There is also a first project to work on, based on the 311 service request data from Open data buffalo. Most of the variables are categorical in nature, the categories of major interest are probably

-Status, subject, reason, type, object type, council district, police district, neighborhood

-There is also a couple of date variables

It looks like status, subject, reason, type and object type might be interesting categorical variables. Which of these can be described by a limited number of categories? Which are not described easily as a limited number of outcomes

Figure out which of the outcomes fit into a limited number of categories, and then produce some graphs explaining how they are related to council district, police district, neighborhood.

Do different districts and/or neighborhoods tend to different numbers of 311 calls?

Do different months have different number of 311 reports? What about different years?

This is a two week long project. Show your results visually, make sure everything is labeled clearly in your ipynb file. Write a clear discussion of each finding and be sure the sections of the notebook are all clearly labeled.

Set up a repository for this project on GitHUB, save your ipynb file there, the pdf version of the ipynb and also a ReadMe file describing the project

Turn in the pdf of your notebook and also a link to the repository on GitHub.

Upload the pdf version of the ipynb file.

What I am really looking for in this project is:

a.) Some interesting insights into 311 calls- search for at least one

b.) A clear visual demonstration of that insight

c.) Documentation of the use of GitHub

Import Data, set up libraries and other options

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

pd.set_option("display.max_columns", 200)
```

In [ ]:
```
infile = "311_Service_Requests_20240221.csv"
df311 =pd.read_csv(infile)
df311.head()
```

```
<ipython-input-463-683123bb5431>:2: DtypeWarning: Columns (0) have mixed types. Speci
fy dtype option on import or set low_memory=False.
  df311 =pd.read_csv(infile)
```

Out[ ]:

	Case Reference	Open Date	Closed Date	Status	Subject	Reason	Type	Object Type	Address Number	
0	514722	07/19/2010 12:06:00 PM	07/19/2010 02:15:00 PM	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	NaN	F
1	1000802971	02/12/2018 06:47:00 AM	02/12/2018 10:33:00 AM	Closed	Buffalo Police Department	Police	Police Issue (Req_Serv)	Street	NaN	I
2	541218	01/11/2011 09:12:00 AM	01/26/2011 10:01:00 AM	Closed	Utilities	National Grid	Streetlights (Req_Serv)	Street	NaN	
3	1000708750	07/10/2017 11:36:00 AM	07/10/2017 02:27:00 PM	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	NaN	
4	1000892866	08/18/2018 03:44:00 PM	08/20/2018 10:00:00 PM	Closed	Dept of Public Works	Forestry	Tree Planting Request (Req_Serv)	Property	NaN	

Understand the data I am dealing with

In [ ]:
```
df311.shape
```

Out[ ]:
```
(1112730, 34)
```

Thats beefy

In [ ]:
```
df311.columns
```

Out[ ]:
```
Index(['Case Reference', 'Open Date', 'Closed Date', 'Status', 'Subject',
       'Reason', 'Type', 'Object Type', 'Address Number', 'Address Line 1',
       'Address Line 2', 'City', 'State', 'Zipcode', 'Property ID', 'Location',
       'Latitude', 'Longitude', 'Council District', 'Council District 2011',
       'Police District', 'Census Tract', 'Census Block Group', 'Census Block',
       'Neighborhood', 'X Coordinate', 'Y Coordinate', '2010 Census Tract',
       '2010 Census Block Group', '2010 Census Block', 'TRACTCE20',
       'GEOID20_tract', 'GEOID20_blockgroup', 'GEOID20_block'],
      dtype='object')
```

In [ ]:
```
df311.dtypes
```

Out[ ]:
```
Case Reference             object
Open Date                  object
Closed Date                object
Status                     object
Subject                    object
Reason                     object
Type                       object
Object Type                object
Address Number             object
Address Line 1             object
Address Line 2             object
City                       object
State                      object
Zipcode                    object
Property ID                object
Location                   object
Latitude                  float64
Longitude                 float64
Council District           object
Council District 2011      object
Police District            object
Census Tract               object
Census Block Group         object
Census Block               object
Neighborhood               object
X Coordinate              float64
Y Coordinate              float64
2010 Census Tract          object
2010 Census Block Group    object
2010 Census Block          object
TRACTCE20                  object
GEOID20_tract              object
GEOID20_blockgroup         object
GEOID20_block              object
dtype: object
```

In [ ]:
```
df311.describe()
```

Out[ ]:

	Latitude	Longitude	X Coordinate	Y Coordinate
count	1.019924e+06	1.019924e+06	5.677520e+05	5.677520e+05
mean	4.299835e+01	-7.899725e+01	-7.135284e+06	4.525047e+06
std	1.296937e-02	2.080884e-02	3.614622e+06	1.711193e+06
min	4.283199e+01	-7.900000e+01	-8.784200e+06	-8.776799e+06
25%	4.300000e+01	-7.900000e+01	-8.779050e+06	5.290542e+06
50%	4.300000e+01	-7.900000e+01	-8.775132e+06	5.297801e+06
75%	4.300000e+01	-7.900000e+01	-8.772702e+06	5.301150e+06
max	4.300000e+01	-7.879899e+01	5.302890e+06	5.306748e+06

Prepare my dat for Analysis with plots

I am going to keep Open Date to refernce Dates and the catagorical variables : Status, subject, reason, type, object typethen compare their relationship with council district, police district, neighborhood.

In [ ]:
```
df311.columns
```

Out[ ]:
```
Index(['Case Reference', 'Open Date', 'Closed Date', 'Status', 'Subject',
       'Reason', 'Type', 'Object Type', 'Address Number', 'Address Line 1',
       'Address Line 2', 'City', 'State', 'Zipcode', 'Property ID', 'Location',
       'Latitude', 'Longitude', 'Council District', 'Council District 2011',
       'Police District', 'Census Tract', 'Census Block Group', 'Census Block',
       'Neighborhood', 'X Coordinate', 'Y Coordinate', '2010 Census Tract',
       '2010 Census Block Group', '2010 Census Block', 'TRACTCE20',
       'GEOID20_tract', 'GEOID20_blockgroup', 'GEOID20_block'],
      dtype='object')
```

1. Get rid of 'Case Reference' I want to understand a general tread
2. Keep open date to reference Dates remove closed dates
3. Status subject reason type and object type are useful for understanding a general trend as they describe the nature of the call
4. I would like to understand the realtions of these variables to council district, police district, neighborhood

In [ ]:
```
working311 = df311[['Open Date','Status', 'Subject',
       'Reason', 'Type', 'Object Type', 'Council District',
       'Police District',
       'Neighborhood']]
working311.head()
```

Out[ ]:

	Open Date	Status	Subject	Reason	Type	Object Type	Council District	Police District	Neighborh
0	07/19/2010 12:06:00 PM	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNO
1	02/12/2018 06:47:00 AM	Closed	Buffalo Police Department	Police	Police Issue (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNO
2	01/11/2011 09:12:00 AM	Closed	Utilities	National Grid	Streetlights (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNO
3	07/10/2017 11:36:00 AM	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNO
4	08/18/2018 03:44:00 PM	Closed	Dept of Public Works	Forestry	Tree Planting Request (Req_Serv)	Property	UNKNOWN	UNKNOWN	UNKNO

In [ ]:
```
working311.dtypes
```

Out[ ]:
```
Open Date           object
Status              object
Subject             object
Reason              object
Type                object
Object Type         object
Council District    object
Police District     object
Neighborhood        object
dtype: object
```

Lets figure out a way to change the data type of the Date to be a Date type in python.

They look like Datetime so import the libaries moduels. And convert it to date time.

In [ ]:
```
from datetime import datetime

