

Family Violence Offences in Victoria - Python/Power BI

The Victoria Crime Statistics Agency has some fantastic datasets available to the public. I downloaded a group of 6 tables outlining police offences across Victoria between 2013 and 2023.

The Data

In these tables, you can see every offence, the local government area and police region where that offence occurred, the subgroup of the offence, and more.

I decided to focus on all family violence offences within two of the downloaded datasets to answer the questions; which local government areas (LGA) have the most and least family violence offences, what subgroup of offences are occurring, and how does the number of family violence offences compare to the total number of offences within these LGAs?

Table #1:

	A	B	C	D	E	F
1	Year	Year ending	Police Region	Local Government Area	Offence Count	Rate per 100,000 population
39	2023	March	2 Eastern	Whitehorse	8,358	4,771.0
40	2023	March	2 Eastern	Wodonga	3,750	8,468.8
41	2023	March	2 Eastern	Yarra Ranges	7,130	4,504.9
42	2023	March	2 Eastern	Total	112,231	6,461.8
43	2023	March	3 Southern Metro	Bayside	5,152	5,008.0
44	2023	March	3 Southern Metro	Cardinia	6,832	5,363.6
45	2023	March	3 Southern Metro	Casey	21,435	5,502.5
46	2023	March	3 Southern Metro	Frankston	13,196	9,300.4
47	2023	March	3 Southern Metro	Glen Eira	7,097	4,628.1
48	2023	March	3 Southern Metro	Greater Dandenong	18,071	11,225.1
49	2023	March	3 Southern Metro	Kingston	10,951	6,774.0
50	2023	March	3 Southern Metro	Mornington Peninsula	9,768	5,711.9
51	2023	March	3 Southern Metro	Port Phillip	13,015	11,990.6
52	2023	March	3 Southern Metro	Stonnington	11,310	10,352.7
53	2023	March	3 Southern Metro	Total	116,827	7,182.8
54	2023	March	4 Western	Ararat	1,426	12,103.7
55	2023	March	4 Western	Ballarat	11,097	9,396.7
56	2023	March	4 Western	Buloke	392	6,439.5
57	2023	March	4 Western	Composse	3,662	9,468.2

Table #3:

	A	B	C	D	E	F	G	H	I
1	Year	Year ending	Local Government Area	Postcode	Suburb/Town Name	Offence Division	Offence Subdivision	Offence Subgroup	Offence Count
2	2023	March	Alpine	3691	Dederang	A Crimes against the person	A20 Assault and related offences	A212 Non-FV Serious assault	1
3	2023	March	Alpine	3691	Dederang	A Crimes against the person	Other crimes against the person	Other crimes against the person	1
4	2023	March	Alpine	3691	Dederang	B Property and deception offences	B40 Theft	B42 Steal from a motor vehicle	2
5	2023	March	Alpine	3691	Dederang	B Property and deception offences	B40 Theft	B49 Other theft	1
6	2023	March	Alpine	3691	Dederang	D Public order and security offences	D10 Weapons and explosives offences	D11 Firearms offences	1
7	2023	March	Alpine	3691	Glen Creek	B Property and deception offences	B50 Deception	B53 Obtain benefit by deception	1
8	2023	March	Alpine	3697	Gundowring	B Property and deception offences	B10 Arson	B11 Cause damage by fire	1
9	2023	March	Alpine	3691	Kancoona	B Property and deception offences	B30 Burglary/Break and enter	B322 Non-residential non-aggravated burglary	1
10	2023	March	Alpine	3691	Running Creek	A Crimes against the person	A70 Stalking, harassment and threatening behaviour	A731 FV Threatening behaviour	1
11	2023	March	Alpine	3691	Upper Gundowring	B Property and deception offences	B40 Theft	B49 Other theft	1
12	2023	March	Alpine	3697	Tawonga	A Crimes against the person	A20 Assault and related offences	A211 FV Serious assault	1
13	2023	March	Alpine	3697	Tawonga	A Crimes against the person	A20 Assault and related offences	A231 FV Common assault	1
14	2023	March	Alpine	3697	Tawonga	A Crimes against the person	A70 Stalking, harassment and threatening behaviour	A722 Non-FV Harassment and private nuisance	3
15	2023	March	Alpine	3697	Tawonga	B Property and deception offences	B30 Burglary/Break and enter	B321 Residential non-aggravated burglary	1
16	2023	March	Alpine	3697	Tawonga	B Property and deception offences	B40 Theft	B42 Steal from a motor vehicle	2
17	2023	March	Alpine	3697	Tawonga	B Property and deception offences	B50 Deception	B53 Obtain benefit by deception	1
18	2023	March	Alpine	3697	Tawonga	D Public order and security offences	D20 Disorderly and offensive conduct	D23 Offensive conduct	1
19	2023	March	Alpine	3698	Tawonga South	A Crimes against the person	Other crimes against the person	Other crimes against the person	2
20	2023	March	Alpine	3698	Tawonga South	B Property and deception offences	B20 Property damage	B21 Criminal damage	2

Cleaning

I started the query with importing the necessary libraries and the csv file for table #3:

```
In [1]: import pandas as pd
import numpy as np
import matplotlib as plt

In [6]: df1 = pd.read_csv(r"C:\Users\user\Desktop\Crime_Data_Project\Crime_Stats_Table_3.csv")
df1
```

Out[6]:

	Year	Year ending	Local Government Area	Postcode	Suburb/Town Name	Offence Division	Offence Subdivision	Offence Subgroup	Offence Count
0	2023	March	Alpine	3691	Dederang	A Crimes against the person	A20 Assault and related offences	A212 Non-FV Serious assault	1
1	2023	March	Alpine	3691	Dederang	A Crimes against the person	Other crimes against the person	Other crimes against the person	1
2	2023	March	Alpine	3691	Dederang	B Property and deception offences	B40 Theft	B42 Steal from a motor vehicle	2
3	2023	March	Alpine	3691	Dederang	B Property and deception offences	B40 Theft	B49 Other theft	1
						D Public order and	D10 Weapons and		

Then, I needed to filter the dataset to only include family violence offences. In these tables, this is indicated by “FV” at the beginning of the offence name. After filtering for FV, I noticed that the filter included offences that were labeled “non-FV, so I needed to filter out those offences as well.

```
In [13]: FV_Table_w_non = df1[df1['Offence Subgroup'].str.contains('FV')]

In [22]: df1FV = FV_Table_w_non[FV_Table_w_non['Offence Subgroup'].str.contains('Non-FV') == False]
```

I wanted to create a table that listed the total amount of all FV offences (sum of all the FV offence subgroups) for each LGA.

When I ran the line of code to group by LGA and sum the column “offence count”, an error was returned.

```
In [36]: count = df1FV.groupby('Local Government Area')['Offence Count'].sum()
print(count)
```

```
-----
KeyError                                Traceback (most recent call last)
Cell In[36], line 1
----> 1 count = df1FV.groupby('Local Government Area')['Offence Count'].sum()
      2 print(count)

File ~\anaconda3\Lib\site-packages\pandas\core\groupby\generic.py:1416, in DataFrameGroupBy._getitem__(self, key)
    1407 if isinstance(key, tuple) and len(key) > 1:
    1408     # if len == 1, then it becomes a SeriesGroupBy and this is actually
    1409     # valid syntax, so don't raise warning
    1410     warnings.warn(
    1411         "Indexing with multiple keys (implicitly converted to a tuple "
    1412         "of keys) will be deprecated, use a list instead.",
    1413         FutureWarning,
    1414         stacklevel=find_stack_level(),
    1415     )
-> 1416 return super()._getitem__(key)
```

After some troubleshooting, I discovered the column name had a rogue blank space at the beginning of the title. I ran a strip method to fix it and then discovered the datatype for that column was “object” instead of integer, so I fixed that with a to_numeric method.

```
In [39]: df1FV.columns = df1FV.columns.str.strip()
```

```
In [40]: df1FV.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 34667 entries, 8 to 359122
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Year                  34667 non-null  int64   
1   Year ending           34667 non-null  object  
2   Local Government Area  34667 non-null  object  
3   Postcode              34667 non-null  int64   
4   Suburb/Town Name      34667 non-null  object  
5   Offence Division      34667 non-null  object  
6   Offence Subdivision   34667 non-null  object  
7   Offence Subgroup      34667 non-null  object  
8   Offence Count         34667 non-null  object  
dtypes: int64(2), object(7)
memory usage: 2.6+ MB
```

```
In [50]: df1FV["Offence Count"] = pd.to_numeric(df1FV["Offence Count"])
```

After these transformations, I was able to create a table showing the sum of FV offences grouped by LGA. And I subsequently removed the columns “Year” and “Postcode” from this table using the drop() method.

```
In [52]: count = df1FV.groupby('Local Government Area').sum()
print(count)
```

	Year	Postcode	Offence Count
Local Government Area			
Alpine	284693	526070	318
Ararat	280564	469401	1011
Ballarat	1883531	3127194	5337
Banyule	1534078	2344788	4233
Bass Coast	1447530	2843404	2742
...
Wodonga	351264	642137	2430
Wyndham	950787	1429196	10760
Yarra	761076	1160353	3420
Yarra Ranges	2426473	4179649	5575
Yarriambiack	367434	619937	463

[79 rows x 3 columns]

And to finish, I saved the finished product as a csv.

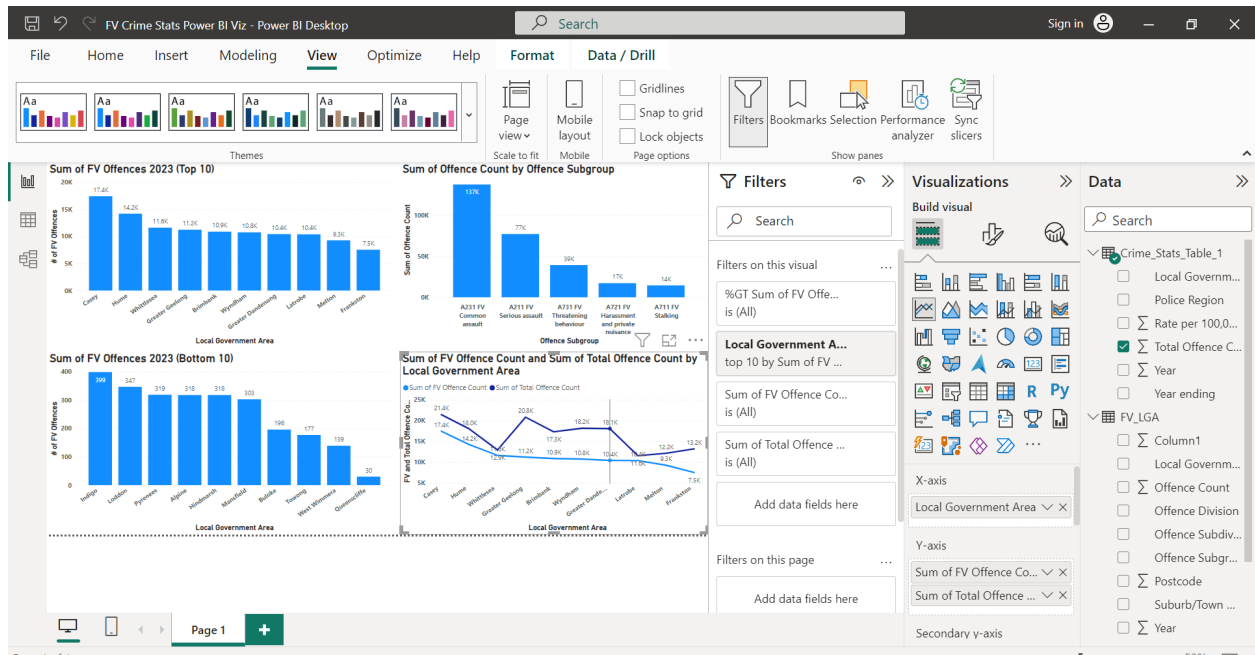
```
In [77]: count.to_csv(r"C:\Users\user\Desktop\Crime_Data_Project\FV_Dataframes\FV_LGA_Sums.csv")
```

Visualisations

Using Power BI, I joined three tables to use for my visualisations: the two tables from above and the offence count sum table I created in my Jupyter Notebook. When I was establishing the relationships between these three tables, the connection with Table 1 wasn't working properly, so I opened Power Query Editor to have a look. It turns out that the LGA column cells all had a blank space at the beginning of the text, so I clicked “split column” and split the column with the delimiter of a blank space. This fixed the issue, as the three tables were all connected by LGA.

I created four visualisations: top 10 LGAs for FV offences, bottom 10 LGA for FV offences, Types of FV offences across all LGAs, and a graph showing the FV offences compared to total offences within the top 10 LGAs for FV offences.

A backend snapshot of building a graph:



The finalised visualisations show that Casey had the most FV offences in 2023, Queenscliffe had the least FV offences, and the most common FV offence was “common assault”. The last graph also shows an interesting breakdown of FV offences versus total offences.

