GEORGE MASON UNIVERSITY

FINAL PROJECT

AIT-580-005

Submitted by
Satyam Singh

G01389368

Crime Analysis in San Francisco using Python, R and SQL

Abstract

This paper revolves around the crime statistic of a popular city call San Francisco in USA. As we know, data has become an integral part of every domain in life, we must take this opportunity and reap the benefits of it. Hence are objective in this research would be to find all patterns and trends in the dataset which can help fellow citizens take better decisions in their day-to-day life.

Crimes involving theft, assault, Fraud, and many more can have a big impact on the life of a common man. With increase in globalization, several students and Working professionals leave their home and are staying far away from their families. Going to a completely different place can be overwhelming for them and their loved ones. But I believe that with proper analysis and research a person can make the best decision suited for him.

In this research, our aim is to gain information about San Francisco's crime statistics relating to each area. This can not only help citizens be wary of certain neighbourhoods but can alert the Law and Enforcement to deploy more Officials in these areas.

Introduction

So, our data comes directly from San Francisco's Police department's Official repository. This indicates that our data is genuine and has not been tampered with. This is important because we should know the source of our dataset and if its reliable or not. Based on our data we draw conclusions and if the data is not true then our conclusions will be False.

In our case the data set has a total of 659,000 rows and 26 columns with its start date from 1st January 2018 to October 2022. The data is collected via their Official Portal (Filed Online) or In-person. Before going ahead and uploading the data, a supervisor checks and gives it a pass.

The data is updated on an hourly basis and the database is managed by OpenData. The dataset has 26 columns each providing useful information.

Data Types

Number	Column Name	Description	Type
1.	**Incident Datetime**	The date and time when the incident occurred	Date & Time
2.	**Incident Date**	The date the incident occurred	Date & Time
3.	**Incident Time**	The time the incident occurred	Plain Text
4.	**Incident Year**	The year the incident occurred, provided as a convenience for filtering	Plain Text
5.	**Incident Day of Wee k**	The day of week the incident occurred	Plain Text
6.	**Report Datetime**	Distinct from Incident Datetime, Report Datetime is when the report was filed.	Date & Time
7.	**Row ID**	A unique identifier for each row of data in the dataset	Plain Text
8.	**Incident ID**	This is the system generated identifier for incident reports. Incident IDs and Incident Numbers both uniquely identify reports, but Incident Numbers are used when referencing cases and report documents.	Plain Text
9.	**Incident Number**	The number issued on the report, sometimes interchangeably referred to as the Case Number. This number is used to reference cases and report documents.	Plain Text
10.	**CAD Number**	The Computer Aided Dispatch (CAD) is the system used by the Department of Emergency Management (DEM) to dispatch officers and other public safety personnel.	Plain Text
11.	**Report Type Code**	A system code for report types, these have corresponding descriptions within the dataset.	Plain Text
12.	**Report Type Descri ption**	The description of the report type, can be one of: Initial; Initial Supplement; Vehicle Initial; Vehicle Supplement; Coplogic Initial; Coplogic Supplement	Plain Text
13.	**Filed Online**	Non- emergency police reports can be filed online by members of the public using SFPD’s self-service reporting system called Coplogic	Checkbox

14.	**Incident Code**	Incident Codes are the system codes to describe a type of incident.	Plain Text
15.	**Incident Category**	A category mapped on to the Incident Code used in statistics and reporting. Mappings provided by the Crime Analysis Unit of the Police Department.	Plain Text
16.	**Incident Subcategory**	A subcategory mapped to the Incident Code that is used for statistics and reporting. Mappings are provided by the Crime Analysis Unit of the Police Department.	Plain Text
17.	**Incident Description**	The description of the incident that corresponds with the Incident Code. These are generally self-explanatory.	Plain Text
18.	**Resolution**	The resolution of the incident at the time of the report. Can be one of: • Cite or Arrest Adult • Cite or Arrest Juvenile* • Exceptional Adult • Exceptional Juvenile* • Open or Active • Unfounded.	Plain Text
19.	**Intersection**	The 2 or more street names that intersect closest to the original incident separated by a backward slash (\).	Plain Text
20.	**CNN**	The unique identifier of the intersection for reference back to other related basemap datasets.	Plain Text
21.	**Police District**	The Police District where the incident occurred. District boundaries can be reviewed in the link below.	Plain Text
22.	**Analysis Neighborhood**	This field is used to identify the neighborhood where each incident occurs.	Plain Text
23.	**Supervisor District**	There are 11 members elected to the Board of Supervisors in San Francisco, each representing a geographic district. The Board of Supervisors is the legislative body for San Francisco. The districts are numbered 1 through 11.	Number
24.	**Latitude**	The latitude coordinate in WGS84, spatial reference is EPSG:4326	Number
25.	**Longitude**	The longitude coordinate in WGS84, spatial reference is EPSG:4326	Number
26.	**Point**	Geolocation in OGC WKT format (e.g, POINT(37.4,-122.3)	Point

Research Questions

1. What day has the greatest number of crimes committed?

2. Which is the most common type of crime?

3. Unsolved cases are the most under which category of crime?

To solve these questions, we must understand our data and its types. As our data set is very large, we would have to make multiple manipulations to fit our requirements. As we know, our data has 651,578 rows and 34 columns in total, we must go through all the columns and have a general idea about how the data is inserted in the dataset. We must check each columns usefulness regarding our requirements. As we have the exact day of an incident happening, we can filter our data to get the exact answer for Question 1. Likewise, we can use methods to select different column combinations which will help us find answers to the other questions.

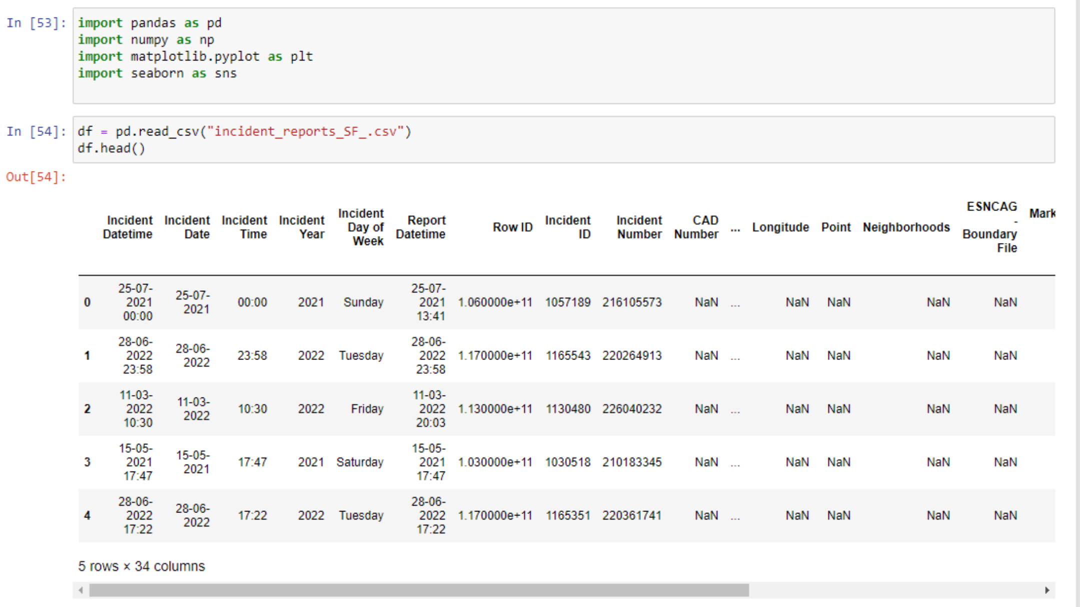
The main objective of these questions is for the well-being of citizens of San Francisco and to suggest or recommend the Police department about the frequencies of crimes in the city. We can ask different questions which may be more accurate or helpful for the users and the answers can be found through Data Analytics.