working311["Open Date"]=pd.to_datetime(working311["Open Date"])
```

```
<ipython-input-471-f6140a70ffc2>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  working311["Open Date"]=pd.to_datetime(working311["Open Date"])
```

In [ ]:
```
working311.dtypes
```

Out[ ]:
```
Open Date           datetime64[ns]
Status                      object
Subject                     object
Reason                      object
Type                        object
Object Type                 object
Council District            object
Police District             object
Neighborhood                object
dtype: object
```

In [ ]:
```
working311.shape
```

Out[ ]:
```
(1112730, 9)
```

In [ ]:
```
working311.isna().sum()
```

Out[ ]:
```
Open Date              0
Status                 0
Subject                0
Reason                 0
Type                   0
Object Type            0
Council District    1797
Police District        0
Neighborhood           0
dtype: int64
```

This is such a small amount comapred to 1112730 I am just going to drop all the rows where Council District is null.

In [ ]:
```
working311=working311.loc[~working311["Council District"].isna()]
```

In [ ]:
```
working311.shape
```

Out[ ]:
```
(1110933, 9)
```

Which variables have limited catagories?

In [ ]:
```
working311.columns
```

Out[ ]:
```
Index(['Open Date', 'Status', 'Subject', 'Reason', 'Type', 'Object Type',
       'Council District', 'Police District', 'Neighborhood'],
      dtype='object')
```

In [ ]:
```
len(working311.columns)
```

Out[ ]:
```
9
```

In [ ]:
```
working311["Council District"].unique()
```

Out[ ]:
```
array(['UNKNOWN', 'LOVEJOY', 'FILLMORE', 'SOUTH', 'NORTH', 'NIAGARA',
       'DELAWARE', 'UNIVERSITY', 'MASTEN', 'ELLICOTT'], dtype=object)
```

In [ ]:
```
len(working311["Council District"].unique())
```

Out[ ]:
```
10
```

10 catagoreis, I am going to keep the "Unknown" catagory to account for all the unidentified values.

In [ ]:
```
working311["Open Date"].unique()
```

Out[ ]:
```
array(['2010-07-19T12:06:00.000000000', '2018-02-12T06:47:00.000000000',
       '2011-01-11T09:12:00.000000000', ...,
       '2024-02-20T09:22:00.000000000', '2024-02-20T09:07:00.000000000',
       '2024-02-20T08:36:00.000000000'], dtype='datetime64[ns]')
```

In [ ]:
```
len(working311["Open Date"].unique())
```

Out[ ]:
```
839254
```

This is not a catagorical variable as it continually increases as time goes on but I just wanted to obseve

In [ ]:
```
working311["Status"].unique()
```

Out[ ]:
```
array(['Closed', 'Open'], dtype=object)
```

In [ ]:
```
len(working311["Status"].unique())
```

Out[ ]:
```
2
```

In [ ]:
```
working311["Subject"].unique()
```

Out[ ]:
```
array(['Dept of Public Works', 'Buffalo Police Department', 'Utilities',
       'DPIS', 'Assessment & Taxation', 'Dept of Parking',
       'Buffalo Municipal Housing Authority', 'Buffalo Fire Department',
       'City Clerk', 'Office of the Mayor',
       'Office of Strategic Planning', 'Dept of Law',
       'Community Services & Rec. Program.', 'New Americans',
       'Human Resources', 'Management Information Systems', 'Test',
       'Knowledge Management'], dtype=object)
```

In [ ]:
```
len(working311["Subject"].unique())
```

Out[ ]:
```
18
```

This one is failry small I will see how my graphs looks.

In [ ]:
```
working311["Reason"].unique()
```

Out[ ]:
```
array(['Streets', 'Police', 'National Grid', 'Forestry', 'Rodent Control',
       'Sanitation', 'Housing', 'Buffalo Sewer Authority',
       'Animal Shelter', 'City Parks', 'Taxation',
       'Buffalo Water Authority', 'Engineering - Street Repairs',
       'Parking Violations Bureau', 'BMHA', 'BFD', 'Licenses',
       'City Clerk Issue', 'Engineering - Traffic',
       'Citizen Services - Quick Response Teams',
       'Citizens Services - Clean City',
       'Citizen Services - Save Our Streets', 'Real Estate',
       'Moving Violations', 'Administration', 'Streets/Sanitation',
       'Citizen Services - Graffiti', 'Telecommunications',
       'Freedom of Information', 'Adjudication - Ordinance Violation',
       'Assessment & Taxation', 'Harbor Master', 'Personnel',
       'Community Based Orgs', 'Buildings Division', 'Immigration', 'HR',
       'OSP', 'COB APP Issues', 'Youth Bureau',
       'Citizen Services - Thrive Program', 'Rodent_Pest Control',
       'Citizen Services - Good Neighbor', 'Assessment', 'Fair Housing',
       'Citizen Services - Tele-Engagement', 'Test',
       'Citizen Services - Weed & Seed', 'Knowledge',
       'Citizens Services - BreakOut & Love Buffalo', 'ADA'], dtype=object)
```

In [ ]:
```
len(working311["Reason"].unique())
```

Out[ ]:
```
51
```

This one is even larger but for a data set of a 1 million rows we can consider this to be a limited number of catagories.

In [ ]:
```
working311["Type"].unique()
```

Out[ ]:
```
array(['Sweeper (Req_Serv)', 'Police Issue (Req_Serv)',
       'Streetlights (Req_Serv)', 'Tree Planting Request (Req_Serv)',
       'Rodents (Req_Serv)', 'Excess Trash (Req_Serv)',
       'Totes Replace (Req_Serv)', 'Tree Removal (Req_SERV)',
       'Housing Violations (Req_Serv)', 'Street Flooding (Req_Serv)',
       'Dead Animal Removal (Req_Serv)', 'Pest (Req_Serv)',
       'Totes Combo (Req_Serv)', 'Animals (Req_Serv)',
       'Snow Removal Inspection (Req_Serv)', 'Sewer (Req_Serv)',
       'City Parks (Req_Serv)', 'Recycling Missed Pick Up (Req_Serv)',
       'User Fee (Req_Serv)', 'Taxation Issue (Req_Serv)',
       'Tree Trimming Request (Req_Serv)', 'Bulk Trash (Req_Serv)',
       'Water (Req_Serv)', 'Street Snow Plowing (Req_Serv)',
       'Olmsted Parks (Req_Serv)', 'Garbage Missed Pick Up (Req_Serv)',
       'PW Vacant Lot (Req_Serv)', 'Pot Hole (Req_Serv)',
       'PVB Single Meter (Req_Serv)', 'Tree Other (Req_Serv)',
       'Recycling Tote Deliver (Req_Serv)',
       'Pools and Splashpads (Req_Serv)', 'BMHA Issue (Req_Serv)',
       'PVB Pay Station (Req_Serv)', 'Paving (Req_Serv)',
       'Quality of Life Issue (Req_Serv)', 'Fire (Req_Serv)',
       'Lead Paint Inspection (Req_Serv)', 'Totes Deliver (Req_Serv)',
       'Electronic Waste (Req_Serv)', 'Boarding request (Req_Serv)',
       'Electrical Issue (Req_Serv)', 'Fridge (Req_Serv)',
       'Trash Ordinance Violation (Req_Serv)', 'Stump Removal (Req_Serv)',
       'Rental Registration (Req_Serv)', 'Street Salting (Req_Serv)',
       'Cave In (Req_Serv)', 'Christmas Tree (Req_Serv)',
       'Dog License Issue (Req_Serv)',
       'Street Snow Plowing Issue (Req_Serv)',
       'Recycling Bin Pickup (Req_Serv)', 'Totes Audit (Req_Serv)',
       'Totes Pickup (Req_Serv)', 'Fallen Tree Blocking RoW (Req_Serv)',
       'Pick and Pay (Req_Serv)', 'City Clerk Issue (Req_Serv)',
       'Graffiti Private Property (Req_Serv)', 'Water Issue (Req_Serv)',
       'Recycling Tote Combo (Req_Serv)',
       'Leaves / Lawn Debris (Req_Serv)',
       'Totes Abandon Pickup (Req_Serv)', 'Snow-Bank Removal (Req_Serv)',
       'Illegal Dumping PrivateProperty (Req_Serv)',
       'Recycling Tote Replace (Req_Serv)', 'Open311 Housing',
       'Sign Maintenance (Req_Serv)', 'Sign Hazards (Req_Serv)',
       'Water_Billing_Meter (Req_Serv)', 'Sidewalks (Req_Serv)',
       'Parking Issues (Req_Serv)',
       'Missed Pickup 2_Piece Large Trash (Req_Serv)',
       'Illegal Dumping Street (Req_Serv)', 'QRT Snow Removal (Req_Serv)',
       'Other Hole in Road (Req_Serv)', 'QRT Other Issue (Req_Serv)',
       'Great American Clean-Up (Req_Serv)',
       'Save Our Streets Program (Req_Serv)',
       'Pavement Marking Lines (Req_Serv)',
       'OSP Illegal Dumping (Req_Serv)',
       'Buffalo Traffic Violations (Req_Serv)',
       'Abandoned Vehicles (Req_Serv)', 'Illegal Dumping Curb (Req_Serv)',
       'Recycling Tote Pickup (Req_Serv)', 'Bridge Issue (Req_Serv)',
       'Recycling Tote Abandon Pickup (Req_Serv)',
       'Tree Trimming Quality Issue (Req_Serv)',
       'Basement Flooding (Req_Serv)',
       'Damaged Street Light Pole (Req_Serv)', 'Curbs (Req_Serv)',
       'Recycling - Escalated Questions (Req_Serv)',
       'Rain Barrels (Req_Serv)', 'Fallen Tree Inspection (Req_Serv)',
       'Fair Housing Issue (Req_Serv)', 'Inrem Real Estate (Req_Serv)',
       'Water Tested (Req_Serv)', 'Tree Removal Quality Issue (Req_serv)',
       'PW Neighborhood Traffic Calming (Req_Serv)',
       'Right of Way Issue (Req_Serv)',
       'Damage from Street Worker (Req_Serv)', 'City Property (Req_Serv)',
```

```
'Curb - Metal Protruding (Req_Serv)',
'Abandoned Vehicle Inspection (Req_Serv)',
'Private Property (Req_Serv)', 'Telecommunications (Req_Serv)',
'Basketball Hoop in RoW (Req_Serv)',
'PW Ongoing Construction (Req_Serv)',
'FOIL Records City Clerk (Req_Serv)', 'Illegal Dumping (Req_Serv)',
'Neighborhood CleanUp (Req_Serv)',
'Signal Timing Issue City (Req_Serv)',
'Assessment Issue (Req_Serv)',
'Other Adjudication Issue (Req_Serv)',
'Vehicle Blocking Snow Plow (Req_Serv)',
'Signal Other Issue (Req_Serv)', 'BFD Snow on Hydrant (Req_Serv)',
'Stump Removal Quality Issue (Req_Serv)', 'PVB Mobile App Ticket',
'Fire Hydrant Issue (Req_Serv)',
'Signal Out or Flashing (Req_Serv)',
'City Park Tree Issue (Req_Serv)',
'Tree Removal per Inspector (Req_Serv)', 'PVB Mobile App Issue',
'Waterfront Issues (Req_Serv)',
'Totes Report of Removed Broken Tote (Req_Serv)',
'Damage from Snow Removal (Req_Serv)',
'Engineering Operational (Req_Serv)',
'Park Garbage Pickup (Req_Serv)', 'Sneakers Hanging (Req_Serv)',
'Parks City (Req_Serv)', 'CBO (Req_Serv)',
'PW Missing Manhole Cover (Req_Serv)',
'FOIL Records Board of Ethics (Req_Serv)',
'Building Maintenance (Req_Serv)', 'Bike Rack (Req_Serv)',
'Obscene PW Engineering (Req_Serv)',
'FOIL Records Assessment & Taxation (Req_Serv)',
'NYS New Americans (Req_Serv)',
'Compensation & Benefits (Req_Serv)', 'Other (Req_Serv)',
'OSP Other Issue (Req_Serv)',
'QRT Illegal Dumping on Viaduct (Req_Serv)',
'Signal Issue DoT (Req_Serv)', 'Obscene City Property (Req_Serv)',
'Snow Removal Bus Stop/Shelter (Req_Serv)',
'FOIL Records EDPIS (Req_Serv)',
'CityHall_CityCourt Maintenance (Req_Serv)',
'Obscene Private Property (Req_Serv)',
'Special Event Totes (Req_Serv)',
'Tree Planting Quality Issue (Req_Serv)', 'PW Traffic (Req_Serv)',
'PW Engineering (Req_Serv)', 'Snow Removal (Req_Serv)',
'FOIL Records AdmFinance_Policy_UrbanAff (Req_Serv)',
'Recreation Center (Req_Serv)',
'FOIL Records Real Estate (Req_Serv)', 'Open311 Graffiti',
'FOIL Records Police Dept (Req_Serv)',
'Obscene PW Traffic (Req_Serv)', 'Small Cell (Req_Serv)',
'FOIL Records Law Dept (Req_Serv)',
'FOIL Records Division of Purchase (Req_Serv)',
'Tree Removal Challenge (Req_Serv)',
'Open311 APP Issues (Req_Serv)', 'Obscene Other (Req_Serv)',
'Youth Bureau Issue (Req_Serv)',
'FOIL Records Adjudication (Req_Serv)',
'FOIL Records Fire Dept (Req_Serv)',
'Thrive Work Development (Req_Serv)',
'FOIL Records Planning Board (Req_Serv)',
'FOIL Records Human Resources (Req_Serv)',
'BFD Fire Prevention (Req_Serv)',
'PVB School-Zone Camera Citations (Req_Serv)',
'Chicken Coop (Req_Serv)',
'FOIL Records Citizens Services (Req_Serv)',
'FOIL Records Traffic Violations (Req_Serv)',
```

