The Distribution of the Air Pollution Burden in Chicago

NU-Project 1

How we decided on this topic

- Environmental justice is a well discussed topic in the Chicago area and we were interested to know which areas of the city were most affected by air pollution.
- We were interested to know if the environmental burden lay on certain neighborhoods in the city, according to racial distribution, income, population, etc.
- We found that the burden of air pollution was unequally distributed - on both the North and South side of Chicago.

Questions That Interested Us

We identified the following questions from the data, that were most interesting for us:

1. Which Chicago neighborhoods submitted the highest number of air pollution complaints in 2010?
2. Did the number of air pollution complaints vary by neighborhood income level?
3. Did the number of air pollution complaints vary by the presence of minority communities?

The next few slides answer these questions using data representations.

Where & how we found the data

We utilized 3 different data sets:

a. **Chicago Metropolitan Agency for Planning (CMAP)**
    i. 2010 Census Data summarized to Chicago Community area - provides race, gender, housing demographics by neighborhood.
b. **Chicago Data Portal**
    i. Census data encapsulating indicators of public health - provides poverty, unemployment, and income demographics by neighborhood from years 2008-2012.
c. **Chicago Department of Public Health Portal (CDPH)**
    i. Provides types and details of environmental complaints during CY2010 by *street address*

Data Exploration

- After identifying the datasets we wanted to use from government portals, we decided to identify a common factor in all the data sets
- The common factor was neighborhoods. While the CMAP data and Census data had a 'neighborhood' tab, the CDPH website didn't have any such demarcation
- We narrowed down our search to just air pollution complaints and then divided the CDPH data set among each member of the team, who then found the corresponding neighborhood for each address in the CDPH data

Cleanup process

- We used Jupyter Lab to read the original CSV files of CMAP, CDPH and Chicago Data Portal
- We visualized the data frame to confirm if the neighborhood tab was the same on both CSVs
- We wanted to get rid of the neighborhoods which had a different name and would disrupt our data set. We decided to merge the data sets to combine the neighborhoods with the same name
- We merged the CMAP, CDPH and Chicago Data Portal data sets using the common "neighborhood" tab, on Jupyter Notebook
- Following the cleanup, we had 248 rows * 26 columns

Our Code for Data Merging

1.

```
In [4]:
import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as st
import numpy as np
```

```
In [20]:
# Study data files
census_data_path = "../Resources/Census_Data.csv"

census_data = pd.read_csv(census_data_path)

# Combine the data into a single dataset
#extension = 'csv'

census_data
```

2.

```
environment_data_path = "../Resources/Chicago_data.csv"

environment_data = pd.read_csv(environment_data_path)

# Combine the data into a single dataset
#extension = 'csv'

environment_data
```

3.

```
combined_data = pd.merge(environment_data, census_data, on="Neighborhood")
combined_data
```

4.

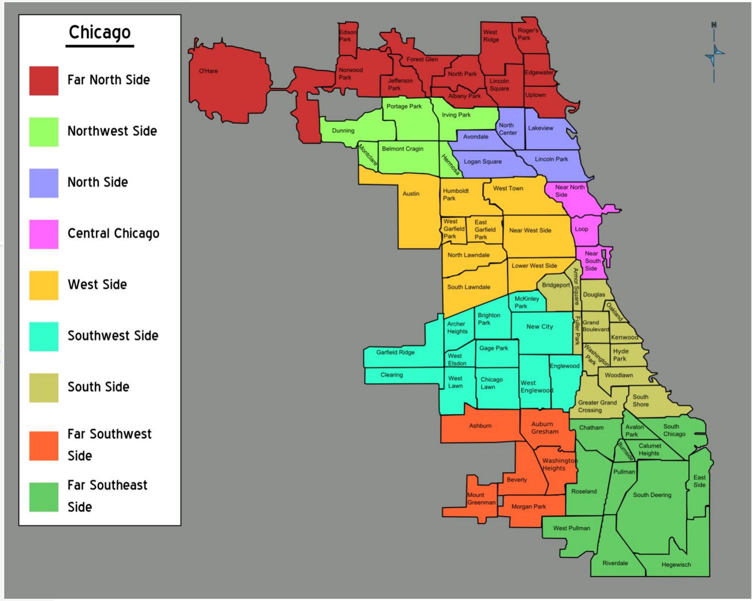
```
combined_data.to_csv("../Resources/merged_data.csv")
combined_data.head()
```

Q1. Which neighborhoods submitted the most air pollution complaints in 2010?

