

Introduction

This notebook has analysed crime in the city of Gwent from 2020-2022 and in the city of Metropoliton area(2021).In notebook, correlation between crimes will be discovered. In addition, we will review methods of forecasting and attempt to forecast number of crimes in 2020, 2021 and 2022.

Furthermore, I will look into crime seasonality, discover crime patterns and correlations between different types of crimes. In addition, we will determine the most dangerous and the safest location in the city of Gwent.Last but not the least, we will compare the data between the during lockdown and after lockdown.

Project Plan:

1. Data reporting for Gwent and metropoliton area.
2. Data analysis for Gwent and metropoliton area.
3. Data visualisation for Gwent and Metropoliton area.
4. Advanced analysis and visualisation for both area.

DATA REPORTING

▼ Import Libraries

```
# Libraries
import pandas as pd
from pandas import Series, DataFrame
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
# Visualization Libraries
import matplotlib
%matplotlib inline

figsize = (10,5)
import seaborn as sns
import datetime
import plotly.graph_objects as go
import plotly.express as px
import folium
from folium import plugins
%matplotlib inline

import folium
import calendar
from datetime import datetime
```

Exploratory Data Analysis

1. Data Collection
2. Data Cleaning
3. Univariate Analysis
4. Bivariate Analysis
5. Data Collection: Taking Gwent police Data(2020-2022) and metro politon police data(2022) from data.police.uk website.First and foremost, Combine the csv files with the help of python code.

Load and clean data

```
csv1 = pd.read_csv("/content/2020-01-gwent-street.csv")

csv2 = pd.read_csv("/content/2020-02-gwent-street.csv")

csv3 = pd.read_csv("/content/2020-03-gwent-street.csv")

csv4 = pd.read_csv("/content/2020-04-gwent-street.csv")

csv5 = pd.read_csv("/content/2020-05-gwent-street.csv")

csv6 = pd.read_csv("/content/2020-06-gwent-street.csv")

csv7 = pd.read_csv("/content/2020-07-gwent-street.csv")

csv8 = pd.read_csv("/content/2020-08-gwent-street.csv")

csv9 = pd.read_csv("/content/2020-09-gwent-street.csv")

csv10 = pd.read_csv("/content/2020-10-gwent-street.csv")

csv11 = pd.read_csv("/content/2020-11-gwent-street.csv")

csv12 = pd.read_csv("/content/2020-12-gwent-street.csv")

csv13 = pd.read_csv("/content/2021-01-gwent-street.csv")

csv14 = pd.read_csv("/content/2021-02-gwent-street.csv")

csv15 = pd.read_csv("/content/2021-03-gwent-street.csv")

csv16 = pd.read_csv("/content/2021-04-gwent-street.csv")

csv17 = pd.read_csv("/content/2021-05-gwent-street.csv")

csv18 = pd.read_csv("/content/2021-06-gwent-street.csv")

csv19 = pd.read_csv("/content/2021-07-gwent-street.csv")

csv20 = pd.read_csv("/content/2021-08-gwent-street.csv")

csv21 = pd.read_csv("/content/2021-09-gwent-street.csv")

csv22 = pd.read_csv("/content/2021-10-gwent-street.csv")

csv23 = pd.read_csv("/content/2021-11-gwent-street.csv")

csv24 = pd.read_csv("/content/2021-12-gwent-street.csv")

csv25 = pd.read_csv("/content/2022-01-gwent-street.csv")

csv26 = pd.read_csv("/content/2022-02-gwent-street.csv")
```

```

csv27 = pd.read_csv("/content/2022-03-gwent-street.csv")

csv28 = pd.read_csv("/content/2022-04-gwent-street.csv")

csv29 = pd.read_csv("/content/2022-05-gwent-street.csv")

csv30 = pd.read_csv("/content/2022-06-gwent-street.csv")

csv31 = pd.read_csv("/content/2022-07-gwent-street.csv")

csv32 = pd.read_csv("/content/2022-08-gwent-street.csv")

csv33 = pd.read_csv("/content/2022-09-gwent-street.csv")

csv34 = pd.read_csv("/content/2022-10-gwent-street.csv")

csv35 = pd.read_csv("/content/2021-01-metropolitan-street.csv")

csv36 = pd.read_csv("/content/2021-02-metropolitan-street.csv")

csv37 = pd.read_csv("/content/2021-03-metropolitan-street.csv")

csv38 = pd.read_csv("/content/2021-04-metropolitan-street.csv")

csv39 = pd.read_csv("/content/2021-05-metropolitan-street.csv")

csv40 = pd.read_csv("/content/2021-06-metropolitan-street.csv")

csv41 = pd.read_csv("/content/2021-07-metropolitan-street.csv")

csv42 = pd.read_csv("/content/2021-08-metropolitan-street.csv")

csv43= pd.read_csv("/content/2021-09-metropolitan-street.csv")

csv44 = pd.read_csv("/content/2021-10-metropolitan-street.csv")

csv45 = pd.read_csv("/content/2021-11-metropolitan-street.csv")

csv46 = pd.read_csv("/content/2021-12-metropolitan-street.csv")

```

After loading the data i will combine the files with the help of python. df.metro = final data for 2021 in metropolitan area. df.gwent = final data for (2020-2022) in metropolitan area. df.2021 = final data 2021 in Gwent area area. df.2022 = final data for 2022(after lockdown) in Gwent area(January-October) df.lockdown = final data for lockdown year 2021(January - october) in Gwent area.

```

df.metro = concate_data_metro = pd.concat([csv35,csv36,csv37,csv38,csv39,csv40,csv41,csv42,csv43,csv44,csv45,csv46], ignore_index=True)

df_gwent = concate_data_gwent = pd.concat([csv1,csv2,csv3,csv4,csv5,csv6,csv7,csv8,csv9,csv10,csv11,csv12,csv13,csv14,csv15,csv16,csv17], ignore_index=True)

df_2021 = concate_data_2021 = pd.concat([csv27,csv28,csv29,csv30,csv31,csv32,csv33,csv34,csv35,csv36,csv37,csv38,csv39,csv40,csv41,csv42,csv43,csv44,csv45,csv46], ignore_index=True)

df_2022 = concate_data_2022 = pd.concat([csv35,csv36,csv37,csv38,csv39,csv40,csv41,csv42,csv43,csv44,csv45,csv46], ignore_index=True)

```

		Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location
0		329241a52223c1ee7b99532d287f6e358a449f59419d6f...	2020-01	Gwent Police	Gwent Police	-3.193791	51.798553	On or near South Bank
1		e57731703cb1e3bfed11ca978c7b3e1ad2c6958867d2e0...	2020-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
2		2bfac884c84fc875f715ce298bde2784e810bf76ef8d79...	2020-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
3		c56c46311a3bb490cbe39b056106196cd9da1f05ce13b5...	2020-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
4		86159a37f6a53330ec3375b316c0c2a4a51523a874208f...	2020-01	Gwent Police	Gwent Police	-3.192296	51.799638	On or near Twyncynghord Road
...	
180007		fb6095b6b49da27fc9ff90efc07a37b337fbce5d10b0b...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180008		3f5c746de0bf3b89bbb1c3ac68887883e2a0c8b1ece8f5...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180009		60c24a531f497286f7ca14d4395d355bfc7de0a65e87f6...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180010		664891d55137c7f2a9a56a4e7612271c663d381ed6bd91...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180011		63c32fb5d32db845311d2711ec3e324b6bd24a1447cd53...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location

180012 rows × 12 columns

df_2021 = concat_data1 = pd.concat([csv13,csv14,csv15,csv16,csv17,csv18,csv19,csv20,csv21,csv22,csv23,csv24], ignore_index=True)

df_lockdown = concat_lock = pd.concat([csv13,csv14,csv15,csv16,csv17,csv18,csv19,csv20,csv21,csv22], ignore_index=True)
df_lockdown

		Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location
0		NaN	2021-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbur Terrac
1		NaN	2021-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbur Terrac
2		NaN	2021-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbur Terrac
3		NaN	2021-01	Gwent Police	Gwent Police	-3.192296	51.799638	On or ne Twyncynghord Roa
4		NaN	2021-01	Gwent Police	Gwent Police	-3.197188	51.794859	On or near Hes Gwyn Llŷ
...	
51233	81d2067c2a6b45fab4cc2582c08ff6c65c0175a0725921...		2021-10	Gwent Police	Gwent Police	NaN	NaN	No Locatio
51234	5e6037c273269a73af5717d1472e7da19f2fe136c473e7...		2021-10	Gwent Police	Gwent Police	NaN	NaN	No Locatio

There are 51630 rows and 12 columns in df_2022. There are 60830 rows and 12 columns in df_2021. There are 51238 rows and 12 columns in df_lockdown. There are 180012 rows and 12 columns in df_gwent. There are 51630 rows and 12 columns in df_metro.

```
df_2022 = concat_data3 = pd.concat([csv25,csv26,csv27,csv28,csv29,csv30,csv31,csv32,csv33,csv34], ignore_index=True)
```

```
df_2022.shape
```

```
(51630, 12)
```

```
df_2021.shape
```

```
(60863, 12)
```

```
df_lockdown.shape
```

```
(51238, 12)
```

```
df_gwent.shape
```

```
(180012, 12)
```

```
df.metro.shape
```

```
(1072876, 12)
```

Then I have extracted Month and year from the Month column

```
year = df_gwent["Month"].values
year = [my_str.split("-")[0] for my_str in year]
df_gwent["year"] = year
df_gwent['year'] = pd.DatetimeIndex(df_gwent['year']).year
df_gwent
```

		Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location
0		329241a52223c1ee7b99532d287f6e358a449f59419d6f...	2020-01	Gwent Police	Gwent Police	-3.193791	51.798553	On or near South Bank
1		e57731703cb1e3bfed11ca978c7b3e1ad2c6958867d2e0...	2020-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
2		2bfa884c84fc875f715ce298bde2784e810bf76ef8d79...	2020-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
3		c56c46311a3bb490cbe39b056106196cddda1f05ce13b5...	2020-01	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
4		86159a37f6a53330ec3375b316c0c2a4a51523a874208f...	2020-01	Gwent Police	Gwent Police	-3.192296	51.799638	On or near Twyncynghordy Road
...	
180007		fb6095b6b49da27fc9ff90efc07a37b337fbce5d10b0b...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180008		3f5c746de0bf3b89bbb1c3ac68887883e2a0c8b1ece8f5...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180009		60c24a531f497286f7ca14d4395d355bfc7de0a65e87f6...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180010		664891d55137c7f2a9a56a4e7612271c663d381ed6bd91...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location
180011		63c32fb5d32db845311d2711ec3e324b6bd24a1447cd53...	2022-10	Gwent Police	Gwent Police	NaN	NaN	No Location

180012 rows × 13 columns

```
year = df_2021["Month"].values
year = [my_str.split("-")[0] for my_str in year]
df_2021["year"] = year
df_2021['year'] = pd.DatetimeIndex(df_2021['year']).year
```

```
year = df_2022["Month"].values
year = [my_str.split("-")[0] for my_str in year]
df_2022["year"] = year
df_2022 ['year'] = pd.DatetimeIndex(df_2022['year']).year
```

```
year = df_lockdown["Month"].values
year = [my_str.split("-")[0] for my_str in year]
df_lockdown["year"] = year
df_lockdown['year'] = pd.DatetimeIndex(df_lockdown['year']).year
```

```
year = df_metro["Month"].values
year = [my_str.split("-")[0] for my_str in year]
df_metro["year"] = year
df_metro['year'] = pd.DatetimeIndex(df_metro['year']).year
```

```
df_lockdown['Month'] = pd.to_datetime(df_lockdown['Month']).dt.strftime('%B')
df_lockdown
```

		Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location
0			NaN January	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
1			NaN January	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
2			NaN January	Gwent Police	Gwent Police	-3.193205	51.798910	On or near Lansbury Terrace
3			NaN January	Gwent Police	Gwent Police	-3.192296	51.799638	On or near Twyncyngordy Road
4			NaN January	Gwent Police	Gwent Police	-3.197188	51.794859	On or near Heol Gwyn Lliw
...	
51233	81d2067c2a6b45fab4cc2582c08ff6c65c0175a0725921...		October	Gwent Police	Gwent Police	NaN	NaN	No Location
51234	5e6037c273269a73af5717d1472e7da19f2fe136c473e7...		October	Gwent Police	Gwent Police	NaN	NaN	No Location
51235	c1450d2df3437ef9e526a8af90594af637dad9015c8db4...		October	Gwent Police	Gwent Police	NaN	NaN	No Location
51236	a9cffaa4bccada376a337767acd6d42c627663608ce747...		October	Gwent Police	Gwent Police	NaN	NaN	No Location
51237	e09315e3eb8ae15813fa17cb04e8d14808e205590718b1...		October	Gwent Police	Gwent Police	NaN	NaN	No Location

```
df_gwent['Month'] = pd.to_datetime(df_gwent['Month']).dt.strftime('%B')
```

```
df_2021['Month'] = pd.to_datetime(df_2021['Month']).dt.strftime('%B')
```

```
df_2022['Month'] = pd.to_datetime(df_2022['Month']).dt.strftime('%B')
```

```
df_metro['Month'] = pd.to_datetime(df_metro['Month']).dt.strftime('%B')
```

