note:** I have created many notebooks where I do some work and then write the results to Cloud Object Storage (COS.) I then read those files into other notebooks. So there is not one notebook which runs straight through. Also, because of fits and starts, some of the relevant notebooks have abandoned, duplicated, or repetitively re-used code that does not flow sequentially from top to bottom. I have provided links to the code. And have highlighted key code and results in this discussion. But I have not completely commented the code itself.

### Technical Difficulties

Because both Washington and Oregon used ArcGIS as a repository for their neighborhood and liquor licensing files, and becuase IBM has a formal relationship with ESRI, I decided to use ESRI ArcGIS for my Geocoding. I signed up for a developer account and used the service to geocode King County. This was a steep learning curve. And there were account issues that delayed my proceeding with the geocoding for a week or so.

I also decided to use Geopandas and Geoplot for my data wrangling. These were able to read the shape files and made maping easy. However, these tools presented another steep learning curve.

Probably the most frustrating technical issue was getting ArcGIS and Geopandas to load in IBM Watson Studio Notebook. Whereas the Coursera modules explained how to use !conda and !pipto load packages, Watson Studio provides custom environments where your modules are automatically loaded at initiation of the runtime. Even then there was an open source conflict loading geopandas 0.5.1 and fiona 1.8.6 in Watson Studio. After opening a case with IBM I was able

to resolve the conflict by loading geopandas 0.5.1 via the custom environment and then using ! pip to load fiona 1.8.6.

```
channels:
- conda-forge
- defaults
# Please add conda packages here
dependencies:
#- fiona=1.8.6
- geopandas=0.5.1
- geoplot=0.3.0
- pysal=1.14.4
- folium=0.7.0
- python=3.6.8
- nodejs=11.14.0
- ipywidgets=7.5.0
# Please add pip packages here
# To add pip packages, please comment out the next line
#- pip:
```

As a result of these environment building issues, I had to figure out how to store results in Cloud Object Storage. Another learning curve.

**Process**

**Downloading from liquor licensing public records**

**Notebook**

The Washington State licensing data is located here:

```
!wget -O OnPremise.xls https://lcb.wa.gov/sites/default/files/publications/Public_Records/2019/On%20Premise.xls
```

Created Pandas Dataframe. Eliminated unnecessary columns. Reduced the file to King County only. And then created a column called "Premises" which concatenated the address fields which will be used for geocoding:

```
WA_KING['Premises'] = WA_KING['Loc Address'].astype(str) + ' , ' +  WA_KING['Loc City'] + ' , ' + WA_KING['Loc St']
WA_KING.head()
```

| | Tradename | Loc Address | Loc City | Loc St | Loc Zip | Business Startup Date | Privilege | County | Status | Premises |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 'AMONOSI MEXICAN KITCHEN | 15321 MAIN ST NE STE 201 | DUVALL | WA | 980198574 | 20180327 | BEER/WINE REST - BEER/WINE | KING | ACTIVE (ISSUED) | 15321 MAIN ST NE STE 201 , DUVALL ... |
| 1 | /NA:M/ THAI CUISINE | 1404 34TH AVE | SEATTLE | WA | 981223334 | 20090608 | SPIRITS/BR/WN REST LOUNGE + | KING | ACTIVE (ISSUED) | 1404 34TH AVE , SEATTLE ... |
| 2 | 00 PIZZA | 31217 124TH AVE SE STE 6 | AUBURN | WA | 980923618 | 20121015 | BEER/WINE REST - BEER/WINE | KING | ACTIVE (ISSUED) | 31217 124TH AVE SE STE 6 , AUBURN ... |
| 3 | 028 BARBER SHOP | 14031 NE WOODINVILLE DUVALL RD | WOODINVILLE | WA | 980728504 | 20181115 | SNACK BAR | KING | ACTIVE (ISSUED) | 14031 NE WOODINVILLE DUVALL RD , WOODINVILLE ... |
| 7 | 108 VIETNAMESE CAJUN CRAWFISH BROILER | 18114 E VALLEY HWY | KENT | WA | 980321001 | 20141105 | BEER/WINE REST - BEER | KING | ACTIVE (ISSUED) | 18114 E VALLEY HWY , KENT ... |

After re-indexing, create csv and upload to COS.