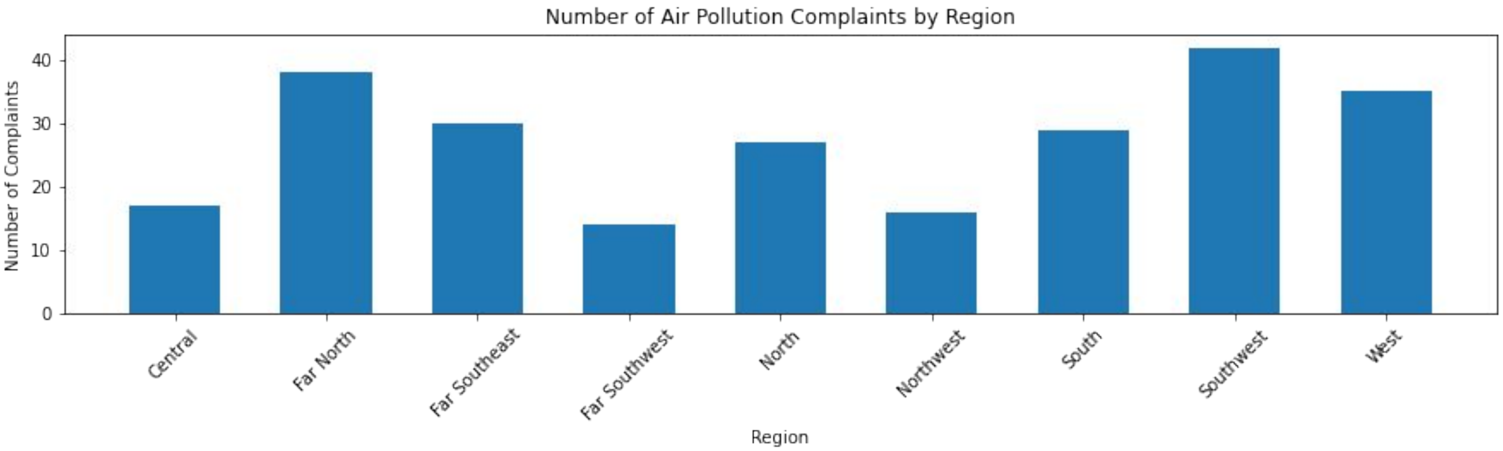
Number of complaints by region and neighborhood

```python
# set up the larger regions based on https://commons.wikimedia.org/wiki/File:Chicago_neighborhoods_map.png
far_north = ["Edison Park", "Norwood Park", "Jefferson Park", ... ]
...

# create new df with region info for each complaint added
regions_df = pollution_df
regions_df["Region"] = ""
for index, neighborhood in enumerate(pollution_df["Neighborhood"]):
    if neighborhood in far_north:
        regions_df.loc[index, "Region"] = "Far North"
    elif neighborhood in northwest:
...
```

```python
# create new groupby to see number of complaints per region
grouped_region = regions_df.groupby(by="Region")
grouped_region_complaint = pd.DataFrame()
grouped_region_complaint["Complaint Count"] = grouped_region["COMPLAINT TYPE"].count
grouped_region_complaint.reset_index(inplace=True)

# x and y values for graph
regions = grouped_region_complaint["Region"]
complaint_num = grouped_region_complaint["Complaint Count"]
xticks = [value for value in regions]

# create bar graph
plt.figure(figsize=(15, 3))
plt.bar(regions, complaint_num, align='center', width=0.6)
plt.xticks(xticks, regions, rotation=45)

# formatting
```