Renaming columns

```
df_gwent= df_gwent.rename(columns={"Crime ID":"crime_id","Crime type":"crime_type","Reported by":"reported_by","Falls within":"area_group"});  
df_gwent
```

	crime_id	Month	reported_by	falls_within	Longitude	Latitude	Location
0	329241a52223c1ee7b99532d287f6e358a449f59419d6...	January	Gwent Police	Gwent Police	-3.193791	51.798553	On Southgate
1	e57731703cb1e3bfed11ca978c7b3e1ad2c6958867d2e0...	January	Gwent Police	Gwent Police	-3.193205	51.798910	On Land
2	2bfa884c84fc875f715ce298bde2784e810bf76ef8d79...	January	Gwent Police	Gwent Police	-3.193205	51.798910	On Land
3	c56c46311a3bb490cbe39b056106196cd9da1f05ce13b5...	January	Gwent Police	Gwent Police	-3.193205	51.798910	On Land
4	86159a37f6a53330ec3375b316c0c2a4a51523a874208f...	January	Gwent Police	Gwent Police	-3.192296	51.799638	Twynycyr
...
180007	fb6095b6b49da27fc9ff90efc07a37b337fbce5d10b0b...	October	Gwent Police	Gwent Police	NaN	NaN	No Location
180008	3f5c746de0bf3b89bbb1c3ac68887883e2a0c8b1ece8f5...	October	Gwent Police	Gwent Police	NaN	NaN	No Location
180009	60c24a531f497286f7ca14d4395d355bfc7de0a65e87f6...	October	Gwent Police	Gwent Police	NaN	NaN	No Location
180010	664891d55137c7f2a9a56a4e7612271c663d381ed6bd91...	October	Gwent Police	Gwent Police	NaN	NaN	No Location
180011	63c32fb5d32db845311d2711ec3e324b6bd24a1447cd5...	October	Gwent Police	Gwent Police	NaN	NaN	No Location

180012 rows × 13 columns

`gt_2021 <- gt_2021 %>% column_labels(cime_id = "CIME_ID", cime_type = "CIME_Type", esca_name = "ESCAName", base_outcome_category =`

— 5 —

```
Month          object
reported_by    object
falls_within   object
Longitude     float64
Latitude      float64
Location       object
LSOA code      object
LSOA.name      object
crime_type     object
last_outcome_category object
Context        float64
year           int64
dtype: object
```

```
df_2021.dtypes
```

crime_id	object
Month	object
Reported by	object
Falls within	object
Longitude	float64
Latitude	float64
Location	object
LSOA code	object
LSOA.name	object
crime_type	object
last_outcome_category	object
Context	float64
year	int64
dtype:	object

```
df_gwent.tail()
```

		crime_id	Month	reported_by	falls_within	Longitude	Latitude	Location	LSOA code
180007		fb6095b6b49da27fc9ff90efc07a37b337fbce5d10b0b...	October	Gwent Police	Gwent Police	NaN	NaN	Nc	Location
180008		3f5c746de0bf3b89bbb1c3ac68887883e2a0c8b1ece8f...	October	Gwent Police	Gwent Police	NaN	NaN	Nc	Location
180009		60c24a531f497286f7ca14d4395d355bfc7de0a65e87f...	October	Gwent Police	Gwent Police	NaN	NaN	Nc	Location
180010		664891d55137c7f2a9a56a4e7612271c663d381ed6bd91...	October	Gwent Police	Gwent Police	NaN	NaN	Nc	Location
180011		63c32fb5d32db845311d2711ec3e324b6bd24a1447cd53...	October	Gwent Police	Gwent Police	NaN	NaN	Nc	Location

```
df_lockdown.tail()
```

		crime_id	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code
51233		81d2067c2a6b45fab4cc2582c08ff6c65c0175a0725921...	October	Gwent Police	Gwent Police	NaN	NaN	No Location	NaN
51234		5e6037c273269a73af5717d1472e7da19f2fe136c473e7...	October	Gwent Police	Gwent Police	NaN	NaN	No Location	NaN
51235		c1450d2df3437ef9e526a8af90594af637dad9015c8db4...	October	Gwent Police	Gwent Police	NaN	NaN	No Location	NaN
51236		a9cffaa4bccada376a337767acd6d42c627663608ce747...	October	Gwent Police	Gwent Police	NaN	NaN	No Location	NaN
51237		e09315e3eb8ae15813fa17cb04e8d14808e205590718b1...	October	Gwent Police	Gwent Police	NaN	NaN	No Location	NaN

Missing Value

```
df_gwent.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180012 entries, 0 to 180011
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   crime_id        129007 non-null   object  
 1   Month            180012 non-null   object  
 2   reported_by     180012 non-null   object  
 3   falls_within    180012 non-null   object  
 4   Longitude        179333 non-null   float64 
 5   Latitude         179333 non-null   float64 
 6   Location          180012 non-null   object  
 7   LSOA code        179333 non-null   object  
 8   LSOA.name        179333 non-null   object  
 9   crime_type       180012 non-null   object  
 10  last_outcome_category 129007 non-null   object  
 11  Context           0 non-null      float64 
 12  year              180012 non-null   int64  
dtypes: float64(3), int64(1), object(9)
memory usage: 17.9+ MB
```

```
df_gwent.nunique()
```

Column	Non-Null Count
crime_id	128970
Month	12
reported_by	1
falls_within	1
Longitude	13722
Latitude	13473
Location	7001
LSOA code	1243
LSOA.name	1243
crime_type	14
last_outcome_category	12
Context	0
year	3
dtype:	int64

```
df_gwent.isnull().sum()
```

Column	Missing Values
crime_id	51005
Month	0
reported_by	0
falls_within	0
Longitude	679
Latitude	679
Location	0
LSOA code	679
LSOA.name	679
crime_type	0
last_outcome_category	51005
Context	180012
year	0
dtype:	int64

For Gwent data, crime_id and last_outcome_category column have 51005 missing values; Longitude, Latitude, LSOA code, LSOA.name have 679 missing values and Context has 180012 missing values.

```
df_2021.isnull().sum()
```

Column	Missing Values
crime_id	17574
Month	0
Reported by	0
Falls within	0
Longitude	228
Latitude	228
Location	0
LSOA code	228
LSOA.name	228
crime_type	0
last_outcome_category	17574

```
Context          60863
year              0
dtype: int64
```

```
df_2022.isnull().sum()
```

crime_id	9450
Month	0
Reported by	0
Falls within	0
Longitude	233
Latitude	233
Location	0
LSOA code	233
LSOA.name	233
crime_type	0
last_outcome_category	9450
Context	51630
year	0
dtype: int64	

```
df_lockdown.isnull().sum()
```

crime_id	15753
Month	0
Reported by	0
Falls within	0
Longitude	180
Latitude	180
Location	0
LSOA code	180
LSOA.name	180
crime_type	0
last_outcome_category	15753
Context	51238
year	0
dtype: int64	

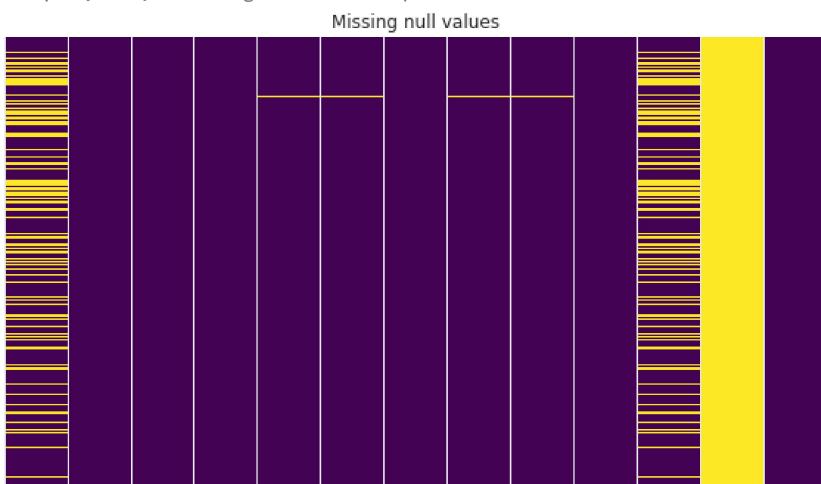
```
df.metro.isnull().sum()
```

crime_id	301494
Month	0
Reported by	0
Falls within	0
Longitude	15209
Latitude	15209
Location	0
LSOA code	15209
LSOA.name	15209
crime_type	0
last_outcome_category	301494
Context	1072876
year	0
dtype: int64	

Display missing values by heatmap

```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (10,6)
sns.heatmap(df_gwent.isnull(),yticklabels = False, cbar = False , cmap = 'viridis')
plt.title("Missing null values")
```

Text(0.5, 1.0, 'Missing null values')



It can easily be seen that Context column has large missing values that shows yellow color and Longitude, Latitude, LSOA code, LSOA.name have small missing values. Also, the remaining column has no missing values.

epi
hall
Lc
LS
SC
cri

Percentage of missing values per column

```
missing_percentages = (df_gwent.isna().sum().sort_values(ascending=False)/(len(df_gwent)))*100  
missing_percentages
```

```
Context          100.000000
crime_id        28.334222
last_outcome_category 28.334222
Longitude       0.377197
Latitude        0.377197
LSOA code       0.377197
LSOA.name       0.377197
Month           0.000000
reported_by    0.000000
falls_within   0.000000
Location        0.000000
crime_type     0.000000
year            0.000000
dtype: float64
```

100 percent data is missing for Context column and 28% data is missing for crime_id and last_outcome_category

```
missing_percentages = missing_percentages[missing_percentages != 0]
import matplotlib
matplotlib.rcParams['figure.figsize'] = (10,6)
missing_percentages.plot(kind = 'barh')
plt.title("Missing percentage of null values")
```

```
Text(0.5, 1.0, 'Missig percentage of null values')
```

Missig percentage of null values

LSOA.name						
LSOA.name						

Observation :

Dealing with the null values

The columns having highest percentage of null values are : Context, Crime ID, LSOA code, LSOA.name these are not much helpful . So, it can be dropped and added some features in the dataset after the Analysis.

```
data_gwent = df_gwent.drop(df_gwent.columns[[0, 2, 3, 7, 8, 10,11]],axis = 1)
data_gwent
```

	Month	Longitude	Latitude	Location	crime_type	year
0	January	-3.193791	51.798553	On or near South Bank	Public order	2020
1	January	-3.193205	51.798910	On or near Lansbury Terrace	Public order	2020
2	January	-3.193205	51.798910	On or near Lansbury Terrace	Vehicle crime	2020
3	January	-3.193205	51.798910	On or near Lansbury Terrace	Violence and sexual offences	2020
4	January	-3.192296	51.799638	On or near Twyncynghordy Road	Violence and sexual offences	2020
...
180007	October	NaN	NaN	No Location	Violence and sexual offences	2022
180008	October	NaN	NaN	No Location	Violence and sexual offences	2022
180009	October	NaN	NaN	No Location	Violence and sexual offences	2022
180010	October	NaN	NaN	No Location	Violence and sexual offences	2022
180011	October	NaN	NaN	No Location	Violence and sexual offences	2022

180012 rows × 6 columns

```
data_2021 = df_2021.drop(df_2021.columns[[0, 2, 3, 7, 8, 10,11]],axis = 1)
data_2021
```

	Month	Longitude	Latitude	Location	crime_type	year
0	January	-3.193205	51.798910	On or near Lansbury Terrace	Anti-social behaviour	2021
1	January	-3.193205	51.798910	On or near Lansbury Terrace	Anti-social behaviour	2021
2	January	-3.193205	51.798910	On or near Lansbury Terrace	Anti-social behaviour	2021
3	January	-3.192296	51.799638	On or near Twyncynghordy Road	Anti-social behaviour	2021
4	January	-3.197188	51.794859	On or near Heol Gwyn Lliw	Anti-social behaviour	2021
...
60858	December	NaN	NaN	No Location	Violence and sexual offences	2021
60859	December	NaN	NaN	No Location	Violence and sexual offences	2021
60860	December	NaN	NaN	No Location	Violence and sexual offences	2021
60861	December	NaN	NaN	No Location	Violence and sexual offences	2021
60862	December	NaN	NaN	No Location	Violence and sexual offences	2021