### Geocoding locations

## Notebook

Download WA_KING.csv and create Dataframe. Then extract a list of only the Premises addresses and eliminate duplicates (because geocoding costs money.)

```
PremisesList = WA_KING['Premises'].tolist()
PremisesList = list(set(PremisesList))
```

ESRI ArcGIS allows for bulk geocoding. They recommend about 150 requests at a time. So the list needs to be broken into chunks of 150. I created an array of 24 lists:

```
# How many elements each
# list should have
n = 150

# using list comprehension
PremisesListChunk = [PremisesList[i * n:(i + 1) * n] for i in range((len(PremisesList) + n - 1) // n )]
```

Then by hand I fed each of the 24 lists to ArcGIS. An initial attempt with the Portland data made me exceed my monthly allocation. So it took some correspondence with customer service to fix the problem. I spend about $12 of my own money getting this done. The ArcGIS returns json so I used json_normalize to convert back to a Dataframe. After defining my credentials this is the code that called the ArcGIS API:

```
results = batch_geocode(PremisesListChunk[24])
jresult = json_normalize(results)
```

ArcGIS returns 62 columns of stuff. After eliminating information irrelevant to this analysis I need to concatenate the original query from the list so I can do a merge with the Drinking Establishment dataframe.

```
PremisesDFChunk = pd.DataFrame(PremisesListChunk[24])
frames = [PremisesDFChunk, jresult]
dfsplitconcat = pd.concat(frames, axis=1)
```

This created 24 dataframes which I then concatenated into one.

```
WA_Locations =
pd.concat([WA_0,WA_1,WA_1,WA_2,WA_3,WA_4,WA_5,WA_6,WA_7,WA_8,WA_9,WA_10,WA_11,WA_12,WA_13,WA_14,WA_15,WA_16,WA_17,WA_18,WA_19,WA_20,WA_21,WA_22,WA_23,WA_24])
```

Next, I needed to merge the geocoded information with the original Drinking Establishment dataframe. This will restore the duplicates that were eliminated for geocoding. 3834 geocoded locations were merged with 4152 drinking establishments. (I made the assumption that if a drinking establishment had more than one license at a location both licenses should count. As a drinker, having more options at a location is a benefit.)

```
WA_KING_merge = WA_KING.merge(WA_Locations, left_on='Premises', right_on='0')
```

I uploaded the merged file to COS as a csv.

### Reviewing Geocoded Data

## Notebook

You wouldn't expect all the geocoded information form ArcGIS to be accurate.

| | Tradename | Loc Address | Loc City | Loc St | Loc Zip | Business Startup Date | Privilege | County | Status | Premises | 0 | address | attributes.MetroArea | attributes.Nbrhd | attributes.Subregion | location.x | location.y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 'AMONOS! MEXICAN KITCHEN | 15321 MAIN ST NE STE 201 | DUVALL | WA | 980198574 | 20180327 | BEER/WINE REST - BEER/WINE | KING | ACTIVE (ISSUED) | 15321 MAIN ST NE STE 201, DUVALL ... | 15321 MAIN ST NE STE 201, DUVALL ... | 15321 Main St NE, Duvall, Washington, 98019 | Seattle Metro Area | NaN | King County | -121.986716 | 47.738966 |
| 1 | /NA:M/ THAI CUISINE | 1404 34TH AVE | SEATTLE | WA | 981223334 | 20090608 | SPIRITS/BR/WN REST LOUNGE + | KING | ACTIVE (ISSUED) | 1404 34TH AVE, SEATTLE ... | 1404 34TH AVE, SEATTLE ... | 1404 34th Ave, Seattle, Washington, 98122 | Seattle Metro Area | Madrona | King County | -122.289263 | 47.613085 |
| 2 | 00 PIZZA | 31217 124TH AVE SE STE 6 | AUBURN | WA | 980923618 | 20121015 | BEER/WINE REST - BEER/WINE | KING | ACTIVE (ISSUED) | 31217 124TH AVE SE STE 6, AUBURN ... | 31217 124TH AVE SE STE 6, AUBURN ... | 31217 124th Ave SE, #6, Auburn, Washington, 98092 | Seattle Metro Area | Lea Hill | King County | -122.176475 | 47.321882 |
| 3 | 028 BARBER SHOP | 14031 NE WOODINVILLE DUVALL RD | WOODINVILLE | WA | 980728504 | 20181115 | SNACK BAR | KING | ACTIVE (ISSUED) | 14031 NE WOODINVILLE DUVALL RD, WOODINVILLE ... | 14031 NE WOODINVILLE DUVALL RD, WOODINVILLE ... | 14031 NE Woodinville Duvall Rd, Woodinville, W... | Seattle Metro Area | Town Center | King County | -122.151612 | 47.754461 |
| 4 | 108 VIETNAMESE CAJUN CRAWFISH BROILER | 18114 E VALLEY HWY | KENT | WA | 980321001 | 20141105 | BEER/WINE REST - BEER | KING | ACTIVE (ISSUED) | 18114 E VALLEY HWY, KENT ... | 18114 E VALLEY HWY, KENT ... | 18114 E Valley Hwy, Kent, Washington, 98032 | Seattle Metro Area | NaN | King County | -122.221090 | 47.439955 |