```
       'Good Neighbor (Req_Serv)', 'Obscene Parks City (Req_Serv)',
       '2020 Reassessment', 'FairHsg Other (Req_Serv)',
       'Tele-Engagement Citizen Services (Req_Serv)',
       'FOIL Records Compensation and Benefits (Req_Serv)',
       'FOIL Records Parking (Req_Serv)',
       'Tele-Engagement Board of Block Clubs (Req_Serv)',
       'FOIL Records Community Srvs & RecProg (Req_Serv)',
       'FOIL Records Civil Service (Req_Serv)', 'Pest_Private Trap',
       'FOIL Records Strategic Planning (Req_Serv)',
       'Tele-Engagement Community Services (Req_Serv)',
       'Obscene Schools (Req_Serv)',
       'FOIL Records Audit Control (Req_Serv)',
       'Parks Olmsted (Req_Serv)', 'Security Deposit (Req_Serv)',
       'Tele-Engagement Strategic Planning (Req_Serv)',
       'ReOpen Buffalo (Req_Serv)', 'Erie Basin Marina (Req_Serv)',
       'Tele-Engagement BETC (Req_Serv)',
       'Obscene Parks Olmsted (Req_Serv)',
       'FOIL Records Office of the Mayor (Req_Serv)',
       'Tele-Engagement Police (Req_Serv)',
       'FOIL Records Preservation Board (Req_Serv)', 'Test2',
       'Parking Meter Issue (Req_Serv)', 'QRT Vacant Lot (Req_Serv)',
       'Test_before', 'Pavement Markings Other (Req_Serv)',
       'Ordinance Violation (Req_Serv)',
       'Recycling Bin Delivery (Req_Serv)',
       'Parks Usage Survey (Req_Serv)', 'Block Club CleanUp (Req_Serv)',
       'Housing Violations (Prov_Info)', 'Obscene Thruway (Req_Serv)',
       'Weed & Seed Program (Req_Serv)', 'Schools (Req_Serv)',
       'Escalated Question', 'Neighborhood Watch Signs (Req_Serv)',
       'Thruway (Req_Serv)', 'RalphWilson Park Construction (Req_Serv)',
       'Scrap_It Curbside (Req_Serv)',
       'QRT Illegal Dumping on Vacant Lot (Req_Serv)',
       'BreakOut & LoveBuffalo (Req_Serv)', 'ADA-Other (Req_Serv)',
       'ADA-PW Sidewalks (Req_Serv)'], dtype=object)
```

In [ ]:
```
len(working311["Type"].unique())
```

Out[ ]:
```
223
```

Fairly large amount of catagories, however still limited when we look at this number next to the number of rows.

reference:'Police District', 'Neighborhood'

In [ ]:
```
working311["Police District"].unique()
```

Out[ ]:
```
array(['UNKNOWN', 'District D', 'District A', 'District E', 'District B',
       'District C'], dtype=object)
```

In [ ]:
```
len(working311["Police District"].unique())
```

Out[ ]:
```
6
```

In [ ]:
```
working311["Neighborhood"].unique()
```

Out[ ]:
```
array(['UNKNOWN', 'Riverside', 'Hopkins-Tifft', 'South Park', 'Kenfield',
       'Black Rock', 'Schiller Park', 'Grant-Amherst', 'Elmwood Bryant',
       'Lovejoy', 'Fillmore-Leroy', 'First Ward', 'Elmwood Bidwell',
       'Central Park', 'Kaisertown', 'Fruit Belt', 'Broadway Fillmore',
       'Parkside', 'North Park', 'MLK Park', 'Genesee-Moselle',
       'West Side', 'Ellicott', 'Kensington-Bailey', 'Allentown',
       'West Hertel', 'Delavan Grider', 'Upper West Side',
       'Seneca-Cazenovia', 'Hamlin Park', 'Seneca Babcock',
       'Lower West Side', 'Masten Park', 'Pratt-Willert', 'Central',
       'University Heights'], dtype=object)
```

In [ ]:
```
len(working311["Neighborhood"].unique())
```

Out[ ]:
```
36
```

In [ ]:
```
len(working311["Object Type"].unique())
```

Out[ ]:
```
5
```

In [ ]:
```
working311["Object Type"].unique()
```

Out[ ]:
```
array(['Street', 'Property', 'Individual', 'Unknown', 'Organisation'],
      dtype=object)
```

In [ ]:
```
#Lets add a seperate year and month column
working311["Year"]=pd.DatetimeIndex(working311["Open Date"]).year
working311["Month"]=pd.DatetimeIndex(working311["Open Date"]).month
```

In [ ]:
```
working311.shape
```

Out[ ]:
```
(1110933, 11)
```

produce some graphs explaining how they are related to council district, police district, neighborhood.

In [ ]:
```
working311.head()
```

Out[ ]:

	Open Date	Status	Subject	Reason	Type	Object Type	Council District	Police District	Neighborhood
0	2010-07-19 12:06:00	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
1	2018-02-12 06:47:00	Closed	Buffalo Police Department	Police	Police Issue (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
2	2011-01-11 09:12:00	Closed	Utilities	National Grid	Streetlights (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
3	2017-07-10 11:36:00	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
4	2018-08-18 15:44:00	Closed	Dept of Public Works	Forestry	Tree Planting Request (Req_Serv)	Property	UNKNOWN	UNKNOWN	UNKNOWN

Lets organize and remind myself what I should be comparing again.

Index(['Open Date', 'Status', 'Subject', 'Reason', 'Type', 'Object Type', 'Council District', 'Police District', 'Neighborhood'], dtype='object')

'Status' 2, 'Subject' 18, 'Reason' 51, 'Type' 223, 'Object Type 5'

'Council District' 10, 'Police District'6, 'Neighborhood'36

Lets Just make sure they are

In [ ]:
```
Compare_1 = working311.groupby("Police District").Type.value_counts(normalize=False)
```

In [ ]:
```
Compare_1 = pd.DataFrame(Compare_1)
Compare_1
```

Out[ ]:

		Type
Police District	**Type**	
District A	**Housing Violations (Req_Serv)**	26054
	Totes Replace (Req_Serv)	21202
	Garbage Missed Pick Up (Req_Serv)	8530
	Pot Hole (Req_Serv)	8124
	Recycling Missed Pick Up (Req_Serv)	6054
...	**...**	...
UNKNOWN	**Tele-Engagement Citizen Services (Req_Serv)**	1
	Tele-Engagement Community Services (Req_Serv)	1
	Tele-Engagement Police (Req_Serv)	1
	Test2	1
	Weed & Seed Program (Req_Serv)	1

1085 rows × 1 columns

In [ ]:
```
Compare_1.loc['District A',:].sum()
```

Out[ ]:
```
Type    176476
dtype: int64
```

This would be hard to describe with a limited number of catagories. It has a length of 1085 with 223 types and 6 police districts. Dealing with each catagory visually would be messy.

In [ ]:
```
Compare_2 = working311.groupby("Police District").Reason.value_counts(normalize=False)
```

In [ ]:
```
Compare_2 = pd.DataFrame(Compare_2)
Compare_2
```

Out[ ]:

Police District	Reason	Reason
District A	Sanitation	55622
	Housing	29475
	Engineering - Street Repairs	17437
	Streets	15248
	Forestry	9511
...	...	...
UNKNOWN	Knowledge	4
	Citizens Services - BreakOut & Love Buffalo	2
	Citizen Services - Weed & Seed	1
	Fair Housing	1
	Test	1

266 rows × 1 columns

I may try to graph this and see what tranpires. But I may just do away with it becuase it doesn't seem as though the outcomes fit easyly into a limited number of catagory for Reasons.

In [ ]:
```
Compare_3 = working311.groupby("Police District").Status.value_counts(normalize=False)
```

In [ ]:
```
Compare_3=pd.DataFrame(Compare_3)