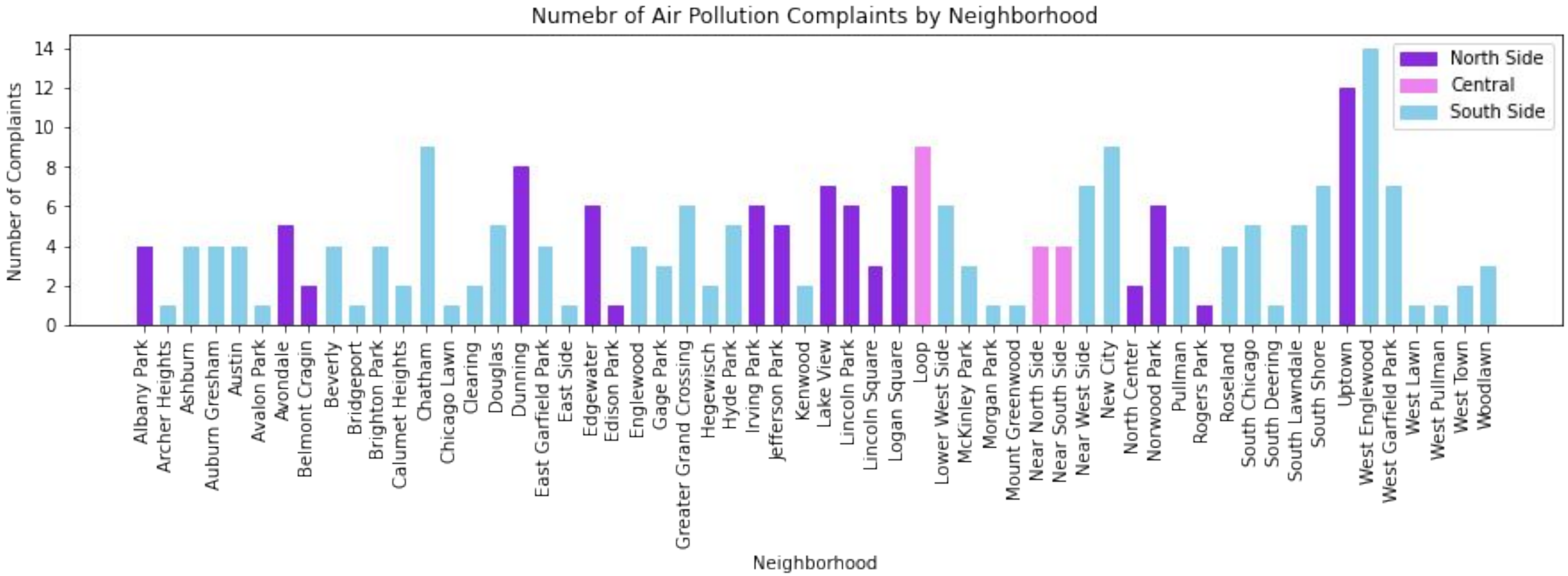
Number of complaints by region



Key Takeaways

1. Regions with the highest number of complaints: Southwest and far north
2. No noticeable trend pointing towards areas with highest complaints
3. Focused on neighborhoods next to see if there were any differences

Number of complaints by neighborhood



Key Takeaways

1. No obvious differences between neighborhoods or the rougher grouping of north vs south
2. Neighborhoods with the most complaints: Uptown and West Englewood

Heatmap of Complaints by Neighborhood

Step 1: Grouped merged dataset by neighborhood

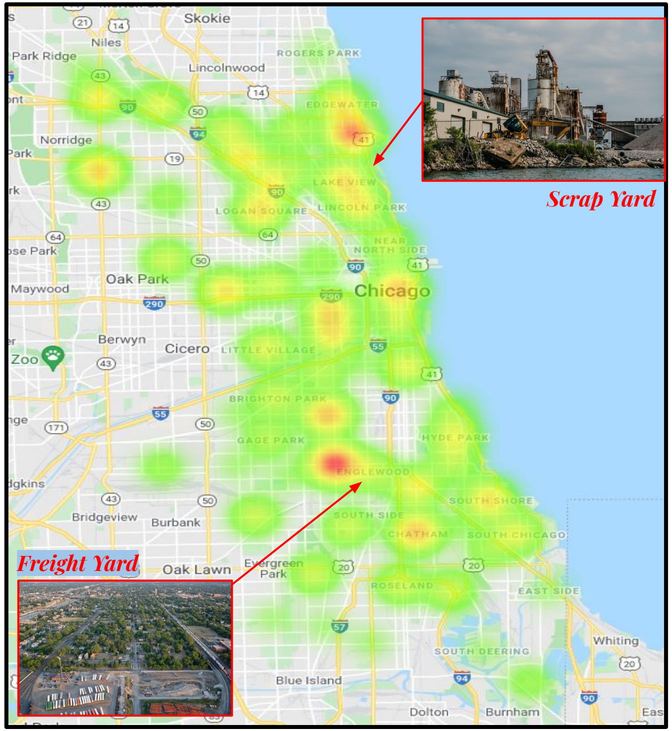
```python
# Load merged dataset into Jupyter Notebook
merged_dataset = "Resources/merged_data.csv"
air_pollution_df = pd.read_csv(merged_dataset)
# Create dataframe showing number of complaints by neighborhood in descending order
neighborhood = air_pollution_df.groupby(["Neighborhood"])
complaint_count = neighborhood["COMPLAINT ID"].count()
neighborhood_df = pd.DataFrame({"Number of Complaints": complaint_count})
neighborhood_df = neighborhood_df.sort_values(["Number of Complaints"], ascending = False).reset_index()
# Add columns to neighborhood_df for location data
neighborhood_df["Region"] = neighborhood_df["Neighborhood"].astype(str) + ", Chicago, IL"
neighborhood_df["Latitude"] = ""
neighborhood_df["Longitude"] = ""
```