First I convert the dataframe to a geodataframe using Geopandas. This converts the long and lat columns to location objects. The crs provides the spacial reference to actual locations on earth.

```
geometry = [Point(xy) for xy in zip(WA_KING_merge["long"],WA_KING_merge["lat"] )]
crs = {'init': 'epsg:2285'} #http://www.spatialreference.org/ref/epsg/2285/
WA_gdf = GeoDataFrame(WA_KING_merge,crs=crs, geometry=geometry)
```

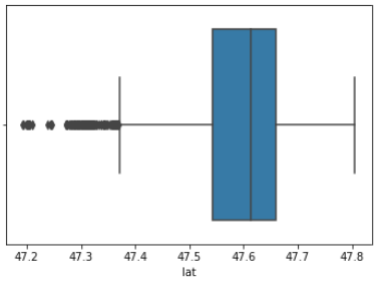
Then I look to see that a long and lat was assigned to each location.

```
missing_data = WA_gdf.isnull()
for column in missing_data.columns.values.tolist():
      print(column)
      print (missing_data[column].value_counts())
      print("")
long
False    4152
Name: long, dtype: int64

lat
False    4152
Name: lat, dtype: int64
```

So I'm good there. Then I did some analysis of the range of long and lat. I used the box plot. It turns out there were some extreme outliers so I eliminated all the entries that were outside the reasonable range of King County. Here is some sample code:

```
sns.boxplot(x=WA_gdf_1['lat'])
```




```
WA_gdf_1 = WA_gdf_1.drop(WA_gdf_1[(WA_gdf_1['lat'] >= 48.0) | (WA_gdf_1['lat'] <= 47.0) | (WA_gdf_1['long'] >= -121.6)].index)
```

And then eliminate any entry that ArcGIS did not flag as "King County."

```
WA_gdf_1 = WA_gdf.drop(WA_gdf[(WA_gdf['attributes.Subregion'] != 'King County')].index)
```

Finally I wanted to save my geodataframe to COS. I needed to save it as JSON file. GeoJSON retains the geocoded object.

```
WA_gdf_1.to_file("WA_gdf_1.geojson", driver="GeoJSON")
```

**GeoPlot and Clustering**

## Notebook

Now that the Drinking Establishment information is cleaned up, it's time to see what we're working with. First I need to download the neighborhoods file from King County. I used the ArcGIS API. It turns out that this was kept in ESRIJSON driver which is supported by fiona 1.8.6. Geopandas loads

1.8.4 by default. So I had to install fiona via !pip after the environment was built. Here is the download:

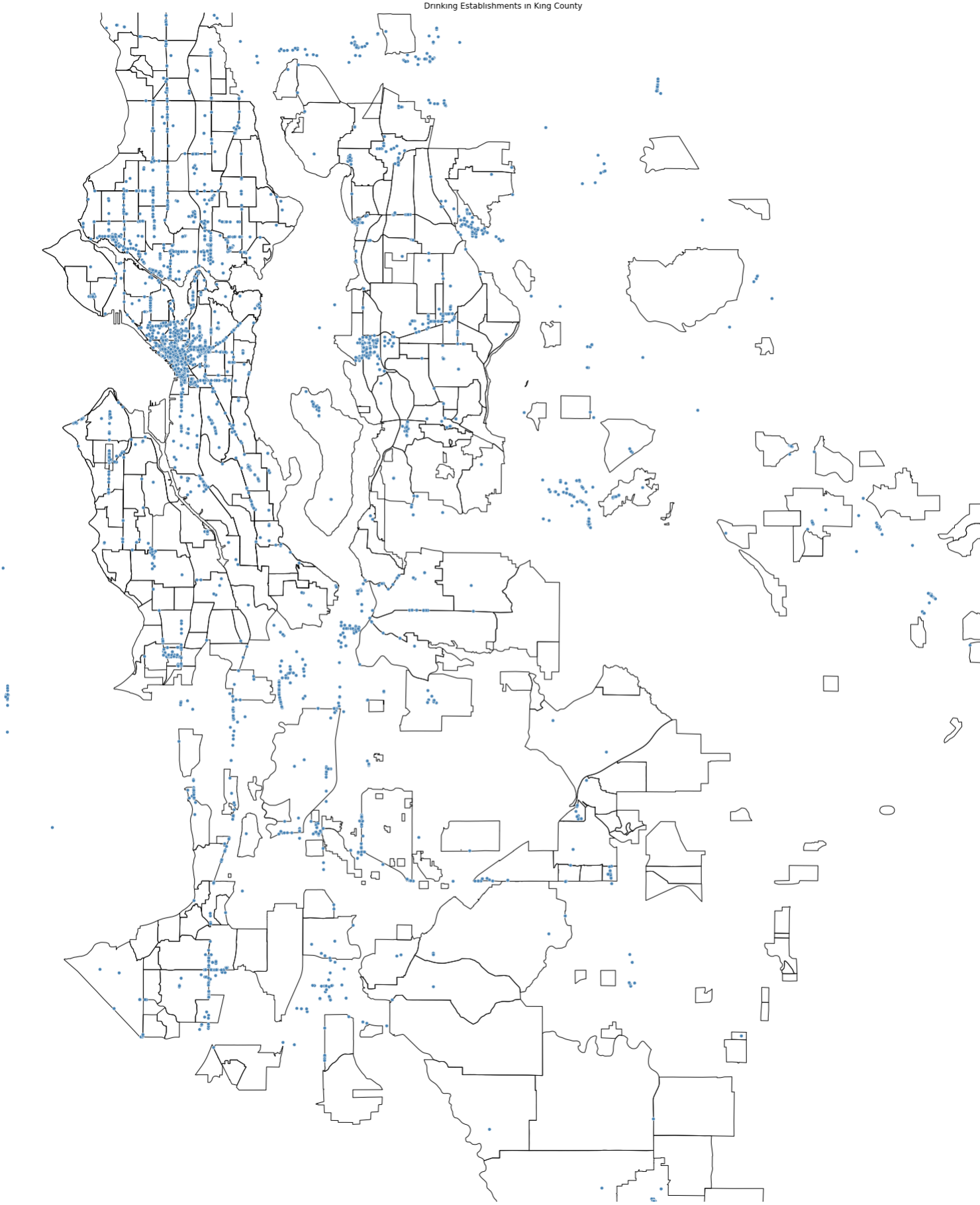
```
url =
"https://gisdata.kingcounty.gov/arcgis/rest/services/OpenDataPortal/transportation___base/MapSer
ver/384/query?where=1%3D1&outFields=*&outSR=4326&f=json"
data = requests.get(url)
b = bytes(data.content)
with fiona.BytesCollection(b) as f:
    crs = f.crs
    df_SEA = geopandas.GeoDataFrame.from_features(f, crs=crs)
    print(df_SEA.head())
        NEIGHBORHOOD_NAME  NEIGH_NUM  OBJECTID    SHAPE_Area  SHAPE_Length  \
0   University District        37         1  4.948493e+07  40436.205860
1          Wallingford        38         2  4.210762e+07  32080.778300
2             Interbay       101         3  3.372042e+07  59930.630161
3  West Campus - Fed Way       227         4  6.639272e+07  36341.429112
4       Berkshire Glen        245         5  7.048653e+06  10597.297569

                                                  geometry
0  POLYGON ((-122.3189590288485 47.67250069638226...
1  POLYGON ((-122.346114180476 47.66505276831814,...
2  POLYGON ((-122.397942744914 47.66481700558359,...
3  POLYGON ((-122.348030734308 47.3151893486264, ...
4  POLYGON ((-122.1764963769378 47.32992973688438...
```

Here I created a scatter plot of all the locations and the neighborhoods:

```
fig = plt.figure(figsize=(60,36))
plt.rcParams['figure.figsize'] = (60,36)
ax1 = plt.subplot(121, projection=gcrs.AlbersEqualArea(central_latitude=47.5,
central_longitude=-122.1))
geoplot.polyplot(df_SEA, ax=ax1, projection=gcrs.AlbersEqualArea())
geoplot.pointplot(WA_gdf_1, projection=gcrs.AlbersEqualArea(),
                  edgecolor='white', linewidth=0.5, zorder=10,
                  ax=ax1)
plt.title("Drinking Establishments in King County")
plt.savefig("King_County_Dist.png", bbox_inches='tight', pad_inches=0)
```

Drinking Establishments in King County



Some initial observations:

- Not all the drinking establishments are located in defined Neighborhoods. So the King County Neighborhood file is not comprehensive of the entire geography.
- Only a few neighborhoods or locations have a significant number of drinking establishments within walking distance.