Compare_3
```

Out[ ]:

		Status
Police District	**Status**	
District A	**Closed**	175801
	Open	675
District B	**Closed**	149414
	Open	698
District C	**Closed**	198554
	Open	786
District D	**Closed**	259651
	Open	1047
District E	**Closed**	230324
	Open	823
UNKNOWN	**Closed**	91945
	Open	1215

In [ ]:
```
Compare_4 = working311.groupby("Police District")['Object Type'].value_counts(normaliz
```

In [ ]:
```
Compare_4=pd.DataFrame(Compare_4)
Compare_4
```

Out[ ]:

		Object Type
Police District	**Object Type**	
District A	**Property**	168230
	Street	8246
District B	**Property**	150111
	Street	1
District C	**Property**	199338
	Street	2
District D	**Property**	260698
District E	**Property**	231147
UNKNOWN	**Street**	63145
	Property	17933
	Unknown	6838
	Individual	5092
	Organisation	152

In [ ]:
```
Compare_5 = working311.groupby("Police District").Type.value_counts(normalize=False)
```

In [ ]:
```
Compare_5 =pd.DataFrame(Compare_5)
Compare_5
```

Out[ ]:

		Type
Police District	**Type**	
District A	**Housing Violations (Req_Serv)**	26054
	Totes Replace (Req_Serv)	21202
	Garbage Missed Pick Up (Req_Serv)	8530
	Pot Hole (Req_Serv)	8124
	Recycling Missed Pick Up (Req_Serv)	6054
...	**...**	...
UNKNOWN	**Tele-Engagement Citizen Services (Req_Serv)**	1
	Tele-Engagement Community Services (Req_Serv)	1
	Tele-Engagement Police (Req_Serv)	1
	Test2	1
	Weed & Seed Program (Req_Serv)	1

1085 rows × 1 columns

In [ ]:
```
Compare_A2 = working311.groupby("Neighborhood").Reason.value_counts(normalize=False)
```

In [ ]:
```
Compare_A2 =pd.DataFrame(Compare_A2)
Compare_A2
```

Out[ ]:

		Reason
Neighborhood	**Reason**	
Allentown	**Sanitation**	5706
	Housing	2901
	Engineering - Street Repairs	1853
	Parking Violations Bureau	1550
	Streets	1420
...	**...**	...
West Side	**Community Based Orgs**	3
	Citizen Services - Good Neighbor	2
	Telecommunications	2
	Fair Housing	1
	OSP	1

1362 rows × 1 columns

In [ ]:
```
Compare_A1 = working311.groupby("Neighborhood").Type.value_counts(normalize=False)
```

In [ ]:
```
Compare_A1 = pd.DataFrame(Compare_A1)
Compare_A1
```

Out[ ]:

		Type
Neighborhood	**Type**	
Allentown	**Housing Violations (Req_Serv)**	1905
	Totes Replace (Req_Serv)	1674
	Parking Issues (Req_Serv)	1348
	Pot Hole (Req_Serv)	1039
	Garbage Missed Pick Up (Req_Serv)	934
...	**...**	...
West Side	**PVB School-Zone Camera Citations (Req_Serv)**	1
	Pavement Markings Other (Req_Serv)	1
	Schools (Req_Serv)	1
	Special Event Totes (Req_Serv)	1
	Tree Removal Challenge (Req_Serv)	1

5450 rows × 1 columns

In [ ]:
```
Compare_A5 = working311.groupby("Neighborhood").Type.value_counts(normalize=False)
```

In [ ]:
```
Compare_A5 = pd.DataFrame(Compare_A5)
Compare_A5
```

Out[ ]:

		Type
Neighborhood	**Type**	
Allentown	**Housing Violations (Req_Serv)**	1905
	Totes Replace (Req_Serv)	1674
	Parking Issues (Req_Serv)	1348
	Pot Hole (Req_Serv)	1039
	Garbage Missed Pick Up (Req_Serv)	934
...	**...**	...
West Side	**PVB School-Zone Camera Citations (Req_Serv)**	1
	Pavement Markings Other (Req_Serv)	1
	Schools (Req_Serv)	1
	Special Event Totes (Req_Serv)	1
	Tree Removal Challenge (Req_Serv)	1

5450 rows × 1 columns

In [ ]:
```
Compare_A4 = working311.groupby("Neighborhood")['Object Type'].value_counts(normalize=
```

In [ ]:
```
Compare_A4 = pd.DataFrame(Compare_A4)
Compare_A4
```

Out[ ]:

		Object Type
Neighborhood	**Object Type**	
Allentown	**Property**	17515
Black Rock	**Property**	15937
Broadway Fillmore	**Property**	63542
	Street	1
Central	**Property**	16922
Central Park	**Property**	18995
Delavan Grider	**Property**	25309
Ellicott	**Property**	9125
Elmwood Bidwell	**Property**	47801
Elmwood Bryant	**Property**	30056
Fillmore-Leroy	**Property**	26772
First Ward	**Property**	10696
Fruit Belt	**Property**	13365
Genesee-Moselle	**Property**	45475
Grant-Amherst	**Property**	17429
Hamlin Park	**Property**	22920
Hopkins-Tifft	**Property**	29318
	Street	8245
Kaisertown	**Property**	24106
Kenfield	**Property**	26241
Kensington-Bailey	**Property**	52308
Lovejoy	**Property**	39716
	Street	1
Lower West Side	**Property**	15962
	Street	1
MLK Park	**Property**	20450
Masten Park	**Property**	34905
North Park	**Property**	54754
Parkside	**Property**	25742
Pratt-Willert	**Property**	11889
Riverside	**Property**	37894
Schiller Park	**Property**	39938
Seneca Babcock	**Property**	15118

		Object Type
Neighborhood	**Object Type**	
Seneca-Cazenovia	**Property**	30175
South Park	**Property**	48729
	Street	1
UNKNOWN	**Street**	63145
	Property	16727
	Unknown	6838
	Individual	5092
	Organisation	152
University Heights	**Property**	38077
Upper West Side	**Property**	36746
West Hertel	**Property**	12497
West Side	**Property**	34306

In [ ]:
```
Compare_A3 = working311.groupby("Neighborhood").Status.value_counts(normalize=False)
```

In [ ]:
```
Compare_A3 = pd.DataFrame(Compare_A3)
Compare_A3
```

Out[ ]:

		Status
Neighborhood	**Status**	
Allentown	**Closed**	17426
	Open	89
Black Rock	**Closed**	15864
	Open	73
Broadway Fillmore	**Closed**	63309
...	**...**	...
Upper West Side	**Open**	145
West Hertel	**Closed**	12451
	Open	46
West Side	**Closed**	34160
	Open	146

72 rows × 1 columns

In [ ]:
```
Compare_B2 = working311.groupby("Council District").Reason.value_counts(normalize=Fals
```

In [ ]:
```
Compare_B2 = pd.DataFrame(Compare_B2)
Compare_B2
```

Out[ ]:

		Reason
Council District	**Reason**	
DELAWARE	**Sanitation**	5346
	Streets	2517
	Housing	2202
	Engineering - Street Repairs	1967
	Parking Violations Bureau	988
...	**...**	...
UNKNOWN	**Knowledge**	5
	ADA	2
	Citizens Services - BreakOut & Love Buffalo	2
	Immigration	2
	Test	1

352 rows × 1 columns

In [ ]:
```
Compare_B1 = working311.groupby("Council District").Type.value_counts(normalize=False)
```

In [ ]:
```
Compare_B1 = pd.DataFrame(Compare_B1)
```

In [ ]:
```
Compare_B1
```

Out[ ]:

		Type
Council District	**Type**	
DELAWARE	**Totes Replace (Req_Serv)**	1420
	Garbage Missed Pick Up (Req_Serv)	1264
	Housing Violations (Req_Serv)	1228
	Street Snow Plowing (Req_Serv)	1124
	Pot Hole (Req_Serv)	1108
...	**...**	...
UNKNOWN	**Tele-Engagement Citizen Services (Req_Serv)**	1
	Tele-Engagement Community Services (Req_Serv)	1
	Tele-Engagement Police (Req_Serv)	1
	Test2	1
	Thruway (Req_Serv)	1

1442 rows × 1 columns

In [ ]:
```
Compare_B5 = working311.groupby("Council District").Type.value_counts(normalize=False)
```

In [ ]:
```
Compare_B5 = pd.DataFrame(Compare_B5)
```

In [ ]:
```
Compare_B5
```

Out[ ]:

		Type
Council District	**Type**	
DELAWARE	**Totes Replace (Req_Serv)**	1420
	Garbage Missed Pick Up (Req_Serv)	1264
	Housing Violations (Req_Serv)	1228
	Street Snow Plowing (Req_Serv)	1124
	Pot Hole (Req_Serv)	1108
...	**...**	...
UNKNOWN	**Tele-Engagement Citizen Services (Req_Serv)**	1
	Tele-Engagement Community Services (Req_Serv)	1
	Tele-Engagement Police (Req_Serv)	1
	Test2	1
	Thruway (Req_Serv)	1

1442 rows × 1 columns

In [ ]:
```
Compare_B4 = working311.groupby("Council District")['Object Type'].value_counts(normal
```

In [ ]:
```
Compare_B4 = pd.DataFrame(Compare_B4)
```

In [ ]:
```
Compare_B4
```

Out[ ]:

		Object Type
Council District	**Object Type**	
DELAWARE	**Property**	17946
ELLICOTT	**Property**	21165
FILLMORE	**Property**	24398
LOVEJOY	**Property**	25279
	Street	7343
MASTEN	**Property**	23363
NIAGARA	**Property**	20330
NORTH	**Property**	16527
SOUTH	**Property**	17246
UNIVERSITY	**Property**	18240
UNKNOWN	**Property**	842963
	Street	64051
	Unknown	6838
	Individual	5092
	Organisation	152

In [ ]:
```
Compare_B3 = working311.groupby("Council District").Status.value_counts(normalize=Fals
```

In [ ]:
```
Compare_B3 = pd.DataFrame(Compare_B3)
```

In [ ]:
```
Compare_B3
```

Out[ ]:

		Status
Council District	**Status**	
DELAWARE	**Closed**	17697
	Open	249
ELLICOTT	**Closed**	20748
	Open	417
FILLMORE	**Closed**	23897
	Open	501
LOVEJOY	**Closed**	32219
	Open	403
MASTEN	**Closed**	23060
	Open	303
NIAGARA	**Closed**	20019
	Open	311
NORTH	**Closed**	16172
	Open	355
SOUTH	**Closed**	17012
	Open	234
UNIVERSITY	**Closed**	17953
	Open	287
UNKNOWN	**Closed**	916912
	Open	2184

Graph all these relationships except the ones containing Types

In [ ]:
```
Compare_B3.drop(labels="UNKNOWN", axis=0)
```

Out[ ]:

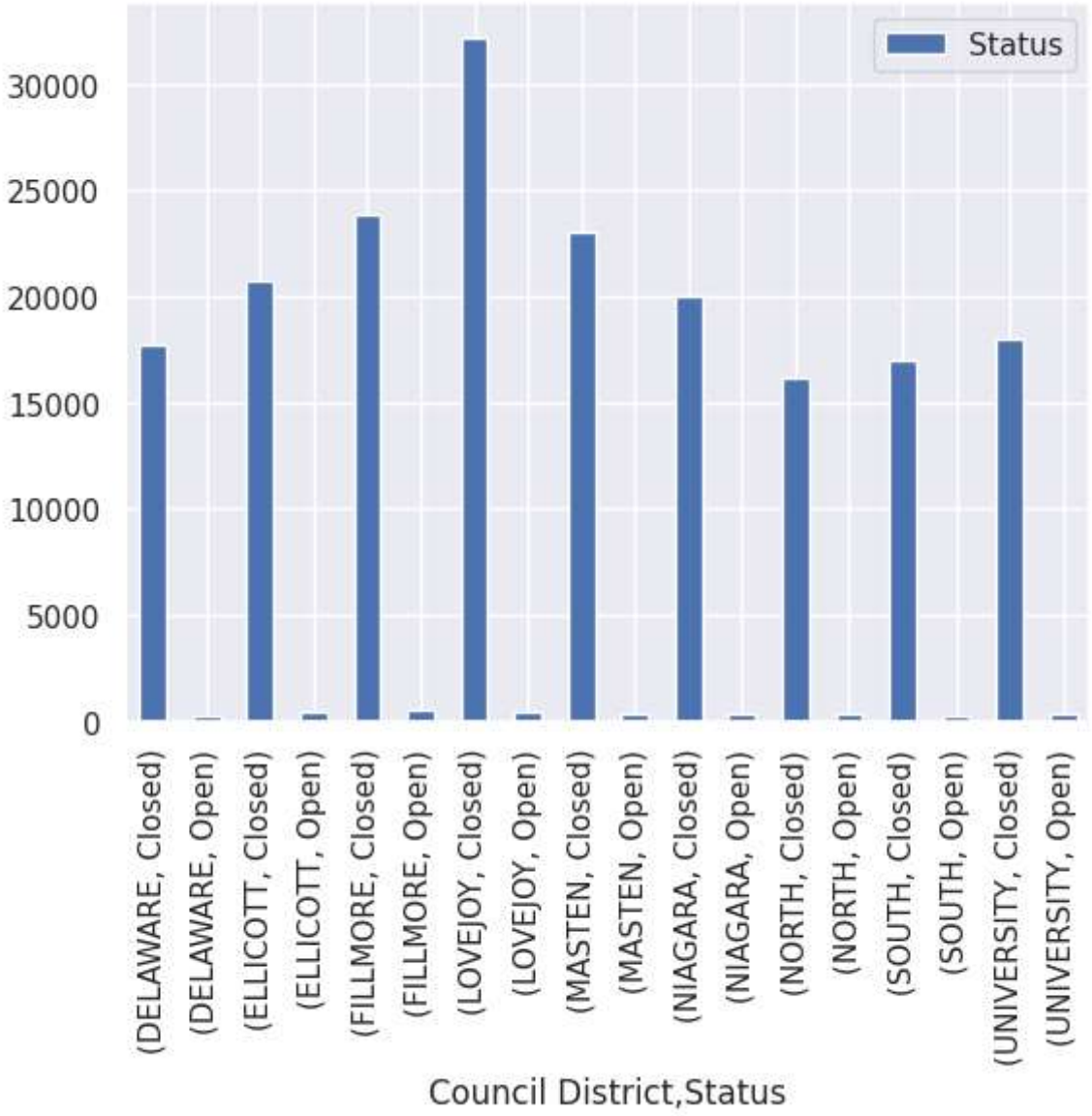
		Status
Council District	**Status**	
DELAWARE	**Closed**	17697
	Open	249
ELLICOTT	**Closed**	20748
	Open	417
FILLMORE	**Closed**	23897
	Open	501
LOVEJOY	**Closed**	32219
	Open	403
MASTEN	**Closed**	23060
	Open	303
NIAGARA	**Closed**	20019
	Open	311
NORTH	**Closed**	16172
	Open	355
SOUTH	**Closed**	17012
	Open	234
UNIVERSITY	**Closed**	17953
	Open	287

Graph the council district, police district, neighborhood vs status

In [ ]:
```
Compare_B3 = Compare_B3.drop(labels="UNKNOWN", axis=0)
```

In [ ]:
```
Compare_B3.plot(kind = 'bar')
```

Out[ ]:
```
<Axes: xlabel='Council District,Status'>
```



I think thats preety, there semms to be very few cases that are open most have been clsoed. Among the closed cases the distribution is roughly uniform. Suggesting the 311 services are optimized to treat all neighbourhoods evenly. I droped the "Unknown" neighbourhood so we can see the impact of neighbourhood and whether or not cases are open or closed. Instead of having 916,000 unknown neighbourhood where we can't observe anything as they aren't identified.

In [ ]:
```
Compare_3
```

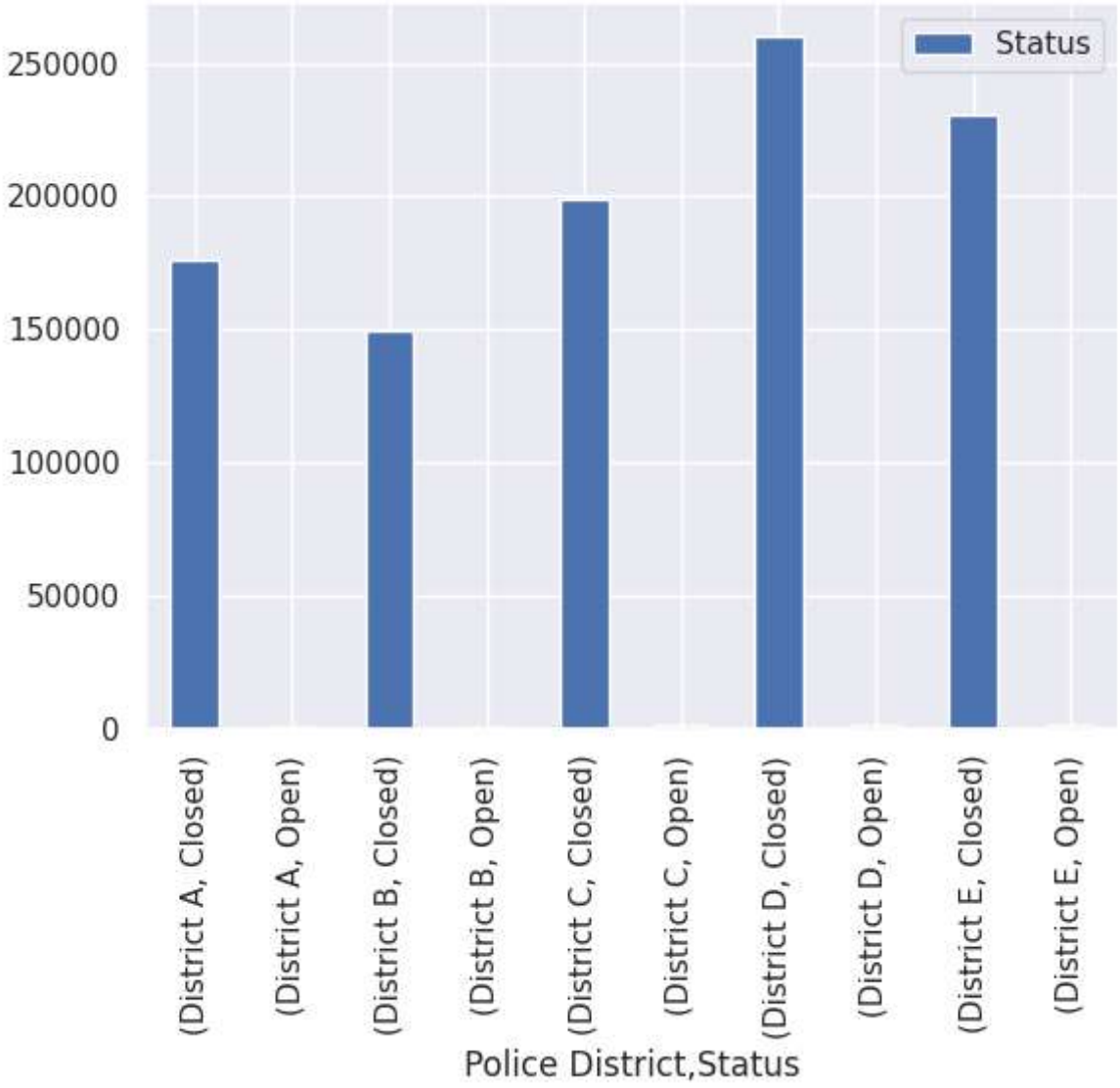
Out[ ]:

		Status
Police District	**Status**	
District A	**Closed**	175801
	Open	675
District B	**Closed**	149414
	Open	698
District C	**Closed**	198554
	Open	786
District D	**Closed**	259651
	Open	1047
District E	**Closed**	230324
	Open	823
UNKNOWN	**Closed**	91945
	Open	1215

In [ ]:
```
Compare_3= Compare_3.drop(labels="UNKNOWN", axis=0)
```

In [ ]:
```
Compare_3.plot(kind = 'bar')
```

Out[ ]:
```
<Axes: xlabel='Police District,Status'>
```



Districkt D seems to have more cases then others but once again police district and council district both semms to have roughly an evely distributed anounts of cases and closed cases.

In [ ]:
```
# Lets do the smae for object Type and police district