Data analytics can help you take better decisions and based on your requirement and needs, your questions might differ. Nevertheless, with today’s technological advancement we can use various software’s and programming languages to find out solutions to problems human mind cannot comprehend.

Analysis Using Python

Importing and Data Cleaning

The first step towards analysis would be to successfully load the dataset into data-frame and install all the required libraries.

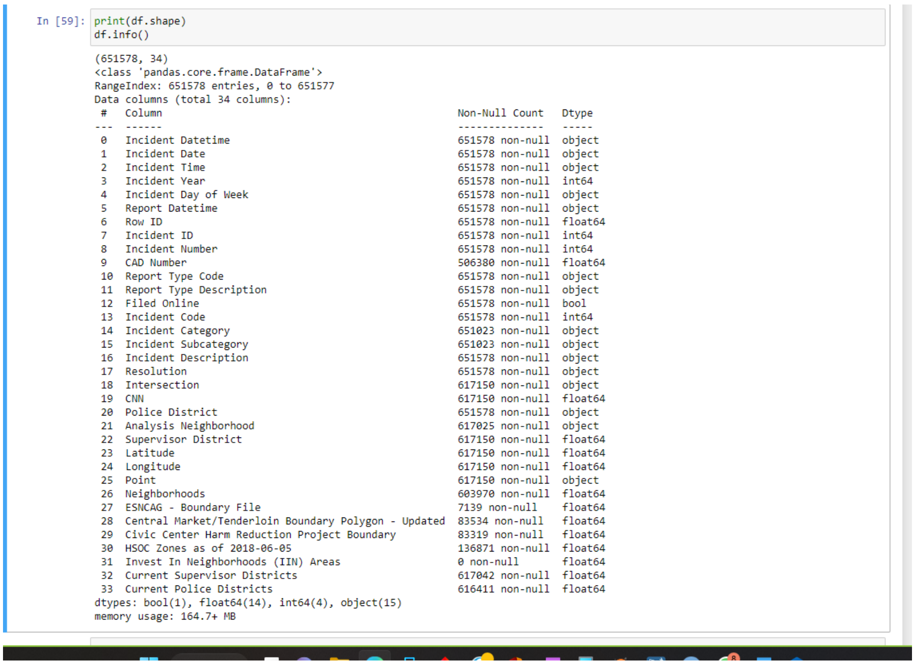


```
In [53]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns

In [54]: df = pd.read_csv("incident_reports_SF_.csv")
         df.head()

Out[54]:
```

	Incident Datetime	Incident Date	Incident Time	Incident Year	Incident Day of Week	Report Datetime	Row ID	Incident ID	Incident Number	CAD Number	...	Longitude	Point	Neighborhoods	ESNCAG - Boundary File	Mark
0	25-07-2021 00:00	25-07-2021	00:00	2021	Sunday	25-07-2021 13:41	1.060000e+11	1057189	216105573	NaN	...	NaN	NaN	NaN	NaN	
1	28-06-2022 23:58	28-06-2022	23:58	2022	Tuesday	28-06-2022 23:58	1.170000e+11	1165543	220264913	NaN	...	NaN	NaN	NaN	NaN	
2	11-03-2022 10:30	11-03-2022	10:30	2022	Friday	11-03-2022 20:03	1.130000e+11	1130480	226040232	NaN	...	NaN	NaN	NaN	NaN	
3	15-05-2021 17:47	15-05-2021	17:47	2021	Saturday	15-05-2021 17:47	1.030000e+11	1030518	210183345	NaN	...	NaN	NaN	NaN	NaN	
4	28-06-2022 17:22	28-06-2022	17:22	2022	Tuesday	28-06-2022 17:22	1.170000e+11	1165351	220361741	NaN	...	NaN	NaN	NaN	NaN	