**Clustering**

There are (more complex) methods to do geographical clustering that take into consideration streets, highways, natural boundaries, etc. In this case, I am only looking for Euclidean distance since walking through urban streets is less restrictive than driving. The proper method for Euclidean distance is haversine. The DBSCAN function takes maximum distance and minimum number of locations to do clustering. The input is in form of an array. Here is sample code:

```
coords = WA_gdf_1.as_matrix(columns=['lat', 'long'])
kms_per_radian = 6371.0088
epsilon = 0.4 / kms_per_radian
db = DBSCAN(eps=epsilon, min_samples=30, algorithm='ball_tree',
metric='haversine').fit(np.radians(coords))
cluster_labels = db.labels_
num_clusters = len(set(cluster_labels))
clusters = pd.Series([coords[cluster_labels == n] for n in range(num_clusters)])
print('Number of clusters: {}'.format(num_clusters))
```

clusters is a list of point pairs (or tuples) for each of the calculated clusters. In order to visualize the results of clustering, I plotted the centroid of each cluster on the map. The code for calculating centroid:

```
df_clustercentroids = pd.DataFrame.from_records([x for x in centroid_points], columns =['lat',
'long'])
print(df_clustercentroids)
```

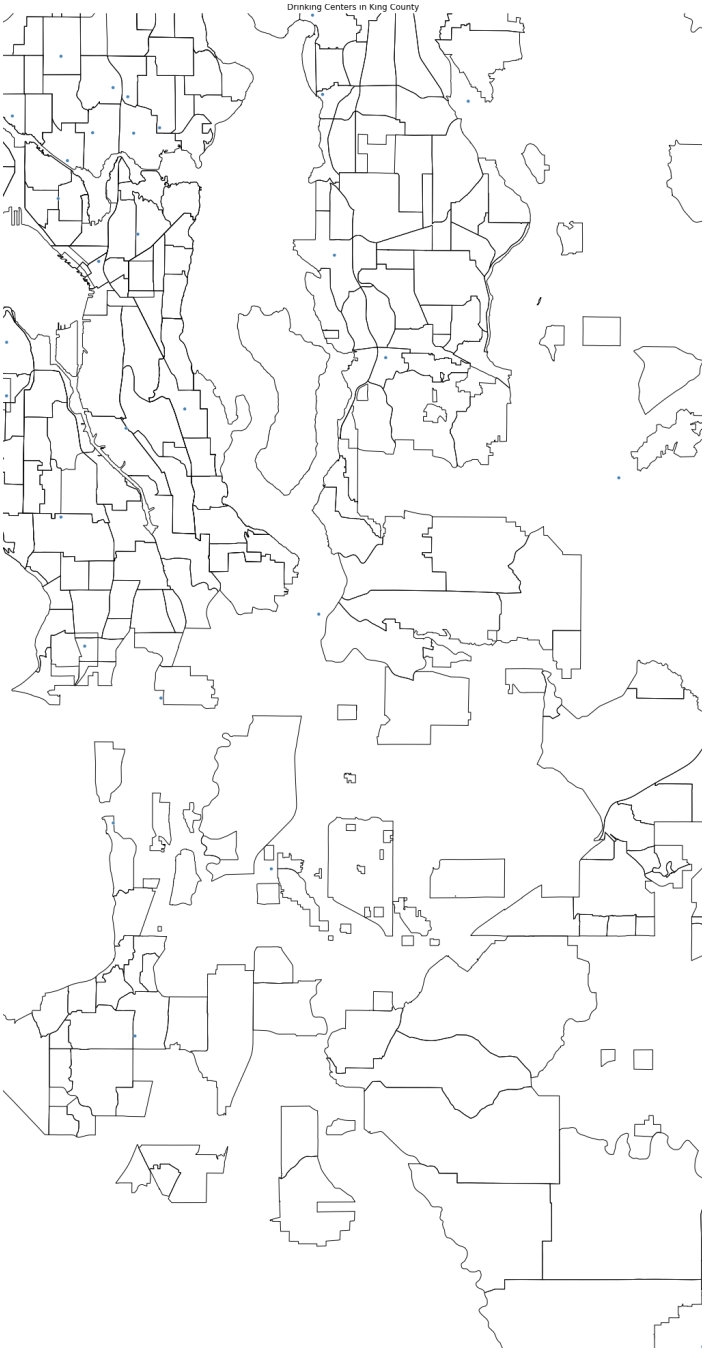
| | lat | long |
|---|---|---|
| 0 | 47.615335 | -122.199263 |
| 1 | 47.612883 | -122.334210 |
| 2 | 47.650800 | -122.351912 |
| 3 | 47.667484 | -122.383431 |
| 4 | 47.676001 | -122.205941 |
| 5 | 47.561816 | -122.386412 |
| 6 | 47.672389 | -122.121365 |
| 7 | 47.466851 | -122.341594 |
| 8 | 47.690262 | -122.355755 |
| 9 | 47.660345 | -122.314018 |

```
10  47.479522 -122.207797
11  47.661440 -122.337205
12  47.447715 -122.297815
```