Compare_4= Compare_4.drop(labels="UNKNOWN", axis=0)
```

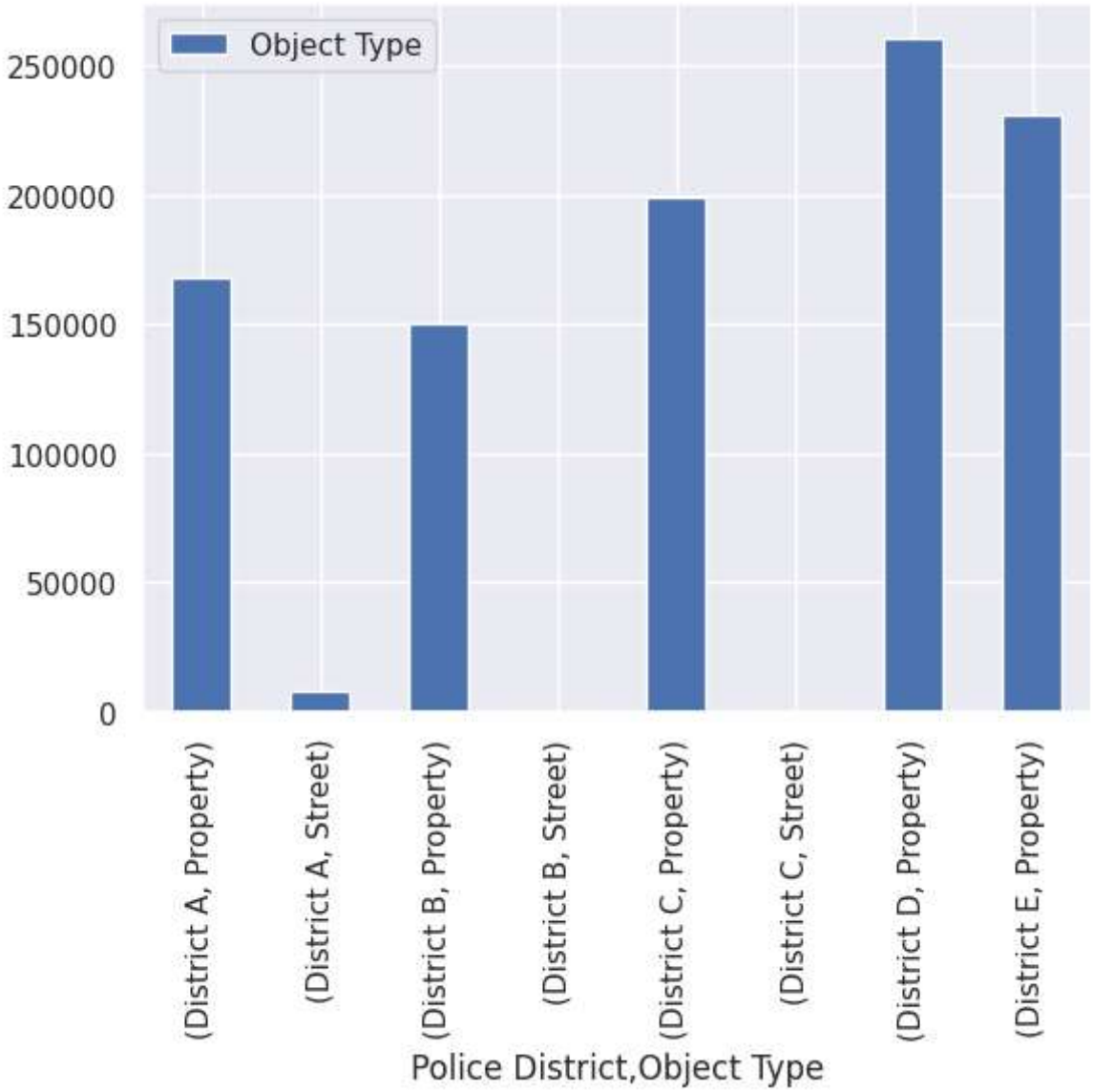
In [ ]:
```
Compare_4
```

Out[ ]:

		Object Type
Police District	**Object Type**	
District A	**Property**	168230
	Street	8246
District B	**Property**	150111
	Street	1
District C	**Property**	199338
	Street	2
District D	**Property**	260698
District E	**Property**	231147

In [ ]:
```
Compare_4.plot(kind = 'bar')
```

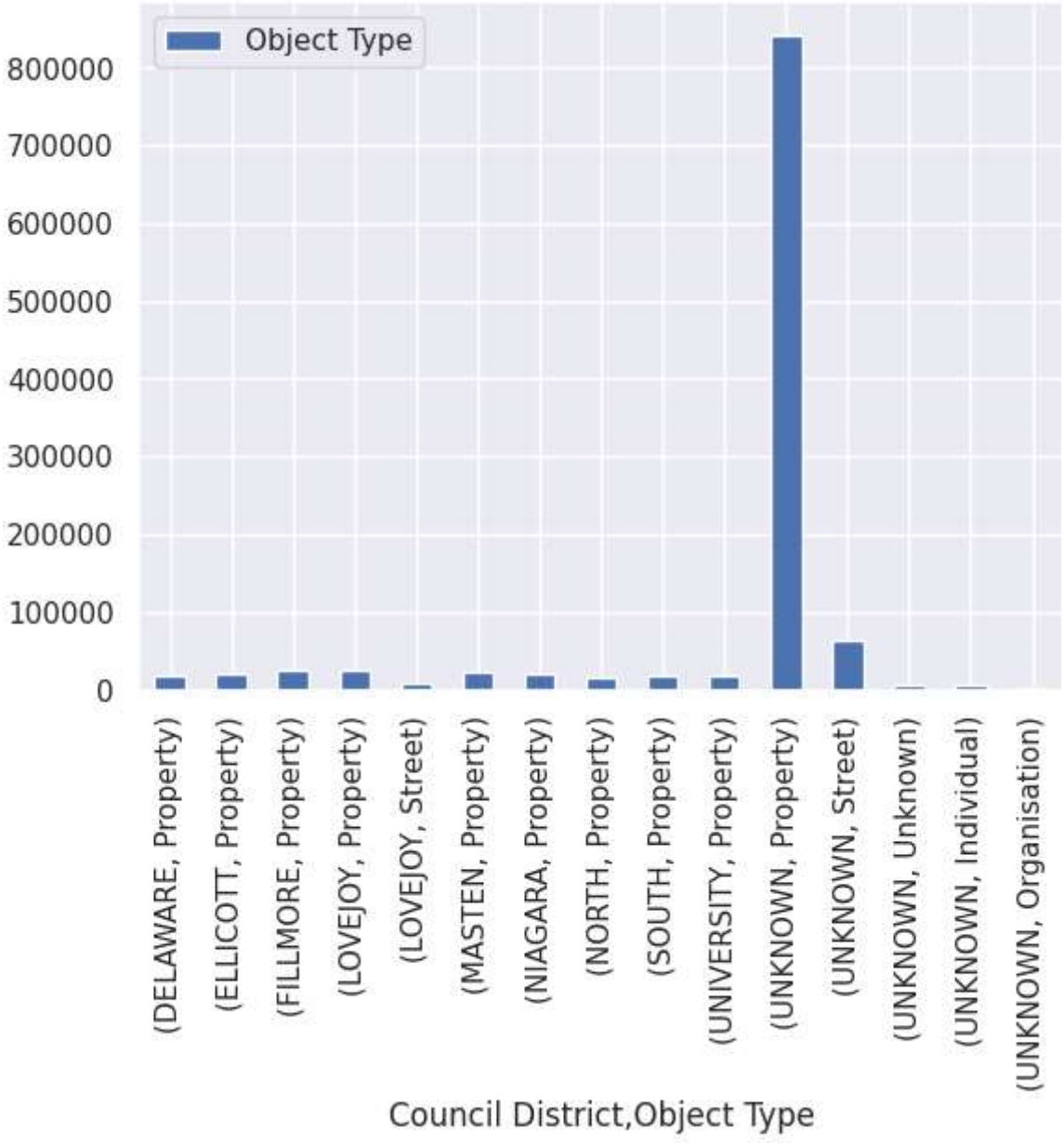
Out[ ]:
```
<Axes: xlabel='Police District,Object Type'>
```



property D and E seems to have higher object type of Property, while only district A has any significat object

In [ ]:
```
Compare_B4.plot(kind = 'bar')
```

Out[ ]:
```
<Axes: xlabel='Council District,Object Type'>
```

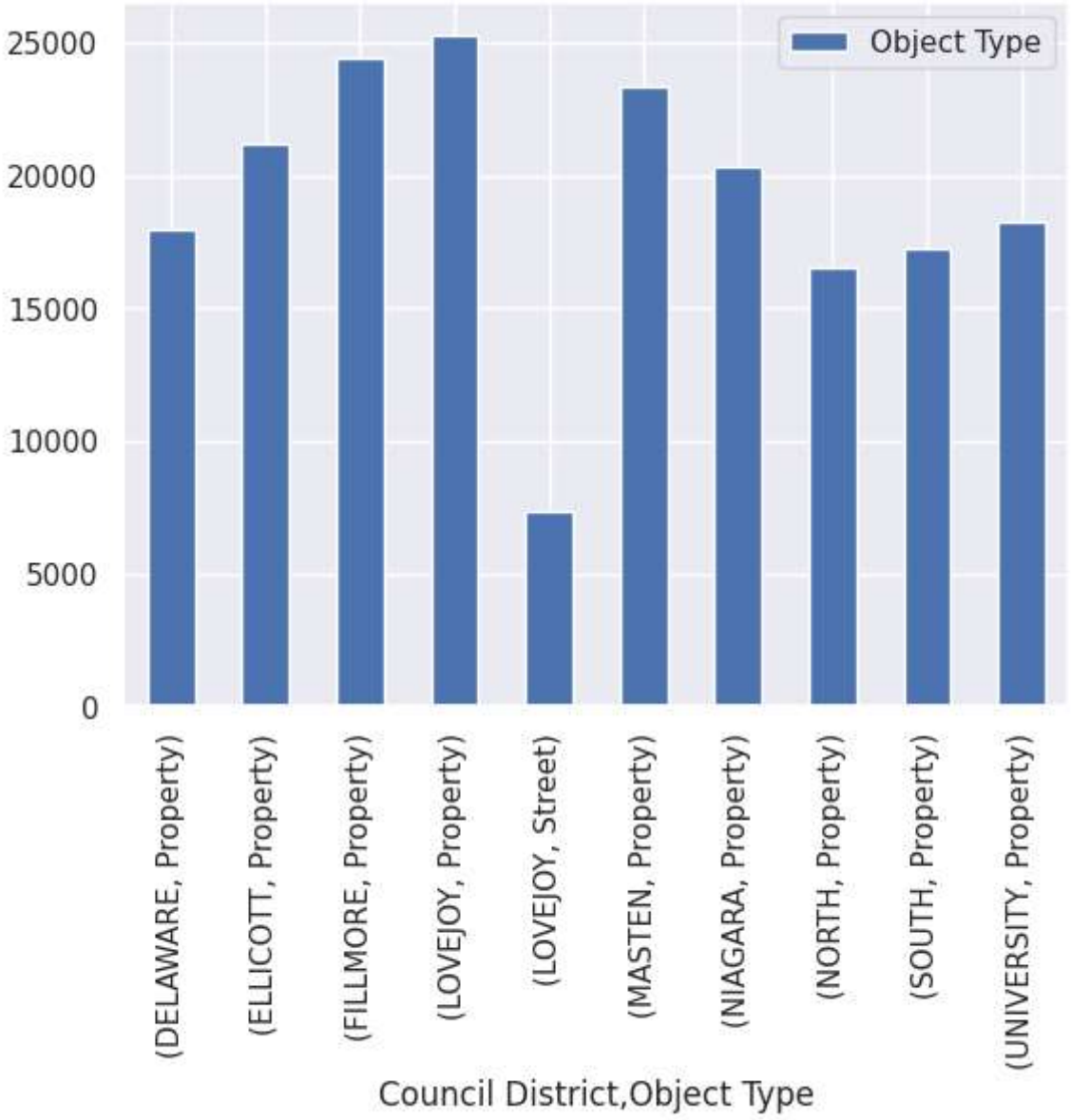


Wanted to show why I am geting rid of it.

In [ ]:
```
Compare_B4 = Compare_B4.drop(labels="UNKNOWN", axis=0)
```

In [ ]:
```
Compare_B4.plot(kind = 'bar')
```

Out[ ]:
```
<Axes: xlabel='Council District,Object Type'>
```



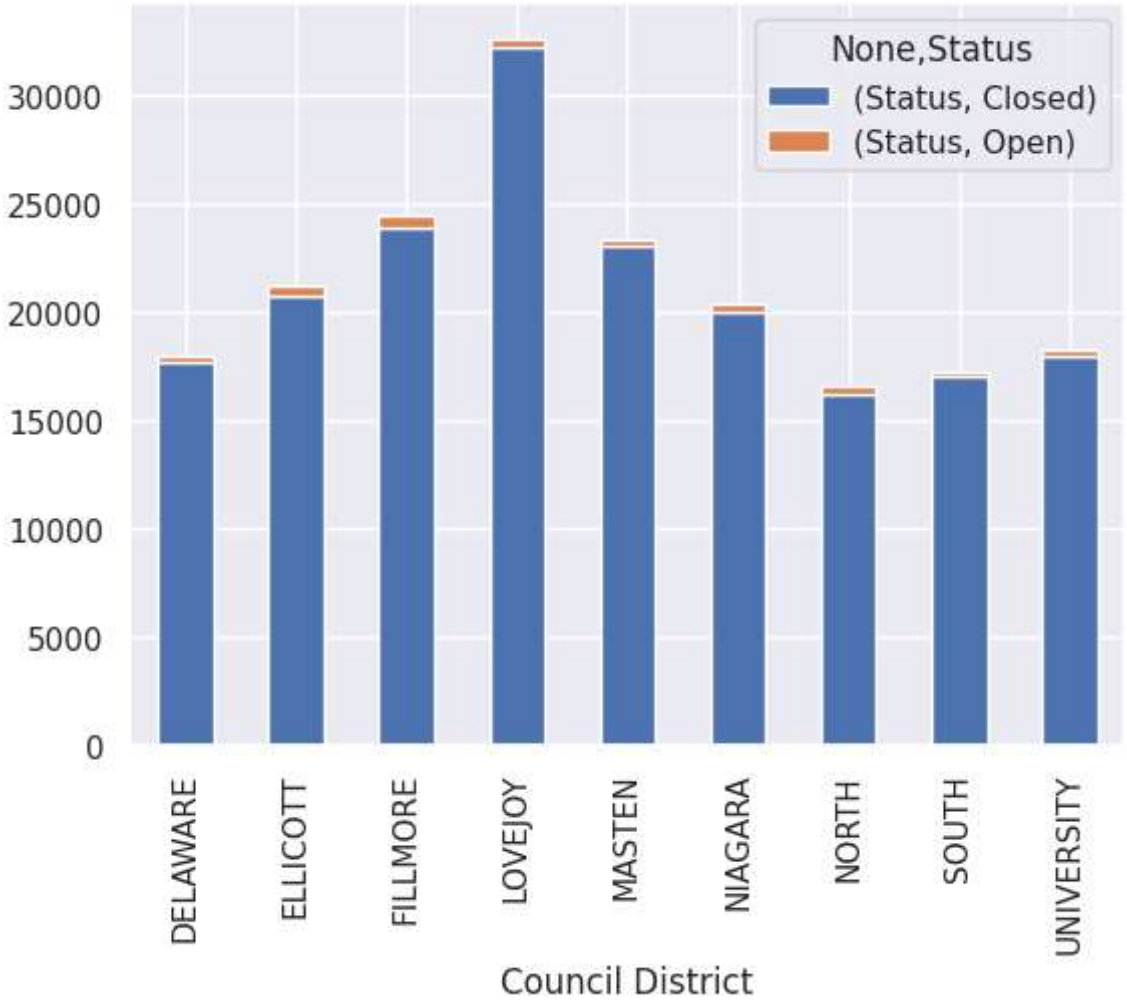