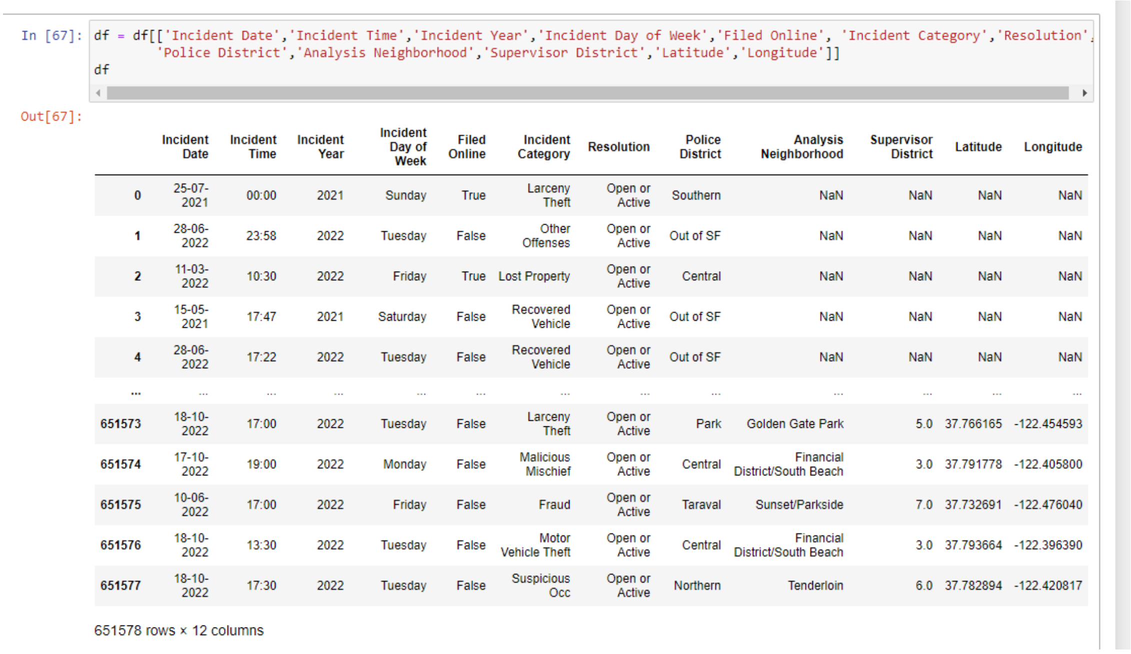
5 rows × 34 columns

After importing the dataset, we check the dimensions of the data along with information about the datatypes of the dataset.



```
In [59]: print(df.shape)
         df.info()

(651578, 34)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651578 entries, 0 to 651577
Data columns (total 34 columns):
 #   Column                                             Non-Null Count   Dtype  
---  ------                                             --------------   -----  
 0   Incident Datetime                                  651578 non-null  object 
 1   Incident Date                                      651578 non-null  object 
 2   Incident Time                                      651578 non-null  object 
 3   Incident Year                                      651578 non-null  int64  
 4   Incident Day of Week                               651578 non-null  object 
 5   Report Datetime                                    651578 non-null  object 
 6   Row ID                                             651578 non-null  float64
 7   Incident ID                                        651578 non-null  int64  
 8   Incident Number                                    651578 non-null  int64  
 9   CAD Number                                         506380 non-null  float64
 10  Report Type Code                                   651578 non-null  object 
 11  Report Type Description                            651578 non-null  object 
 12  Filed Online                                       651578 non-null  bool   
 13  Incident Code                                      651578 non-null  int64  
 14  Incident Category                                  651023 non-null  object 
 15  Incident Subcategory                               651023 non-null  object 
 16  Incident Description                               651578 non-null  object 
 17  Resolution                                         651578 non-null  object 
 18  Intersection                                       617150 non-null  object 
 19  CNN                                                617150 non-null  float64
 20  Police District                                    651578 non-null  object 
 21  Analysis Neighborhood                              617025 non-null  object 
 22  Supervisor District                                617150 non-null  float64
 23  Latitude                                           617150 non-null  float64
 24  Longitude                                          617150 non-null  float64
 25  Point                                              617150 non-null  object 
 26  Neighborhoods                                      603970 non-null  float64
 27  ESNCAG - Boundary File                             7139 non-null    float64
 28  Central Market/Tenderloin Boundary Polygon - Updated  83534 non-null   float64
 29  Civic Center Harm Reduction Project Boundary       83319 non-null   float64
 30  HSOC Zones as of 2018-06-05                        136871 non-null  float64
 31  Invest In Neighborhoods (IIN) Areas                0 non-null       float64
 32  Current Supervisor Districts                       617042 non-null  float64
 33  Current Police Districts                           616411 non-null  float64
dtypes: bool(1), float64(14), int64(4), object(15)
memory usage: 164.7+ MB
```

As there are a total of 34 columns we will go ahead and select only a few that important for analysis and drop the rest.



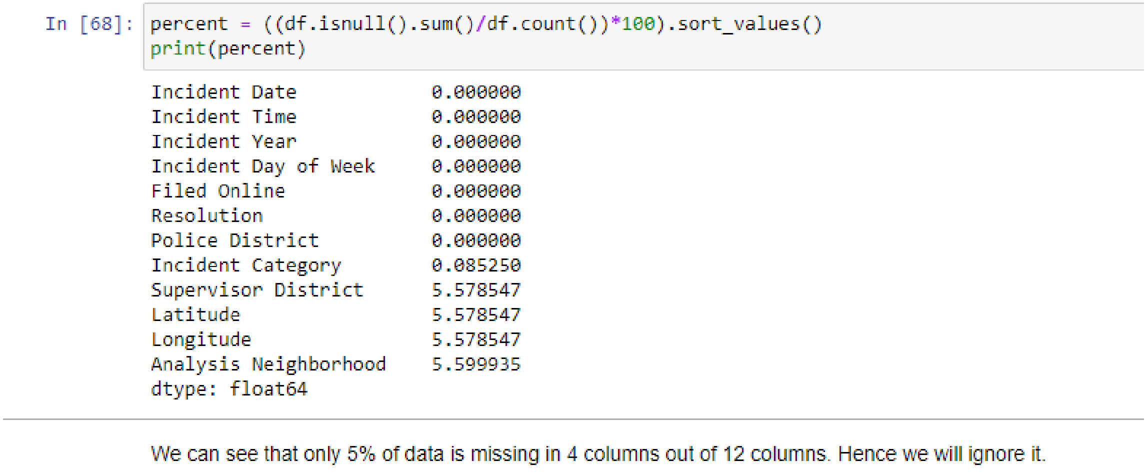
In [67]:
```
df = df[['Incident Date','Incident Time','Incident Year','Incident Day of Week','Filed Online', 'Incident Category','Resolution',
        'Police District','Analysis Neighborhood','Supervisor District','Latitude','Longitude']]
df
```

Out[67]:

	Incident Date	Incident Time	Incident Year	Incident Day of Week	Filed Online	Incident Category	Resolution	Police District	Analysis Neighborhood	Supervisor District	Latitude	Longitude
0	25-07-2021	00:00	2021	Sunday	True	Larceny Theft	Open or Active	Southern	NaN	NaN	NaN	NaN
1	28-06-2022	23:58	2022	Tuesday	False	Other Offenses	Open or Active	Out of SF	NaN	NaN	NaN	NaN
2	11-03-2022	10:30	2022	Friday	True	Lost Property	Open or Active	Central	NaN	NaN	NaN	NaN
3	15-05-2021	17:47	2021	Saturday	False	Recovered Vehicle	Open or Active	Out of SF	NaN	NaN	NaN	NaN
4	28-06-2022	17:22	2022	Tuesday	False	Recovered Vehicle	Open or Active	Out of SF	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
651573	18-10-2022	17:00	2022	Tuesday	False	Larceny Theft	Open or Active	Park	Golden Gate Park	5.0	37.766165	-122.454593
651574	17-10-2022	19:00	2022	Monday	False	Malicious Mischief	Open or Active	Central	Financial District/South Beach	3.0	37.791778	-122.405800
651575	10-06-2022	17:00	2022	Friday	False	Fraud	Open or Active	Taraval	Sunset/Parkside	7.0	37.732691	-122.476040
651576	18-10-2022	13:30	2022	Tuesday	False	Motor Vehicle Theft	Open or Active	Central	Financial District/South Beach	3.0	37.793664	-122.396390
651577	18-10-2022	17:30	2022	Tuesday	False	Suspicious Occ	Open or Active	Northern	Tenderloin	6.0	37.782894	-122.420817

651578 rows × 12 columns

As we are dealing with a huge amount of data, we must take caution of missing values. To avoid any problems like misrepresentation of data or skewness, we check the percent of missing values.



In [68]:
```
percent = ((df.isnull().sum()/df.count())*100).sort_values()
print(percent)
```

```
Incident Date             0.000000
Incident Time             0.000000
Incident Year             0.000000
Incident Day of Week      0.000000
Filed Online              0.000000
Resolution                0.000000
Police District           0.000000
Incident Category         0.085250
Supervisor District       5.578547
Latitude                  5.578547
Longitude                 5.578547
Analysis Neighborhood     5.599935
dtype: float64
```