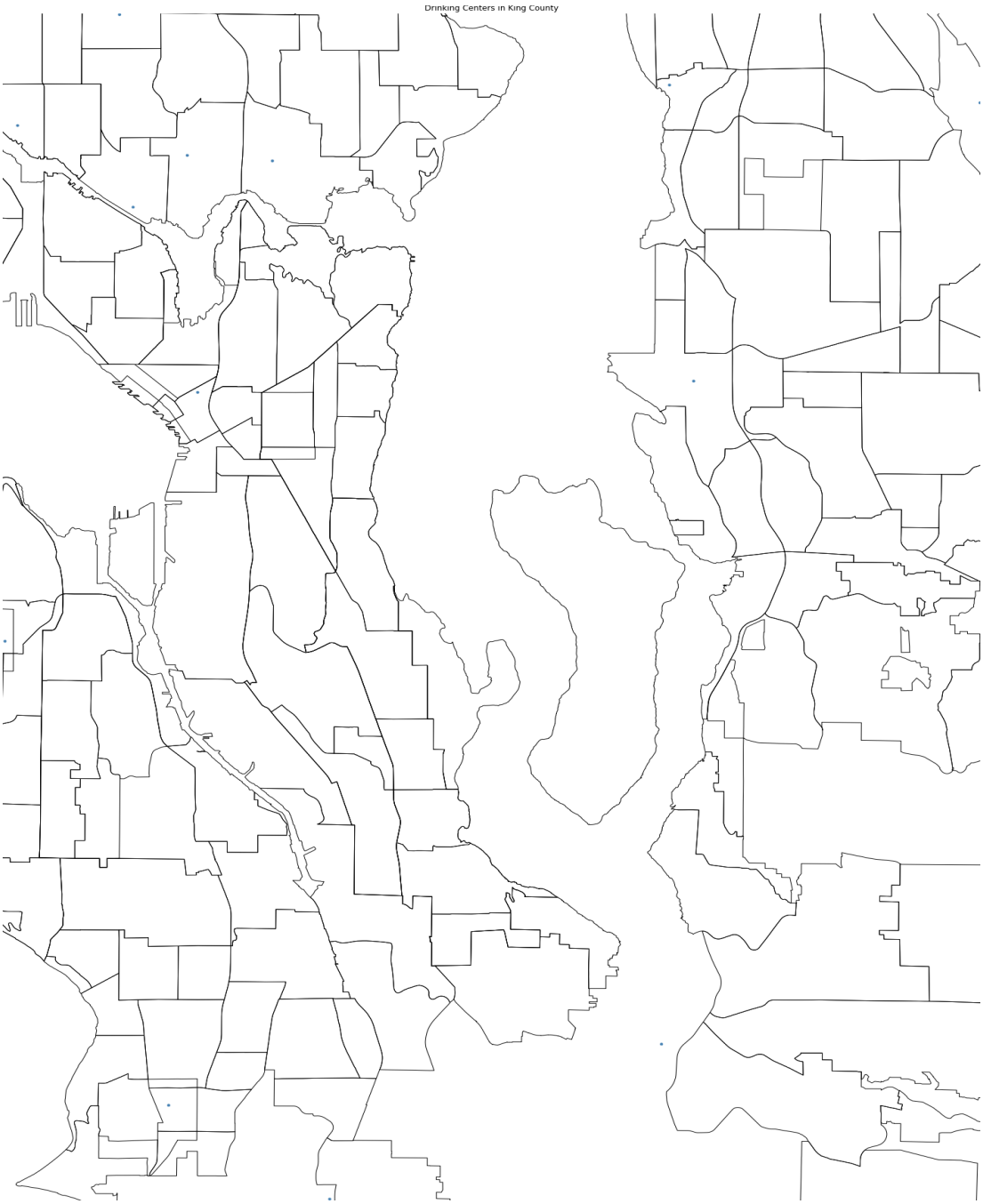
Here are some maps:

**Max Distance = 1/2 mile, Minimum Number in cluster = 10**

**Max Distance = .37 mile, Minimum Number in cluster = 20**



**Max Distance = 1/4 mile, Minimum Number in cluster = 30**



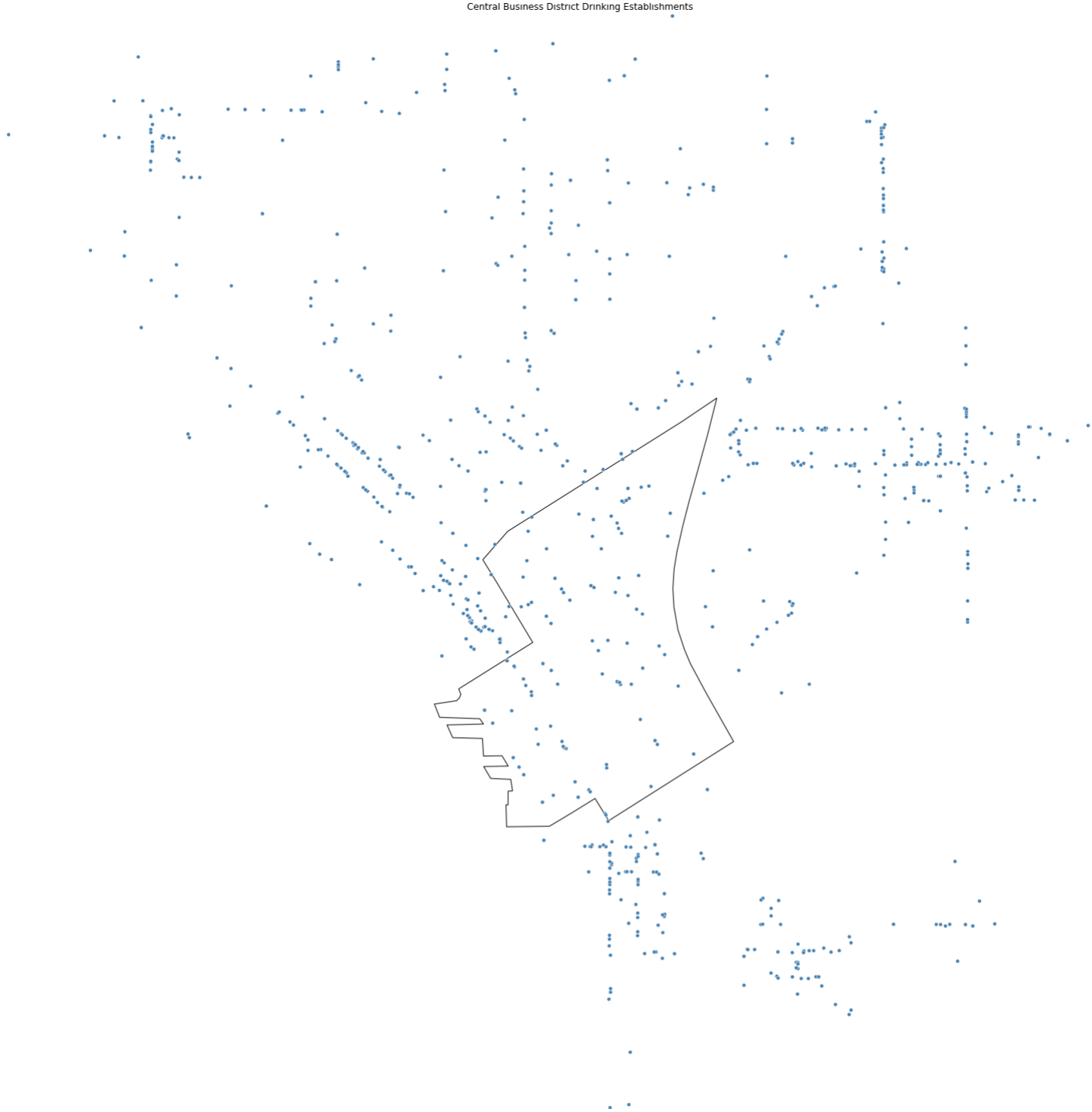

This final clustering shows 13 clusters which seemed like the right number of candidates to determine best neighborhood. If I count the number of drinking establishments of the 13 clusters it is overwhelmingly obvious that the second cluster is the largest.

```
print(len(clustersub[0]))
print(len(clustersub[1]))
print(len(clustersub[2]))
print(len(clustersub[3]))
print(len(clustersub[4]))
```

```
print(len(clustersub[5]))
print(len(clustersub[6]))
print(len(clustersub[7]))
print(len(clustersub[8]))
print(len(clustersub[9]))
print(len(clustersub[10]))
print(len(clustersub[11]))
print(len(clustersub[12]))
```
```
141
1010
68
104
53
49
56
44
37
54
33
44
32
```
And it turns out, the centroid of that cluster (47.612883 -122.334210) is in the Central Business District. CenturyLink Plaza, to be exact. Here is a plot of that cluster and the Central Business District neighborhood.

```
ax1 = plt.subplot(121, projection=gcrs.AlbersEqualArea(central_latitude=47.5,
central_longitude=-122.1))
geoplot.polyplot(gdf_CBD_Shape, ax=ax1, projection=gcrs.AlbersEqualArea())
geoplot.pointplot(gdf_Central_Business_District, projection=gcrs.AlbersEqualArea(),
                  edgecolor='white', linewidth=0.5, zorder=10,
                  ax=ax1)
plt.title("Central Business District Drinking Establishments")
plt.savefig("CBD_Locations.png", bbox_inches='tight', pad_inches=0)
```

Central Business District Drinking Establishments



## 4. Results section where you discuss the results.

I currently live in the Central Business District just a 16 minute walk from CenturyLink Plaza. I don't have a car so I walk everywhere or take a bus or light rail as needed. I am not surprised that Downtown Seattle won the best neighborhood contest. I am surprised by the margin. Seven times bigger than the next largest cluster. Thanks to Amazon, growth in Seattle has been aggressive. New high rise apartments being built all the time. And as I mentioned in my introduction, high tech workers are a thirsty bunch.

Among my drinking friends, downtown is not their favorite place to hang out. They prefer more divey bars. So neighborhoods like Fremont, Greenwood, Ballard, or Georgetown are better for them. I noticed on the Washington State Liquor database, the year the license was opened is one of the data fields. It may be interesting to redo the analysis but give older establishments a heavier weighting.

Also, some people prefer wine. The Washington database specifies where wineries with tasting rooms are located. Filtering for wine may give greater weight to Woodenville where several wineries are clustered.