Overall There semms to be a uniform distribution among the property object type of all the council district while only lovejoy has the stree object type.

Do different districts and/or neighborhoods tend to different numbers of 311 calls?

We can simply stack all the neighborrd counts for each neighbor hood to figure this out.

In [ ]:
```
Compare_B3.unstack().plot(kind ='bar', stacked= True)
```

Out[ ]:
```
<Axes: xlabel='Council District'>
```



In [ ]:
```
Compare_B3
```

Out[ ]:

		Status
Council District	**Status**	
DELAWARE	**Closed**	17697
	Open	249
ELLICOTT	**Closed**	20748
	Open	417
FILLMORE	**Closed**	23897
	Open	501
LOVEJOY	**Closed**	32219
	Open	403
MASTEN	**Closed**	23060
	Open	303
NIAGARA	**Closed**	20019
	Open	311
NORTH	**Closed**	16172
	Open	355
SOUTH	**Closed**	17012
	Open	234
UNIVERSITY	**Closed**	17953
	Open	287

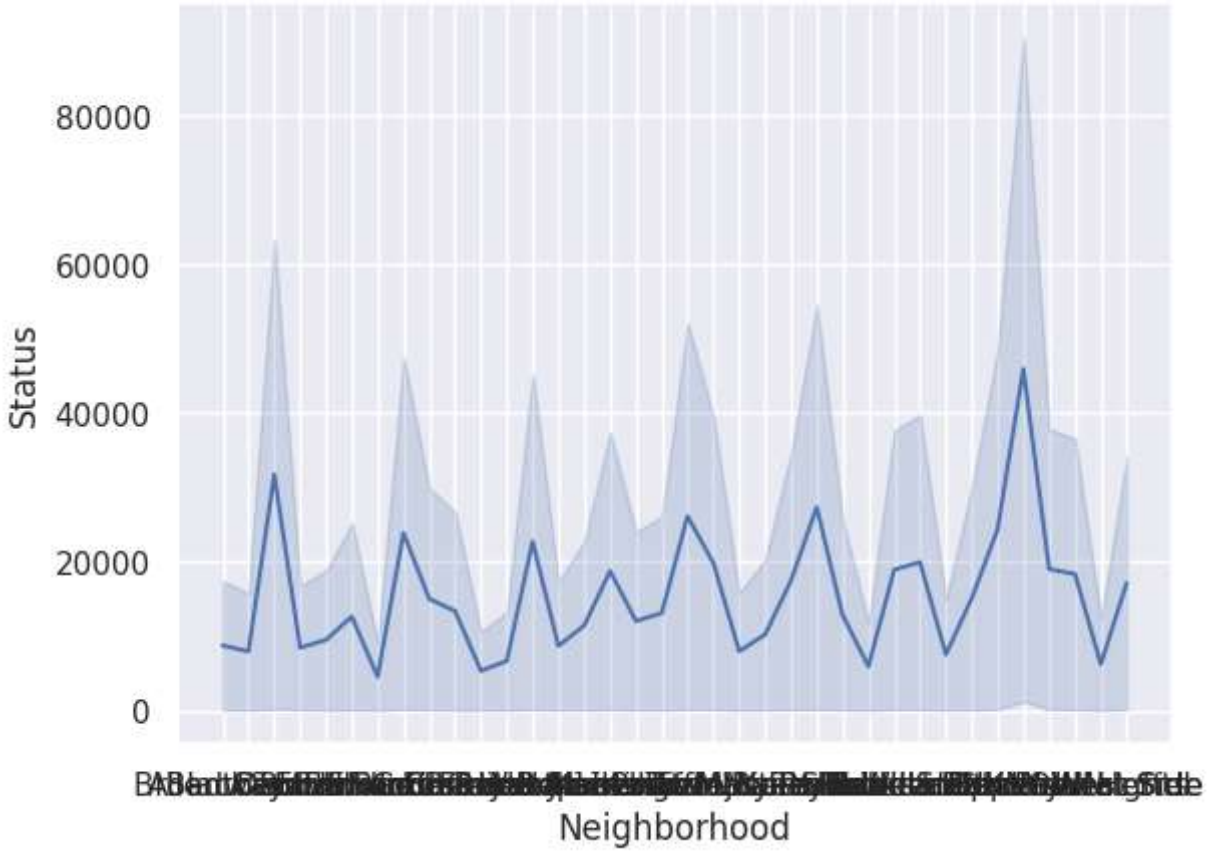
In [ ]:
```
Compare_B3.columns.values.tolist()
```

Out[ ]:
```
['Status']
```

As seen here diffrent neighborood does indeed have diffrent number of calls. In the above cell we can also see where or not the case is closed or open.

In [ ]:
```
sns.set_theme(style="darkgrid")