60863 rows × 6 columns

```
Data_2021= data_2021.fillna(0)
Data_2021
```

	Month	Longitude	Latitude	Location	crime_type	year
0	January	-3.193205	51.798910	On or near Lansbury Terrace	Anti-social behaviour	2021
1	January	-3.193205	51.798910	On or near Lansbury Terrace	Anti-social behaviour	2021
2	January	-3.193205	51.798910	On or near Lansbury Terrace	Anti-social behaviour	2021
3	January	-3.192296	51.799638	On or near Twyncynghordy Road	Anti-social behaviour	2021
4	January	-3.197188	51.794859	On or near Heol Gwyn Lliw	Anti-social behaviour	2021
...
60858	December	0.000000	0.000000	No Location	Violence and sexual offences	2021
60859	December	0.000000	0.000000	No Location	Violence and sexual offences	2021
60860	December	0.000000	0.000000	No Location	Violence and sexual offences	2021
60861	December	0.000000	0.000000	No Location	Violence and sexual offences	2021
60862	December	0.000000	0.000000	No Location	Violence and sexual offences	2021

```
data_2022 = df_2022.drop(df_2022.columns[[0, 2, 3, 7, 8, 10,11]],axis = 1)
```

```
data_lockdown = df_lockdown.drop(df_lockdown.columns[[0, 2, 3, 7, 8, 10, 11]],axis = 1)
```

```
data_metro = df_metro.drop(df_metro.columns[[0, 7, 8, 10,11]],axis = 1)
```

```
df_Lockdown = data_lockdown.fillna(0)
```

```
data_Gw=data_gwent.fillna(0)
data_Gw
```

	Month	Longitude	Latitude	Location	crime_type	year
0	January	-3.193791	51.798553	On or near South Bank	Public order	2020
1	January	-3.193205	51.798910	On or near Lansbury Terrace	Public order	2020
2	January	-3.193205	51.798910	On or near Lansbury Terrace	Vehicle crime	2020
3	January	-3.193205	51.798910	On or near Lansbury Terrace	Violence and sexual offences	2020
4	January	-3.192296	51.799638	On or near Twyncynghordy Road	Violence and sexual offences	2020
...
180007	October	0.000000	0.000000	No Location	Violence and sexual offences	2022
180008	October	0.000000	0.000000	No Location	Violence and sexual offences	2022
180009	October	0.000000	0.000000	No Location	Violence and sexual offences	2022
180010	October	0.000000	0.000000	No Location	Violence and sexual offences	2022
180011	October	0.000000	0.000000	No Location	Violence and sexual offences	2022

180012 rows × 6 columns

After dropping the columns the data has 180012 rows and 6 columns

```
Data_2022=data_2022.fillna(0)
Data_2022
```

	Month	Longitude	Latitude	Location	crime_type	year
0	January	-3.192296	51.799638	On or near Twyncynghordy Road	Anti-social behaviour	2022
1	January	-3.193440	51.800643	On or near Big Lane	Anti-social behaviour	2022
2	January	-3.199344	51.794684	On or near Llandaff Road	Burglary	2022
3	January	-3.192296	51.799638	On or near Twyncynghordy Road	Public order	2022
4	January	-3.193205	51.798910	On or near Lansbury Terrace	Vehicle crime	2022
...
51625	October	0.000000	0.000000	No Location	Violence and sexual offences	2022
51626	October	0.000000	0.000000	No Location	Violence and sexual offences	2022

```
Data_metro = data_metro.fillna(0)
```

```
Data_metro
```

	Month	Reported by	Falls within	Longitude	Latitude	Location
0	January	Metropolitan Police Service	Metropolitan Police Service	0.880386	51.219190	On or near Buck Street
1	January	Metropolitan Police Service	Metropolitan Police Service	0.141143	51.590873	On or near Furze Farm Close
2	January	Metropolitan Police Service	Metropolitan Police Service	0.134947	51.588063	On or near Mead Grove
3	January	Metropolitan Police Service	Metropolitan Police Service	0.134947	51.588063	On or near Mead Grove
4	January	Metropolitan Police Service	Metropolitan Police Service	0.134947	51.588063	On or near Mead Grove
...
1072871	December	Metropolitan Police Service	Metropolitan Police Service	0.000000	0.000000	No Location
1072872	December	Metropolitan Police Service	Metropolitan Police Service	0.000000	0.000000	No Location
1072873	December	Metropolitan Police Service	Metropolitan Police Service	0.000000	0.000000	No Location
1072874	December	Metropolitan Police Service	Metropolitan Police Service	0.000000	0.000000	No Location
1072875	December	Metropolitan Police Service	Metropolitan Police Service	0.000000	0.000000	No Location

1072876 rows × 8 columns

Data Analysis

```
data_Gw.describe()
```

Longitude	Latitude	year
-----------	----------	------

It can easily be observed that Only Longitude and Latitude have numerical figures, and thus, the mean value of Longitude and Latitude are -3.058 and 51.463 respectively for Gwent area.

```
-3.058000 51.463000 2021.000000
```

```
Data_metro.describe()
```

	Longitude	Latitude	year
count	1.072876e+06	1.072876e+06	1072876.0
mean	-1.172884e-01	5.078225e+01	2021.0
std	1.603135e-01	6.090191e+00	0.0
min	-5.540048e+00	0.000000e+00	2021.0
25%	-2.003470e-01	5.146708e+01	2021.0
50%	-1.084545e-01	5.151533e+01	2021.0
75%	-2.236300e-02	5.155560e+01	2021.0
max	1.737687e+00	5.571940e+01	2021.0

It can easily be shown that Only Longitude and Latitude have numerical figures, and thus, the mean value of Longitude and Latitude are -1.173 and 5.078225e+01 respectively for metropoliton area.

```
data_Gw['crime_type'].value_counts()
```

Violence and sexual offences	60536
Anti-social behaviour	51005
Public order	24826
Other theft	9962
Vehicle crime	6831
Shoplifting	6689
Burglary	5743
Drugs	4381
Other crime	4078
Criminal damage and arson	2956
Bicycle theft	825
Possession of weapons	793
Robbery	790
Theft from the person	597
Name: crime_type, dtype: int64	

```
data_Gw['crime_type'].value_counts()/(len(data_Gw))*100
```

Violence and sexual offences	33.628869
Anti-social behaviour	28.334222
Public order	13.791303
Other theft	5.534076
Vehicle crime	3.794747
Shoplifting	3.715863
Burglary	3.190343
Drugs	2.433727
Other crime	2.265405
Criminal damage and arson	1.642113
Bicycle theft	0.458303
Possession of weapons	0.440526
Robbery	0.438860
Theft from the person	0.331645
Name: crime_type, dtype: float64	

Most crime_types are of Violence and sexual offences which is almost 33% of total crime_types as well as the least are of Theft from the person for gwent area.

```
Data_2021.groupby(['year','crime_type']).size()
```

year	crime_type	
2021	Anti-social behaviour	17574
	Bicycle theft	315
	Burglary	1915
	Criminal damage and arson	947
	Drugs	1494
	Other crime	1403
	Other theft	3399
	Possession of weapons	260
	Public order	8112
	Robbery	265
	Shoplifting	2179
	Theft from the person	173
	Vehicle crime	2274
	Violence and sexual offences	20553

dtype: int64

```
Data_metro.groupby(['year','crime_type']).size()
```

year	crime_type	
2021	Anti-social behaviour	301494
	Bicycle theft	21000
	Burglary	53388
	Criminal damage and arson	52684
	Drugs	42819
	Other crime	10229
	Other theft	89488
	Possession of weapons	4855
	Public order	58714
	Robbery	22452
	Shoplifting	33047
	Theft from the person	37591
	Vehicle crime	100188
	Violence and sexual offences	244927

dtype: int64

```
crimes_percentages_2021 = Data_2021['crime_type'].value_counts()/(len(Data_2021))*100
crimes_percentages_2021
```

Violence and sexual offences	33.769285	
Anti-social behaviour	28.874686	
Public order	13.328295	
Other theft	5.584674	
Vehicle crime	3.736260	
Shoplifting	3.580172	
Burglary	3.146411	
Drugs	2.454693	
Other crime	2.305177	
Criminal damage and arson	1.555954	
Bicycle theft	0.517556	
Robbery	0.435404	
Possession of weapons	0.427189	
Theft from the person	0.284245	

Name: crime_type, dtype: float64

```
crimes_percentages.metro = Data.metro['crime_type'].value_counts()/(len(Data.metro))*100
crimes_percentages.metro
```

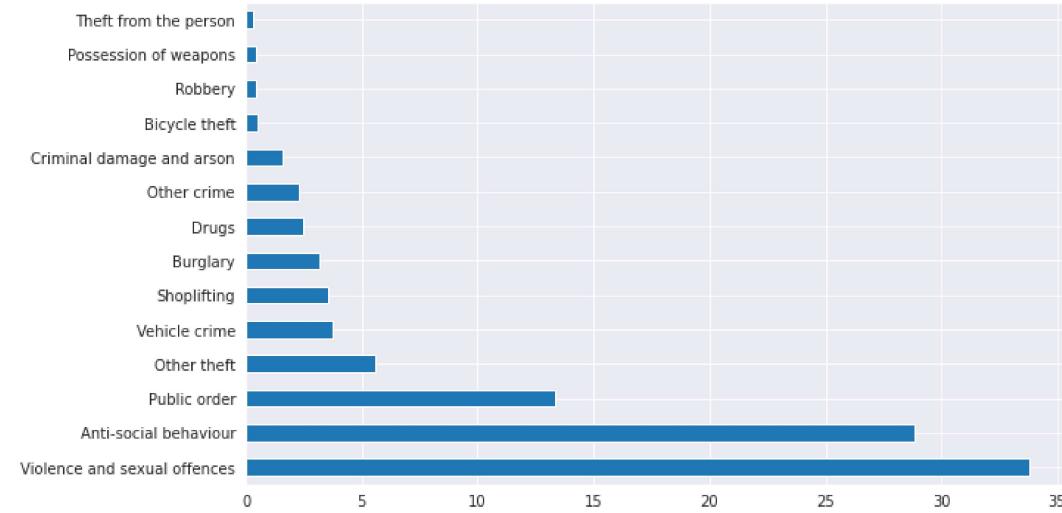
Anti-social behaviour	28.101477	
Violence and sexual offences	22.829013	
Vehicle crime	9.338265	
Other theft	8.340945	
Public order	5.472580	
Burglary	4.976158	
Criminal damage and arson	4.910540	
Drugs	3.991048	
Theft from the person	3.503760	
Shoplifting	3.080225	
Robbery	2.092693	
Bicycle theft	1.957356	

```
Other crime           0.953419
Possession of weapons 0.452522
Name: crime_type, dtype: float64
```

```
crimes_percentages_2021 = crimes_percentages_2021[crimes_percentages_2021 != 0]
import matplotlib
matplotlib.rcParams['figure.figsize'] = (10,6)
crimes_percentages_2021.plot(kind = 'barh')
plt.title("crimes_percentages_2021")
```

Text(0.5, 1.0, 'crimes_percentages_2021')

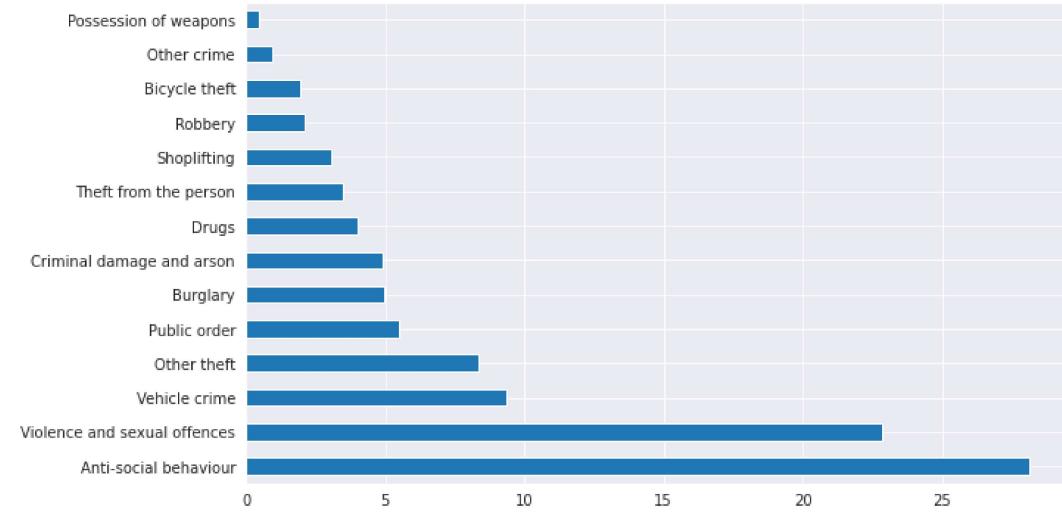
crimes_percentages_2021



```
crimes_percentages_metro = crimes_percentages_metro[crimes_percentages_metro != 0]
import matplotlib
matplotlib.rcParams['figure.figsize'] = (10,6)
crimes_percentages_metro.plot(kind = 'barh')
plt.title("crimes_percentages_metro")
```

Text(0.5, 1.0, 'crimes_percentages_metro')

crimes_percentages_metro



However, the scenario is different for metropolitan area. Here, 22% of crimes were reported by Violence and sexual offences that was 2nd highest. But for gwent it was 33% that was highest.

```
Data_2022.groupby(['year', 'crime_type']).size()
```

year	crime_type	size
2022	Anti-social behaviour	9450

```
Bicycle theft           223
Burglary                1600
Criminal damage and arson    707
Drugs                   1104
Other crime              1225
Other theft               3199
Possession of weapons      278
Public order             8707
Robbery                  273
Shoplifting              2070
Theft from the person     205
Vehicle crime             2005
Violence and sexual offences 20584
dtype: int64
```

```
df_Lockdown.groupby(['year', 'crime_type']).size()
```

```
year  crime_type
2021  Anti-social behaviour      15753
      Bicycle theft                 286
      Burglary                      1563
      Criminal damage and arson     773
      Drugs                          1286
      Other crime                   1137
      Other theft                   2758
      Possession of weapons        205
      Public order                  6699
      Robbery                       228
      Shoplifting                   1790
      Theft from the person         135
      Vehicle crime                1876
      Violence and sexual offences 16749
dtype: int64
```

After lockdown 20584 crimes were reported by Violence and sexual offences in Gwent area but during lockdown 16749 crimes were reported by that offence.

```
data_Gw.dtypes
```

```
Month      object
Longitude   float64
Latitude   float64
Location    object
crime_type  object
year       int64
dtype: object
```

```
data_Gw.count()
```

```
Month      180012
Longitude  180012
Latitude   180012
Location    180012
crime_type  180012
year       180012
dtype: int64
```

```
Data_2021.count()
```

```
Month      60863
Longitude  60863
Latitude   60863
Location    60863
crime_type  60863
year       60863
dtype: int64
```

```
data_Gw.isnull().sum()
```

```
Month      0
Longitude  0
Latitude   0
Location   0
crime_type 0
year       0
dtype: int64
```

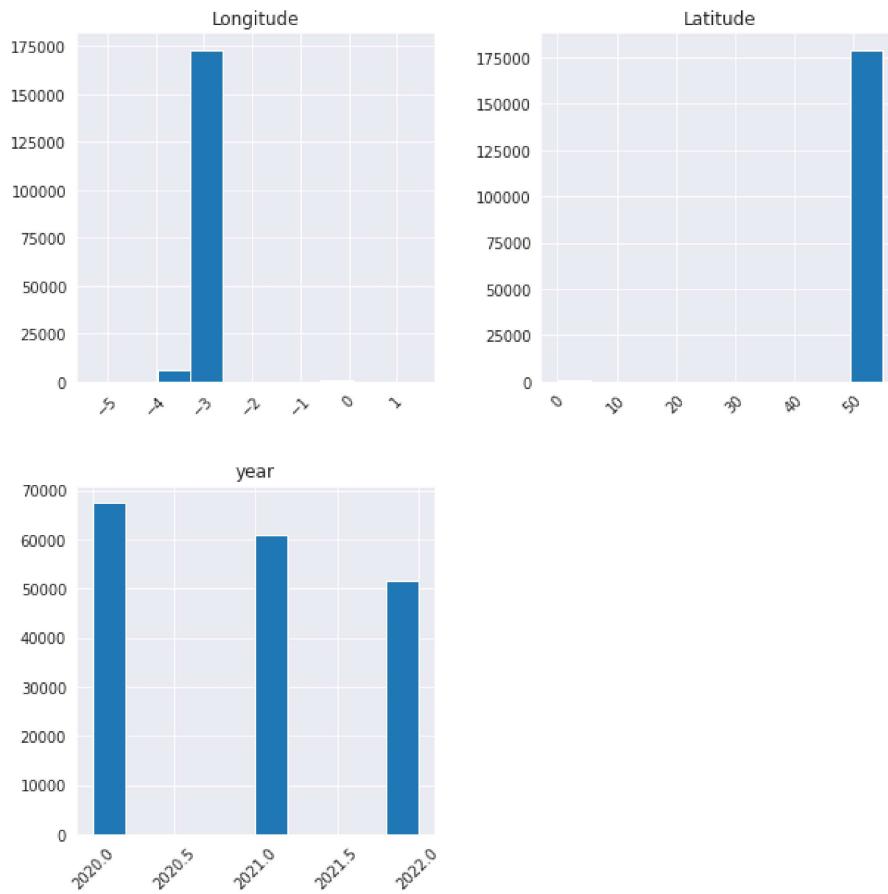
```
data_Gw.describe(include='object')
```

	Month	Location	crime_type
count	180012	180012	180012
unique	12	7001	14
top	April	On or near Parking Area	Violence and sexual offences
freq	19265	6276	60536

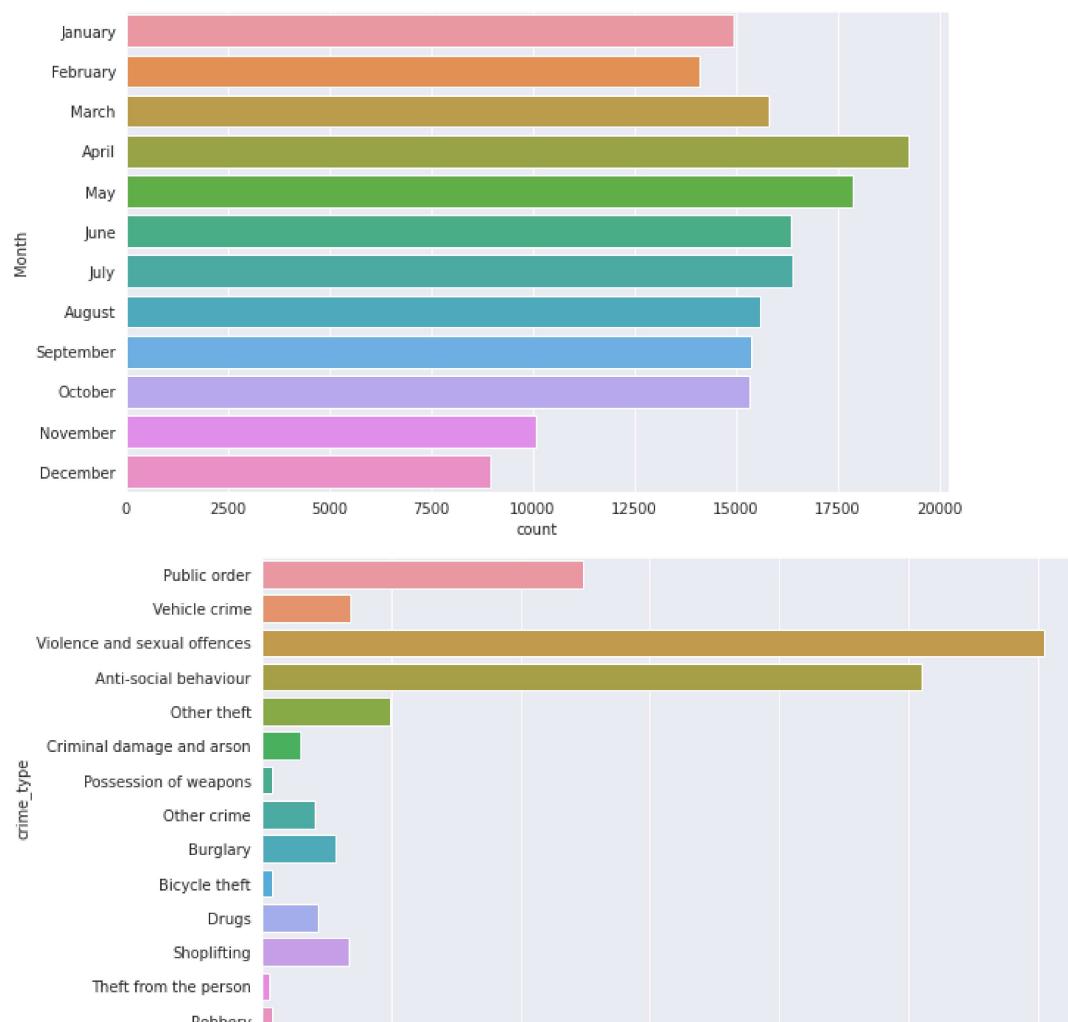
Univariate Analysis

The figures show for numerical variable.

```
data_Gw.hist(figsize=(10,10), xrot=45)
plt.show()
```



```
for column in data_Gw.select_dtypes(include='object'):
    if data_Gw[column].nunique() < 20:
        sns.countplot(y=column, data=data_Gw)
        plt.show()
```



The figures from the above are for categorical variable.

```
for column in data_Gw.select_dtypes(include='object'):
    if data_Gw[column].nunique() < 20:
        display(data_Gw.groupby(column).mean())
```

Month	Longitude	Latitude	year
April	-3.058630	51.491458	2020.835038
August	-3.057783	51.439586	2021.006093
December	-3.053955	51.390970	2020.508103
February	-3.060391	51.471605	2020.997946
January	-3.060166	51.493795	2021.017261
July	-3.061454	51.478671	2020.975826
June	-3.053123	51.421775	2020.950391
March	-3.062248	51.483572	2021.015959
May	-3.056653	51.484972	2020.857951
November	-3.055260	51.443606	2020.504670
October	-3.059602	51.484594	2021.004433
September	-3.056216	51.428234	2020.985355

year	Longitude	Latitude	year
2020			
Anti-social behaviour			23981
Bicycle theft			287
Burglary			2228
Criminal damage and arson			1302
Drugs			1783
Other crime			1450
Other theft			3364
Possession of weapons			255
Public order			8007
Robbery			252
Shoplifting			2440
Theft from the person			219
Vehicle crime			2552
Violence and sexual offences			19399
2021			
Anti-social behaviour			17574
Bicycle theft			315
Burglary			1915
Criminal damage and arson			947
Drugs			1494
Other crime			1403
Other theft			3399
Possession of weapons			260
Public order			8112
Robbery			265
Shoplifting			2179
Theft from the person			173
Vehicle crime			2274
Violence and sexual offences			20553
2022			
Anti-social behaviour			9450
Bicycle theft			223
Burglary			1600
Criminal damage and arson			707
Drugs			1104
Other crime			1225
Other theft			3199
Possession of weapons			278
Public order			8707
Robbery			273
Shoplifting			2070
Theft from the person			205
Vehicle crime			2005
Violence and sexual offences			20584

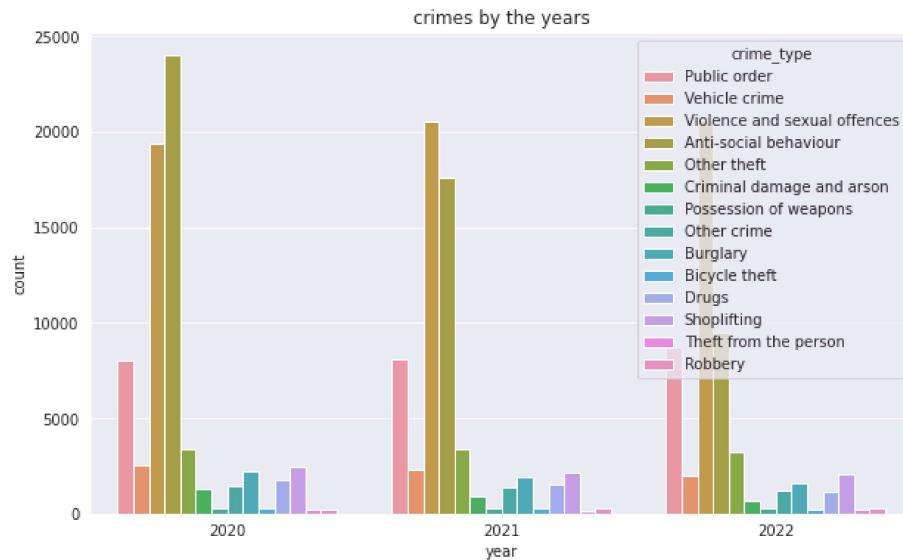
dtype: int64

```
[data_Gw.groupby(['year', 'crime_type']).size() / (len(data_Gw)) * 100][:50]
```

year	crime_type	
2020	Anti-social behaviour	13.321890
	Bicycle theft	0.159434
	Burglary	1.237695
	Criminal damage and arson	0.723285
	Drugs	0.990490
	Other crime	0.805502
	Other theft	1.868764
	Possession of weapons	0.141657
	Public order	4.448037
	Robbery	0.139991
	Shoplifting	1.355465
	Theft from the person	0.121659
	Vehicle crime	1.417683
	Violence and sexual offences	10.776504
2021	Anti-social behaviour	9.762682
	Bicycle theft	0.174988
	Burglary	1.063818
	Criminal damage and arson	0.526076
	Drugs	0.829945
	Other crime	0.779392
	Other theft	1.888207
	Possession of weapons	0.144435
	Public order	4.506366
	Robbery	0.147212
	Shoplifting	1.210475
	Theft from the person	0.096105
	Vehicle crime	1.263249
	Violence and sexual offences	11.417572
2022	Anti-social behaviour	5.249650
	Bicycle theft	0.123881
	Burglary	0.888830
	Criminal damage and arson	0.392752
	Drugs	0.613292
	Other crime	0.680510
	Other theft	1.777104
	Possession of weapons	0.154434
	Public order	4.836900
	Robbery	0.151657
	Shoplifting	1.149923
	Theft from the person	0.113881
	Vehicle crime	1.113815
	Violence and sexual offences	11.434793

dtype: float64]

```
sns.countplot(x='year', hue='crime_type', data=data_Gw)
sns.set(rc={'figure.figsize':(15,6)})
plt.title("crimes by the years")
plt.show()
```



The bar graph shows the crimes by years(2020,2021,2022) for Gwent area.

Overall, the proportion of crimes except Bicycle theft, Robbery and Violence and sexual offences in Gwent area decreased over the time. The figure explains the crime for Anti-social behaviour decreased. The most dramatic change was for Violence and sexual offences. It was increased by nearly 1% in 2021 but in 2022 it was increased only 0.02%. The smallest change was seen for Robbery, where the number increased only from 0.13% to 0.14% and then 0.14% to 0.15%. The only crime for Anti-social behaviour was dramatically decreased within the time. It was going down from approximately from 13% to 5% over the period.

```
pd.value_counts(data_Gw['year'])
```

Year	Count
2020	67519
2021	60863
2022	51630

Name: year, dtype: int64

Locationwise crimes

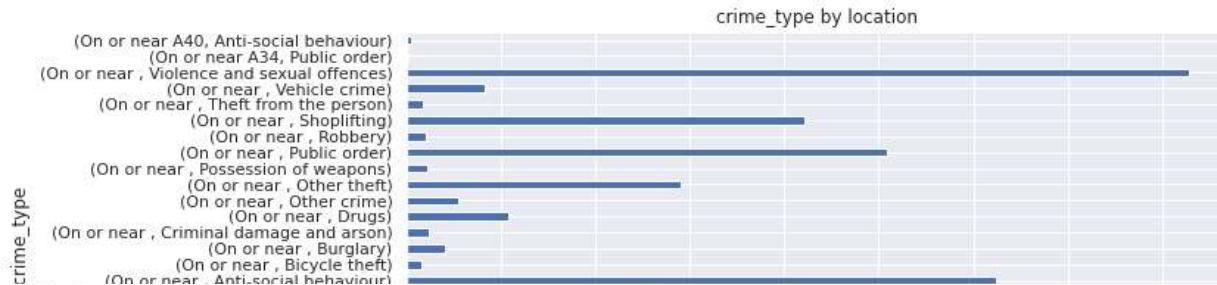
```
data1 = data_Gw.groupby(['Location', 'crime_type']).size()[:30]
data1
```

Location	crime_type	Count
No Location	Anti-social behaviour	78
	Bicycle theft	5
	Burglary	28
	Criminal damage and arson	11
	Drugs	52
	Other crime	21
	Other theft	56
	Possession of weapons	4
	Public order	91
	Robbery	6
	Shoplifting	23
	Theft from the person	3
	Vehicle crime	35
	Violence and sexual offences	266
On or near	Anti-social behaviour	624
	Bicycle theft	15
	Burglary	41
	Criminal damage and arson	24
	Drugs	108
	Other crime	54
	Other theft	291
	Possession of weapons	22
	Public order	509
	Robbery	20
	Shoplifting	422
	Theft from the person	17
	Vehicle crime	83
	Violence and sexual offences	829
On or near A34	Public order	1
On or near A40	Anti-social behaviour	5

dtype: int64

```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (10,6)
data1.plot(kind = 'barh')
plt.title("crime_type by location")
```

```
Text(0.5, 1.0, 'crime_type by location')
```



The diagram shows the highest crimes for either Anti-social behaviour or violence and sexual offences was reported by on or near location. Besides these, every highest crimes occurred on or near location.

```
(No Location, Possession of weapons)
```

Crimes per Month in Gwent area

```
(No Location, Burglary)
```

```
data_Gw.groupby('Month').crime_type.size()
```

Month	Count
April	19265
August	15591
December	8947
February	14116
January	14947
July	16381
June	16348
March	15790
May	17860
November	10064
October	15339
September	15364

Name: crime_type, dtype: int64

```
data_Gw.groupby('crime_type').Month.value_counts()[:120]
```

Crime Type	Month	Count
Anti-social behaviour	April	8157
	May	6217
	March	4594
	June	4411
	July	4047
	...	
Robbery	May	67
	July	64
	March	60
	December	38
	November	37

Name: Month, Length: 120, dtype: int64

```
import matplotlib.pyplot as plt
```

```
# function to add value labels
```

```
def addlabels(x,y):
    for i in range(len(x)):
        plt.text(i, y[i], y[i], ha = 'center',
                 Bbox = dict(facecolor = 'pink', alpha = .8))
```

```
if __name__ == '__main__':
```

```
    # creating data on which bar chart will be plot
```

```
x = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
```

```
y = [14947, 14116, 15790, 19265, 17860, 16348, 16381, 15591, 15364, 15339, 10064, 8947]
```

```
plt.figure(figsize = (12,7))
```

```
# making the bar chart on the data
```

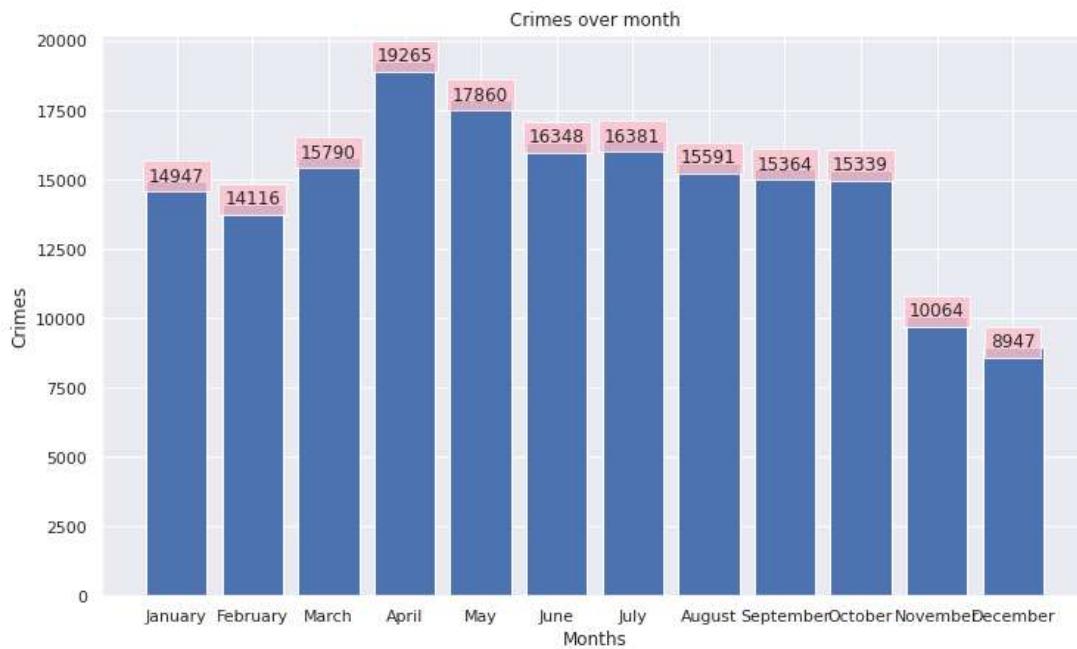
```
plt.bar(x, y)
```

```
# calling the function to add value labels
```

```
addlabels(x, y)
```

```
plt.title("Crimes over month")
```

```
# giving X and Y labels
plt.xlabel("Months")
plt.ylabel("Crimes")
# visualizing the plot
plt.show()
```



The graph represents the highest crimes from January to December from 2020 to 2022 in Gwent area. Overall, the diagram can be explained that April month was the largest occurring crimes month while December was the least. Apart from that, some breakdown crimes can be observed by the graph. For instance, it was 14947 in January but in February it was 14116. It was raised by 15790 in next month. After April it was decreased and it was fluctuated till August. After then it was decreased for the following month.

```
data_Gw.Month
```

```
0      January
1      January
2      January
3      January
4      January
...
180007  October
180008  October
180009  October
180010  October
180011  October
Name: Month, Length: 180012, dtype: object
```

```
data_Gw.Latitude
```

```
0      51.798553
1      51.798910
2      51.798910
3      51.798910
4      51.799638
...
180007  0.000000
180008  0.000000
180009  0.000000
180010  0.000000
180011  0.000000
Name: Latitude, Length: 180012, dtype: float64
```

```
data_Gw.Longitude
```

```

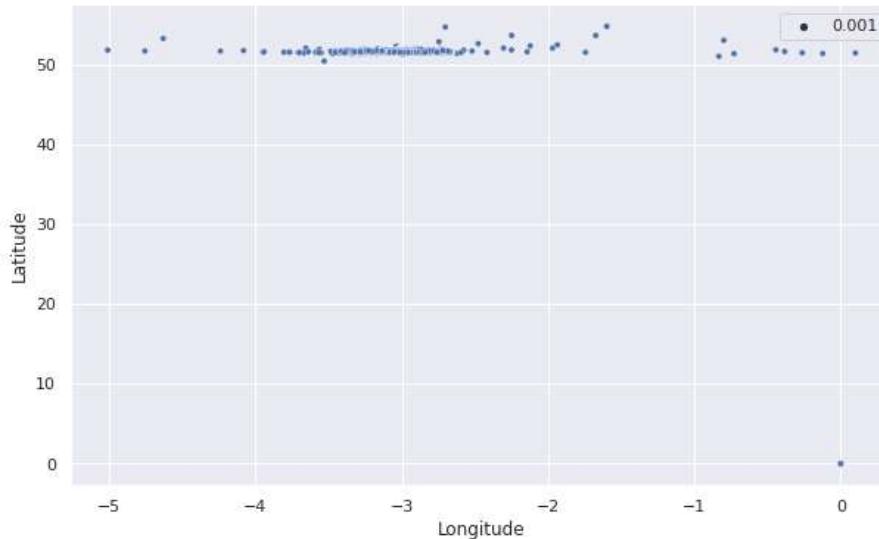
0      -3.193791
1      -3.193205
2      -3.193205
3      -3.193205
4      -3.192296
...
180007    0.000000
180008    0.000000
180009    0.000000
180010    0.000000
180011    0.000000
Name: Longitude, Length: 180012, dtype: float64

```

```
sample_df = data_Gw.sample(int(0.1 * len(data_Gw)))
```

```
sns.scatterplot(x= sample_df.Longitude, y= sample_df.Latitude, size =0.001)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc81976c160>
```



```
Latitude,Longitude = data_Gw.Latitude[0], data_Gw.Longitude[0]
Latitude,Longitude
```

```
(51.798553, -3.193791)
```

It can be seen by the graph, most of the plots were shown in this area:(x,y)=(51.79,-3.19)

Was Robbery more prevalent over Month?

```
crime_type_grp = Data_2021.groupby(['crime_type'])
```

```
df8=crime_type_grp.get_group('Robbery')
df8
```

	Month	Longitude	Latitude	Location	crime_type	year
378	January	-3.247402	51.769592	On or near Park Row	Robbery	2021
379	January	-3.247402	51.769592	On or near Park Row	Robbery	2021
1770	January	-3.092889	51.611881	On or near Fairview Avenue	Robbery	2021
2315	January	-3.022314	51.823853	On or near Prince'S Street	Robbery	2021
2500	January	-3.027050	51.019100	On or near Wards Close	Robbery	2021

```
filt=df8['crime_type'] == 'Robbery'
df8.loc[filt]['Month'].value_counts()
```

```
October      31
April       29
May        24
June       24
September   23
December    23
January     21
February    21
August      21
July        18
March       16
November    14
Name: Month, dtype: int64
```

```
# plot line graph
plt.title ("Robbery by Month in 2021")
x = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
y = [21,21,16,29,24,24,18,21,23,31,14,23]
sns.lineplot(x , y)
plt.show()
```

/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword args to the function: 'label'. The value 'Robbery' was passed to 'label'.



The line graph describes for the crime for Robbery in Gwent area(2021).Overall, the diagram can be seen that there was no change in January and February; May and June, but it was decreased in March.Thus, it can be said that it was fluctuated. October was the highest Robbery occurring month but it was half times decreased for the following month.

```
crime_type_grp['Month'].value_counts().loc['Anti-social behaviour']
```

```
Month
January      2273
March        2260
```

```

February      2100
April        1878
June        1391
May         1362
July        1348
August      1193
November    1072
September   1004
October     944
December    749
Name: Month, dtype: int64

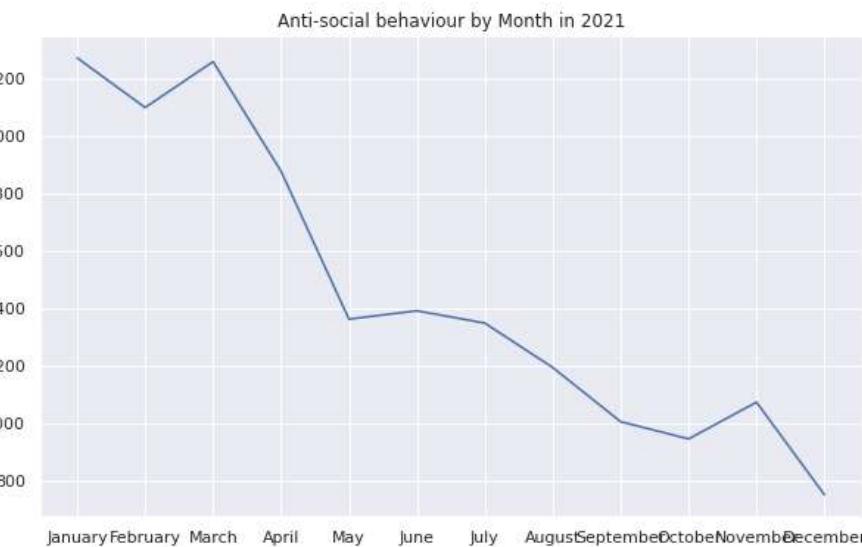
```

```

# plot line graph
plt.title ("Anti-social behaviour by Month in 2021")
x = ['January','February','March','April','May','June','July','August','September','October','November','December']
y = [2273,2100,2260,1878,1362,1391,1348,1193,1004,944,1072,749]
sns.lineplot(x , y)
plt.show()

```

/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments to warnings.warn(



The line graph describes for the crime for Anti-social behaviour in Gwent area(2021).Overall, without some fluctuation it was fallen over the month.

Crimes for Summer and Winter in Gwent area

```

df10 = data_Gw[data_Gw['Month'].isin(['June','July','August'])]
df10

```

	Month	Longitude	Latitude	Location	crime_type	year
30308	June	-3.197188	51.794859	On or near Heol Gwyn Lliw	Anti-social behaviour	2020
30309	June	-3.198548	51.796401	On or near Llangattwg Close	Anti-social behaviour	2020
30310	June	-3.197188	51.794859	On or near Heol Gwyn Lliw	Anti-social behaviour	2020

```
df21 = data_Gw[data_Gw['Month'].isin(['April', 'May', 'July'])]
```

```
df21
```

	Month	Longitude	Latitude	Location	crime_type	year
14107	April	-3.202002	51.797040	On or near Heol Siloam	Anti-social behaviour	2020
14108	April	-3.197188	51.794859	On or near Heol Gwyn Lliw	Anti-social behaviour	2020
14109	April	-3.193440	51.800643	On or near Big Lane	Anti-social behaviour	2020
14110	April	-3.194144	51.800411	On or near Clos Lon Fawr	Anti-social behaviour	2020
14111	April	-3.192296	51.799638	On or near Twyncynghordy Road	Anti-social behaviour	2020
...
164004	July	0.000000	0.000000	No Location	Violence and sexual offences	2022
164005	July	0.000000	0.000000	No Location	Violence and sexual offences	2022
164006	July	0.000000	0.000000	No Location	Violence and sexual offences	2022
164007	July	0.000000	0.000000	No Location	Violence and sexual offences	2022
164008	July	0.000000	0.000000	No Location	Violence and sexual offences	2022

53506 rows × 6 columns

```
dfa = data_Gw [data_Gw['Month'].isin(['April'])]
```

```
dfa
```

	Month	Longitude	Latitude	Location	crime_type	year
14107	April	-3.202002	51.797040	On or near Heol Siloam	Anti-social behaviour	2020
14108	April	-3.197188	51.794859	On or near Heol Gwyn Lliw	Anti-social behaviour	2020
14109	April	-3.193440	51.800643	On or near Big Lane	Anti-social behaviour	2020
14110	April	-3.194144	51.800411	On or near Clos Lon Fawr	Anti-social behaviour	2020
14111	April	-3.192296	51.799638	On or near Twyncynghordy Road	Anti-social behaviour	2020
...
148305	April	0.000000	0.000000	No Location	Violence and sexual offences	2022
148306	April	0.000000	0.000000	No Location	Violence and sexual offences	2022
148307	April	0.000000	0.000000	No Location	Violence and sexual offences	2022
148308	April	0.000000	0.000000	No Location	Violence and sexual offences	2022
148309	April	0.000000	0.000000	No Location	Other crime	2022

19265 rows × 6 columns

```
crime_type_grp3 = dfa.groupby(['crime_type'])
```

```
breakdown_crime_gwent = crime_type_grp3['Month'].value_counts().loc['Anti-social behaviour'].sum()
```

```
breakdown_crime_gwent
```

```
df11=data_Gw[data_Gw['Month'].isin(['December', 'January', 'February'])]
```

```
df11
```

	Month	Longitude	Latitude	Location	crime_type	year
0	January	-3.193791	51.798553	On or near South Bank	Public order	2020
1	January	-3.193205	51.798910	On or near Lansbury Terrace	Public order	2020
2	January	-3.193205	51.798910	On or near Lansbury Terrace	Vehicle crime	2020
3	January	-3.193205	51.798910	On or near Lansbury Terrace	Violence and sexual offences	2020
4	January	-3.192296	51.799638	On or near Twyncynghordy Road	Violence and sexual offences	2020
...
137748	February	0.000000	0.000000	No Location	Violence and sexual offences	2022
137749	February	0.000000	0.000000	No Location	Violence and sexual offences	2022
137750	February	0.000000	0.000000	No Location	Violence and sexual offences	2022
137751	February	0.000000	0.000000	No Location	Violence and sexual offences	2022
137752	February	0.000000	0.000000	No Location	Violence and sexual offences	2022

38010 rows × 6 columns

```
crime_type_grp1 = df10.groupby(['crime_type'])
```

```
df22 = df21.groupby('crime_type').Month.value_counts()
df22
```

crime_type	Month	
Anti-social behaviour	April	8157
	May	6217
	July	4047
Bicycle theft	July	104
	April	82
	May	71
Burglary	May	547
	July	531
	April	453
Criminal damage and arson	April	366
	May	313
	July	246
Drugs	May	457
	July	422
	April	418
Other crime	May	424
	July	390
	April	386
Other theft	July	958
	May	856
	April	819
Possession of weapons	May	81
	April	72
	July	64
Public order	July	2663
	May	2238
	April	2171
Robbery	April	79
	May	67
	July	64
Shoplifting	May	578
	July	554
	April	533
Theft from the person	July	57
	May	53
	April	45
Vehicle crime	April	619
	May	583
	July	570
Violence and sexual offences	July	5711
	May	5375
	April	5065

Name: Month, dtype: int64

```
crime_type_grp2 = df11.groupby(['crime_type'])

summer_anti = crime_type_grp1['Month'].value_counts().loc['Anti-social behaviour'].sum()
summer_anti

12150

winter_anti = crime_type_grp2['Month'].value_counts().loc['Anti-social behaviour'].sum()
winter_anti

9749

summer_vio = crime_type_grp1['Month'].value_counts().loc['Violence and sexual offences'].sum()
summer_vio

16837

winter_vio = crime_type_grp2['Month'].value_counts().loc['Violence and sexual offences'].sum()
winter_vio

13396

summer_pub = crime_type_grp1['Month'].value_counts().loc['Public order'].sum()
summer_pub

7602

winter_pub = crime_type_grp2['Month'].value_counts().loc['Public order'].sum()
winter_pub

4841

summer_vehi = crime_type_grp1['Month'].value_counts().loc['Vehicle crime'].sum()
summer_vehi

1775

winter_vehi = crime_type_grp2['Month'].value_counts().loc['Vehicle crime'].sum()
winter_vehi

1655

summer_oth = crime_type_grp1['Month'].value_counts().loc['Other theft'].sum()
summer_oth

2757

winter_oth = crime_type_grp2['Month'].value_counts().loc['Other theft'].sum()
winter_oth

2224

summer_drugs = crime_type_grp1['Month'].value_counts().loc['Drugs'].sum()
summer_drugs

1173

winter_drugs = crime_type_grp2['Month'].value_counts().loc['Drugs'].sum()
winter_drugs
```

1069

```
summer_shopli = crime_type_grp1['Month'].value_counts().loc['Shoplifting'].sum()
summer_shopli
```

1730

```
winter_shopli = crime_type_grp2['Month'].value_counts().loc['Shoplifting'].sum()
winter_shopli
```

1615

```
summer_Robbery = crime_type_grp1['Month'].value_counts().loc['Robbery'].sum()
summer_Robbery
```

212

```
winter_Robbery = crime_type_grp2['Month'].value_counts().loc['Robbery'].sum()
winter_Robbery
```

173

```
labels = ['Violence and sexual offences', 'Anti_social behaviour', 'Public order', 'Other theft', 'Vehicle crime', 'Shoplifting']
```

```
summer_total = [16837, 12150, 7602, 2757, 1775, 1730, 1173, 212]
```

```
winter_total = [13396, 9749, 4841, 2224, 1655, 1615, 1069, 173]
```

```
x = np.arange(len(labels)) # the label locations
```

```
width = 0.25 # the width of the bars
```

```
fig, ax = plt.subplots()
```

```
rects1 = ax.bar(x - width/2, summer_total, width, label='summer')
```

```
rects2 = ax.bar(x + width/2, winter_total, width, label='winter')
```

```
# Add some text for labels, title and custom x-axis tick labels, etc.
```

```
ax.set_ylabel('crimes')
```

```
ax.set_title('crimes by summer and winter')
```

```
ax.set_xticks(x)
```

```
ax.set_xticklabels(labels)
```

```
ax.legend()
```

```
def autolabel(rects):
```

```
    """Attach a text label above each bar in *rects*, displaying its height."""
```

```
for rect in rects:
```

```
    height = rect.get_height()
```

```
    ax.annotate('{}'.format(height),
```

```
                xy=(rect.get_x() + rect.get_width() / 3, height),
```

```
                xytext=(0, 5), # 3 points vertical offset
```

```
                textcoords="offset points",
```

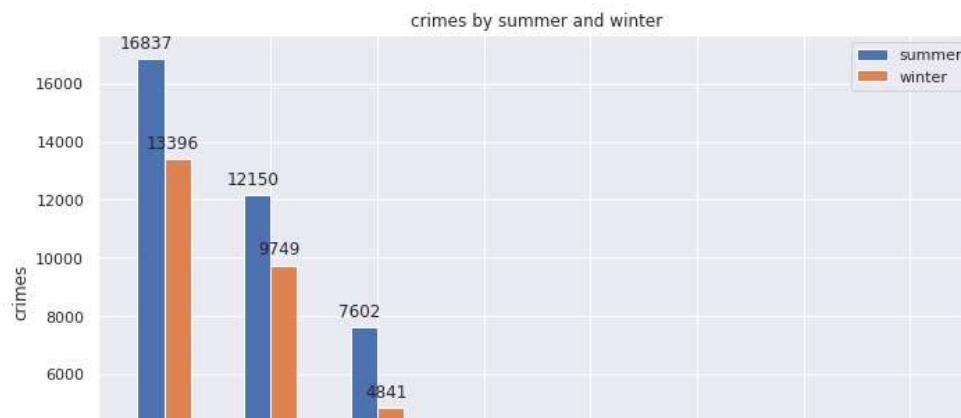
```
                ha='center', va='bottom')
```

```
autolabel(rects1)
```

```
autolabel(rects2)
```

```
fig.tight_layout()
```

```
plt.show()
```



The bar graph explains crimes for Summer ant Winter in Gwent area. Overall, the diagram can be seen that in summer, the crime rate was higher than the winter season. Every highest crime was occurred within the summer period simply because at that time most of the people stayed at outside.

Violence and sexual offences
Offences against the person
Public order
Other theft
Vehicle crime
Shoplifting
Drugs
Robbery

Visualisation

```
list(zip(list(data_Gw.Latitude), list(data_Gw.Longitude)))
```

```
[(51.798553, -3.193791),
 (51.79891, -3.193205),
 (51.79891, -3.193205),
 (51.79891, -3.193205),
 (51.799638, -3.192296),
 (51.794859, -3.197188),
 (51.794859, -3.197188),
 (51.794947, -3.20531),
 (51.804033, -3.212674),
 (51.804033, -3.212674),
 (51.804033, -3.212674),
 (51.803998, -3.211788),
 (51.804716, -3.212707),
 (51.804033, -3.212674),
 (51.805654, -3.208149),
 (51.804033, -3.212674),
 (51.804716, -3.212707),
 (51.803998, -3.211788),
 (51.803683, -3.205674),
 (51.800346, -3.211023),
 (51.797502, -3.20867),
 (51.797502, -3.20867),
 (51.795587, -3.207822),
 (51.796704, -3.210244),
 (51.796704, -3.210244),
 (51.795578, -3.211302),
 (51.798348, -3.200731),
 (51.796497, -3.20757),
 (51.797114, -3.239881),
 (51.79647, -3.226711),
 (51.800227, -3.226813),
 (51.800227, -3.226813),
 (51.800227, -3.226813),
 (51.800905, -3.223061),
 (51.799876, -3.219074),
 (51.799876, -3.219074),
 (51.80165, -3.225735),
 (51.796941, -3.222055),
 (51.80165, -3.225735),
 (51.799823, -3.221625),
 (51.803717, -3.172926),
 (51.803717, -3.172926),
 (51.80213, -3.16617),
 (51.80523, -3.164582),
 (51.802394, -3.175807),
 (51.799078, -3.167512),
 (51.799078, -3.167512),
 (51.802819, -3.171046),
```

```
(51.804912, -3.147503),
(51.802819, -3.171046),
(51.799078, -3.167512),
(51.80523, -3.164582),
(51.799321, -3.183645),
(51.79731, -3.186899),
(51.800817, -3.186846),
(51.79731, -3.186899),
(51.79731, -3.186899),
(51.800817, -3.186846),
```

```
import folium
from folium.plugins import HeatMap
import numpy as np
sample_data_Gw = data_Gw.sample(int(0.01*len(data_Gw)))
lat_lon_pairs = list(zip(list(sample_data_Gw.Latitude), list(sample_data_Gw.Longitude)))
```

```
map = folium.Map()
HeatMap (lat_lon_pairs).add_to(map)
map
```

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

+
—
-



From the folium map it canbe observed that George street bridge,Newport bridge,near Cardiff road, Malpas road, Stowpark area, Alma street, Commercial street, Newport city centre etc. was the most crime occurring area over the period.

```
Data_2022.groupby(['year', 'crime_type']).size()
```

year	crime_type	
2022	Anti-social behaviour	9450
	Bicycle theft	223
	Burglary	1600
	Criminal damage and arson	707
	Drugs	1184
	Other crime	1225
	Other theft	3199
	Possession of weapons	278
	Public order	8707

```

Robbery           273
Shoplifting      2070
Theft from the person 205
Vehicle crime    2005
Violence and sexual offences 20584
dtype: int64

```

```
df_Lockdown.groupby(['year','crime_type']).size()
```

year	crime_type	
2021	Anti-social behaviour	15753
	Bicycle theft	286
	Burglary	1563
	Criminal damage and arson	773
	Drugs	1286
	Other crime	1137
	Other theft	2758
	Possession of weapons	205
	Public order	6699
	Robbery	228
	Shoplifting	1790
	Theft from the person	135
	Vehicle crime	1876
	Violence and sexual offences	16749

dtype: int64

```

import plotly.graph_objects as go
from plotly.subplots import make_subplots

labels = ["Anti-social behaviour","Violence and sexual offences","Public order","Other theft","Vehicle crime","Burglary","Shopliftin
fig = make_subplots(1, 2, specs=[[{'type':'domain'}, {'type':'domain'}]], subplot_titles=['Crimes after lockdown', 'Crimes during lockdown'])
fig.add_trace(go.Pie(labels=labels, values= [9450,20584,8707,3199,2005,1600,2070,1104], scalegroup='one', name="Crimes after lockdown-2022"), 1, 1)
fig.add_trace(go.Pie(labels=labels, values= [15753,16749,6699,2758,1876,1563,1790,1286], scalegroup='one', name="Crimes during lockdown-2021"), 1, 2)

fig.update_layout(title_text='Crimes during and after lockdown')
fig.show()

```

The two pie charts compare the number of crimes during lockdown i.e. 2021 and after lockdown i.e. 2022. Overall, it can be observed that violence and sexual offences after lockdown accounted for the majority of crime, while Anti-social behaviour during lockdown accounted for the most occurring crime.

In detail, without anti-social behaviour and Drugs, most of the crimes was increased after lockdown. During lockdown people had to stay at home, thus, anti-social behaviour was nearly 1.5 times higher than the after lockdown. Besides, there were other crimes, Public order and Burglary, accounting for after lockdown 17.9% and 3.28% respectively whereas during lockdown it was 13.8% and 3.22% respectively.

 Other theft

```
Data_2021.groupby(['year','crime_type']).size()
```

year	crime_type	size()
2021	Anti-social behaviour	17574
	Bicycle theft	315
	Burglary	1915
	Criminal damage and arson	947
	Drugs	1494
	Other crime	1403
	Other theft	3399
	Possession of weapons	260
	Public order	8112
	Robbery	265
	Shoplifting	2179
	Theft from the person	173
	Vehicle crime	2274
	Violence and sexual offences	20553

dtype: int64

```
import plotly.graph_objects as go
from plotly.subplots import make_subplots

labels = ["Anti-social behaviour","Violence and sexual offences","Public order","Other theft","Vehicle crime","Burglary","Shopl:"

fig = make_subplots(1, 2, specs=[[{'type':'domain'}, {'type':'domain'}]]),
                     subplot_titles=['crimes by metropoliton area in 2021','crimes by Gwent area in 2021'])
fig.add_trace(go.Pie(labels=labels, values = [301494,244927,58714,89488,100188,53388,33047,42819,22452]

, scalegroup='one',
                    name="crimes by Metropoliton area in 2021"), 1, 1)
fig.add_trace(go.Pie(labels=labels, values= [17574,20553,8112,3399,2274,1915,2179,1494,265]
, scalegroup='one',
                    name="crimes by Gwent area in 2021"), 1, 2)

fig.update_layout(title_text='Crimes by Metropoliton and Gwent area in 2021 ')
fig.show()
```

Crimes by Metropoliton and Gwent

Crimes by Gwent and Metropoliton area in 2021

The two pie charts compare the number of crimes between Metropoliton area and Gwent area in 2021. Overall, it can be seen that without violence and sexual offences, Public order and Shoplifting the majority of crime occurred in Gwent area over the year.

In detail, first and foremost,in Gwent area, the number of crimes for Public order was more than 2 times higher than that of Metropoliton area within the period. Also. the number of crimes for anti-social behaviour was 2% lower than that of Metropoliton area in 2021.

Secondly, in Metropoliton area, the number of crimes for violence and sexual offences was 9% lower than that of Gwent area.over the year. Apart from that,most of the crimes was increased in Metropoliton area rather than Gwent area.Other crimes i.e. Burglary and Drugs were 3.33% and 2.6%; 5.78% and 4.63% andaccounting in Gwent and Metropoliton area in 2021 respectively.

```
Data_metro.groupby(['year', 'crime_type']).size()
```

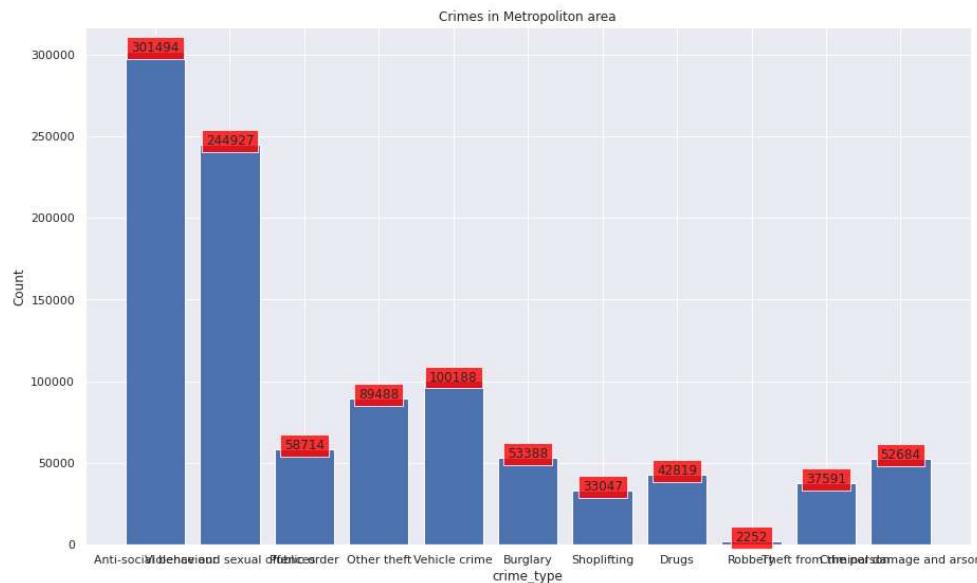
year	crime_type	size()
2021	Anti-social behaviour	301494
	Bicycle theft	21000
	Burglary	53388
	Criminal damage and arson	52684
	Drugs	42819
	Other crime	10229
	Other theft	89488
	Possession of weapons	4855
	Public order	58714
	Robbery	22452
	Shoplifting	33047
	Theft from the person	37591
	Vehicle crime	100188
	Violence and sexual offences	244927

dtype: int64

```
import matplotlib.pyplot as plt

# function to add value labels
def addlabels(x,y):
    for i in range(len(x)):
        plt.text(i, y[i], y[i], ha = 'center',
                 Bbox = dict(facecolor = 'red', alpha = .8))

if __name__ == '__main__':
    # creating data on which bar chart will be plot
    x = ['Anti-social behaviour', 'Violence and sexual offences', 'Public order', 'Other theft', 'Vehicle crime', 'Burglary', 'Shoplifting']
    y = [301494, 244927, 58714, 89488, 100188, 53388, 33047, 42819, 2252, 37591, 52684]
    plt.figure(figsize = (15,9))
    # making the bar chart on the data
    plt.bar(x, y)
    # calling the function to add value labels
    addlabels(x, y)
    plt.title("Crimes in Metropoliton area")
    # giving X and Y labels
    plt.xlabel("crime_type")
    plt.ylabel("Count")
    # visualizing the plot
    plt.show()
```



From the bar chart it can easily be seen that Metropoliton area was the highest crime occuring area in 2021 and Anti-social behaviour was the most highest crime type among other types.

```
df_Metro = Data_metro.groupby(['crime_type', 'Month']).size()[:50]
df_Metro
```

crime_type	Month	Count
Anti-social behaviour	April	27127
	August	23330
	December	16749
	February	29154
	January	33642
	July	26214
	June	27629
	March	30615
	May	26136
	November	17996
	October	20998
	September	21904
Bicycle theft	April	1843
	August	2044
	December	1102
	February	1107
	January	1058
	July	2196
	June	2110
	March	1620
	May	2056
	November	1676
	October	1999
	September	2189
Burglary	April	4203
	August	4403
	December	4712
	February	4062
	January	4299
	July	4262
	June	4286
	March	4715
	May	4235
	November	4954
	October	4812
	September	4445
Criminal damage and arson	April	4308
	August	4609
	December	4257
	February	3551
	January	3634
	July	4698

```

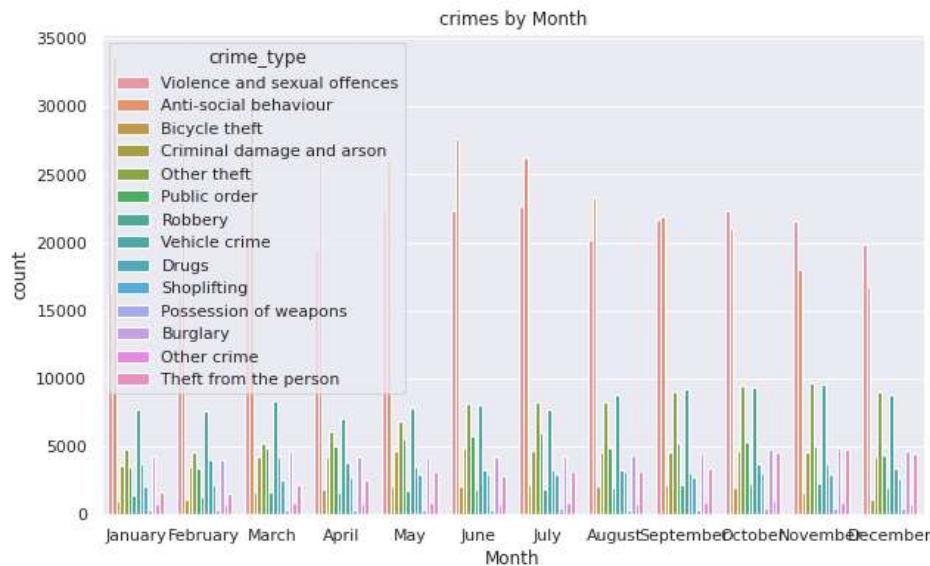
       June      4903
       March     4222
       May       4734
       November   4550
       October    4664
       September   4554
Drugs        April    3791
                  August   3252
dtype: int64

```

```

sns.countplot(x='Month',hue='crime_type',data=Data_metro)
sns.set(rc={'figure.figsize':(20,15)})
plt.title("crimes by Month")
plt.show()

```



from the bar chart it can easily be observed that anti-social behaviour and violence and sexual offences are the highest crime type over the 12 months in Metropoliton area.

Correlation

```

data_Gw_ct = pd.crosstab(data_Gw.crime_type, data_Gw.year, margins=True)
data_Gw_ct.head()

```

	year	2020	2021	2022	All
crime_type					
Anti-social behaviour		23981	17574	9450	51005
Bicycle theft		287	315	223	825
Burglary		2228	1915	1600	5743
Criminal damage and arson		1302	947	707	2956
Drugs		1783	1494	1104	4381

```

#include years from 2020 to 2022
df25 = data_Gw[data_Gw.year >=2020]
df26 = data_Gw[data_Gw.year <=2022]

```

```

#Transposed crosstab & find correlation
data_Gw_ct = pd.crosstab(df25.crime_type, df26.year, margins=False).reindex()
data_Gw_ct = data_Gw_ct.T

```

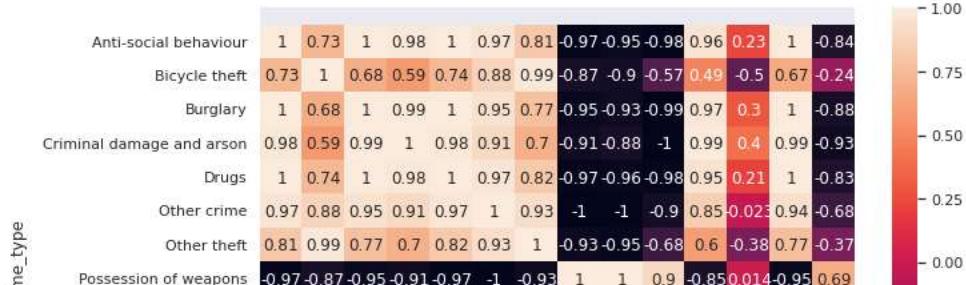
```
type_corr = data_Gw_ct.corr()
type_corr
```

crime_type	Anti-social behaviour	Bicycle theft	Burglary	Criminal damage and arson	Drugs	Other crime	Other theft	Possession of weapons
crime_type								
Anti-social behaviour	1.000000	0.726973	0.997805	0.983979	0.999846	0.967369	0.813839	-0.
Bicycle theft	0.726973	1.000000	0.679901	0.592903	0.738915	0.877233	0.990654	-0.
Burglary	0.997805	0.679901	1.000000	0.993626	0.996488	0.948465	0.773568	-0.
Criminal damage and arson	0.983979	0.592903	0.993626	1.000000	0.980698	0.906698	0.697200	-0.
Drugs	0.999846	0.738915	0.996488	0.980698	1.000000	0.971667	0.823914	-0.
Other crime	0.967369	0.877233	0.948465	0.906698	0.971667	1.000000	0.934515	-0.
Other theft	0.813839	0.990654	0.773568	0.697200	0.823914	0.934515	1.000000	-0.
Possession of weapons	-0.969572	-0.872952	-0.951231	-0.910394	-0.973720	-0.999961	-0.931329	1.
Public order	-0.950499	-0.904354	-0.927833	-0.879872	-0.955807	-0.998213	-0.954114	0.
Robbery	-0.979117	-0.572195	-0.990432	-0.999675	-0.975398	-0.895658	-0.678710	0.
Shoplifting	0.955042	0.490715	0.972580	0.992598	0.949691	0.848761	0.604975	-0.
Theft from the person	0.231184	-0.500000	0.295110	0.400936	0.214070	-0.022869	-0.377203	0.
Vehicle crime	0.996990	0.671543	0.999936	0.994840	0.995475	0.944811	0.766333	-0.
Violence and sexual offences	-0.842568	-0.242693	-0.876388	-0.925092	-0.832984	-0.678611	-0.372745	0.

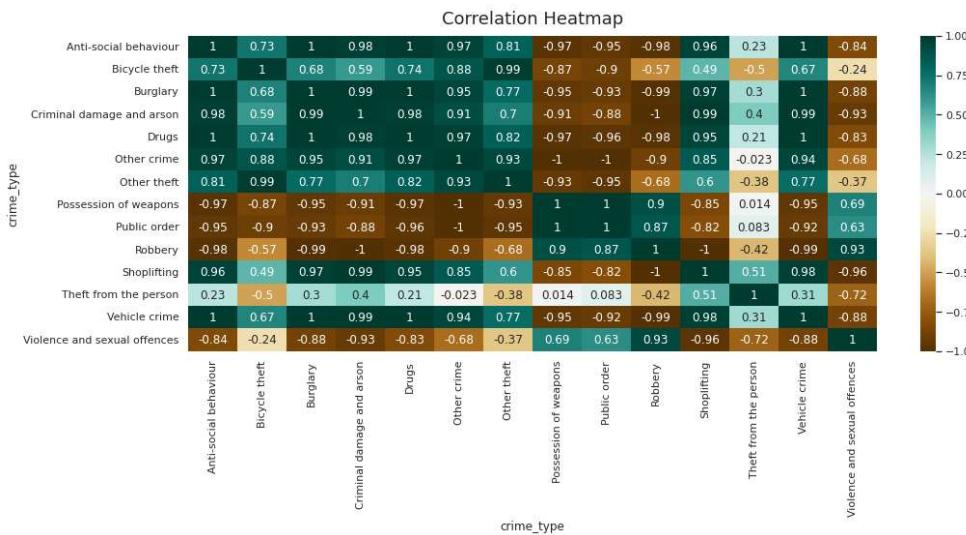


```
#Correlation heatmap
plt.figure(figsize=(10,7))
ax = sns.heatmap(type_corr, annot=True) #notation: "annot" not "annotate"
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
```

(14.5, -0.5)



```
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(type_corr, vmin=-1, vmax=1, annot=True, cmap='BrBG')
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':18}, pad=12);
# save heatmap as .png file
# dpi - sets the resolution of the saved image in dots/inches
# bbox_inches - when set to 'tight' - does not allow the labels to be cropped
plt.savefig('heatmap.png', dpi=300, bbox_inches='tight')
```



Stronger correlation on both ends of the spectrum pops out in darker, weaker correlation in lighter shades.

Each square shows the correlation between the variables on each axis. Correlation ranges from -1 to +1. Values closer to zero means there is no linear trend between the two variables. The closer to 1 the correlation is the more positively correlated they are; that is as one increases so does the other and the closer to 1 the stronger this relationship is. A correlation closer to -1 is similar, but instead of both increasing one variable will decrease as the other increases. The diagonals are all 1/dark green because those squares are correlating each variable to itself (so it's a perfect correlation). For the rest the larger the number and darker the color the higher the correlation between the two variables. The plot is also symmetrical about the diagonal since the same two variables are being paired together in those squares.

For example, Anti-social behaviour has a strong positive correlation with other theft, other crime, Bicycle theft and criminal damage and arson but strongly negative association with robbery, public order, violence and sexual offences, possession of weapons. Again, Burglary is strongly associated with Shoplifting and the correlation coefficient is $r = 0.97$. Also, the people who were involved in Drugs related crime, did not involve in possession of weapons. Hence, $r = -0.97$.

```
top_crimes = Data_metro.groupby(['crime_type'])['Month'].size()
top_crimes = pd.DataFrame(top_crimes).nlargest(10,'Month').reset_index()
top_crimes = list(top_crimes['crime_type'])
```

```
df22 = df21.groupby('crime_type').Month.value_counts()
df22
```

crime_type	Month	
Anti-social behaviour	April	8157
	May	6217
	July	4047
Bicycle theft	July	104
	April	82
	May	71
Burglary	May	547
	July	531
	April	453
Criminal damage and arson	April	366
	May	313
	July	246
Drugs	May	457
	July	422
	April	418
Other crime	May	424
	July	390
	April	386
Other theft	July	958
	May	856
	April	819
Possession of weapons	May	81
	April	72
	July	64
Public order	July	2663
	May	2238
	April	2171
Robbery	April	79
	May	67
	July	64
Shoplifting	May	578
	July	554
	April	533
Theft from the person	July	57
	May	53
	April	45
Vehicle crime	April	619
	May	583
	July	570
Violence and sexual offences	July	5711
	May	5375
	April	5065

Name: Month, dtype: int64

```
data_Gw_loc_ct = pd.crosstab(data_Gw.Location, data_Gw.year, margins=True)
data_Gw_loc_ct.sort_values("All", axis = 0, ascending = False,
                           inplace = True)
data_Gw_loc_ct.head(6)
```

	year	2020	2021	2022	All
	Location				
All		67519	60863	51630	180012
On or near Parking Area		2648	2354	1274	6276
On or near Sports/Recreation Area		2033	1717	842	4592
On or near Supermarket		1758	1567	895	4220
On or near		0	0	3059	3059
On or near Petrol Station		1046	924	552	2522

```
data_Gw_loc_ct = pd.crosstab(data_Gw.Location, data_Gw.year, margins=True)
data_Gw_loc_ct.sort_values("All", axis = 0, ascending = False,
                           inplace = True)
data_Gw_loc_ct.tail(6)
```

	year	2020	2021	2022	All
	Location				
On or near Helen Place		0	1	0	1
On or near Helmsdale		0	0	1	1
On or near Hen Heol Gerrig		0	0	1	1
On or near Hendre Court		0	1	0	1
On or near St Maughans Close		0	0	1	1
On or near Kemy's Place		0	0	1	1

```
#group data by Neighbourhood and count
data_Gw_gp = data_Gw.groupby(['Month','Location']).count()[['crime_type']]
data_Gw_gp= data_Gw_gp.reset_index()
data_Gw_gp.head(7)
```

	Month	Location	crime_type
0	April	No Location	62
1	April	On or near A40	6
2	April	On or near A4042	22
3	April	On or near A4046	12
4	April	On or near A4047	7
5	April	On or near A4049	2
6	April	On or near A4051	5

Advanced Analysis

```
from sklearn import preprocessing
lab=preprocessing.LabelEncoder()
data_Gw['crime_type']=lab.fit_transform(data_Gw['crime_type'])
```

```
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.model_selection import train_test_split
```

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=14)
```

```
kmeans.fit(data_Gw.iloc[:,4:])
```

```
KMeans(n_clusters=14)
```

```
kmeans.cluster_centers_
```

```
array([[1.76085863e-02, 2.02100000e+03],
       [1.29003811e+01, 2.02100000e+03],
       [7.98442497e+00, 2.02049580e+03],
       [4.24256570e+00, 2.02088261e+03],
       [5.84341046e+00, 2.02156551e+03],
       [7.99945993e+00, 2.02200000e+03],
       [1.00439677e+01, 2.02094915e+03],
```

```
[1.28837411e+01, 2.02000000e+03],
[1.18262733e-02, 2.02000000e+03],
[2.31998452e+00, 2.02144631e+03],
[1.29112400e+01, 2.02200000e+03],
[5.69879518e+00, 2.02000000e+03],
[2.30538613e-02, 2.02200000e+03],
[2.36883853e+00, 2.02000000e+03]])
```

```
labels = kmeans.labels_
labels

array([ 2,  2,  7, ..., 10, 10, 10], dtype=int32)
```

```
import numpy as np
unique , counts = np.unique(kmeans.labels_, return_counts=True)
dict_data = dict(zip(unique, counts))
dict_data
```

```
{0: 17889,
1: 22827,
2: 16886,
3: 5784,
4: 7823,
5: 9258,
6: 7551,
7: 21951,
8: 24268,
9: 5169,
10: 22589,
11: 4814,
12: 9673,
13: 3530}
```

```
data_Gw["cluster"] = kmeans.labels_
data_Gw
```

```
import seaborn as sns
sns.lmplot('year','Longitude', data=data_Gw, hue='cluster', palette='coolwarm', size=5, aspect=1, fit_reg=False)
```

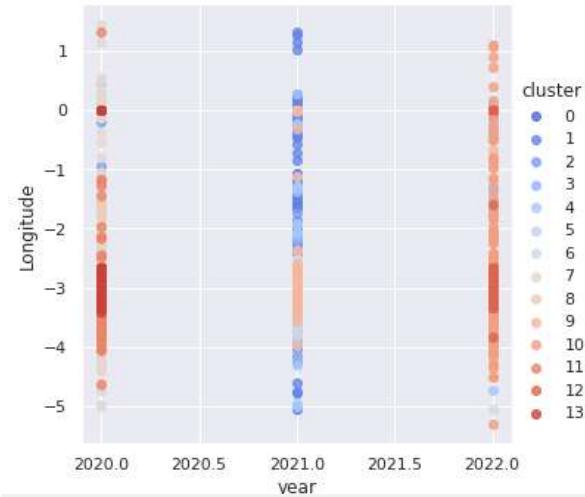
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
```

```
Pass the following variables as keyword args: x, y. From version 0.12, the only valid p
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/regression.py:581: UserWarning:
```

```
The `size` parameter has been renamed to `height`; please update your code.
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc814ec2dc0>
```



After doing cluster the data in Gwent area

data_Gw

	Month	Longitude	Latitude	Location	crime_type	year	cluster
0	January	-3.193791	51.798553	On or near South Bank	8	2020	2
1	January	-3.193205	51.798910	On or near Lansbury Terrace	8	2020	2
2	January	-3.193205	51.798910	On or near Lansbury Terrace	12	2020	7
3	January	-3.193205	51.798910	On or near Lansbury Terrace	13	2020	7
4	January	-3.192296	51.799638	On or near Twyncynghordy Road	13	2020	7
...
180007	October	0.000000	0.000000	No Location	13	2022	10
180008	October	0.000000	0.000000	No Location	13	2022	10
180009	October	0.000000	0.000000	No Location	13	2022	10
180010	October	0.000000	0.000000	No Location	13	2022	10