Step 2: Made Google Geocode API request to extract neighborhood coordinates

```python
# Build partial query URL
base_url = "https://maps.googleapis.com/maps/api/geocode/json"
params = {"key" : gkey}
# Iterate through rows of neighborhood_df
for index, row in neighborhood_df.iterrows():
    # Get address from neighborhood_df
    search_address = row["Region"]
    # Add keyword to params dictionary
    params["address"] = search_address
    # Assemble URL and make API request
    print(f"Retrieving Results for Index {index}: {search_address}.")
    response = requests.get(base_url, params = params).json()
    # Extract location coordinates and save to neighborhood_df
    neighborhood_data = response
    try:
        neighborhood_df.loc[index, "Latitude"] = neighborhood_data["results"][0]["geometry"]["location"]["lat"]
        neighborhood_df.loc[index, "Longitude"] = neighborhood_data["results"][0]["geometry"]["location"]["lng"]
    # Skip row if neighborhood not found
    except (KeyError, IndexError):
        print("Neighborhood not found. Skipping...")
```

Step 3: Made Google Geocode API request to extract neighborhood coordinates

```python
# Configure gmaps
gmaps.configure(api_key = gkey)
# Store latitude and longitude in locations
locations = neighborhood_df[["Latitude", "Longitude"]]
complaints = neighborhood_df["Number of Complaints"].astype(float)
# Plot heatmap
complaints_heatmap = gmaps.figure()
heat_layer = gmaps.heatmap_layer(locations, weights = complaints, dissipating = False, max_intensity = 14,
# Add layer
complaints_heatmap.add_layer(heat_layer)
# Display figure
complaints_heatmap
```



Key Takeaways

1. Complaints were **scattered** across Chicago with no strong regional or neighborhood trends

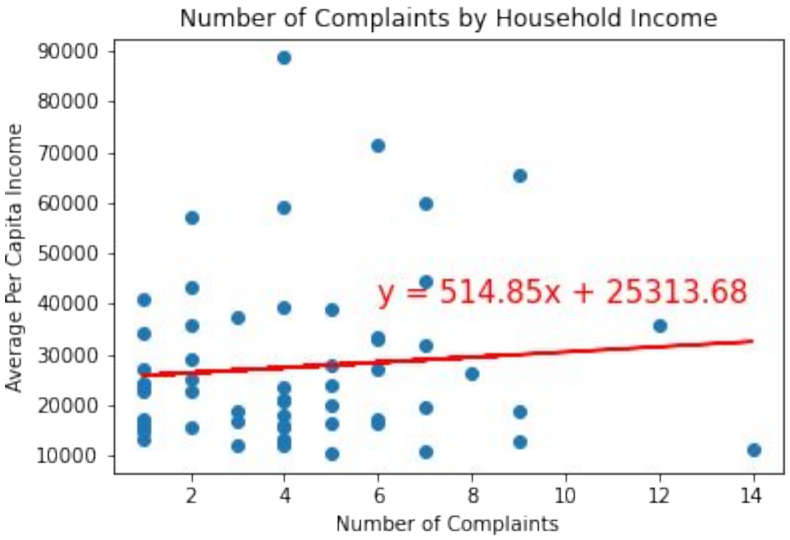
2. West Englewood (SW) and Uptown (N) emerged as **two isolated hotspots** with highest number of complaints

Q2. Did the number of air pollution complaints vary by neighborhood income level?

Complaints by Household Income

```
# import merged dataset
pollution_df = pd.read_csv(path)

# group by neighborhood and find the frequency of complaints per neighborhood
grouped_df = pollution_df.groupby(by="Neighborhood")

# finds the number of complaints and avg income per neighborhood
grouped_complaint["Complaint frequency"] = grouped_df["COMPLAINT TYPE"].count()
grouped_complaint["Avg Per Capita Income"] = grouped_df["PER CAPITA INCOME "].mean()
```

```
# scatter plot X - frequency of complaint Y - income; linear regression
x_values = grouped_complaint['Complaint frequency']
y_values = grouped_complaint['Avg Per Capita Income']

# create linear regression line

# plot scatter plot with linear regression line
plt.scatter(x_values,y_values)
plt.plot(x_values,regress_values,"r-")

# formatting
```



Key Takeaways

1. No correlation between number of complaints and average household income
2. R-square value: 0.00788

Boxplot of Complaints by High- vs. Low-Income Neighborhoods

Step 1: Locate and store per capita income for City of Chicago to use as threshold

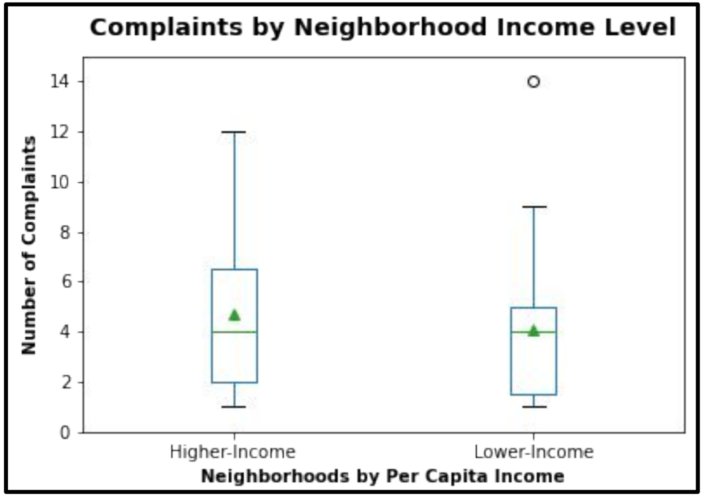
```
# Load merged dataset into Jupyter Notebook
income_dataset = "Resources/Census_data.csv"
income_df = pd.read_csv(income_dataset)
# Identify and store per capita income for Chicago at the city level
chicago_income_df = income_df.loc[income_df["Neighborhood"] == "CHICAGO"]
avg_chicago_income = chicago_income_df["PER CAPITA INCOME "]
avg_chicago_income = int(avg_chicago_income)
avg_chicago_income
```

Step 2: Assign neighborhoods into “Higher-Income” or “Lower-Income” groups

```
# Group neighborhoods in Chicago into lower-income vs higher-income
lower_income_df = air_pollution_df.loc[air_pollution_df["PER CAPITA INCOME "] < avg_chicago_income]
higher_income_df = air_pollution_df.loc[air_pollution_df["PER CAPITA INCOME "] > avg_chicago_income]

# Create dataframe showing number of complaints by lower-income neighborhood in descending order
lower_income_neighborhood = lower_income_df.groupby(["Neighborhood"])
lower_complaint_count = lower_income_neighborhood["COMPLAINT ID"].count()
lower_income_neighborhood_df = pd.DataFrame({"Number of Complaints": lower_complaint_count})
lower_income_neighborhood_df = lower_income_neighborhood_df.sort_values(["Number of Complaints"], ascending = False).
lower_income_neighborhood_df["Neighborhood Income"] = "Lower-Income"
```

Step 3: Create boxplots by neighborhood income groupings

```
neighborhoods = [lower_income_neighborhood_df, higher_income_neighborhood_df]
all_neighborhoods_df = pd.concat(neighborhoods)
all_neighborhoods_df.boxplot(by = "Neighborhood Income", showmeans = True)
plt.title("")
plt.grid(False)
plt.ylim(0, 15)
plt.suptitle("Complaints by Neighborhood Income Level", fontsize = 14, fontweight = "bold")
plt.xlabel("Neighborhoods by Per Capita Income", fontsize = 10, fontweight = "bold")
plt.ylabel("Number of Complaints", fontsize = 10, fontweight = "bold")
plt.savefig("Output/Q2_Boxplot.png")
plt.show()
```



Key Takeaways

1. Higher-income and lower-income neighborhoods had the **same median** number of air pollution complaints

2. The distribution of “Higher-Income” neighborhood complaints is more **right-skewed** indicating a few neighborhoods submitted many complaints

Q3. Did the number of air pollution complaints vary by presence of minority communities?

Neighborhood Demographics

Step 1: Remane the columns

```
# Renaming CMAP columns
cmap_data = cmap_data.rename(columns= {'Geog' : 'Neighborhood',
                                       'Not Hispanic or Latino, White alone' : 'White',
```

Step 2: Pull the columns needed and sort

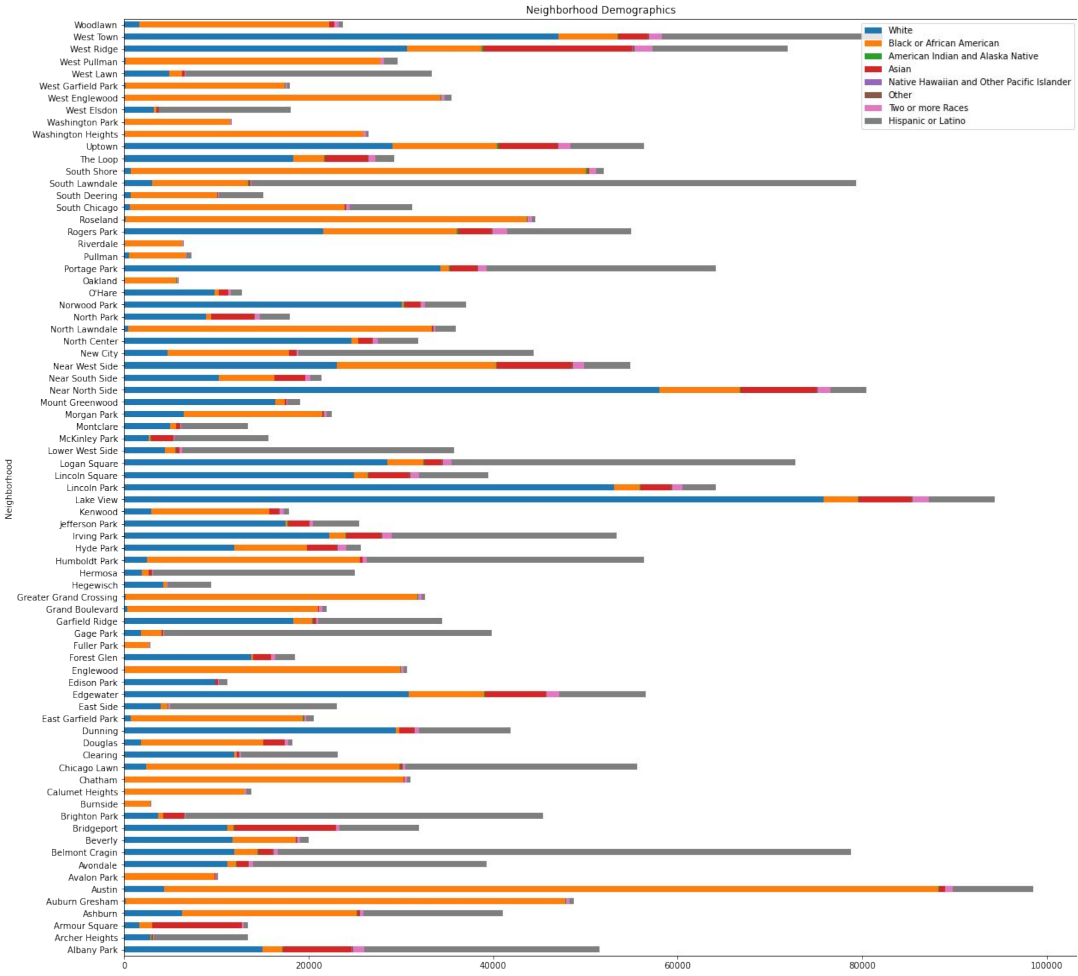
```
race_data = cmap_data.iloc[:, [0, 3,4,5,6,7,8,9,10,67]]
race_data.sort_values('Neighborhood')
```

Step 3: Create Barplot

```
# Neighborhood Demogrphic breakdown
ax = race_data.plot.barh(x="Neighborhood", stacked = True,
                         figsize = (20,20), xlabel = 'Neighborhood',
                         title = 'Neighborhood Demographics')
```

Key Takeaways

1. White, African American and Hispanic or Latino are the most representative demographics
2. While West Englewood and Uptown have the highest number of complaints their demographics are very different



Complaints by Minority Population

Step 1: Grouped merged dataset by neighborhood and do a count to get the number of complaints

```
hood_data = demo_data.groupby(by= 'Neighborhood')
count_df = hood_data.count()
```

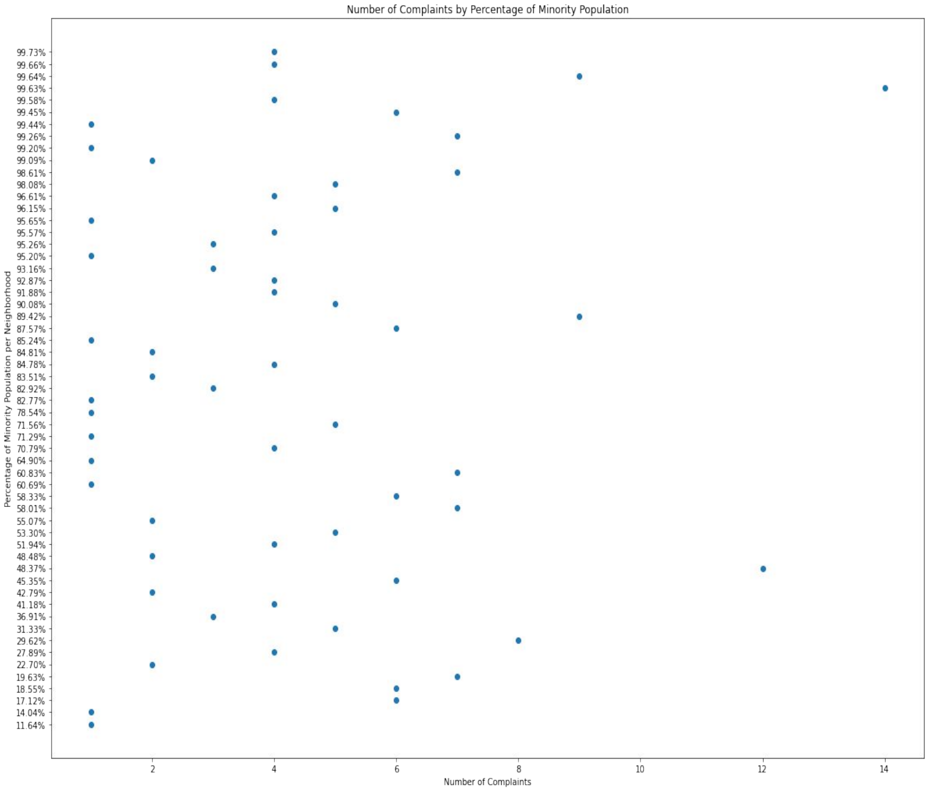
Step 2: Grouped count dataset and race by neighborhood, then pll the columns needed and sort

```
complaint_df = count_df['COMPLAINT ID']
```

```
chart_data = pd.merge(complaint_df, race_data, on = 'Neighborhood')
chart_data = chart_data.iloc[: , [0,1,10]]
chart_data = chart_data.sort_values('Minority Percentage')
```

Step 3: Create Scatterplot

```
# Scatterplot Y-Minority %, x-Complaint Frequency
# Set up x and y values
x_values = chart_data['COMPLAINT ID']
y_values = chart_data['Minority Percentage']
# plot scatterplot
plt.figure(figsize = (20,15))
figure = plt.scatter(x_values,y_values)
#Formating
plt.ylabel('Percentage of Minority Population per Neighborhood')
plt.xlabel('Number of Complaints')
plt.title('Number of Complaints by Percentage of Minority Population')
plt.show()
```



Key Takeaways

1. No correlation between Minority Population and Number of Complaints
2. A few outliers with high minority populations and complaint counts