ax = sns.lineplot(x="Neighborhood", y="Status", data=Compare_A3)
```



This may be a weird way of looking at it with a lineplot however as you can see there are diffrent peaks of varios diffrent levels, so diffffrent neighbourhoods do tend to have diffrent amounts of 311 calls

I tried using other forms of plotting but this seems to be the best.

In [ ]:
```
Compare_A3
```

Out[ ]:

		Status
Neighborhood	**Status**	
Allentown	**Closed**	17426
	Open	89
Black Rock	**Closed**	15864
	Open	73
Broadway Fillmore	**Closed**	63309
...	**...**	...
Upper West Side	**Open**	145
West Hertel	**Closed**	12451
	Open	46
West Side	**Closed**	34160
	Open	146

72 rows × 1 columns

Do different months have different number of 311 reports? What about different years?

In [ ]:
```
working311.head()
```

Out[ ]:

	Open Date	Status	Subject	Reason	Type	Object Type	Council District	Police District	Neighborhood
0	2010-07-19 12:06:00	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
1	2018-02-12 06:47:00	Closed	Buffalo Police Department	Police	Police Issue (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
2	2011-01-11 09:12:00	Closed	Utilities	National Grid	Streetlights (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
3	2017-07-10 11:36:00	Closed	Dept of Public Works	Streets	Sweeper (Req_Serv)	Street	UNKNOWN	UNKNOWN	UNKNOWN
4	2018-08-18 15:44:00	Closed	Dept of Public Works	Forestry	Tree Planting Request (Req_Serv)	Property	UNKNOWN	UNKNOWN	UNKNOWN

In [ ]:
```
yearGroup = working311.groupby('Year').Neighborhood.value_counts()
```

In [ ]:
```
yearGroup = pd.DataFrame(yearGroup)
```

In [ ]:
```
yearGroup
```

Out[ ]:

		Neighborhood
Year	**Neighborhood**	
2008	**UNKNOWN**	3463
	Broadway Fillmore	913
	Genesee-Moselle	718
	West Side	660
	Elmwood Bidwell	646
...	**...**	...
2024	**Pratt-Willert**	134
	First Ward	121
	West Hertel	95
	Ellicott	91
	UNKNOWN	1

612 rows × 1 columns

In [ ]:
```
yearGroup.index.get_level_values(0).unique()
```

Out[ ]:
```
Int64Index([2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018,
            2019, 2020, 2021, 2022, 2023, 2024],
           dtype='int64', name='Year')
```

In [ ]:
```
total_value_sum = yearGroup.groupby(level='Year')['Neighborhood'].sum()
```

In [ ]:
```
total_value_sum = pd.DataFrame(total_value_sum)
```

In [ ]:
```
total_value_sum
```

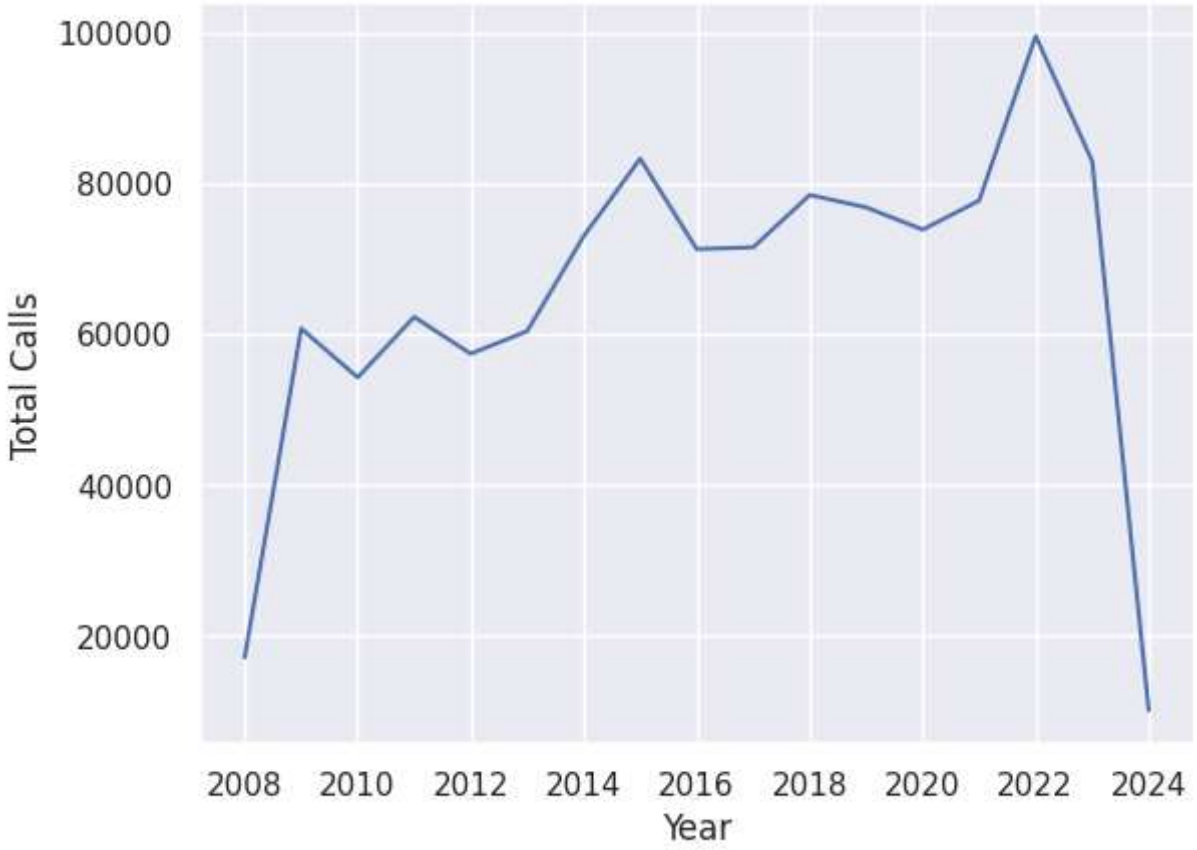
Out[ ]:

Year	Neighborhood
2008	17174
2009	60760
2010	54283
2011	62294
2012	57418
2013	60409
2014	73020
2015	83276
2016	71291
2017	71520
2018	78471
2019	76820
2020	73870
2021	77732
2022	99531
2023	82924
2024	10140

In [ ]:
```
sns.set_theme(style="darkgrid")