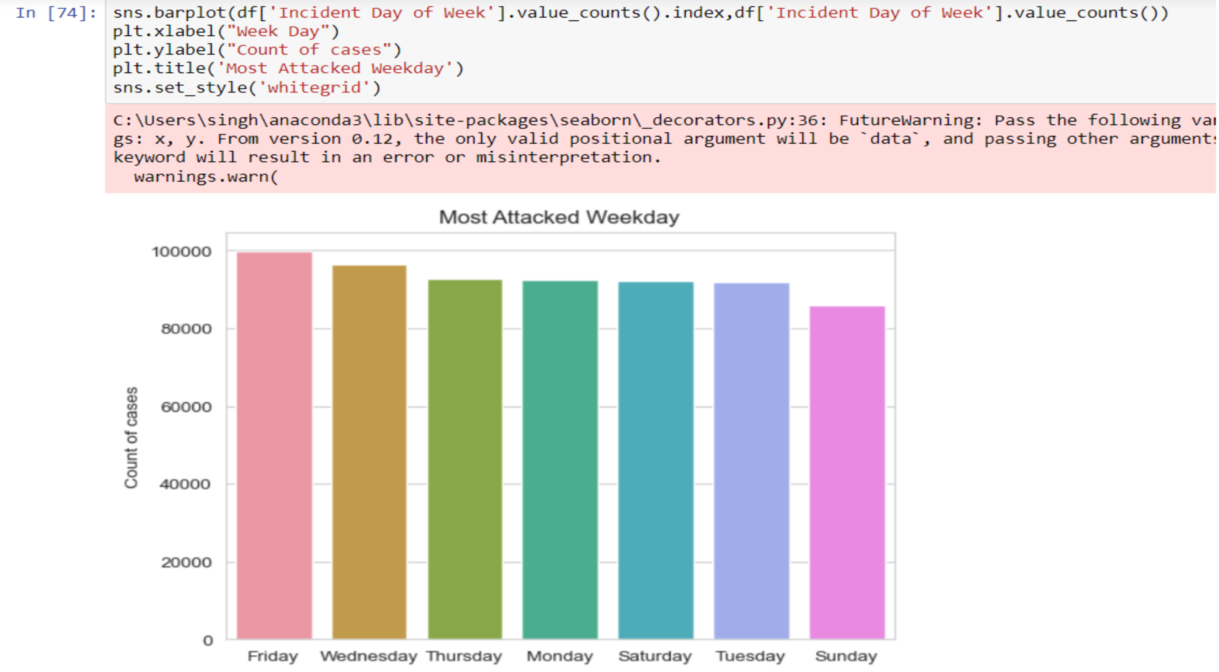
We can see that only 5% of data is missing in 4 columns out of 12 columns. Hence we will ignore it.

As missing data is found only in 4 columns and that too less than 5%, we can ignore and move further for analysis.

Analysis

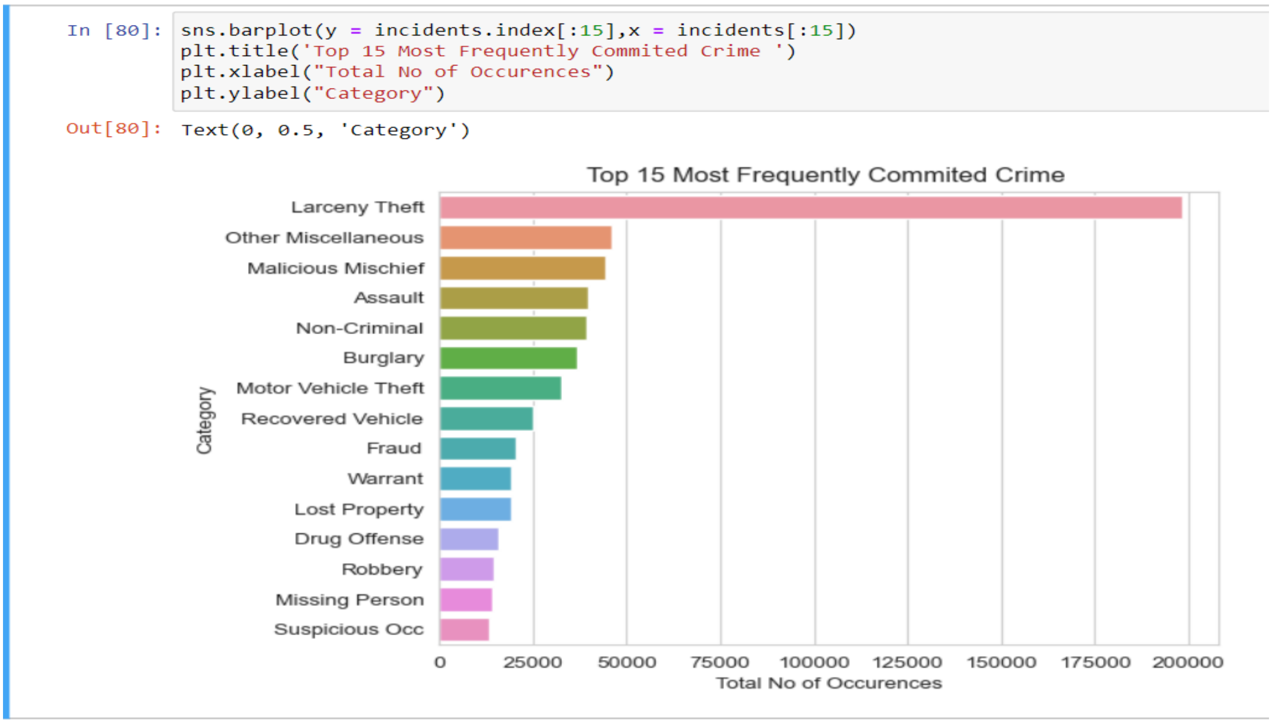
As there are many columns, we can check the plots of each variable. As python is open source, it supports a wide variety of libraries, which makes the job of data manipulation and visualization easy. Pandas, Matplotlib and Seaborn are the main libraries I have used for data exploration and visualization.

- Most Number of crimes are committed on which day?



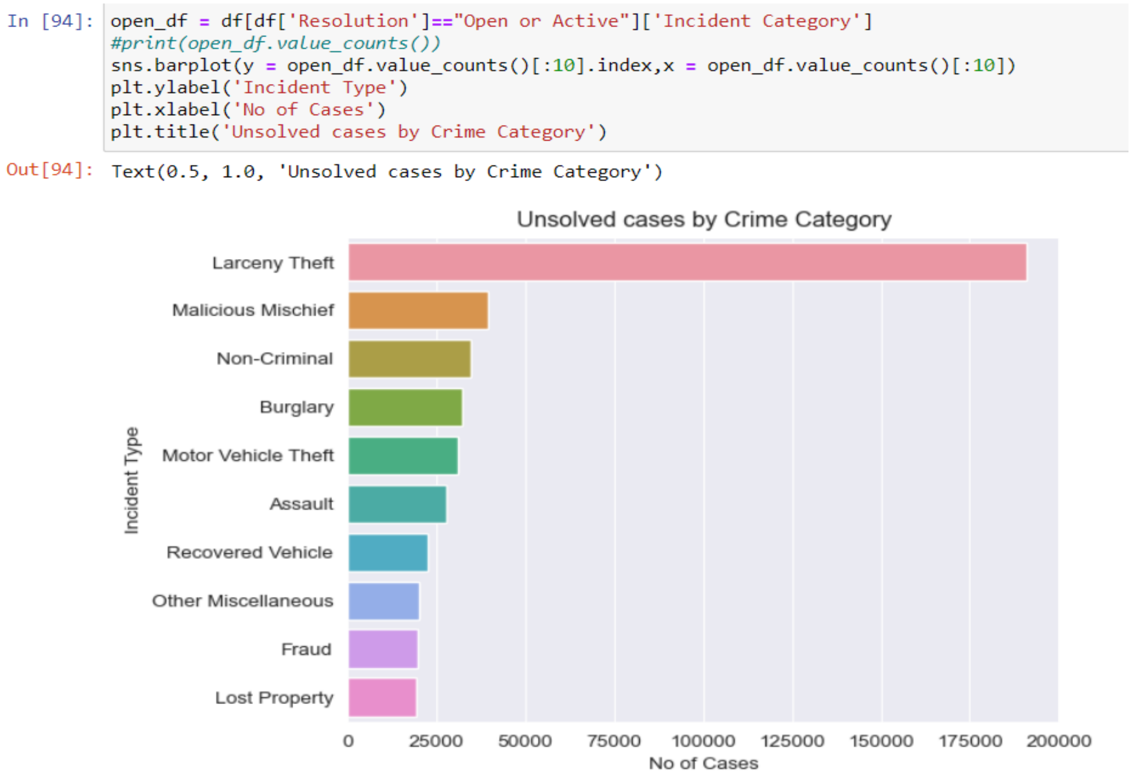
```
In [74]: sns.barplot(df['Incident Day of Week'].value_counts().index,df['Incident Day of Week'].value_counts())
         plt.xlabel("Week Day")
         plt.ylabel("Count of cases")
         plt.title('Most Attacked Weekday')
         sns.set_style('whitegrid')

         C:\Users\singh\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following var
         gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
         keyword will result in an error or misinterpretation.
           warnings.warn(
```

