

CIS 5690

Laia Gonzalez Call

✓ CRIME RATE PREDICTION - LIVE DATA

✓ 1. Obtaining real-time Data

Real-time data streaming is a method of continuously gathering real-time data from multiple external data sources in the form of data streams.

- Gathered data from the official San Francisco Police Department's (SFPD) Website, that is updated automatically daily by 10:00 Pacific.

<https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783>

```
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyspark
  Downloading pyspark-3.3.2.tar.gz (281.4 MB)
    281.4/281.4 MB 2.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
    199.7/199.7 KB 9.1 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.3.2-py2.py3-none-any.whl size=281824025 sha256=625a451b89c606faf6c130d6858645ffeccbd187e
  Stored in directory: /root/.cache/pip/wheels/6c/e3/9b/0525ce8a69478916513509d43693511463c6468db0de237c86
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.5 pyspark-3.3.2
```

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('Load Datafile').getOrCreate()
```

```
url = "https://data.sfgov.org/api/views/wg3w-h783/rows.csv"
```

```
from pyspark import SparkFiles
spark.sparkContext.addFile(url)
```

```
df = spark.read.csv(SparkFiles.get("rows.csv"), header=True, inferSchema= True)
```

```
df
```

```
DataFrame[Incident Datetime: string, Incident Date: string, Incident Time: timestamp, Incident Year: int, Incident Day of Week: string,
Report Datetime: string, Row ID: bigint, Incident ID: int, Incident Number: int, CAD Number: int, Report Type Code: string, Report Type
Description: string, Filed Online: boolean, Incident Code: int, Incident Category: string, Incident Subcategory: string, Incident
Description: string, Resolution: string, Intersection: string, CNN: double, Police District: string, Analysis Neighborhood: string,
Supervisor District: int, Supervisor District 2012: int, Latitude: double, Longitude: double, Point: string, Neighborhoods: int, ESNCA
- Boundary File: int, Central Market/Tenderloin Boundary Polygon - Updated: int, Civic Center Harm Reduction Project Boundary: int,
HSOC Zones as of 2018-06-05: int, Invest In Neighborhoods (IIN) Areas: string, Current Supervisor Districts: int, Current Police
Districts: int]
```

```
df.repartition(1).write.format("csv").mode('overwrite').option("header", "true").save("mydata.csv")
```

✓ 2. Load Data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Creation Folium Maps
import folium
from folium.plugins import HeatMap, MarkerCluster, HeatMapWithTime
```

```
from sklearn.pipeline import Pipeline
import sklearn
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split
from datetime import date
from pandas.tseries.holiday import USFederalHolidayCalendar
import holidays
```

```
df = pd.read_csv('/content/mydata.csv/part-00000-08508669-e5ae-43fa-9a30-b96e93e3bcea-c000.csv', error_bad_lines = False)
```

```
<ipython-input-8-7571edcff695>:1: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future versio
```

```
df = pd.read_csv('/content/mydata.csv/part-00000-08508669-e5ae-43fa-9a30-b96e93e3bcea-c000.csv', error_bad_lines = False)
b'Skipping line 9683: expected 35 fields, saw 37\n'
b'Skipping line 23425: expected 35 fields, saw 37\nSkipping line 25739: expected 35 fields, saw 37\n'
b'Skipping line 37034: expected 35 fields, saw 37\nSkipping line 37693: expected 35 fields, saw 37\nSkipping line 40517: expected 35 fie
b'Skipping line 53909: expected 35 fields, saw 37\nSkipping line 62141: expected 35 fields, saw 37\n'
b'Skipping line 106436: expected 35 fields, saw 37\nSkipping line 106827: expected 35 fields, saw 37\n'
b'Skipping line 147947: expected 35 fields, saw 37\n'
b'Skipping line 266178: expected 35 fields, saw 37\nSkipping line 270261: expected 35 fields, saw 37\nSkipping line 277210: expected 35
b'Skipping line 319087: expected 35 fields, saw 37\nSkipping line 326781: expected 35 fields, saw 37\n'
b'Skipping line 357089: expected 35 fields, saw 37\n'
b'Skipping line 378165: expected 35 fields, saw 37\n'
b'Skipping line 406313: expected 35 fields, saw 37\n'
b'Skipping line 429229: expected 35 fields, saw 37\n'
b'Skipping line 447489: expected 35 fields, saw 37\nSkipping line 457134: expected 35 fields, saw 37\n'
b'Skipping line 502113: expected 35 fields, saw 37\n'
b'Skipping line 525456: expected 35 fields, saw 37\nSkipping line 533503: expected 35 fields, saw 37\nSkipping line 537233: expected 35
b'Skipping line 546964: expected 35 fields, saw 37\n'
b'Skipping line 577497: expected 35 fields, saw 37\nSkipping line 588463: expected 35 fields, saw 37\n'
b'Skipping line 597980: expected 35 fields, saw 37\n'
b'Skipping line 608534: expected 35 fields, saw 37\nSkipping line 611060: expected 35 fields, saw 37\n'
b'Skipping line 627373: expected 35 fields, saw 37\nSkipping line 628518: expected 35 fields, saw 37\nSkipping line 629981: expected 35
b'Skipping line 659808: expected 35 fields, saw 37\n'
b'Skipping line 677695: expected 35 fields, saw 37\n'
b'Skipping line 704674: expected 35 fields, saw 37\n'
```

3. Checking Column Names and Dropping Unnecessary Data

```
df.shape
```

```
(705203, 35)
```

```
df.tail(3)
```

	Incident Datetime	Incident Date	Incident Time	Incident Year	Incident Day of Week	Report Datetime	Row ID	Incident ID	Incident Number	CAD Number	...	Longi
705200	2023/03/12 06:00:00 PM	2023/03/12	2023-03-15T18:00:00.000Z	2023	Sunday	2023/03/12 11:21:00 PM	125339807021	1253398	230176530	230713067.0	...	-122.43
705201	2023/03/07 06:30:00 PM	2023/03/07	2023-03-15T18:30:00.000Z	2023	Tuesday	2023/03/08 09:36:00 AM	125345906374	1253459	236045448	NaN	...	-122.41
705202	2023/03/12 04:45:00 PM	2023/03/12	2023-03-15T16:45:00.000Z	2023	Sunday	2023/03/13 01:19:00 AM	125342007021	1253420	230176734	230720094.0	...	-122.43

3 rows x 35 columns

```
# iterating the columns
for col in df.columns:
    print(col)
```

```
Incident Datetime
Incident Date
Incident Time
```

```

Incident Year
Incident Day of Week
Report Datetime
Row ID
Incident ID
Incident Number
CAD Number
Report Type Code
Report Type Description
Filed Online
Incident Code
Incident Category
Incident Subcategory
Incident Description
Resolution
Intersection
CNN
Police District
Analysis Neighborhood
Supervisor District
Supervisor District 2012
Latitude
Longitude
Point
Neighborhoods
ESNCAG - Boundary File
Central Market/Tenderloin Boundary Polygon - Updated
Civic Center Harm Reduction Project Boundary
HSOC Zones as of 2018-06-05
Invest In Neighborhoods (IIN) Areas
Current Supervisor Districts
Current Police Districts

```

```
# Remove columns
```

```
df = df.drop(columns=['Incident Date', 'Incident Time', 'Incident Year', 'Report Datetime', 'Row ID', 'Incident ID', 'Incident Number',
                     'CAD Number', 'Report Type Code', 'Report Type Description', 'Filed Online', 'Incident Code', 'Incident Subcategory', 'CNN',
                     'Analysis Neighborhood', 'Supervisor District', 'Supervisor District 2012', 'Point', 'Neighborhoods', 'ESNCAG - Boundary Fi
                     'Central Market/Tenderloin Boundary Polygon - Updated', 'Civic Center Harm Reduction Project Boundary',
                     'HSOC Zones as of 2018-06-05', 'Invest In Neighborhoods (IIN) Areas', 'Current Supervisor Districts', 'Current Police Dis
```

```
# iterating the columns
```

```
for col in df.columns:
    print(col)
```

```

↳ Incident Datetime
Incident Day of Week
Incident Category
Incident Description
Resolution
Intersection
Police District
Latitude
Longitude

```

```
df.isnull().sum()
```

```

↳ Incident Datetime      0
Incident Day of Week     0
Incident Category       585
Incident Description     0
Resolution              0
Intersection            37399
Police District         0
Latitude                37399
Longitude               37399
dtype: int64

```

```
df = df.dropna(subset=['Incident Category', 'Latitude', 'Longitude'])
```

```
df.isnull().sum()
```

```

↳ Incident Datetime      0
Incident Day of Week     0
Incident Category       0
Incident Description     0
Resolution              0
Intersection            0
Police District         0
Latitude                0

```

```
Longitude
dtype: int64      0
```

```
df.shape
```

```
(667232, 9)
```

```
df.sort_values(by=['Incident Datetime'])
```

↗

	Incident Datetime	Incident Day of Week	Incident Category	Incident Description	Resolution	Intersection	Police District	Latitude	Longitude
274798	2018/01/01 01:00:00 AM	Monday	Larceny Theft	Theft, Other Property, >\$950	Open or Active	NEW MONTGOMERY ST \ MARKET ST \ MONTGOMERY ST	Central	37.788721	-122.402066
341200	2018/01/01 01:00:00 AM	Monday	Larceny Theft	Theft, From Locked Vehicle, >\$950	Open or Active	BADEN ST \ MANGELS AVE	Ingleside	37.733111	-122.439752
365083	2018/01/01 01:00:00 AM	Monday	Assault	Battery	Cite or Arrest Adult	ELWOOD ST \ MASON ST	Tenderloin	37.786659	-122.409775
244970	2018/01/01 01:00:00 AM	Monday	Other Miscellaneous	False Report of Emergency	Open or Active	08TH ST \ MISSION ST	Southern	37.777457	-122.413158
239646	2018/01/01 01:00:00 AM	Monday	Lost Property	Lost Property	Open or Active	VIA BUFANO \ COLUMBUS AVE \ FILBERT ST	Central	37.801125	-122.411376
...	...	...	...	...	...	...	...	...	...
705121	2023/03/13 12:30:00 AM	Monday	Non-Criminal	Mental Health Detention	Open or Active	SUNSET BLVD \ YORBA ST	Taraval	37.735023	-122.494015

4. Split Data into Train and Test

4.1. Split the data into train and test set and save it in two differents files:

- train.csv
- test.csv

```
#split the data into train and test set
train,test = train_test_split(df, test_size=0.20, random_state=10)
#save the data
train.to_csv('train.csv',index=False)
test.to_csv('test.csv',index=False)
```

4.2 Train Set

```
train.shape
```

```
(533785, 9)
```

```
#total crimes in each category
train['Incident Category'].value_counts()
```

↗

Larceny Theft	156547
Other Miscellaneous	39108
Malicious Mischief	36507
Assault	34128
Non-Criminal	32911
Burglary	31419
Motor Vehicle Theft	28570
Fraud	17686
Warrant	16219
Recovered Vehicle	15925
Drug Offense	13732
Lost Property	13104
Robbery	12512
Missing Person	12218
Suspicious Occ	11553
Disorderly Conduct	9202
Offences Against The Family And Children	7830
Miscellaneous Investigation	5803
Traffic Violation Arrest	5471

Other	4594
Other Offenses	4056
Weapons Offense	3851
Weapons Carrying Etc	3054
Stolen Property	2794
Case Closure	2220
Forgery And Counterfeiting	1904
Arson	1616
Courtesy Report	1609
Traffic Collision	1297
Vandalism	1210
Fire Report	817
Embezzlement	740
Prostitution	722
Civil Sidewalks	711
Sex Offense	583
Vehicle Impounded	359
Suicide	242
Vehicle Misplaced	219
Drug Violation	183
Rape	135
Human Trafficking (A), Commercial Sex Acts	98
Suspicious	79
Homicide	74
Liquor Laws	62
Motor Vehicle Theft?	57
Gambling	21
Human Trafficking, Commercial Sex Acts	17
Weapons Offence	15
Human Trafficking (B), Involuntary Servitude	1

Name: Incident Category, dtype: int64

```
train.drop(train.index[train['Incident Category'] == 'Human Trafficking (B), Involuntary Servitude'], inplace = True)
```

```
#total crimes in each category
train['Incident Category'].value_counts()
```



Larceny Theft	156547
Other Miscellaneous	39108
Malicious Mischief	36507
Assault	34128
Non-Criminal	32911
Burglary	31419
Motor Vehicle Theft	28570
Fraud	17686
Warrant	16219
Recovered Vehicle	15925
Drug Offense	13732
Lost Property	13104
Robbery	12512
Missing Person	12218
Suspicious Occ	11553
Disorderly Conduct	9202
Offences Against The Family And Children	7830
Miscellaneous Investigation	5803
Traffic Violation Arrest	5471
Other	4594
Other Offenses	4056
Weapons Offense	3851
Weapons Carrying Etc	3054
Stolen Property	2794
Case Closure	2220
Forgery And Counterfeiting	1904
Arson	1616
Courtesy Report	1609
Traffic Collision	1297
Vandalism	1210
Fire Report	817
Embezzlement	740
Prostitution	722
Civil Sidewalks	711
Sex Offense	583
Vehicle Impounded	359
Suicide	242
Vehicle Misplaced	219
Drug Violation	183
Rape	135
Human Trafficking (A), Commercial Sex Acts	98
Suspicious	79
Homicide	74
Liquor Laws	62
Motor Vehicle Theft?	57

```

Gambling                21
Human Trafficking, Commercial Sex Acts  17
Weapons Offence         15
Name: Incident Category, dtype: int64

```

```
train['Police District'].unique()
```

```

array(['Bayview', 'Mission', 'Park', 'Southern', 'Ingleside', 'Central',
      'Out of SF', 'Taraval', 'Tenderloin', 'Northern', 'Richmond'],
      dtype=object)

```

```
train['Resolution'].unique()
```

```

array(['Open or Active', 'Cite or Arrest Adult', 'Unfounded',
      'Exceptional Adult'], dtype=object)

```

```
train['Incident Day of Week'].unique()
```

```

array(['Saturday', 'Sunday', 'Monday', 'Wednesday', 'Friday', 'Thursday',
      'Tuesday'], dtype=object)

```

4.2 Test Set

```
test.shape
```

```
(133447, 9)
```

```
#total crimes in each category
```

```
train['Resolution'].value_counts()
```

```

Open or Active          422402
Cite or Arrest Adult    106972
Unfounded               2969
Exceptional Adult       1441
Name: Resolution, dtype: int64

```

```
test.describe()
```

```

      Latitude      Longitude
count  133447.000000  133447.000000
mean     37.769209    -122.424061
std       0.024490     0.026441
min       37.707988    -122.511295
25%       37.754707    -122.434602
50%       37.775873    -122.417783
75%       37.785829    -122.407337
max       37.829991    -122.363743

```

5. Visualization

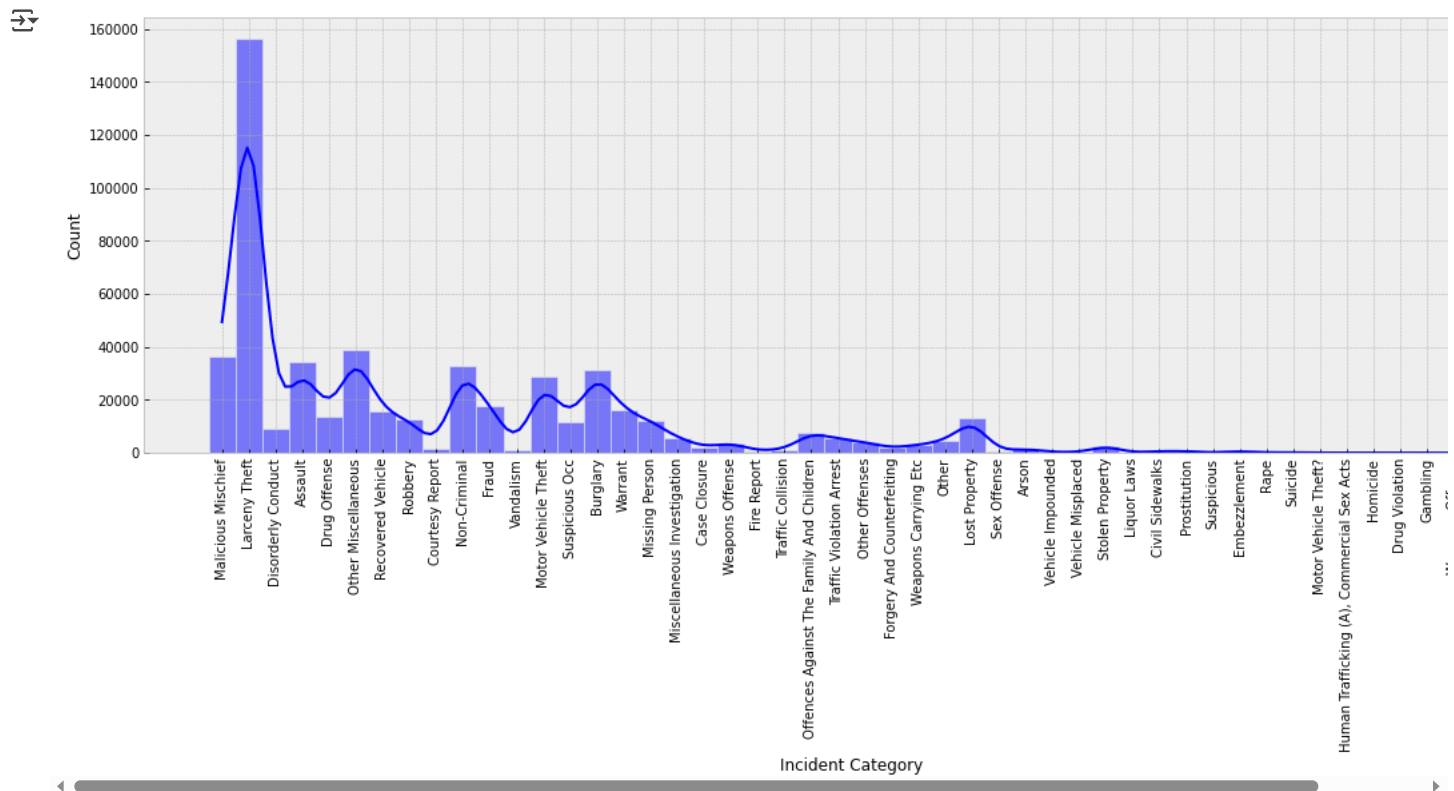
```

def visualization_crime(value, tight=False):
    with plt.style.context('bmh'):
        fig, ax = plt.subplots(1, 1, figsize=(19, 6))
        sns.histplot(x=value, data=train, kde=True, ax=ax)
        ax.tick_params(axis='x', rotation=90)
        if tight == True:
            fig.tight_layout()

```

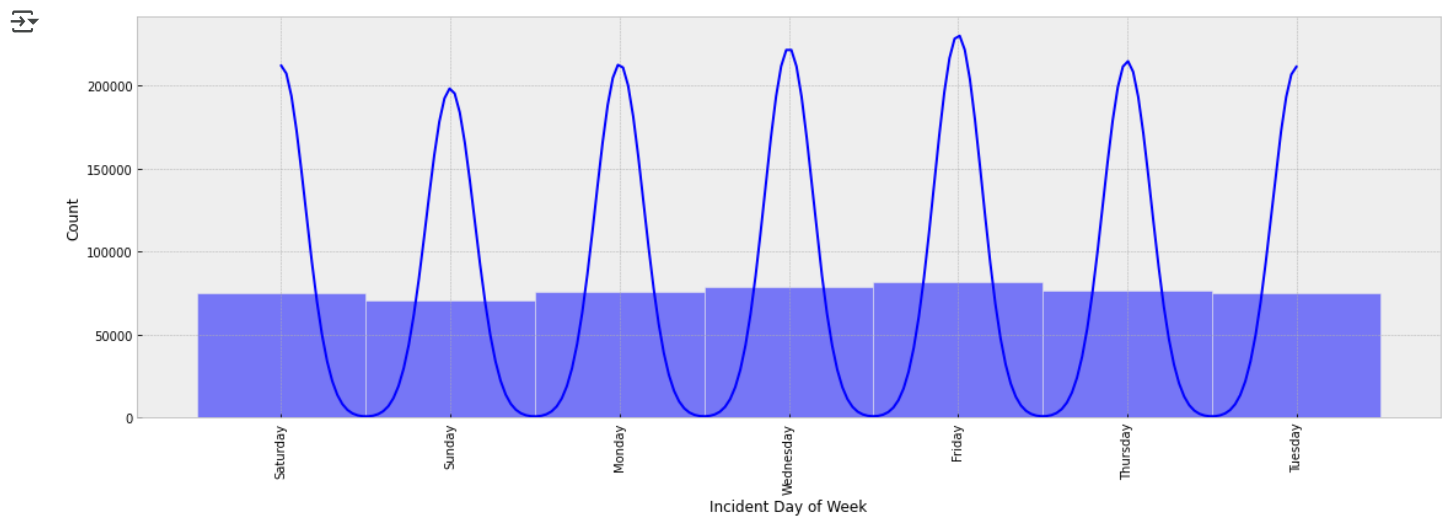
5.1. Incident Category Visualization

```
visualization_crime('Incident Category', tight=False)
```



5.2. Incident Day of the Week Visualization

```
visualization_crime('Incident Day of Week', tight=False)
plt.show()
```

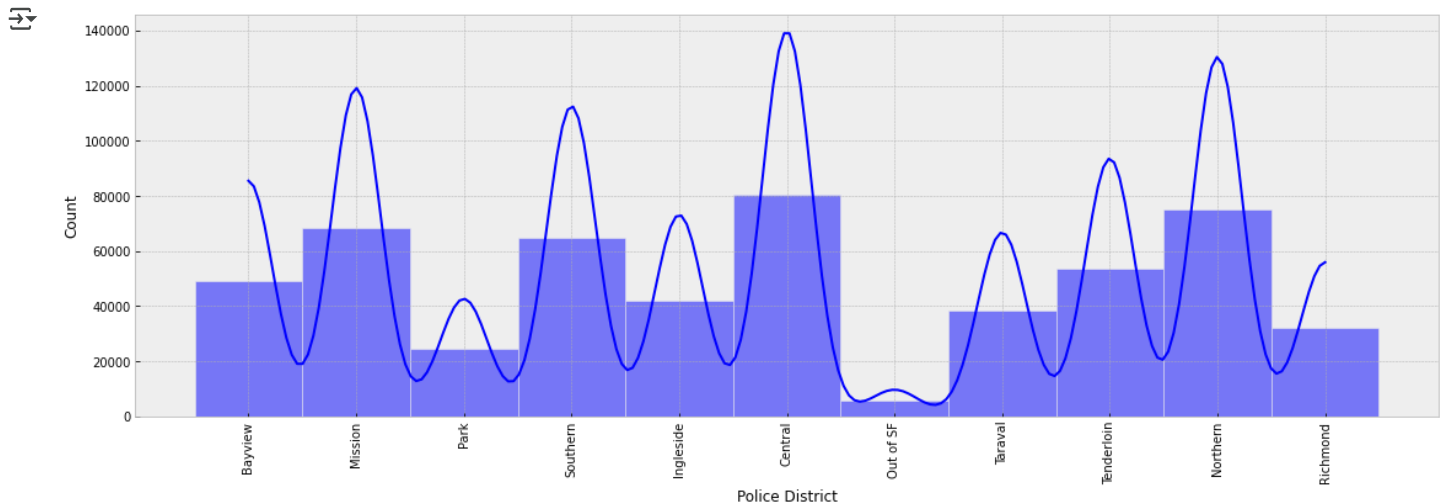


```
#total crimes for each day of the week
train['Incident Day of Week'].value_counts()
```

```
Friday      81893
Wednesday   79099
Thursday    76300
Monday      75660
Saturday    75307
Tuesday     75075
Sunday      70450
Name: Incident Day of Week, dtype: int64
```

5.3. Incident Police District Visualization

```
visualization_crime('Police District')
plt.show()
```



✓ Latitude and longitude scatter chart

- As we have the Latitude and Longitude of the Crimes Locations
- In this way, the shape of the scatter plot displays the shape of the city. This will become visually clearer as the number of data increases
- The following scatter plot shows the location of the crimes.

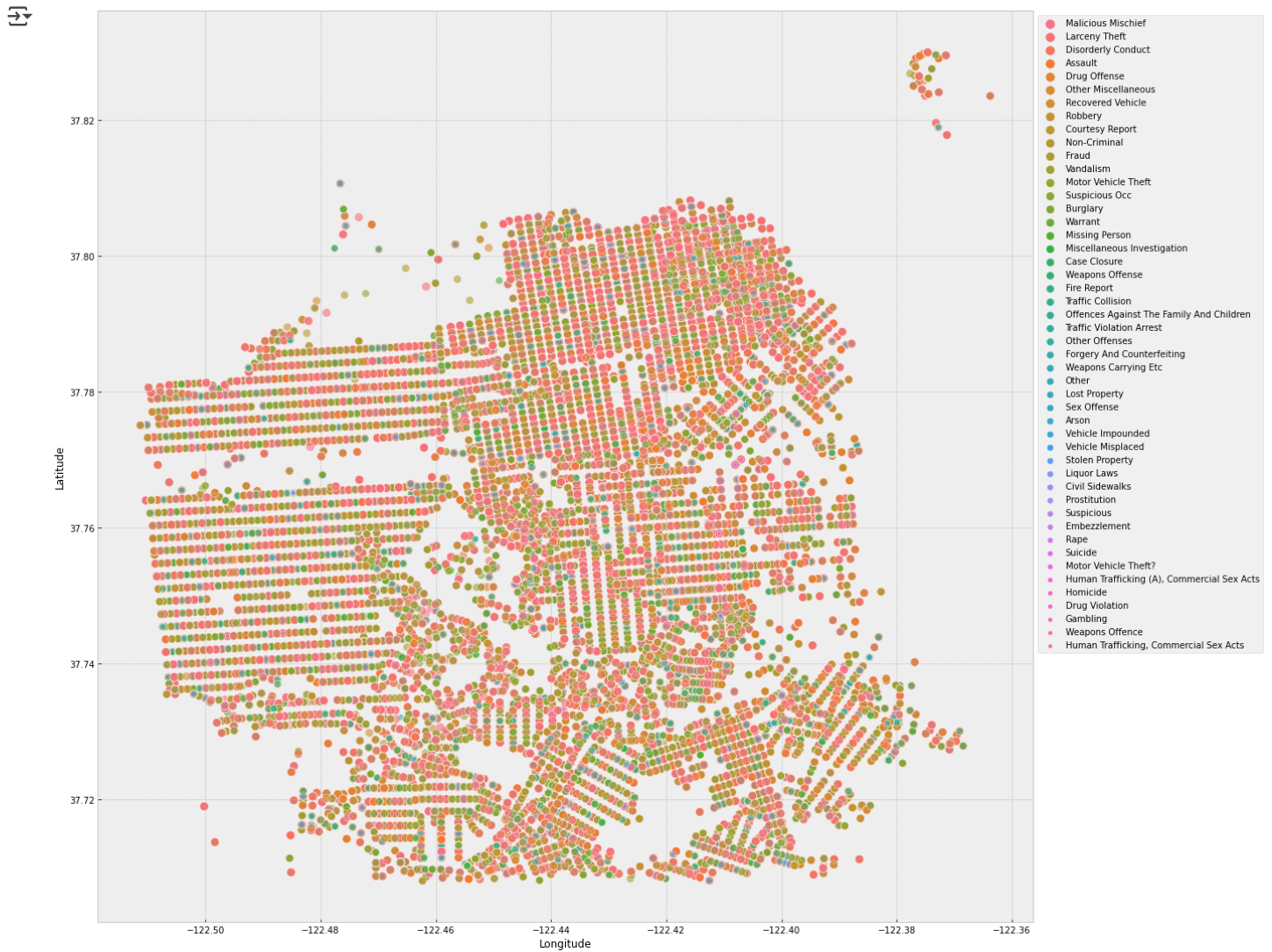
Relationship between Coordinates (Longitude and Latitude) and Incident Category

```
train.rename(columns={"Incident Category": "Category"})
```

	Incident Datetime	Incident Day of Week	Category	Incident Description	Resolution	Intersection	Police District	Latitude	Longitude
295386	2018/04/21 11:30:00 PM	Saturday	Malicious Mischief	Malicious Mischief, Vandalism to Property	Open or Active	THORNTON AVE \ SAN BRUNO AVE	Bayview	37.730318	-122.404706
418933	2018/11/11 01:00:00 PM	Sunday	Larceny Theft	Theft, From Locked Vehicle, >\$950	Open or Active	SOUTH VAN NESS AVE \ PLUM ST	Mission	37.770876	-122.418029
401668	2019/06/22 01:19:00 PM	Saturday	Disorderly Conduct	Terrorist Threats	Open or Active	THORNTON AVE \ SAN BRUNO AVE	Bayview	37.730318	-122.404706
468608	2019/05/06 06:02:00 PM	Monday	Assault	Battery, former Spouse or Dating Relationship	Cite or Arrest Adult	WOODSIDE AVE \ IDORA AVE	Park	37.746106	-122.454664
495219	2019/10/23 11:04:00 PM	Wednesday	Drug Offense	Marijuana, Possession For Sale	Open or Active	TREAT AVE \ 16TH ST \ HARRISON ST	Mission	37.765448	-122.413105
...	...	...	...	...	...	...	...	...	...
559486	2019/08/05 11:00:00 AM	Monday	Malicious Mischief	Malicious Mischief, Vandalism to	Open or Active	CABRILLO ST \ 07TH AVE	Richmond	37.775321	-122.464942

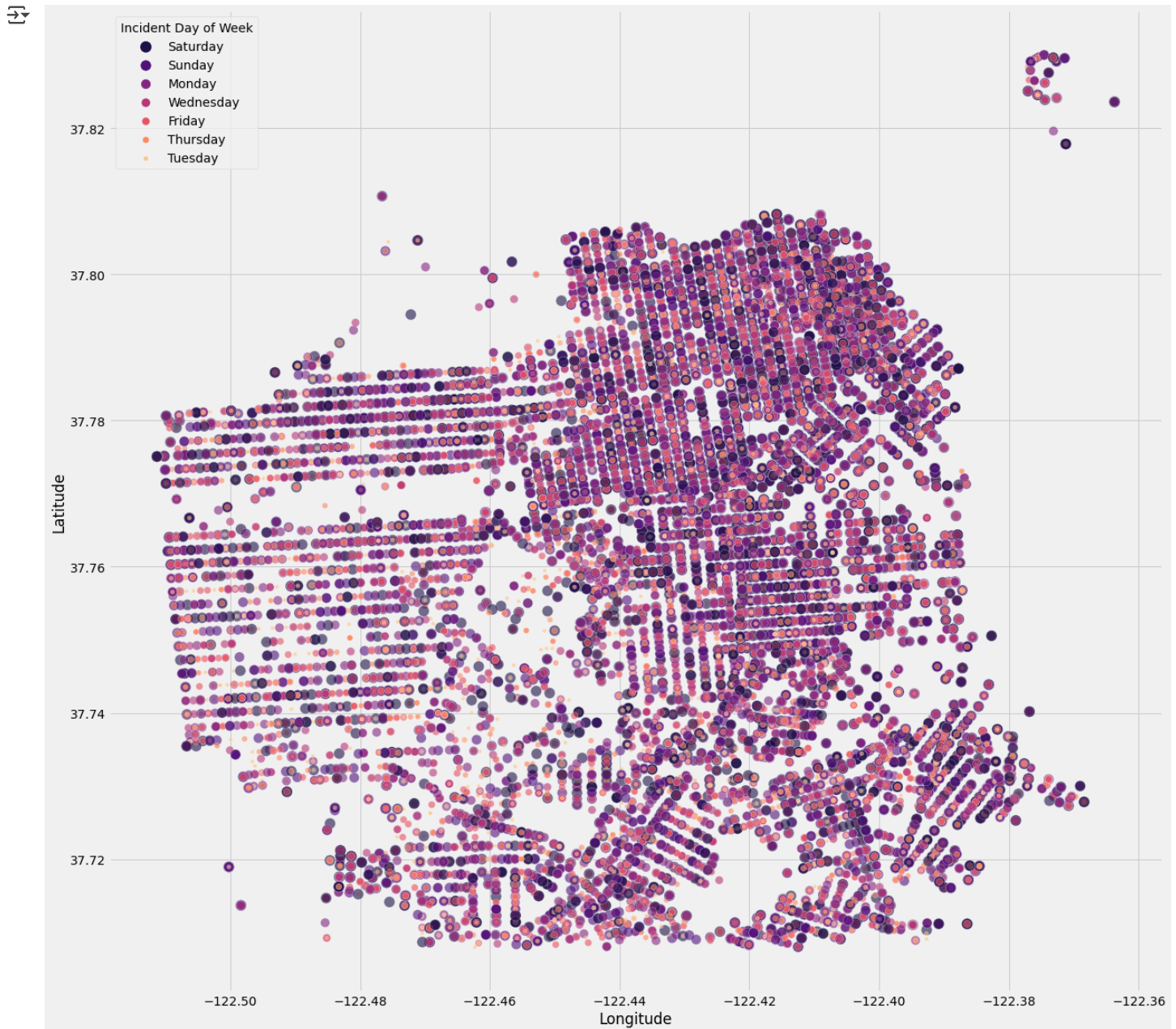
```
with plt.style.context('bmh'):
    fig, ax = plt.subplots(1, 1, figsize=(19, 19))
    sns.scatterplot(data=train.iloc[:250000], x='Longitude', y='Latitude', alpha=0.6, palette='husl', hue='Incident Category',
                    size='Incident Category', sizes=(15, 100))
    plt.legend(bbox_to_anchor=(1.0, 1.0), loc='upper left')

plt.show()
```



Relationship between Day of the Week and Coordinates

```
with plt.style.context('fivethirtyeight'):
    fig, ax = plt.subplots(1, 1, figsize=(19, 19))
    sns.scatterplot(data=train.iloc[:50000], x='Longitude', y='Latitude', alpha=0.6, palette='magma', hue='Incident Day of Week',
                    size='Incident Day of Week', sizes=(20, 150), markers=True)
plt.show()
```



6. Folium

Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map.

```
pip install folium
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: folium in /usr/local/lib/python3.9/dist-packages (0.12.1.post1)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.9/dist-packages (from folium) (0.6.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from folium) (1.22.4)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from folium) (2.25.1)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.9/dist-packages (from folium) (3.1.2)
```

```
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from jinja2>=2.9->folium) (2.1.2)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.9/dist-packages (from requests->folium) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->folium) (2022.12.7)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->folium) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->folium) (1.26.15)
```

```
def query_heat_map(query, location=[37.770496, -122.479895]):
    train_query = train.query(query).loc[:, ['Latitude', 'Longitude']]
    if train_query.shape[0] == 0:
        print('Data not working!')
    m = folium.Map(location=location, zoom_start=13, tiles='stamenterrain')
    train_query_geo_list = train_query.values.tolist()
    HeatMap(train_query_geo_list, blur=2, radius=3).add_to(m)

    # Save html of the Map
    m.save('SFCrimesgeo.html')
    return m
```

Shows the Category "Assault" with a Resolution of 'Cite or Arrest' where it has more concentration

```
train.rename(columns={'Incident Category': 'Incident_Category'}, inplace=True)
```

```
# iterating the columns
for col in train.columns:
    print(col)
```

```
➡ Incident Datetime
Incident Day of Week
Incident_Category
Incident Description
Resolution
Intersection
Police District
Latitude
Longitude
```

```
query_heat_map("Incident_Category=='Assault' & Resolution=='Cite or Arrest Adult'")
```

 Make this Notebook Trusted to load map: File -> Trust Notebook
+
-



Leaflet (<https://leafletjs.com>) | Map tiles by Stamen Design (<http://stamen.com>), under CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0>). Data by © OpenStreetMap (<http://openstreetmap.org>), under CC BY SA (<http://creativecommons.org/licenses/by-sa/3.0>).

6. Timestamp operations

Read Dates as Timestamp, and split and convert the Timestamp data.

- What time of the day has the highest crime rate using the `.count()` statistic.

```
train_timestamp = pd.read_csv('train.csv', parse_dates=True, index_col='Incident Datetime')
```

```
train_timestamp.loc['2023-02-21'].sort_values(by=['Incident Datetime'])
```



Incident Datetime	Incident Day of Week	Incident Category	Incident Description	Resolution	Intersection	Police District	Latitude	Longitude
2023-02-21 00:00:00	Tuesday	Assault	Battery	Cite or Arrest Adult	SCOTT ST \ HAIGHT ST	Northern	37.771479	-122.435373
2023-02-21 00:00:00	Tuesday	Malicious Mischief	Malicious Mischief, Graffiti, Real or Personal...	Cite or Arrest Adult	SCOTT ST \ HAIGHT ST	Northern	37.771479	-122.435373
2023-02-21 00:00:00	Tuesday	Missing Person	Missing Adult	Open or Active	MOSCOW ST \ PERSIA AVE	Ingleside	37.719712	-122.428805
2023-02-21 00:00:00	Tuesday	Motor Vehicle Theft	Vehicle, Stolen, Auto	Open or Active	GEARY ST \ JONES ST	Central	37.786730	-122.413161
2023-02-21 00:00:00	Tuesday	Other Miscellaneous	Resisting, Delaying, or Obstructing Peace Off...	Cite or Arrest Adult	SCOTT ST \ HAIGHT ST	Northern	37.771479	-122.435373
...	...	...	...	...	...	...	...	...
2023-02-21 23:00:00	Tuesday	Larceny Theft	Theft, Bicycle, \$200-\$950	Open or Active	TAYLOR ST \ BUSH ST	Central	37.789740	-122.412076

```
train_timestamp.index

DatetimeIndex(['2018-04-21 23:30:00', '2018-11-11 13:00:00',
               '2019-06-22 13:19:00', '2019-05-06 18:02:00',
               '2019-10-23 23:04:00', '2021-07-17 23:40:00',
               '2019-07-29 21:42:00', '2022-08-08 21:38:00',
               '2018-11-23 03:30:00', '2022-08-06 17:15:00',
               ...
               '2018-04-23 07:50:00', '2022-07-08 05:00:00',
               '2018-05-19 14:00:00', '2022-05-17 18:00:00',
               '2021-04-26 21:15:00', '2019-08-05 11:32:00',
               '2022-05-24 07:34:00', '2020-11-16 20:00:00',
               '2018-10-30 12:00:00', '2019-06-10 20:30:00'],
              dtype='datetime64[ns]', name='Incident Datetime', length=533785, freq=None)

train_timestamp.loc['2023'].mean()

<ipython-input-46-044d8a541452>:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated
  train_timestamp.loc['2023'].mean()
Latitude      37.768614
Longitude     -122.423341
dtype: float64

train_timestamp.loc['2023'].count()

Incident Day of Week    18140
Incident Category       18140
Incident Description    18140
Resolution              18140
Intersection            18140
Police District         18140
Latitude                18140
Longitude               18140
dtype: int64

train_timestamp.loc['2022'].count()

Incident Day of Week    100397
Incident Category       100397
Incident Description    100397
Resolution              100397
Intersection            100397
Police District         100397
Latitude                100397
Longitude               100397
dtype: int64

mc = train_timestamp.groupby(level=0)

mc.count().head(6)
```



Incident Datetime	Incident Day of Week	Incident Category	Incident Description	Resolution	Intersection	Police District	Latitude	Longitude
2018-01-01 00:00:00	66	66	66	66	66	66	66	66
2018-01-01 00:01:00	14	14	14	14	14	14	14	14
2018-01-01 00:10:00	1	1	1	1	1	1	1	1
2018-01-01								

7. SigmaClip

Used to remove outliers

```
train_plot = pd.read_csv('train.csv', parse_dates=True)

train.rename(columns={'Incident Datetime': 'Date','Incident Day of Week': 'DayOfWeek','Incident Category': 'Incident_Category',
                    'Incident Description': 'Description','Police District': 'Police_District', }, inplace=True)

frequency_c = pd.DataFrame(train_plot['Incident Category'].value_counts(normalize=True))
frequency_c.reset_index(inplace=True)
frequency_c.rename({'index': 'Incident Category', 'Incident Category': 'Freq'}, axis=1, inplace=True)
frequency_c['Cumsum_Freq'] = frequency_c['Freq'].cumsum()

frequency_c = frequency_c.set_index('Incident Category')

freq_quantiles = np.percentile(frequency_c['Cumsum_Freq'], [25, 50, 75])

qt = freq_quantiles[1]

qt
0.9760746367919667

sigma = 0.74 * (freq_quantiles[2] - freq_quantiles[0])

sigma
0.11708354487293578

query_freq = frequency_c.query("(Cumsum_Freq > @qt - 5 * @sigma) & (Cumsum_Freq < @qt + 5 * @sigma)")

query_freq.reset_index(inplace=True)

query_freq_list = list(query_freq['Incident Category'])

query_freq_list
['Malicious Mischief',
'Assault',
'Non-Criminal',
'Burglary',
'Motor Vehicle Theft',
'Fraud',
'Warrant',
'Recovered Vehicle',
'Drug Offense',
'Lost Property',
'Robbery',
'Missing Person',
'Suspicious Occ',
'Disorderly Conduct',
'Offences Against The Family And Children',
'Miscellaneous Investigation',
```

```
'Traffic Violation Arrest',
'Other',
'Other Offenses',
'Weapons Offense',
'Weapons Carrying Etc',
'Stolen Property',
'Case Closure',
'Forgery And Counterfeiting',
'Arson',
'Courtesy Report',
'Traffic Collision',
'Vandalism',
'Fire Report',
'Embezzlement',
'Prostitution',
'Civil Sidewalks',
'Sex Offense',
'Vehicle Impounded',
'Suicide',
'Vehicle Misplaced',
'Drug Violation',
'Rape',
'Human Trafficking (A), Commercial Sex Acts',
'Suspicious',
'Homicide',
'Liquor Laws',
'Motor Vehicle Theft?',
'Gambling',
'Human Trafficking, Commercial Sex Acts',
'Weapons Offence',
'Human Trafficking (B), Involuntary Servitude']
```

```
train_sigma = train.loc[train['Incident_Category'].isin(query_freq_list)]
```

```
train_sigma['Date'] = pd.to_datetime(train_sigma['Date'])
```

 <ipython-input-65-c65f3f0fcb62>:1: 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/user_guide/indexing.html#returning-a-view-versus-a-copy
train_sigma['Date'] = pd.to_datetime(train_sigma['Date'])

```
test['Incident Datetime'] = pd.to_datetime(test['Incident Datetime'])
```

```
#total crimes in each major category
train_sigma['Incident_Category'].value_counts()
```

 Malicious Mischief 36507
Assault 34128
Non-Criminal 32911
Burglary 31419
Motor Vehicle Theft 28570
Fraud 17686
Warrant 16219
Recovered Vehicle 15925
Drug Offense 13732
Lost Property 13104
Robbery 12512
Missing Person 12218
Suspicious Occ 11553
Disorderly Conduct 9202
Offences Against The Family And Children 7830
Miscellaneous Investigation 5803
Traffic Violation Arrest 5471
Other 4594
Other Offenses 4056
Weapons Offense 3851
Weapons Carrying Etc 3054
Stolen Property 2794
Case Closure 2220
Forgery And Counterfeiting 1904
Arson 1616
Courtesy Report 1609
Traffic Collision 1297
Vandalism 1210
Fire Report 817
Embezzlement 740
Prostitution 722

Civil Sidewalks	711
Sex Offense	583
Vehicle Impounded	359
Suicide	242
Vehicle Misplaced	219
Drug Violation	183
Rape	135
Human Trafficking (A), Commercial Sex Acts	98
Suspicious	79
Homicide	74
Liquor Laws	62
Motor Vehicle Theft?	57
Gambling	21
Human Trafficking, Commercial Sex Acts	17
Weapons Offence	15

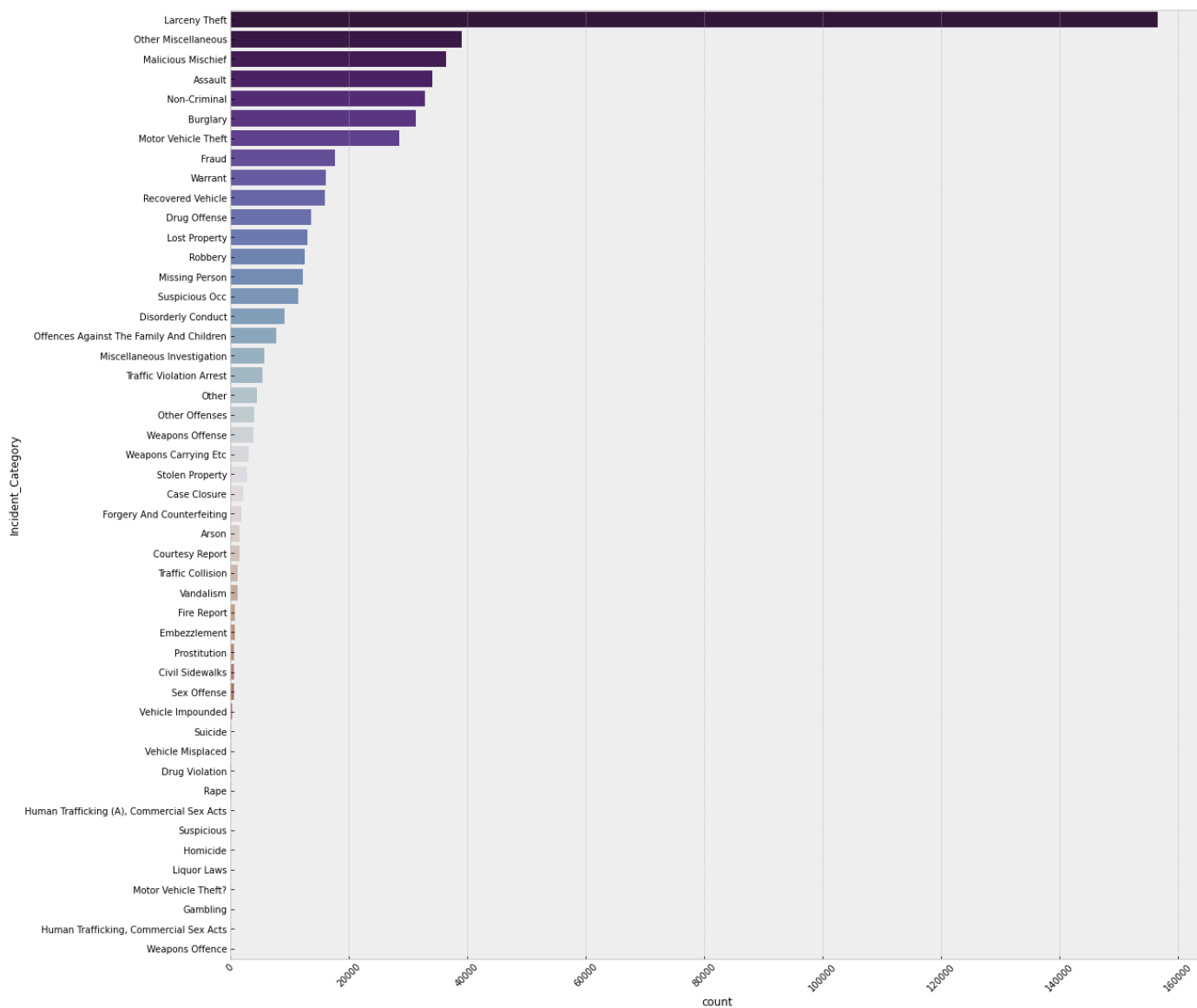
Name: Incident_Category, dtype: int64

8. Second Exploratory Data Analysis

Which Crime is the most common?

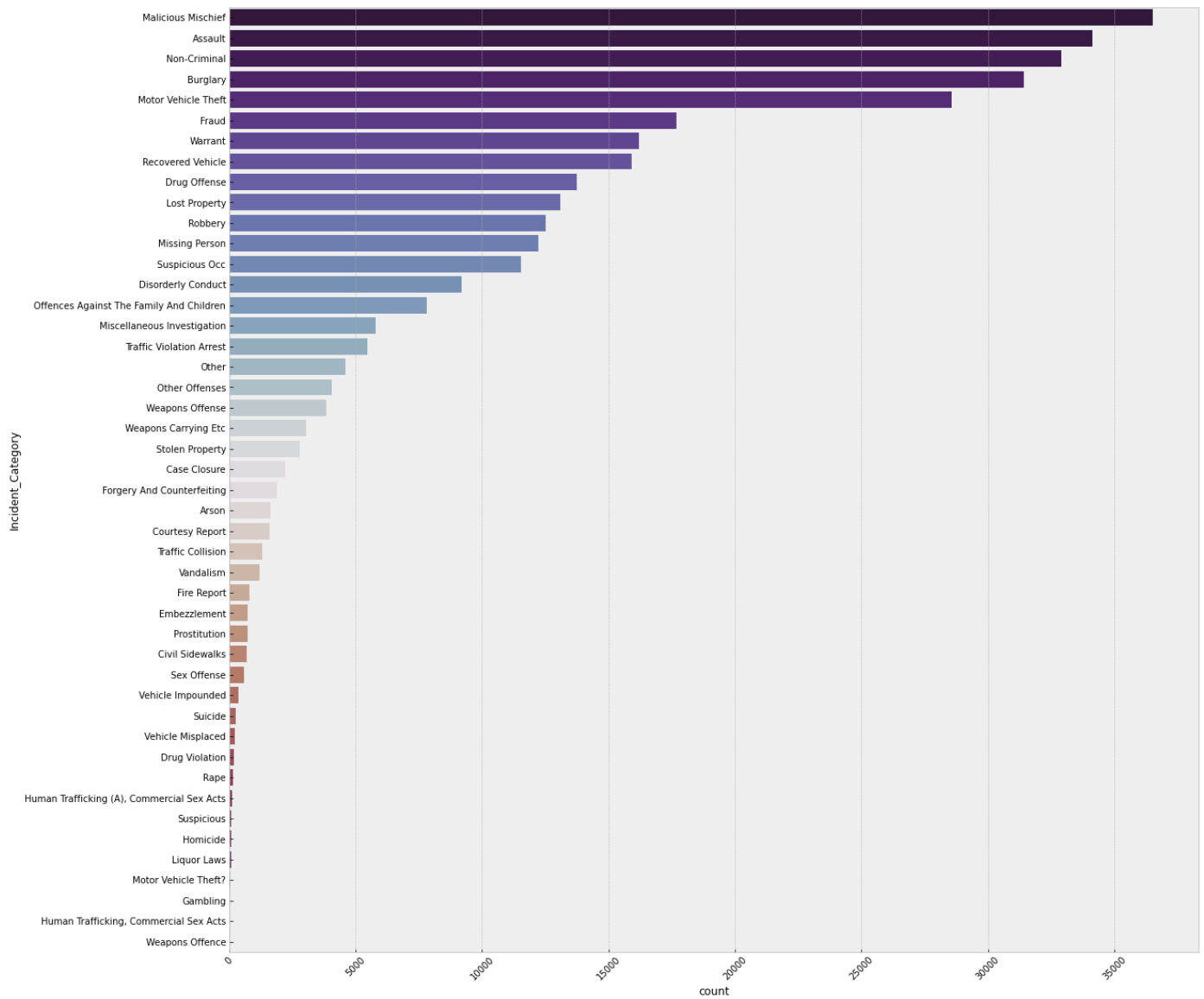
```
with plt.style.context('bmh'):
    fig, ax = plt.subplots(1, 1, figsize=(19, 19))
    sns.countplot(y='Incident_Category', data=train, order=train['Incident_Category'].value_counts().index, palette='twilight_shifted')
    ax.tick_params(axis='x', rotation=45)

plt.show()
```



```
with plt.style.context('bmh'):
    fig, ax = plt.subplots(1, 1, figsize=(19, 19))
    sns.countplot(y='Incident_Category', data=train_sigma, order=train_sigma['Incident_Category'].value_counts().index, palette='twilight_s')
    ax.tick_params(axis='x', rotation=45)

    plt.show()
```



✓ Which incidents are most frequently dealt with by each police station?

This can be made clearer by showing the police stations as location points on the map.

```
cross_tab_train = pd.crosstab(train_sigma['Incident_Category'], train_sigma['Police_District'])
```

```
cross_tab_train = pd.crosstab(train['Incident_Category'], train['Police_District'])
cross_tab_train.style.background_gradient(cmap="Blues", text_color_threshold=0.5)
```



Police_District	Bayview	Central	Ingleside	Mission	Northern	Out of SF	Park	Richmond	Southern	Taraval	Tenderloin
Incident_Category											
Arson	267	199	115	238	230	1	52	84	200	89	141
Assault	3982	3872	2684	5593	3845	124	1044	1146	4464	1820	5554
Burglary	2492	4534	2846	3645	5450	35	2128	2241	4210	2462	1376
Case Closure	151	332	150	118	185	252	151	147	469	138	127
Civil Sidewalks	0	22	0	160	21	0	490	8	7	0	3
Courtesy Report	65	50	33	61	43	1114	23	37	47	43	93
Disorderly Conduct	1112	1033	835	1421	1269	63	317	464	1074	692	922
Drug Offense	439	810	372	1513	952	21	337	170	1806	209	7103
Drug Violation	25	15	17	37	6	3	4	1	28	5	42
Embezzlement	113	97	26	93	54	20	13	37	129	40	118
Fire Report	174	66	78	120	59	0	40	48	110	58	64
Forgery And Counterfeiting	138	288	146	251	269	58	77	109	254	167	147
Fraud	1297	2535	1595	1894	2289	853	695	1244	1941	2282	1061
Gambling	2	2	5	6	0	0	0	0	0	4	2
Homicide	11	2	6	9	11	3	0	4	11	9	8
Human Trafficking (A), Commercial Sex Acts	3	17	0	14	6	2	0	0	34	18	4
Human Trafficking, Commercial Sex Acts	0	0	1	9	2	0	0	0	2	1	2
Larceny Theft	9433	33687	9351	16538	30111	486	7571	12605	17214	11272	8279
Liquor Laws	2	18	6	24	2	0	3	0	4	2	1
Lost Property	704	2643	648	1818	1633	507	395	645	1751	973	1387
Malicious Mischief	3619	5872	3131	4758	5292	59	1795	2205	4740	2762	2274
Miscellaneous Investigation	745	479	698	680	691	61	226	400	659	662	502
Missing Person	1015	1068	1173	1392	1192	352	1262	660	1599	1250	1255
Motor Vehicle Theft	4894	2249	3989	3764	3146	270	1486	1809	2854	2794	1315
Motor Vehicle Theft?	3	7	9	8	5	2	0	5	5	9	4
Non-Criminal	2372	4731	2774	3979	3767	178	1684	2024	4375	2651	4376
Offences Against The Family And Children	1100	735	994	919	883	77	235	424	837	863	763
Other	327	590	304	469	580	1	201	206	607	317	992
Other Miscellaneous	3512	5589	2996	6656	4577	201	1329	1756	5324	2032	5136
Other Offenses	505	362	394	457	493	57	294	260	442	276	516
Prostitution	7	130	8	375	9	0	1	0	59	44	89
Rape	10	9	7	17	13	4	7	5	40	8	15
Recovered Vehicle	4074	776	1790	1837	1591	182	559	823	1920	1087	1286
Robbery	1351	1903	1249	1955	1530	59	268	485	1362	609	1741
Sex Offense	36	76	33	116	81	2	29	24	89	27	70
Stolen Property	281	488	172	353	422	66	95	114	381	128	294
Suicide	20	31	25	26	32	0	10	26	38	21	13
Suspicious	6	9	4	5	5	0	4	7	34	4	1
Suspicious Occ	1298	1527	870	1331	1161	195	485	825	1398	971	1492
Traffic Collision	212	116	131	155	150	11	40	87	126	183	86
Traffic Violation Arrest	477	546	532	1211	481	24	274	283	671	322	650
Vandalism	147	138	140	161	162	4	73	72	164	119	30
Vehicle Impounded	440	24	50	20	20	4	0	40	22	0	44

Vehicle Impounded	110	24	39	20	39	1	0	40	32	9	11
Vehicle Misplaced	11	31	5	48	32	0	12	22	19	29	10
Warrant	1278	1958	840	3005	1590	118	630	396	2406	540	3458
Weapons Carrying Etc	562	292	278	576	229	57	69	59	365	125	442

```
cross_tab_train.style.background_gradient(cmap="Blues", text_color_threshold=0.5)
```



Police_District	Bayview	Central	Ingleside	Mission	Northern	Out of SF	Park	Richmond	Southern	Taraval	Tenderloin
Incident_Category											
Arson	267	199	115	238	230	1	52	84	200	89	141
Assault	3982	3872	2684	5593	3845	124	1044	1146	4464	1820	5554
Burglary	2492	4534	2846	3645	5450	35	2128	2241	4210	2462	1376
Case Closure	151	332	150	118	185	252	151	147	469	138	127
Civil Sidewalks	0	22	0	160	21	0	490	8	7	0	3
Courtesy Report	65	50	33	61	43	1114	23	37	47	43	93
Disorderly Conduct	1112	1033	835	1421	1269	63	317	464	1074	692	922
Drug Offense	439	810	372	1513	952	21	337	170	1806	209	7103
Drug Violation	25	15	17	37	6	3	4	1	28	5	42
Embezzlement	113	97	26	93	54	20	13	37	129	40	118
Fire Report	174	66	78	120	59	0	40	48	110	58	64
Forgery And Counterfeiting	138	288	146	251	269	58	77	109	254	167	147
Fraud	1297	2535	1595	1894	2289	853	695	1244	1941	2282	1061
Gambling	2	2	5	6	0	0	0	0	0	4	2
Homicide	11	2	6	9	11	3	0	4	11	9	8
Human Trafficking (A), Commercial Sex Acts	3	17	0	14	6	2	0	0	34	18	4
Human Trafficking, Commercial Sex Acts	0	0	1	9	2	0	0	0	2	1	2
Larceny Theft	9433	33687	9351	16538	30111	486	7571	12605	17214	11272	8279
Liquor Laws	2	18	6	24	2	0	3	0	4	2	1
Lost Property	704	2643	648	1818	1633	507	395	645	1751	973	1387
Malicious Mischief	3619	5872	3131	4758	5292	59	1795	2205	4740	2762	2274
Miscellaneous Investigation	745	479	698	680	691	61	226	400	659	662	502
Missing Person	1015	1068	1173	1392	1192	352	1262	660	1599	1250	1255
Motor Vehicle Theft	4894	2249	3989	3764	3146	270	1486	1809	2854	2794	1315
Motor Vehicle Theft?	3	7	9	8	5	2	0	5	5	9	4
Non-Criminal	2372	4731	2774	3979	3767	178	1684	2024	4375	2651	4376
Offences Against The Family And Children	1100	735	994	919	883	77	235	424	837	863	763
Other	327	590	304	469	580	1	201	206	607	317	992
Other Miscellaneous	3512	5589	2996	6656	4577	201	1329	1756	5324	2032	5136
Other Offenses	505	362	394	457	493	57	294	260	442	276	516
Prostitution	7	130	8	375	9	0	1	0	59	44	89
Rape	10	9	7	17	13	4	7	5	40	8	15
Recovered Vehicle	4074	776	1790	1837	1591	182	559	823	1920	1087	1286
Robbery	1351	1903	1249	1955	1530	59	268	485	1362	609	1741
Sex Offense	36	76	33	116	81	2	29	24	89	27	70
Stolen Property	281	488	172	353	422	66	95	114	381	128	294
Suicide	20	31	25	26	32	0	10	26	38	21	13
Suspicious	6	9	4	5	5	0	4	7	34	4	1
Suspicious Occ	1298	1527	870	1331	1161	195	485	825	1398	971	1492
Traffic Collision	212	116	131	155	150	11	40	87	126	183	86
Traffic Violation Arrest	477	546	532	1211	481	24	274	283	671	322	650
Vandalism	147	138	140	161	162	4	73	72	164	119	30
Vehicle Impounded	440	24	50	20	20	4	0	40	22	0	44