The initial goal of this exercise was to find the best neighborhood for drinkers. If I were to expand the scope to create an app that would calculate the "drunk score" for any location a different methodology would be necessary. Given a point, you would simply calculate the number of drinking establishments within a certain distance. Whereas DBSCAN used Machine Learning to cluster locations. If I were developing that app, I would normalize the data so CenturyLink Plaza was 100.

Also, there is a ton of valuable information in the ESRI databases. Probably ratings, cost ranges, cuisines, etc. The geocoding even showed what neighborhood the location was in. I saved all that information in COS so if I wanted to expand on this study, I could.

## 5. Discussion section where you discuss any observations you noted and any recommendations you can make based on the results.

Besides the learning curves and technical difficulties, I spent a fair amount of time learning code. As I proceeded, it became obvious that I was learning and implementing new technology that didn't end up being apropos to the goal of the project. Also, whereas the geocoding information in ESRI was thorough and reliable, the neighborhood shapefile from King County was not comprehensive. I needed to adjust my approach constantly.

For example, the State of Washington database was exclusively retail public houses. Conversely, the Oregon database included retail locations. I needed to create a dictionary of licensing codes and do a look-up to eliminate retail locations.

Another reason I didn't include Portland in the analysis is I don't think the licensing data would be normalized between Seattle and Portland. Washington and Oregon have different laws so it isn't necessarily comparing apples to apples.

And finally, a more comprehensive approach to the problem would be to include public transportation and ride hailing services in the calculation. There are some neighborhoods that may not be as walkable but are an easy light rail ride from a location with plenty of drinking establishments.

## 6. Conclusion section where you conclude the report.

This Coursera coursework has been challenging but an extremely effective introduction to data science. At first I thought the introductory sections when they described the profession were frivolous. But in fact, once you begin a project, you quickly see how you need to adjust your approach to the problem as you uncover nuances of the data. Also it was good to understand that the desired outcome was not always possible. You really don't know what result you can achieve until you dive in.

Here's a picture of Sally and I enjoying a Mac and Jack at Urbane. Just a half a block from CenturyLink Plaza. Celebrating getting this project done.


SANTA FE 5K