- Most committed crime?



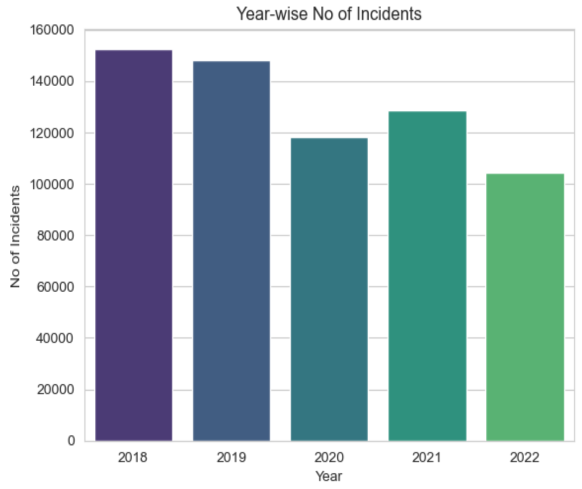
```
In [80]: sns.barplot(y = incidents.index[:15],x = incidents[:15])
         plt.title('Top 15 Most Frequently Commited Crime ')
         plt.xlabel("Total No of Occurences")
         plt.ylabel("Category")

Out[80]: Text(0, 0.5, 'Category')
```

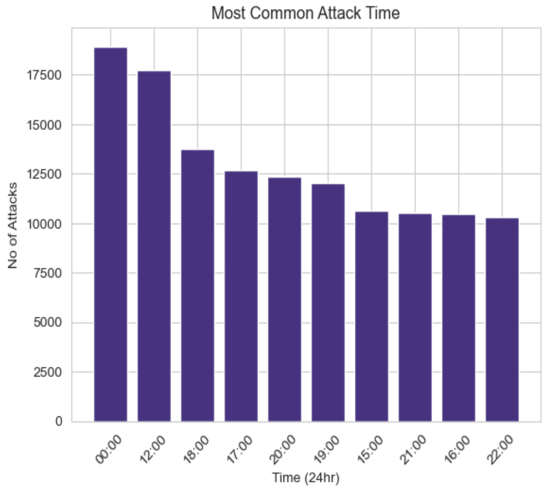
- Unsolved cases are the most under which category of crime?

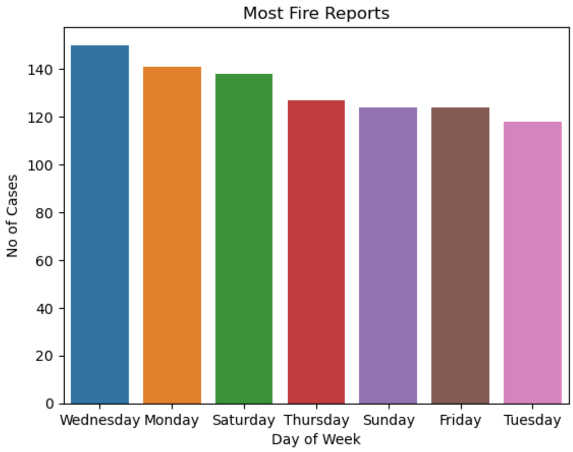


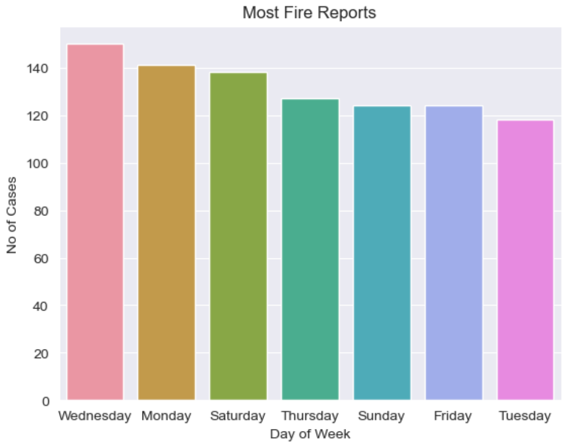
```
In [94]: open_df = df[df['Resolution']=="Open or Active"]['Incident Category']
         #print(open_df.value_counts())
         sns.barplot(y = open_df.value_counts()[:10].index,x = open_df.value_counts()[:10])
         plt.ylabel('Incident Type')
         plt.xlabel('No of Cases')
         plt.title('Unsolved cases by Crime Category')

Out[94]: Text(0.5, 1.0, 'Unsolved cases by Crime Category')
```

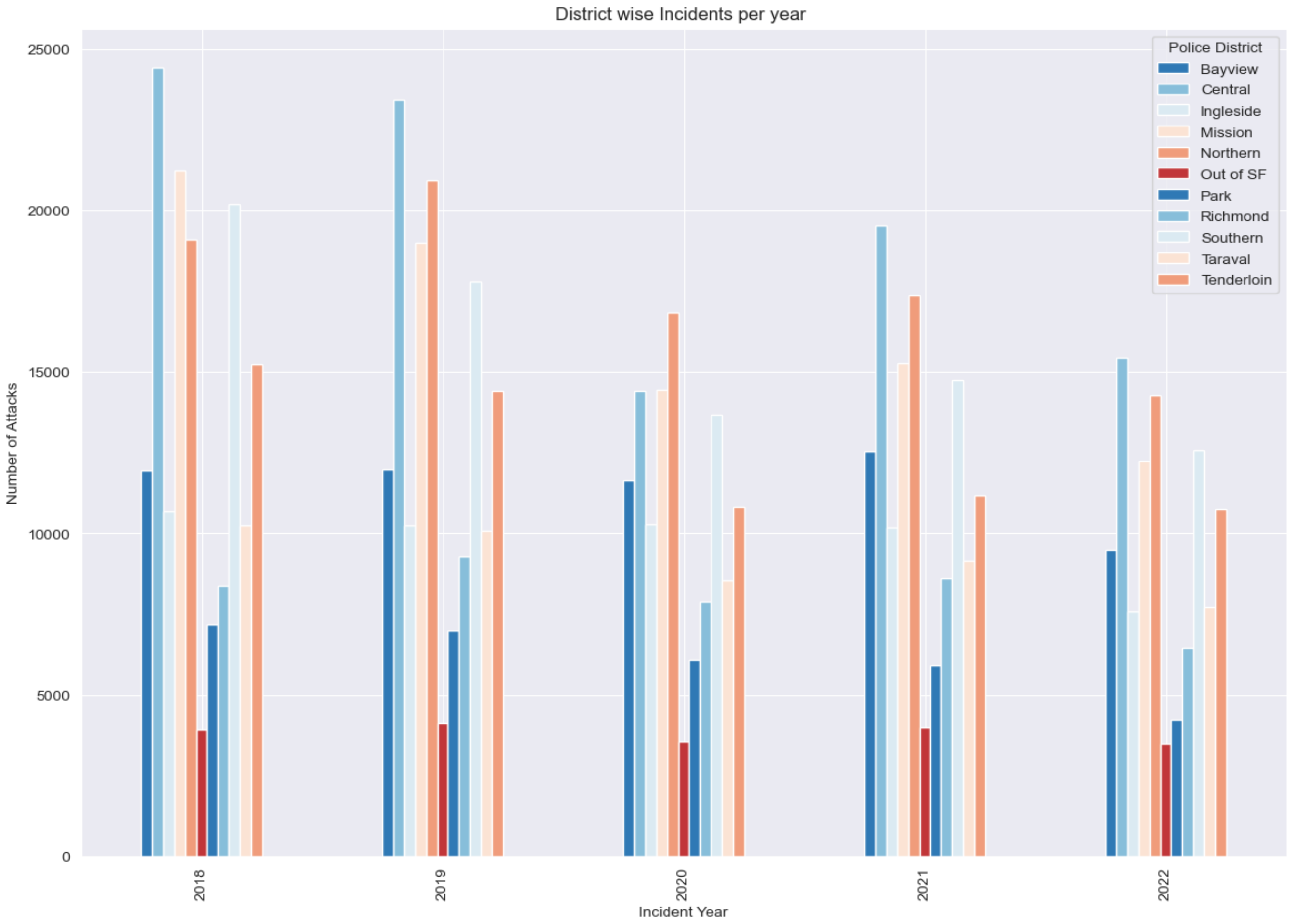
Some more Interesting visualizations






Most Fire Reports
No of Cases
0
20
40
60
80
100
120
140
Wednesday
Monday
Saturday
Thursday
Sunday
Friday
Tuesday
Day of Week


Most Fire Reports
No of Cases
0
20
40
60
80
100
120
140
Wednesday
Monday
Saturday
Thursday
Sunday
Friday
Tuesday
Day of Week

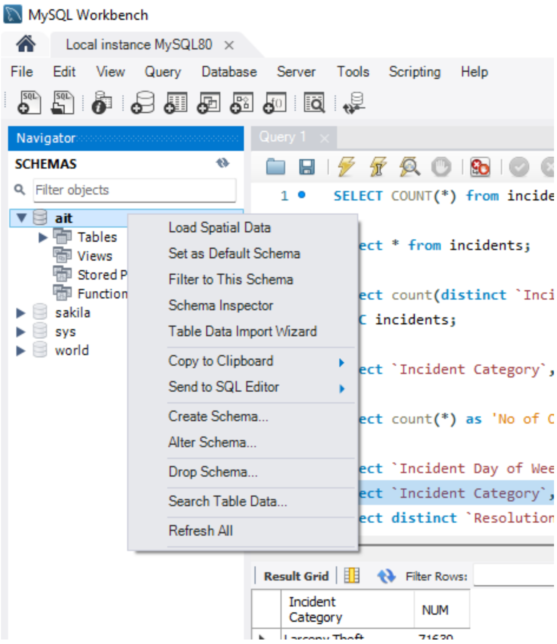

District wise Incidents per year
Number of Attacks
0
5000
10000
15000
20000
25000
2018
2019
2020
2021
2022
Incident Year
Police District
Bayview
Central
Ingleside
Mission
Northern
Out of SF
Park
Richmond
Southern
Taraval
Tenderloin

Analysis Using MySQL Workbench

MySQL is an open-source relational database management system (RDBMS). SQL stands for Structured Query Language and is mainly used to retrieve insightful data from a large database.

Data Importing

To import a database, we first need to create a schema which will hold our database rows and columns in a table. This schema can hold multiple tables and if needed we can merge multiple tables.



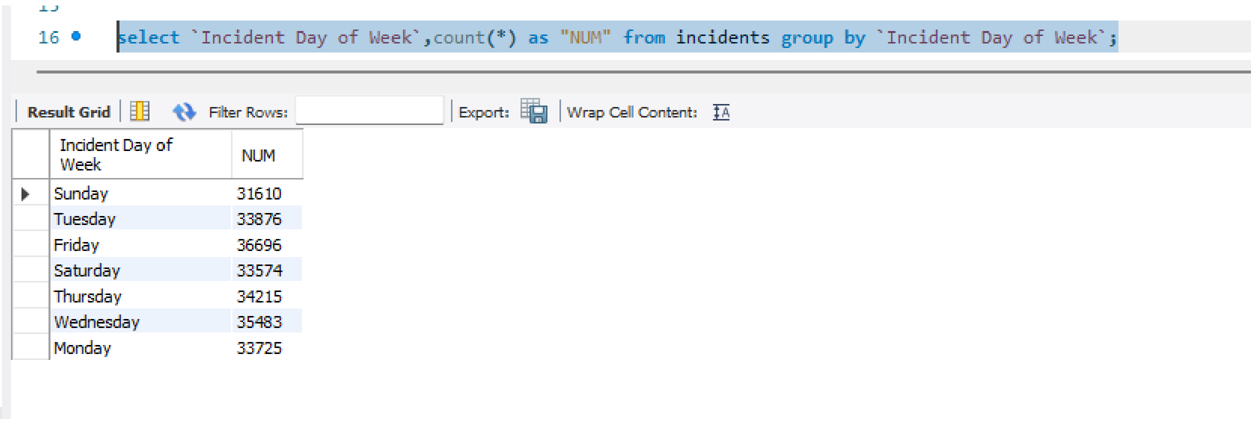
```
SELECT COUNT(*) from incide
ect * from incidents;
ect count(distinct `Inci
( incidents;
ect `Incident Category`,
ect count(*) as 'No of C
ect `Incident Day of Wee
ect `Incident Category`,
ect distinct `Resolution
```

Incident Category	NUM
Larceny Theft	71630

We can create a schema using the “Create Schema” option and select “Table data import wizard” from the options menu. Follow the steps to select the CSV file and make this Schema as default.Once we have our table loaded, we can run specific queries to retrieve valuable data.

- Most Number of crimes are committed on which day?

QUERY: select `Incident Day of Week`,count(*) as "NUM" from incidents group by `Incident Day of Week`;

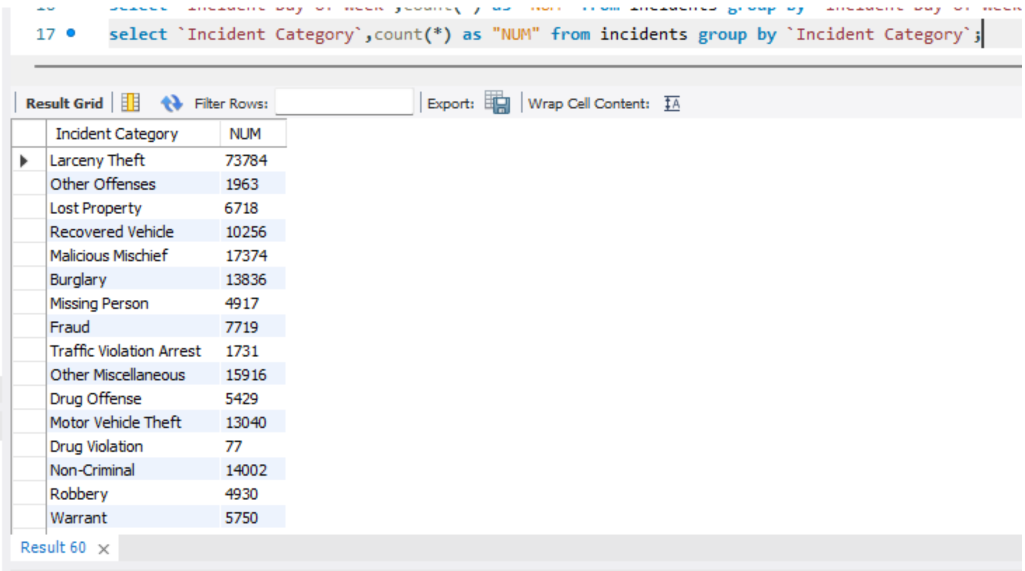


```
select `Incident Day of Week`,count(*) as "NUM" from incidents group by `Incident Day of Week`;
```

Incident Day of Week	NUM
Sunday	31610
Tuesday	33876
Friday	36696
Saturday	33574
Thursday	34215
Wednesday	35483
Monday	33725

- Most committed crime?

QUERY: select `Incident Category`,count(*) as "NUM" from incidents group by `Incident Category`;

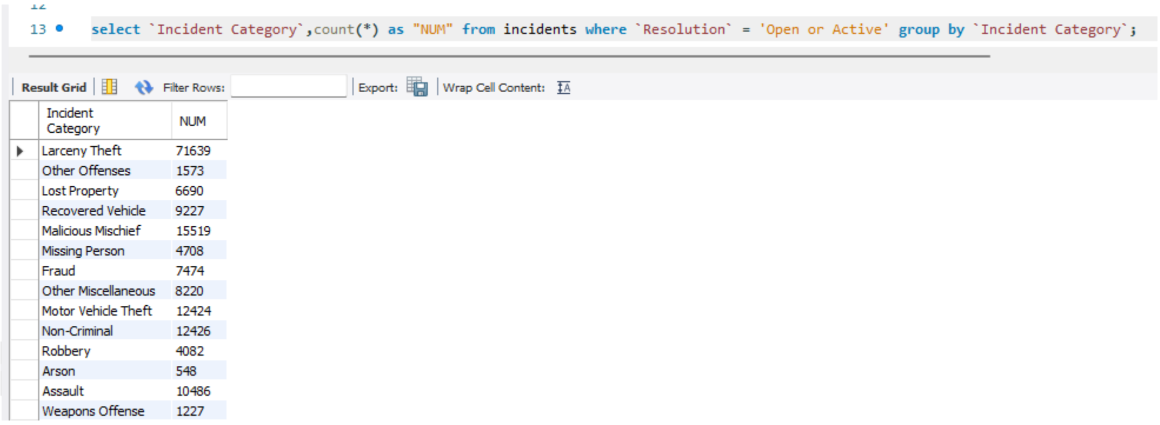


```
select `Incident Category`,count(*) as "NUM" from incidents group by `Incident Category`;
```

Incident Category	NUM
Larceny Theft	73784
Other Offenses	1963
Lost Property	6718
Recovered Vehicle	10256
Malicious Mischief	17374
Burglary	13836
Missing Person	4917
Fraud	7719
Traffic Violation Arrest	1731
Other Miscellaneous	15916
Drug Offense	5429
Motor Vehicle Theft	13040
Drug Violation	77
Non-Criminal	14002
Robbery	4930
Warrant	5750

- Unsolved cases are the most under which category of crime?

QUERY: select `Incident Category`,count(*) as "NUM" from incidents where `Resolution` = 'Open or Active' group by `Incident Category`;



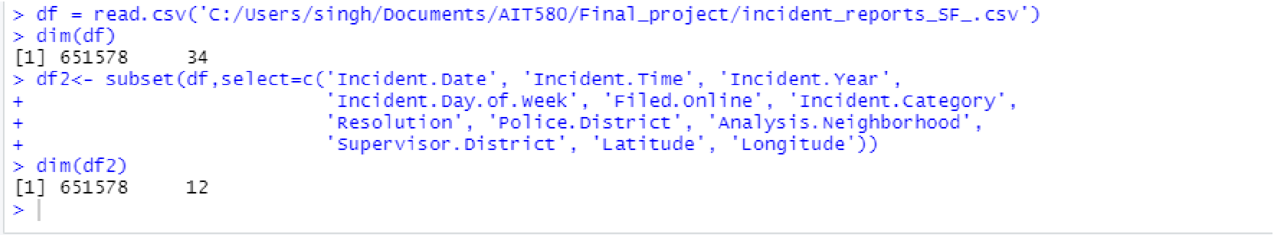
```
select `Incident Category`,count(*) as "NUM" from incidents where `Resolution` = 'Open or Active' group by `Incident Category`;
```

Incident Category	NUM
Larceny Theft	71639
Other Offenses	1573
Lost Property	6690
Recovered Vehicle	9227
Malicious Mischief	15519
Missing Person	4708
Fraud	7474
Other Miscellaneous	8220
Motor Vehicle Theft	12424
Non-Criminal	12426
Robbery	4082
Arson	548
Assault	10486
Weapons Offense	1227

Analysis Using R

R is a programming language used for statistical analysis and graphics. Like python, R also has many libraries through which you can perform various functions on data. Its mostly used to find out the statistics of the data. We can perform descriptive and inferential statistics through R.

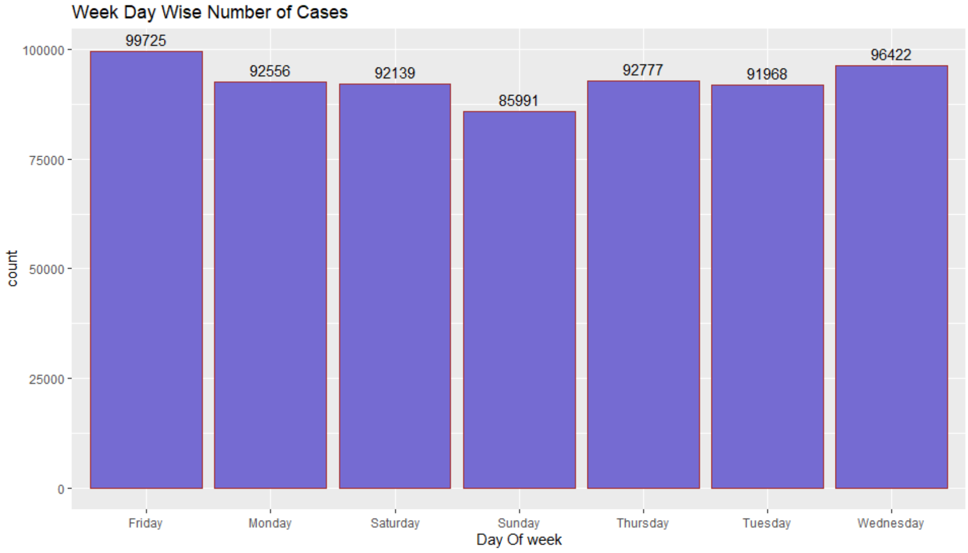
Import and selecting a subset of data



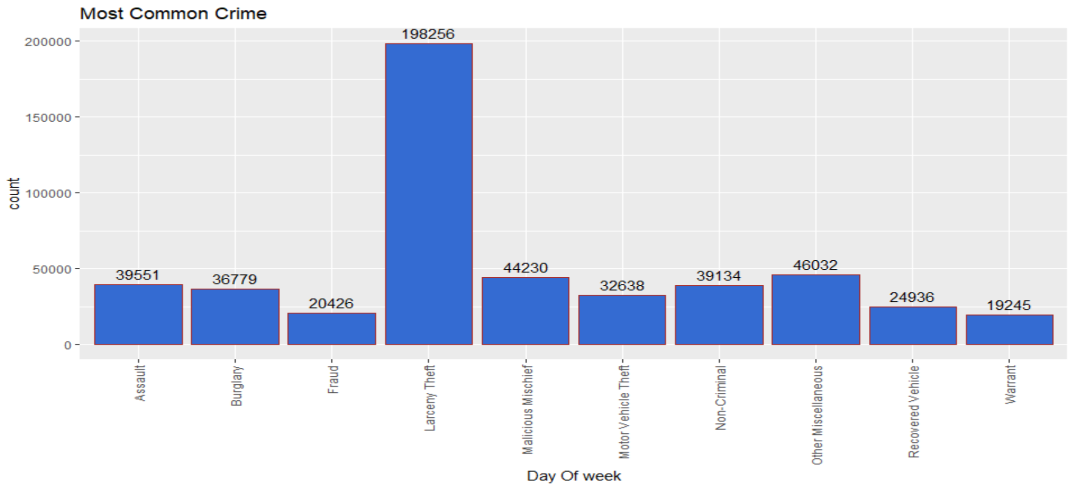
```
> df = read.csv('C:/Users/singh/Documents/AIT580/Final_project/incident_reports_SF_.csv')
> dim(df)
[1] 651578     34
> df2<- subset(df,select=c('Incident.Date', 'Incident.Time', 'Incident.Year',
+                          'Incident.Day.of.Week', 'Filed.Online', 'Incident.Category',
+                          'Resolution', 'Police.District', 'Analysis.Neighborhood',
+                          'Supervisor.District', 'Latitude', 'Longitude'))
> dim(df2)
[1] 651578     12
> 
```

Analysis

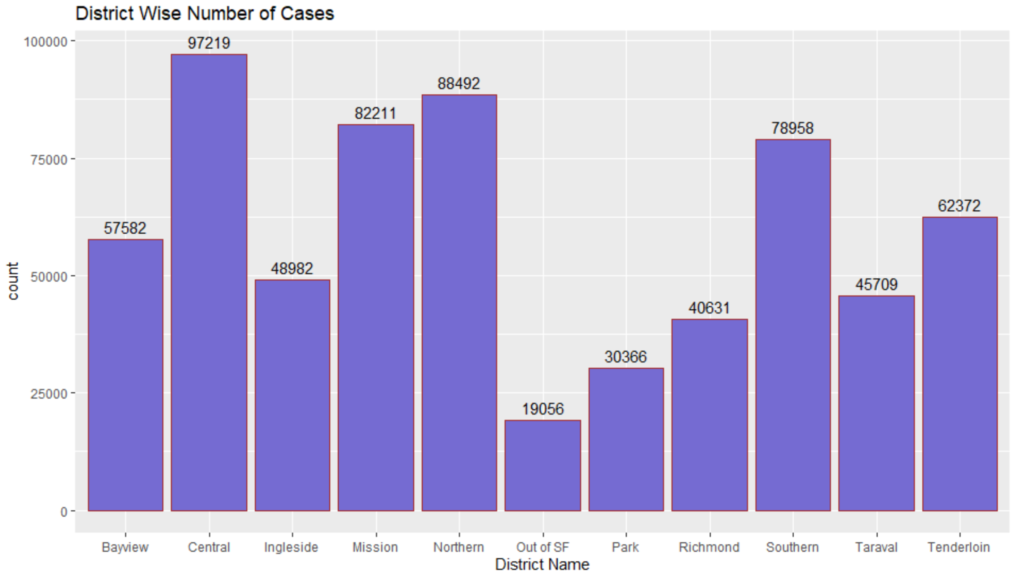
- Most Common Day for Committing Crimes?



- Most committed crime?



- Number of Cases in each Police District



The graphs are very clear in R and give us good understanding of how data is distributed.

Conclusion

In the process to find out the answers to our research question, we have found out many interesting patterns and trends. Larceny Theft is the most common crime reported in San Francisco, this means that a large part of society is not able to meet their daily expenses and are turning their face towards unlawful acts. The government and civil bodies with this information can launch various programs and seminars where people can get opportunities to meet their ends. Along with this, we can also present this to our citizens, make them aware of the common atrocities happening in their neighbourhood. Information like this can be lifesaving as people would then take precautionary measures before going out. The Police and Enforcement can be more watchful over certain areas

We have used technologies like R, Python and SQL to give us information about San Francisco City. In all the platforms we have found out that Friday is one of the most common days for crime. So Policing activity should be increased during this day and people also should take extra steps for their safety before going out.

The most open or unsolved cases are also for Theft, as this is one of most committed crimes, it is no surprise that due to it being the most unsolved crime, it makes it easier for criminals to commit as chances are they will be left unsolved and hence it is most committed. This analysis gives us a deeper understanding of the trends of crime and how each one affects the other. We can ask several questions and get answers which make sense. All effects have a cause, and this can be found out using simple data analysis.

References

- Police Department Incident Reports: 2018 to Present | DataSF | City and County of San Francisco (sfgov.org)
- https://wrlc-gm.primo.exlibrisgroup.com/permalink/01WRLC_GML/19u1omk/cdi_proquest_journals_2526965676
- https://wrlc-gm.primo.exlibrisgroup.com/permalink/01WRLC_GML/19u1omk/cdi_crossref_primary_10_1016_j_procs_2018_05_075
- https://wrlc-gm.primo.exlibrisgroup.com/permalink/01WRLC_GML/19u1omk/cdi_proquest_journals_1771058740
- https://library.gmu.edu/
- https://ohiostate.pressbooks.pub/engrtechcomm/chapter/strategies-for-conducting-research/
- https://dsc.gmu.edu/
- https://www3.nd.edu/~pkamat/pdf/graphs.pdf