ax = sns.lineplot(x="Year", y="Neighborhood", data=total_value_sum)
ax.set(xlabel='Year', ylabel='Total Calls')
```

Out[ ]:
```
[Text(0.5, 0, 'Year'), Text(0, 0.5, 'Total Calls')]
```



The number decreases significantly at the end, that just means it's not a complete year from tableaue experience.

Lets do the same for the months

In [ ]:
```
MonthGroup = working311.groupby('Month').Neighborhood.value_counts()
```

In [ ]:
```
MonthGroup = pd.DataFrame(yearGroup)
```

In [ ]:
```
total_value_sum2 = MonthGroup.groupby(level='Month')['Neighborhood'].sum()
```

In [ ]:
```
total_value_sum2 = pd.DataFrame(total_value_sum2)
```

In [ ]:
```
total_value_sum2
```

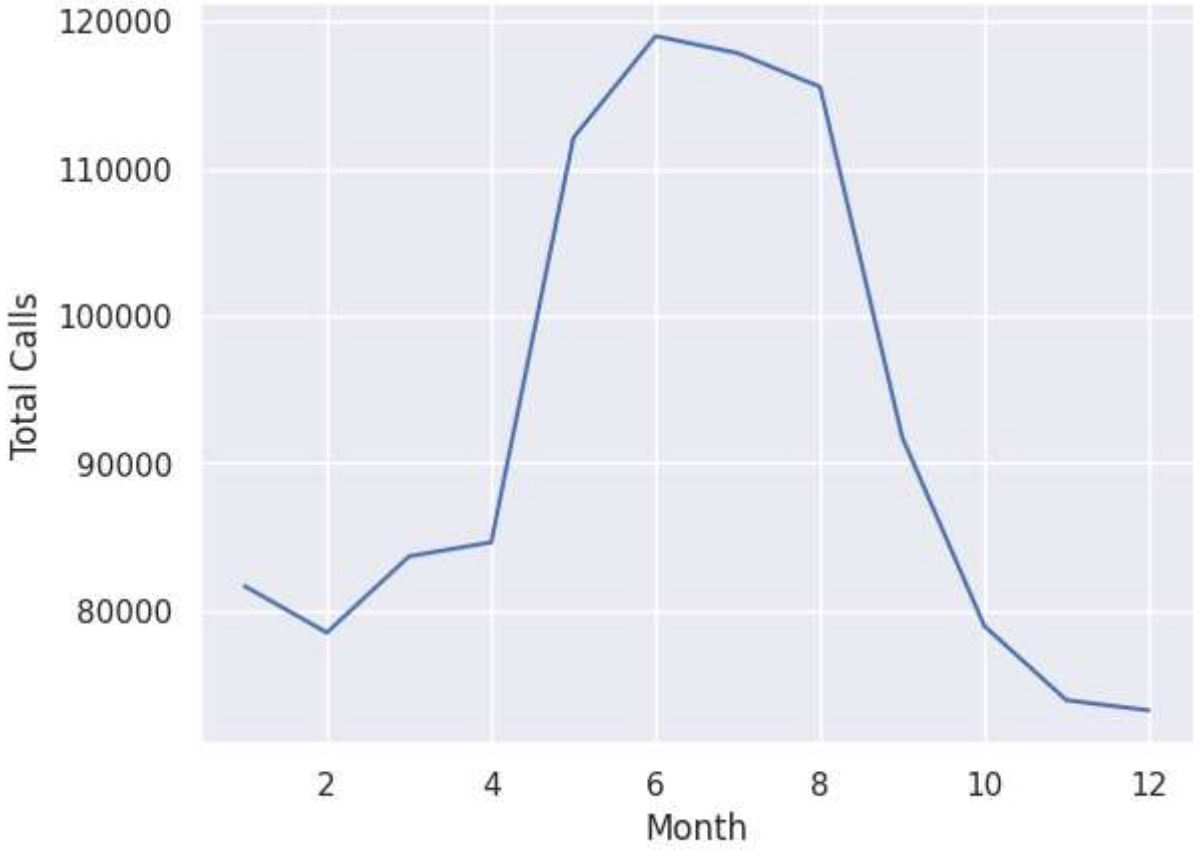
Out[ ]:

Month	Neighborhood
1	81670
2	78519
3	83694
4	84645
5	112100
6	118997
7	117839
8	115553
9	91784
10	78965
11	73927
12	73240

In [ ]:
```
sns.set_theme(style="darkgrid")

ax = sns.lineplot(x='Month', y="Neighborhood", data=total_value_sum2)
ax.set(xlabel='Month', ylabel='Total Calls')
```

Out[ ]:
```
[Text(0.5, 0, 'Month'), Text(0, 0.5, 'Total Calls')]
```



Looks like a bell shaped curve almost identical to a normal distribution shares certain properties of it , such as most of the calls are clustured in the middle. However it's not really the mean so it's inappopriate to say it's normal distribution but I thought that it's very interesting that this graph shares the same properties.

In [ ]: