# Importing the Dataset

In [8]:
```
import seaborn as sns
import pandas as pd
import numpy as np
```

In [2]:
```
df_311calls=pd.read_csv('/content/311_Service_Requests.csv')
df_311calls.columns = df_311calls.columns.str.strip()
df_311calls.head()
```

```
<ipython-input-2-3474cc85883d>:1: DtypeWarning: Columns (0) have mixed types. Specify
dtype option on import or set low_memory=False.
  df_311calls=pd.read_csv('/content/311_Service_Requests.csv')
```

Out[2]:

| | Case Reference | Open Date | Closed Date | Status | Subject | Reason | Type | Object Type | Address Number |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1001911955 | 08/24/2023 01:54:00 PM | 08/31/2023 07:02:00 AM | Closed | Dept of Public Works | Sanitation | Missed Pickup 2_Piece Large Trash (Req_Serv) | Property | 241 |
| **1** | 1001900925 | 07/18/2023 02:04:00 PM | 07/20/2023 10:57:00 AM | Closed | DPIS | Housing | Housing Violations (Req_Serv) | Property | 325 |
| **2** | 1001804417 | 03/06/2023 10:47:00 AM | 03/15/2023 08:38:00 AM | Closed | Utilities | National Grid | Streetlights (Req_Serv) | Property | 201 |
| **3** | 523945-1001902298 | 07/21/2023 10:30:00 PM | 07/24/2023 11:29:00 AM | Closed | Buffalo Police Department | Police | Police Issue (Req_Serv) | Property | 257 |
| **4** | 518978-1001851078 | 05/13/2023 11:30:00 AM | 05/15/2023 11:29:00 AM | Closed | Buffalo Police Department | Police | Police Issue (Req_Serv) | Property | 82 |

5 rows × 33 columns

# Determining the Categorical Variables

Based on the following code, Status and Object Type are the easiest to be described by a limited number of categories. The others cannot be easily described by a limited number of categories because there are many different possible values for them.

In [3]:
```
df_311calls["Status"].unique()
```

Out[3]:
```
array(['Closed', 'Open'], dtype=object)
```

In [4]:
```
df_311calls["Subject"].unique()
```

Out[4]:
```
array(['Dept of Public Works', 'DPIS', 'Utilities',
       'Buffalo Police Department', 'Dept of Parking', 'Dept of Law',
       'Buffalo Fire Department', 'Buffalo Municipal Housing Authority',
       'Office of the Mayor', 'Office of Strategic Planning',
       'Assessment & Taxation', 'City Clerk',
       'Community Services & Rec. Program.', 'Human Resources',
       'Knowledge Management', 'Management Information Systems', 'Test',
       'New Americans'], dtype=object)
```

In [5]:
```
df_311calls["Reason"].unique()
```

Out[5]:
```
array(['Sanitation', 'Housing', 'National Grid', 'Police',
       'Engineering - Traffic', 'Engineering - Street Repairs',
       'Forestry', 'Moving Violations', 'Freedom of Information',
       'Rodent_Pest Control', 'Buffalo Water Authority', 'Streets',
       'Parking Violations Bureau', 'Buffalo Sewer Authority',
       'City Parks', 'BFD', 'BMHA', 'Animal Shelter',
       'Citizen Services - Quick Response Teams', 'Streets/Sanitation',
       'Citizen Services - Graffiti', 'Real Estate', 'OSP',
       'Citizens Services - Clean City', 'Assessment', 'Personnel',
       'Taxation', 'Administration', 'Buildings Division',
       'Adjudication - Ordinance Violation', 'City Clerk Issue',
       'Citizen Services - Save Our Streets', 'Harbor Master',
       'Citizen Services - Good Neighbor', 'Youth Bureau',
       'Community Based Orgs', 'Licenses', 'Rodent Control',
       'Fair Housing', 'HR', 'Knowledge', 'COB APP Issues',
       'Assessment & Taxation', 'Telecommunications', 'Test',
       'Immigration', 'Citizen Services - Thrive Program',
       'Citizen Services - Tele-Engagement',
       'Citizen Services - Weed & Seed'], dtype=object)
```

In [6]:
```
df_311calls["Type"].unique()
```

Out[6]:
```
array(['Missed Pickup 2_Piece Large Trash (Req_Serv)',
       'Housing Violations (Req_Serv)', 'Streetlights (Req_Serv)',
       'Police Issue (Req_Serv)', 'Sign Hazards (Req_Serv)',
       'Pick and Pay (Req_Serv)', 'Pot Hole (Req_Serv)',
       'Open311 Housing', 'Fallen Tree Blocking RoW (Req_Serv)',
       'Buffalo Traffic Violations (Req_Serv)',
       'Trash Ordinance Violation (Req_Serv)',
       'Signal Out or Flashing (Req_Serv)',
       'FOIL Records Police Dept (Req_Serv)',
       'Recycling Tote Replace (Req_Serv)', 'Rodents (Req_Serv)',
       'Water Issue (Req_Serv)', 'Street Snow Plowing (Req_Serv)',
       'Street Salting (Req_Serv)', 'Totes Deliver (Req_Serv)',
       'Garbage Missed Pick Up (Req_Serv)',
       'Leaves / Lawn Debris (Req_Serv)', 'Sweeper (Req_Serv)',
       'Totes Replace (Req_Serv)', 'Parking Issues (Req_Serv)',
       'Sign Maintenance (Req_Serv)',
       'FOIL Records Compensation and Benefits (Req_Serv)',
       'Sewer (Req_Serv)', 'FOIL Records EDPIS (Req_Serv)',
       'Tree Other (Req_Serv)', 'City Parks (Req_Serv)',
       'Totes Pickup (Req_Serv)', 'Bulk Trash (Req_Serv)',
       'Damaged Street Light Pole (Req_Serv)', 'Water Tested (Req_Serv)',
       'PVB Single Meter (Req_Serv)', 'Basement Flooding (Req_Serv)',
       'Recycling Missed Pick Up (Req_Serv)',
       'Totes Abandon Pickup (Req_Serv)', 'PW Vacant Lot (Req_Serv)',
       'Fire (Req_Serv)', 'Tree Trimming Request (Req_Serv)',
       'Cave In (Req_Serv)', 'Abandoned Vehicles (Req_Serv)',
       'Pest (Req_Serv)', 'Pavement Marking Lines (Req_Serv)',
       'Illegal Dumping Curb (Req_Serv)', 'BMHA Issue (Req_Serv)',
       'Recycling Tote Deliver (Req_Serv)', 'Totes Combo (Req_Serv)',
       'Dead Animal Removal (Req_Serv)',
       'QRT Illegal Dumping on Viaduct (Req_Serv)',
       'Quality of Life Issue (Req_Serv)', 'Paving (Req_Serv)',
       'Tree Removal (Req_SERV)', 'Animals (Req_Serv)',
       'Stump Removal (Req_Serv)', 'Excess Trash (Req_Serv)',
       'Sidewalks (Req_Serv)', 'Totes Audit (Req_Serv)',
       'Fire Hydrant Issue (Req_Serv)',
       'Damage from Street Worker (Req_Serv)',
       'Right of Way Issue (Req_Serv)', 'QRT Other Issue (Req_Serv)',
       'Street Snow Plowing Issue (Req_Serv)',
       'Illegal Dumping Street (Req_Serv)', 'Electronic Waste (Req_Serv)',
       'Street Flooding (Req_Serv)', 'Recycling Tote Combo (Req_Serv)',
       'FOIL Records Fire Dept (Req_Serv)',
       'Obscene Private Property (Req_Serv)',
       'Fallen Tree Inspection (Req_Serv)',
       'FOIL Records City Clerk (Req_Serv)',
       'Electrical Issue (Req_Serv)', 'User Fee (Req_Serv)',
       'Boarding request (Req_Serv)', 'Inrem Real Estate (Req_Serv)',
       'Curb - Metal Protruding (Req_Serv)',
       'Tree Planting Request (Req_Serv)',
       'Snow Removal Inspection (Req_Serv)', 'Open311 Graffiti',
       'Basketball Hoop in RoW (Req_Serv)', 'City Property (Req_Serv)',
       'Signal Other Issue (Req_Serv)',
       'Illegal Dumping PrivateProperty (Req_Serv)',
       'Recycling Tote Pickup (Req_Serv)',
       'PW Ongoing Construction (Req_Serv)',
       'Recycling - Escalated Questions (Req_Serv)', 'Curbs (Req_Serv)',
       'OSP Other Issue (Req_Serv)', 'Pools and Splashpads (Req_Serv)',
       'PW Neighborhood Traffic Calming (Req_Serv)', 'Pest_Private Trap',
       'PW Missing Manhole Cover (Req_Serv)',
       'Signal Timing Issue City (Req_Serv)',
```

```
'Neighborhood CleanUp (Req_Serv)',
'Damage from Snow Removal (Req_Serv)',
'Assessment Issue (Req_Serv)',
'Engineering Operational (Req_Serv)',
'FOIL Records Human Resources (Req_Serv)',
'PVB Pay Station (Req_Serv)',
'FOIL Records AdmFinance_Policy_UrbanAff (Req_Serv)',
'Other Hole in Road (Req_Serv)', 'Park Garbage Pickup (Req_Serv)',
'Fridge (Req_Serv)', 'Water_Billing_Meter (Req_Serv)',
'Recycling Tote Abandon Pickup (Req_Serv)',
'Olmsted Parks (Req_Serv)', 'Taxation Issue (Req_Serv)',
'Bike Rack (Req_Serv)', 'Fair Housing Issue (Req_Serv)',
'PVB Mobile App Issue', 'Rain Barrels (Req_Serv)',
'Graffiti Private Property (Req_Serv)',
'FOIL Records Planning Board (Req_Serv)',
'Obscene Parks City (Req_Serv)',
'FOIL Records Strategic Planning (Req_Serv)',
'FOIL Records Assessment & Taxation (Req_Serv)',
'Tree Trimming Quality Issue (Req_Serv)',
'Lead Paint Inspection (Req_Serv)', 'Signal Issue DoT (Req_Serv)',
'Stump Removal Quality Issue (Req_Serv)',
'CityHall_CityCourt Maintenance (Req_Serv)',
'Tree Removal Quality Issue (Req_serv)',
'Obscene Other (Req_Serv)', 'Illegal Dumping (Req_Serv)',
'Rental Registration (Req_Serv)',
'Great American Clean-Up (Req_Serv)',
'City Clerk Issue (Req_Serv)', 'Christmas Tree (Req_Serv)',
'Save Our Streets Program (Req_Serv)',
'FOIL Records Citizens Services (Req_Serv)',
'Chicken Coop (Req_Serv)', 'Other (Req_Serv)',
'Recreation Center (Req_Serv)', 'Waterfront Issues (Req_Serv)',
'FOIL Records Traffic Violations (Req_Serv)',
'PW Engineering (Req_Serv)', 'Private Property (Req_Serv)',
'Tree Removal Challenge (Req_Serv)', 'QRT Snow Removal (Req_Serv)',
'Building Maintenance (Req_Serv)',
'BFD Snow on Hydrant (Req_Serv)',
'FOIL Records Parking (Req_Serv)', 'Bridge Issue (Req_Serv)',
'Good Neighbor (Req_Serv)', 'City Park Tree Issue (Req_Serv)',
'BFD Fire Prevention (Req_Serv)',
'FOIL Records Law Dept (Req_Serv)',
'Other Adjudication Issue (Req_Serv)',
'Youth Bureau Issue (Req_Serv)',
'FOIL Records Real Estate (Req_Serv)',
'FOIL Records Adjudication (Req_Serv)',
'FOIL Records Audit Control (Req_Serv)', 'CBO (Req_Serv)',
'Vehicle Blocking Snow Plow (Req_Serv)',
'Snow Removal Bus Stop/Shelter (Req_Serv)',
'FOIL Records Division of Purchase (Req_Serv)',
'PVB Mobile App Ticket', 'PW Traffic (Req_Serv)',
'Totes Report of Removed Broken Tote (Req_Serv)',
'Obscene PW Traffic (Req_Serv)', 'Sneakers Hanging (Req_Serv)',
'OSP Illegal Dumping (Req_Serv)',
'FOIL Records Board of Ethics (Req_Serv)',
'FOIL Records Community Srvs & RecProg (Req_Serv)',
'Dog License Issue (Req_Serv)',
'RalphWilson Park Construction (Req_Serv)',
'FOIL Records Preservation Board (Req_Serv)',
'PVB School-Zone Camera Citations (Req_Serv)',
'Obscene PW Engineering (Req_Serv)',
'FOIL Records Civil Service (Req_Serv)',
```

```
       'Obscene City Property (Req_Serv)',
       'Obscene Parks Olmsted (Req_Serv)', 'Scrap_It Curbside (Req_Serv)',
       'FairHsg Other (Req_Serv)', 'Schools (Req_Serv)',
       'Special Event Totes (Req_Serv)', 'Water (Req_Serv)',
       'Recycling Bin Delivery (Req_Serv)',
       'Parking Meter Issue (Req_Serv)',
       'Compensation & Benefits (Req_Serv)', 'Escalated Question',
       'Parks City (Req_Serv)', 'Parks Usage Survey (Req_Serv)',
       'Open311 APP Issues (Req_Serv)',
       'Abandoned Vehicle Inspection (Req_Serv)',
       'Tree Planting Quality Issue (Req_Serv)',
       'Tree Removal per Inspector (Req_Serv)',
       'Telecommunications (Req_Serv)', 'Small Cell (Req_Serv)', 'Test2',
       'Test_before', 'Recycling Bin Pickup (Req_Serv)',
       'ReOpen Buffalo (Req_Serv)', 'Snow-Bank Removal (Req_Serv)',
       'Erie Basin Marina (Req_Serv)', 'NYS New Americans (Req_Serv)',
       'Snow Removal (Req_Serv)', 'Thrive Work Development (Req_Serv)',
       '2020 Reassessment',
       'Tele-Engagement Board of Block Clubs (Req_Serv)',
       'Security Deposit (Req_Serv)',
       'FOIL Records Office of the Mayor (Req_Serv)',
       'Tele-Engagement Police (Req_Serv)',
       'Tele-Engagement Strategic Planning (Req_Serv)',
       'Tele-Engagement BETC (Req_Serv)',
       'Tele-Engagement Citizen Services (Req_Serv)',
       'Parks Olmsted (Req_Serv)',
       'Tele-Engagement Community Services (Req_Serv)',
       'Obscene Schools (Req_Serv)', 'QRT Vacant Lot (Req_Serv)',
       'Ordinance Violation (Req_Serv)',
       'Pavement Markings Other (Req_Serv)',
       'Housing Violations (Prov_Info)', 'Weed & Seed Program (Req_Serv)',
       'Block Club CleanUp (Req_Serv)',
       'Neighborhood Watch Signs (Req_Serv)',
       'QRT Illegal Dumping on Vacant Lot (Req_Serv)', 'Pot Hol'],
      dtype=object)
```

In [7]:
```
df_311calls["Object Type"].unique()
```

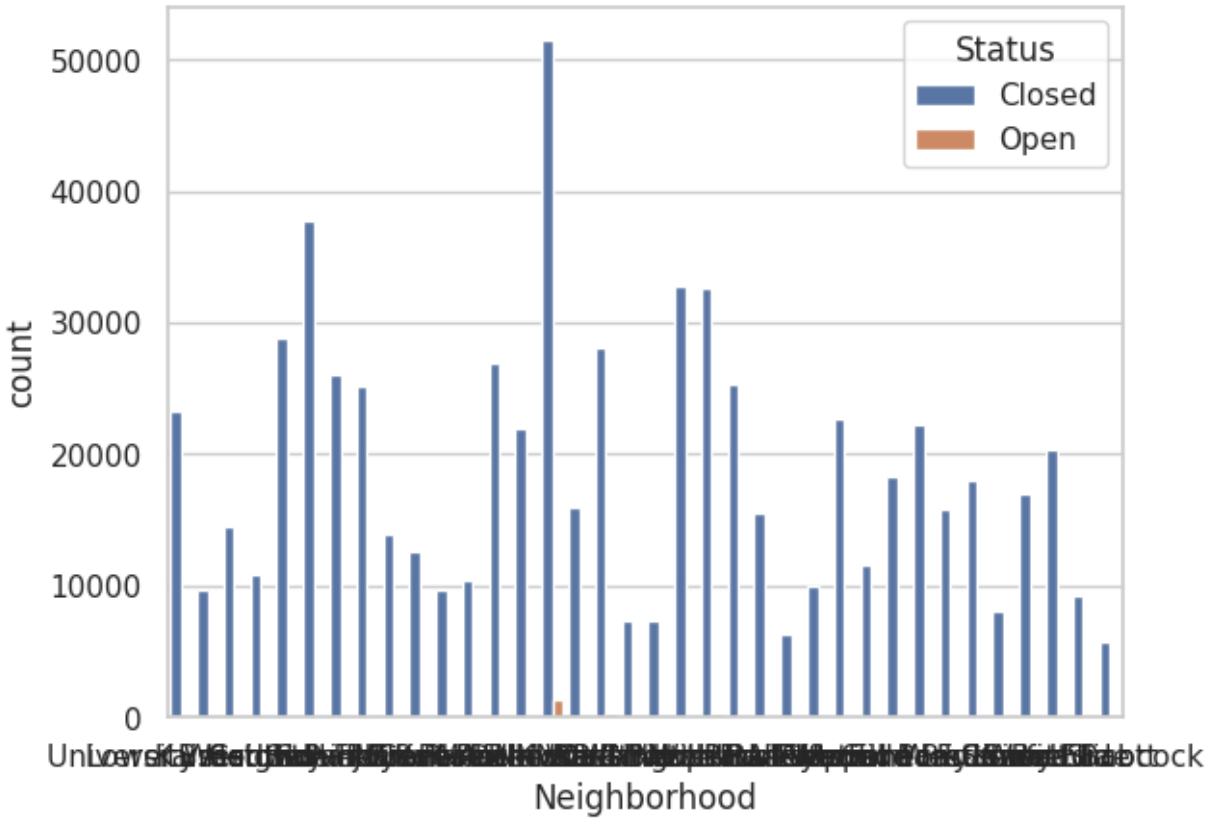
Out[7]:
```
array(['Property', 'Unknown', 'Street', 'Organisation', 'Individual', nan],
      dtype=object)
```

# How the Categorical Variables are related to Council District, Police District, and Neighborhood

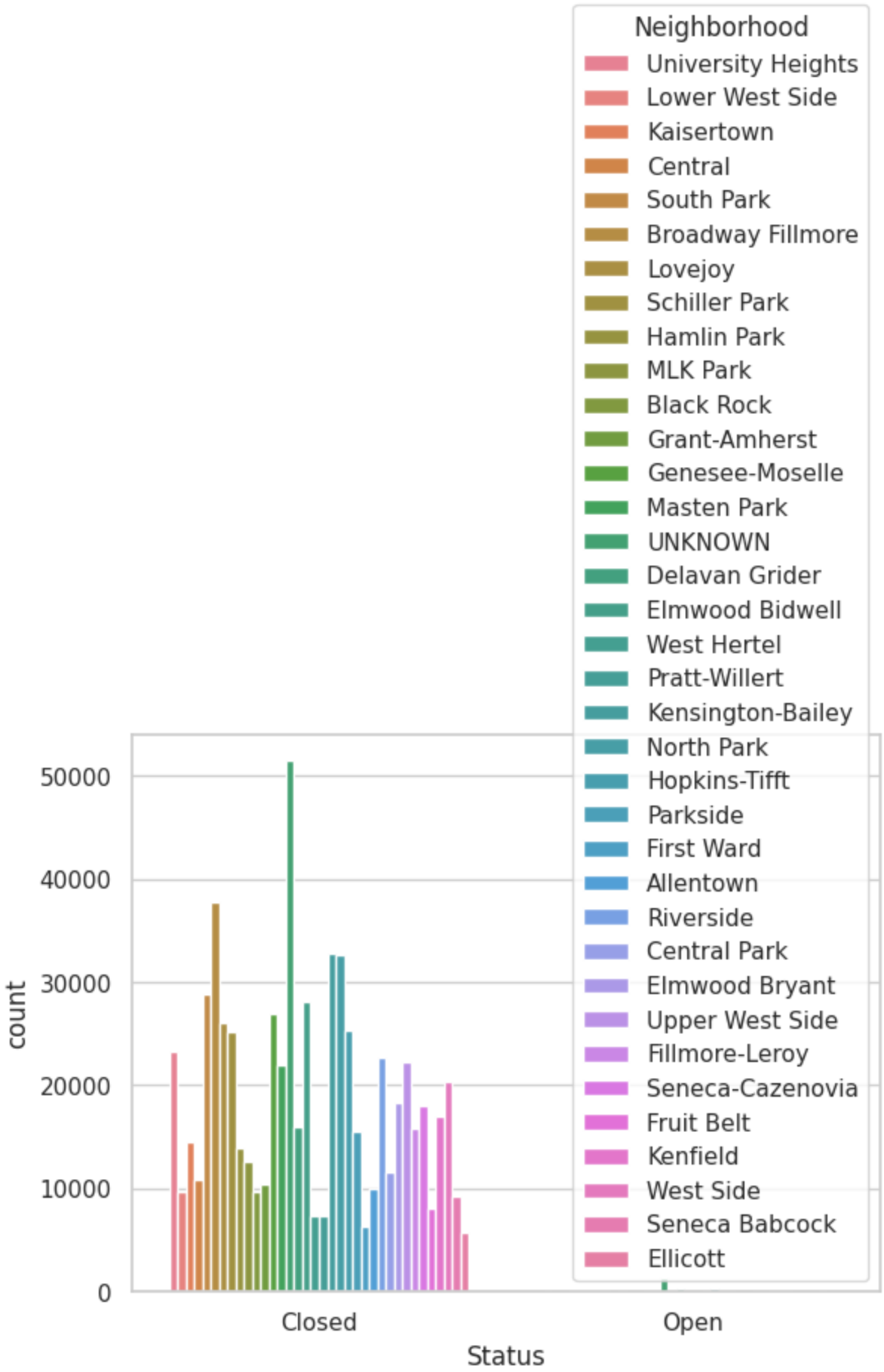
In [20]:
```
df_311calls["Neighborhood"].unique()
```

Out[20]:
```
array(['University Heights', 'Lower West Side', 'Kaisertown', 'Central',
       'South Park', 'Broadway Fillmore', 'Lovejoy', 'Schiller Park',
       'Hamlin Park', 'MLK Park', 'Black Rock', 'Grant-Amherst',
       'Genesee-Moselle', 'Masten Park', 'UNKNOWN', 'Delavan Grider',
       'Elmwood Bidwell', 'West Hertel', 'Pratt-Willert',
       'Kensington-Bailey', 'North Park', 'Hopkins-Tifft', 'Parkside',
       'First Ward', 'Allentown', 'Riverside', 'Central Park',
       'Elmwood Bryant', 'Upper West Side', 'Fillmore-Leroy',
       'Seneca-Cazenovia', 'Fruit Belt', 'Kenfield', 'West Side',
       'Seneca Babcock', 'Ellicott', nan], dtype=object)
```

In [15]:
```
ax = sns.countplot(x="Neighborhood", hue="Status", data=df_311calls)
```




In [14]:
```
ax = sns.countplot(x="Status", hue="Neighborhood", data=df_311calls)
```

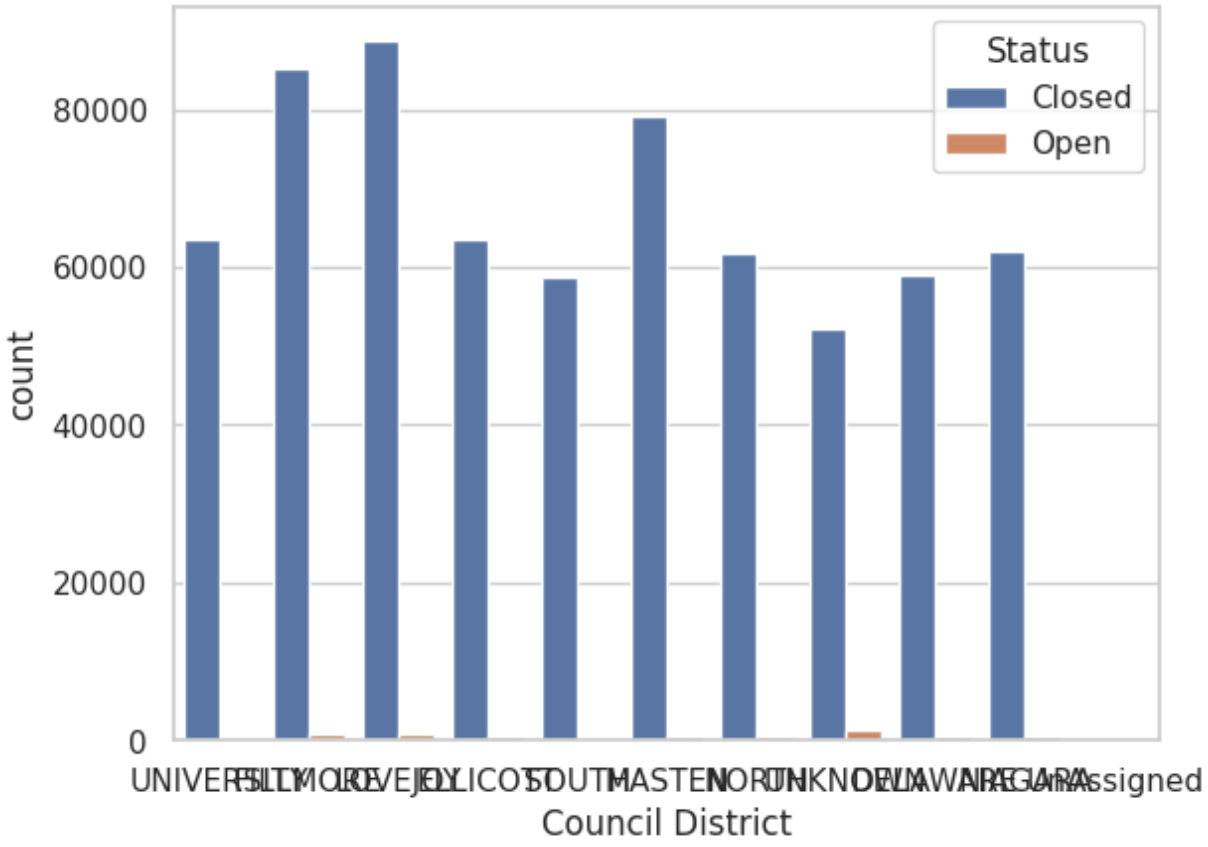

Neighborhood
University Heights
Lower West Side
Kaisertown
Central
South Park
Broadway Fillmore
Lovejoy
Schiller Park
Hamlin Park
MLK Park
Black Rock
Grant-Amherst
Genesee-Moselle
Masten Park
UNKNOWN
Delavan Grider
Elmwood Bidwell
West Hertel
Pratt-Willert
Kensington-Bailey
North Park
Hopkins-Tifft
Parkside
First Ward
Allentown
Riverside
Central Park
Elmwood Bryant
Upper West Side
Fillmore-Leroy
Seneca-Cazenovia
Fruit Belt
Kenfield
West Side
Seneca Babcock
Ellicott
50000
40000
30000
20000
10000
0
count
Closed
Open
Status

The different neighborhoods definitely vary in number of 311 reports. Excluding the unknown entries, Broadway Fillmore has the highest number of reports, while First Ward has the lowest.

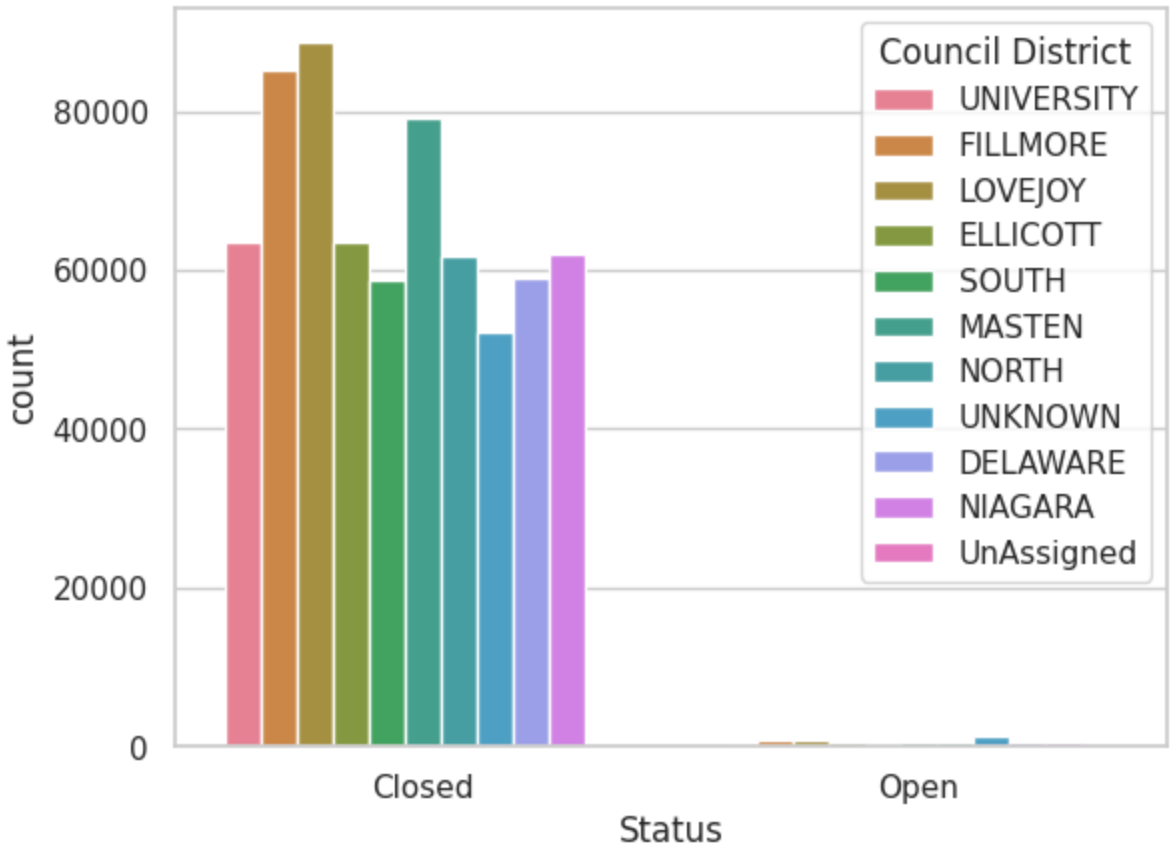
In [16]:
```
df_311calls["Council District"].unique()
```

Out[16]:
```
array(['UNIVERSITY', 'FILLMORE', 'LOVEJOY', 'ELLICOTT', 'SOUTH', 'MASTEN',
       'NORTH', 'UNKNOWN', 'DELAWARE', 'NIAGARA', 'UnAssigned', nan],
      dtype=object)
```

In [18]:
```
ax = sns.countplot(x="Council District", hue="Status", data=df_311calls)
```




In [19]:
```
ax = sns.countplot(x="Status", hue="Council District", data=df_311calls)
```

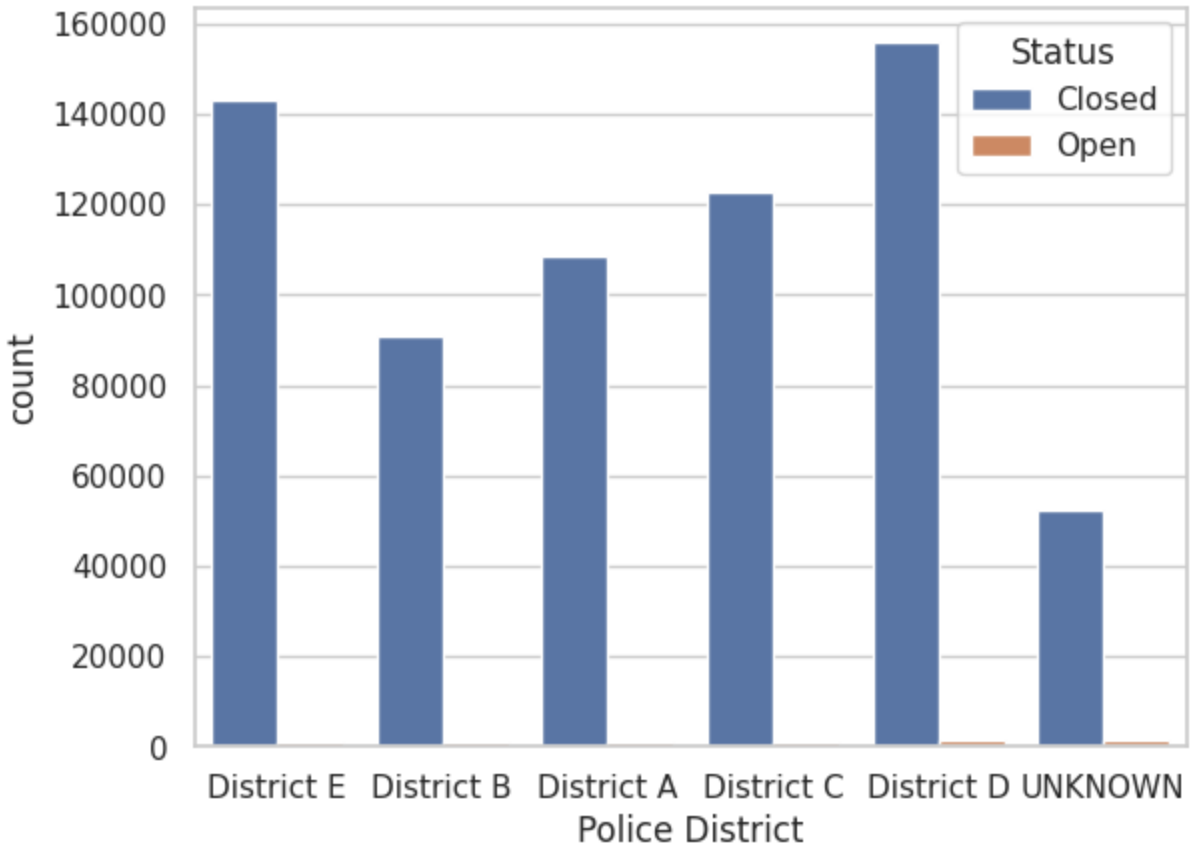



The different Council Districts vary in number of 311 reports, but not as drastically as neighborhoods. Excluding the unknown entries, Lovejoy and Fillmore have the highest number of reports, while South has the lowest.

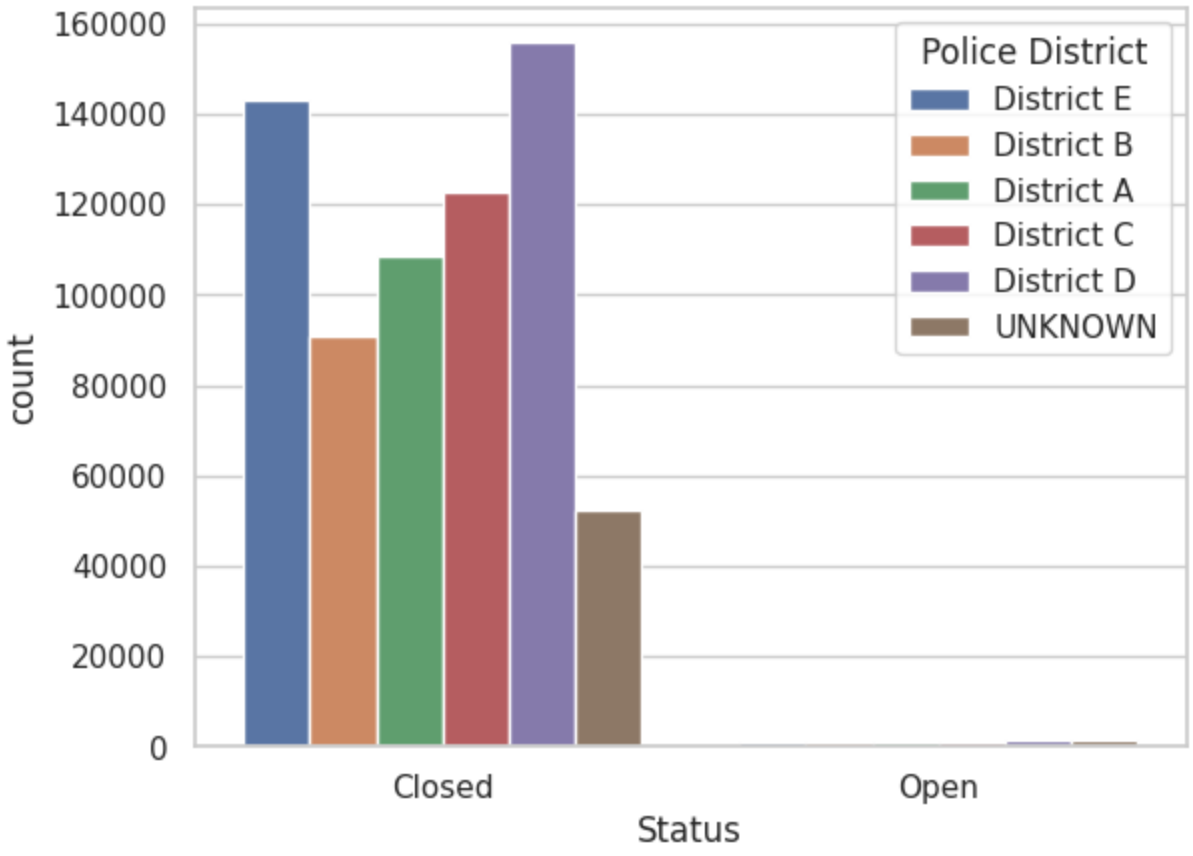
In [17]:
```
df_311calls["Police District"].unique()
```

Out[17]:
```
array(['District E', 'District B', 'District A', 'District C',
       'District D', 'UNKNOWN', nan], dtype=object)
```

In [21]:
```
ax = sns.countplot(x="Police District", hue="Status", data=df_311calls)
```




In [22]:
```
ax = sns.countplot(x="Status", hue="Police District", data=df_311calls)
```

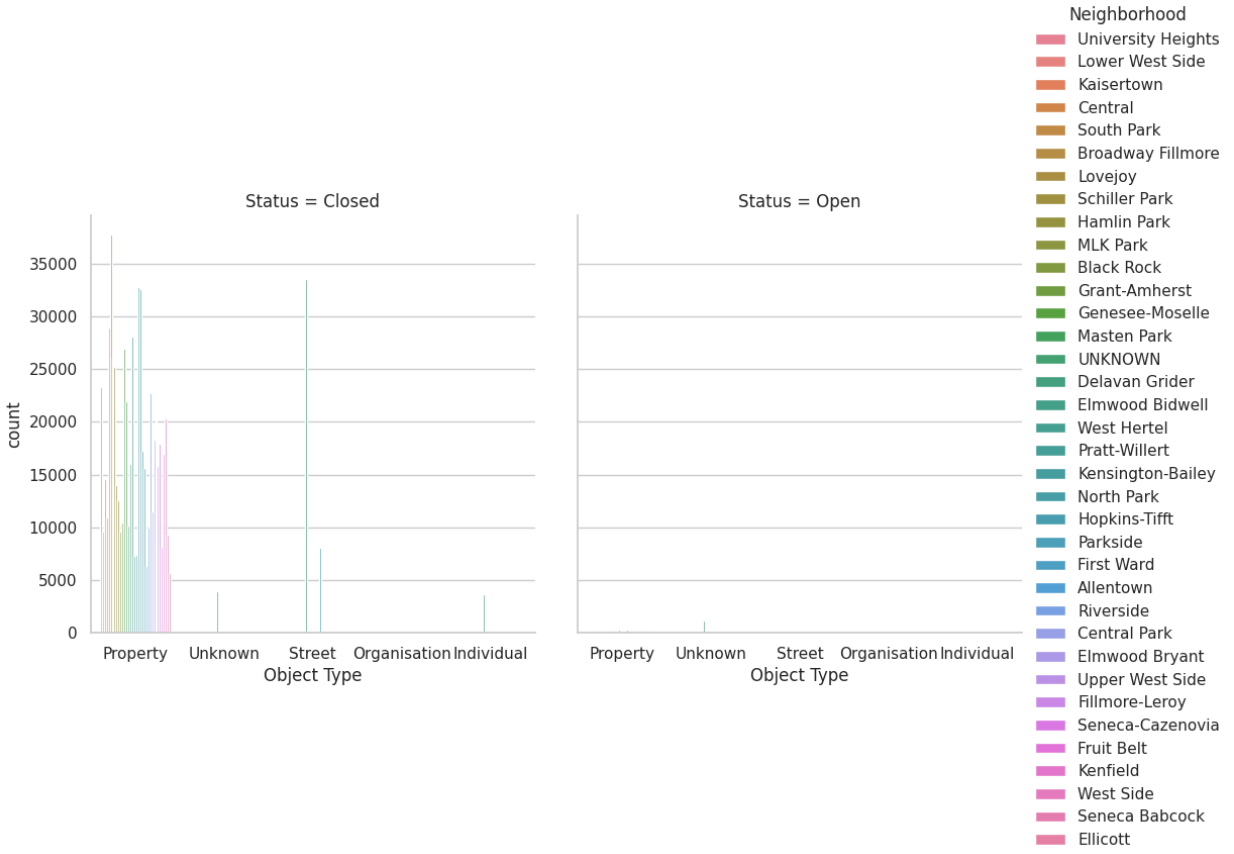


The different Police Districts slightly vary in number of 311 reports. Excluding the unknown entries, District D and E have the highest number of reports, while District B has the lowest.

Status is not a good categorical variable to use in this dataset because the overwhelming majority of reports are closed. It is difficult to even notice any open reports.

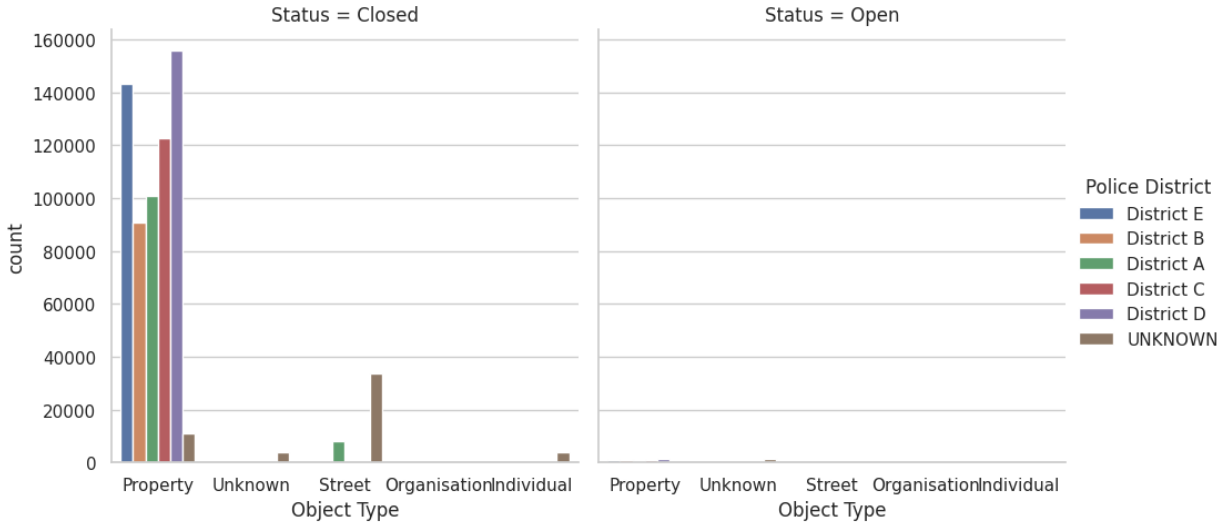
In [24]:
```
g = sns.catplot(x="Object Type", hue="Neighborhood", col="Status",
                data=df_311calls, kind="count");
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:123: UserWarning: Tight l
ayout not applied. The bottom and top margins cannot be made large enough to accommod
ate all axes decorations.
  self._figure.tight_layout(*args, **kwargs)
```




In [25]:
```
g = sns.catplot(x="Object Type", hue="Council District", col="Status",
                data=df_311calls, kind="count");
```


In [26]:
```
g = sns.catplot(x="Object Type", hue="Police District", col="Status",
                data=df_311calls, kind="count");
```



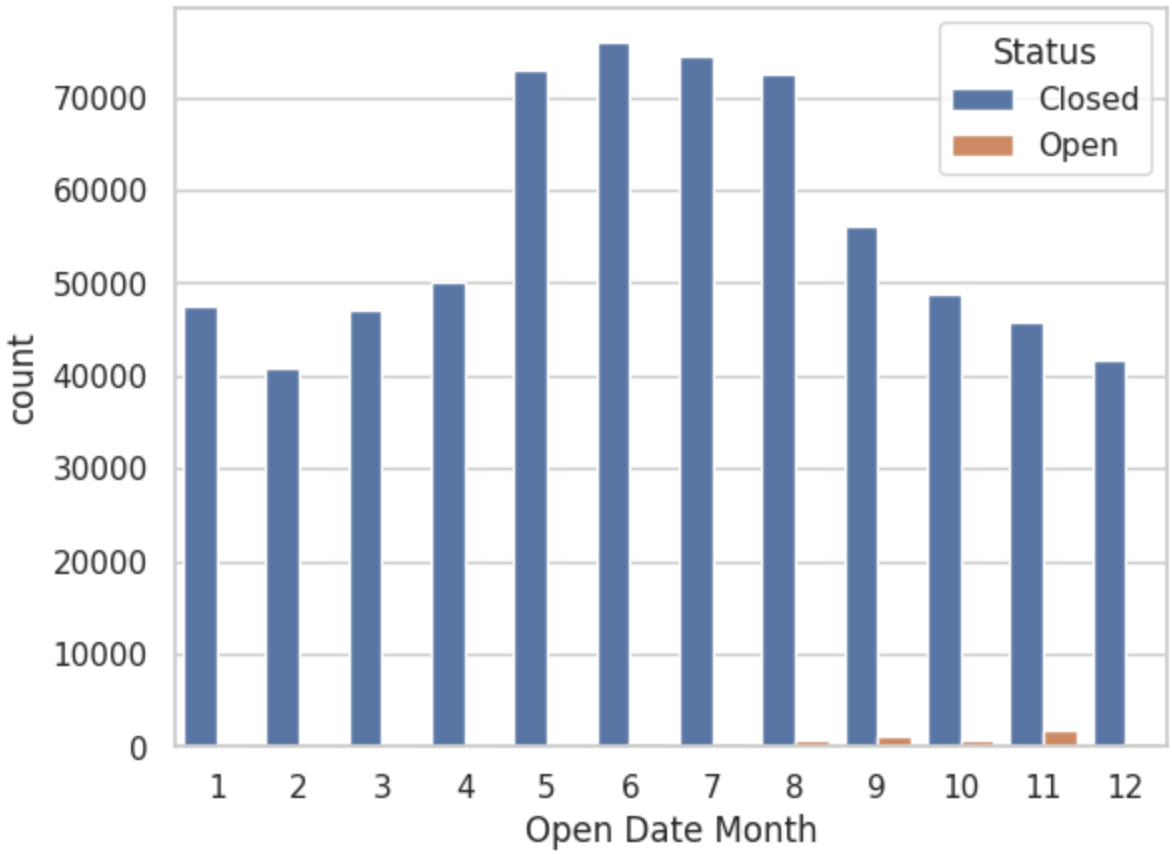

The overwhelming majority of 311 reports are done on property. The majority of the reports where the location is unknown defaults to Street.

# How the Categorical Variables are related to Months/Years

In [27]:
```
df_311calls['Open Date'] = pd.to_datetime(df_311calls['Open Date'])
```

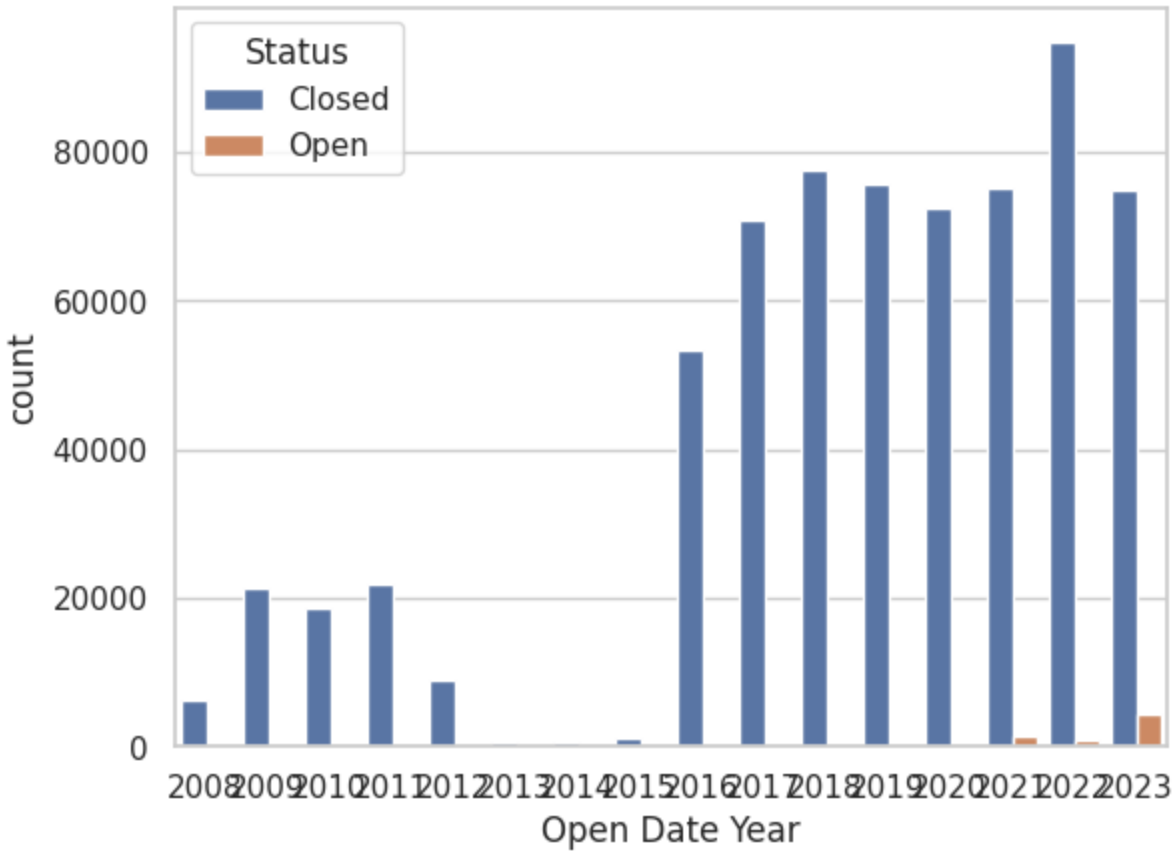
In [28]:
```
df_311calls['Open Date Month'] = df_311calls['Open Date'].dt.month
df_311calls['Open Date Year'] = df_311calls['Open Date'].dt.year
```

In [29]:
```
ax = sns.countplot(x="Open Date Month", hue="Status", data=df_311calls)
```

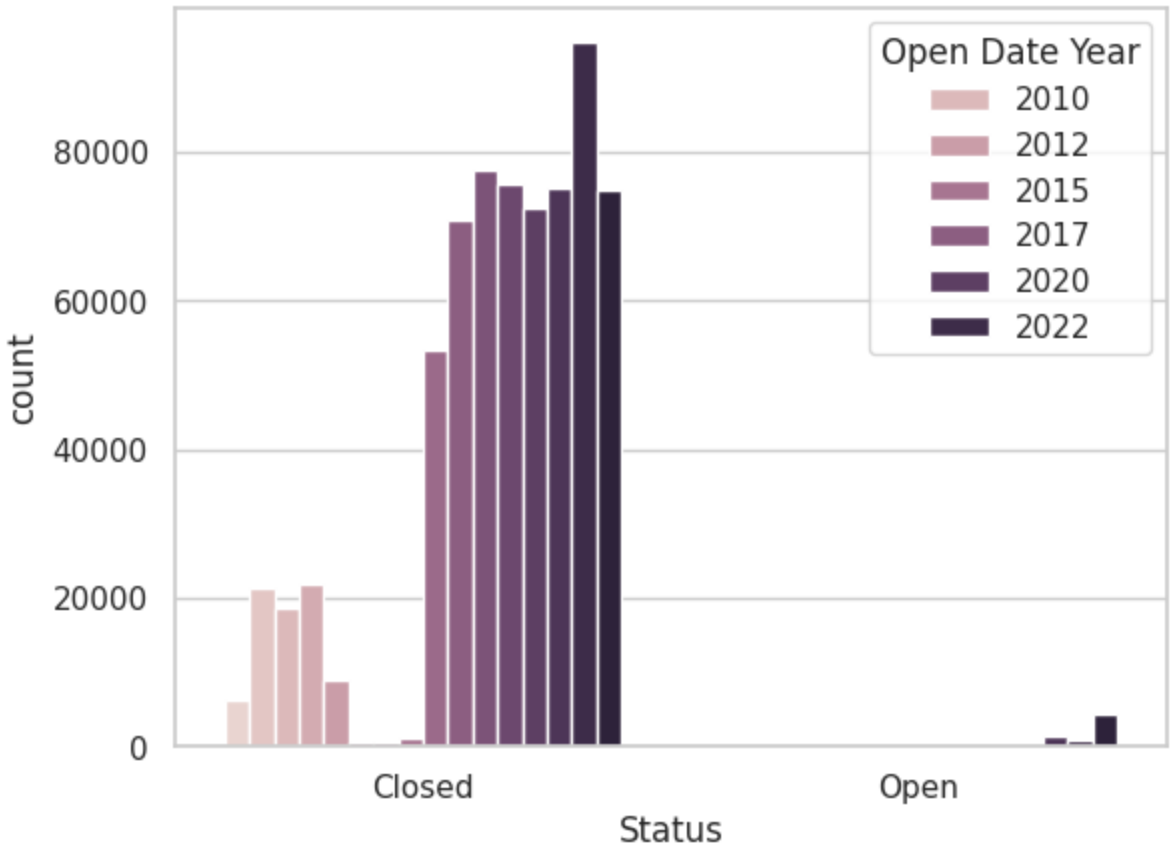



The majority of 311 reports occur over the Summer. This makes sense considering those months mean more general outdoor activity.

In [30]:
```
ax = sns.countplot(x="Open Date Year", hue="Status", data=df_311calls)
```




In [31]:
```
ax = sns.countplot(x="Status", hue="Open Date Year", data=df_311calls)
```



The vast majority of 311 reports occurred within the last 8 years. The number of reports from 2008 to 2015 are considerably less than the number of reports from 2016 to 2023.