Task A: Data Exploration and Auditing

A1. Dataset size

How many rows and columns exist in this dataset?

The dataset contains 385296 rows and 7 columns.

```
crime_statistics = pd.read_csv('Crime_Statistics_SA_2014_2019.csv')
crime_statistics.shape
```

```
Out[2]: (385296, 7)
```

A2. Null values in the dataset

Are there any null values in this dataset?

The dataset contains null values.

```
crime_statistics.isnull().values.any()
```

```
Out[3]: True
```

A3. Data Types

What are the min and max for column 'Reported Date '? Does this column have the correct data type? If no, convert it to an appropriate data type.

Minimum in 'Reported Date' column: 2014-01-01

```
crime_statistics['Reported Date'].min()
```

Maximum in 'Reported Date' column: 2019-12-03

```
crime_statistics['Reported Date'].max()
```

'Reported Date' column is of object type but it should be of datetime type.

```
print(crime_statistics['Reported Date'].dtypes)
```

The above code returns the type: 'object'

```
crime_statistics['Reported Date'] = pd.to_datetime(crime_statistics['Reported Date'])
print(crime_statistics['Reported Date'].dtypes)
```

The above code is to change the type of 'Reported Date' to datetime.

A4. Descriptive statistics

Calculate the statistics for the "Offence Count" column (Find the count, mean, standard deviation, minimum and maximum).

Count: 385296

```
crime_statistics['Offence Count'].count()
```

Mean: 1.164870644906774

```
crime_statistics['Offence Count'].mean()
```

Standard Deviation: 0.560723109395765

```
crime_statistics['Offence Count'].std()
```

Minimum: 1

```
crime_statistics['Offence Count'].min()
```

Maximum: 28

```
crime_statistics['Offence Count'].max()
```

A5. Exploring Offence Level 1 Description

Now look at the Offence Level 1 Description column and answer the following questions

1. How many unique values does "Offence Level 1 Description" column take?

There are 2 unique values in the column 'Offence Level 1 Description'.

```
n_uni_lvl1 = {'Offence Level 1 Description':{'Unique Values':'nunique'}}
crime_statistics.agg(n_uni_lvl1)
```

Out[15]:

	Offence Level 1 Description
Unique Values	2

2. Display the unique values of level 1 offences.

```
uni_lvl2 = {'Offence Level 1 Description':{'Unique Values':'unique'}}
crime_statistics.agg(uni_lvl2)
```

Out[16]:

	Offence Level 1 Description
	Unique Values
0	OFFENCES AGAINST PROPERTY
1	OFFENCES AGAINST THE PERSON

3. How many records do contain "offences against the person"?

86791 records contain "Offences against the person".

```
size1 = crime_statistics.groupby('Offence Level 1 Description').size()
size1
```

```
Out[39]: Offence Level 1 Description
         OFFENCES AGAINST PROPERTY      298505
         OFFENCES AGAINST THE PERSON     86791
         dtype: int64
```

4. What percentage of the records are "offences against the property"?

77.47% of the records are "Offences against property".

```
(size1/crime_statistics['Offence Level 1 Description'].count())*100
```

```
Out[16]: Offence Level 1 Description
         OFFENCES AGAINST PROPERTY      77.474202
         OFFENCES AGAINST THE PERSON    22.525798
         dtype: float64
```

A6. Exploring Offence Level 2 Description

Now look at the Offence Level 2 Description column and answer the following questions

1. How many unique values does "Offence Level 2 Description" column take? Display the unique values of level 2 offences together with their counts (i.e., how many times they have been repeated).

"Offence Level 2 Description" has 9 unique values.

```
n_uni_lvl2 = {'Offence Level 2 Description':{'Unique Values':'nunique'}}
crime_statistics.agg(n_uni_lvl2)
```

Out[19]:

	Offence Level 2 Description
Unique Values	9

```
uni_lvl2 = {'Offence Count':{'Count':'count'}}
count_lvl2 = crime_statistics.groupby('Offence Level 2 Description').agg(uni_lvl2).reset_index()
count_lvl2.columns = count_lvl2.columns.droplevel(0)
count_lvl2.rename(columns = {'':'Offence Level 2 Description'}, inplace = True)
count_lvl2
```

Out[21]:

	Offence Level 2 Description	Count
0	ACTS INTENDED TO CAUSE INJURY	63747
1	FRAUD DECEPTION AND RELATED OFFENCES	11644
2	HOMICIDE AND RELATED OFFENCES	226
3	OTHER OFFENCES AGAINST THE PERSON	12327
4	PROPERTY DAMAGE AND ENVIRONMENTAL	80047
5	ROBBERY AND RELATED OFFENCES	2607
6	SERIOUS CRIMINAL TRESPASS	53888
7	SEXUAL ASSAULT AND RELATED OFFENCES	7884
8	THEFT AND RELATED OFFENCES	152926

2. How many serious criminal trespasses have occurred with more than 1 offence count?

4198 serious criminal trespasses have occurred with more than 1 offence count.

```
cond1 = crime_statistics['Offence Level 2 Description'] == 'SERIOUS CRIMINAL TRESPASS'
cond2 = crime_statistics['Offence Count'] > 1
len(crime_statistics[cond1 & cond2])
```

Out[19]: 4198

Task B: Investigating Offence Count in different suburbs and different years

B1. Investigating the number of crimes per year

Find the number of crimes per year. Plot the graph and explain your understanding of the graph.

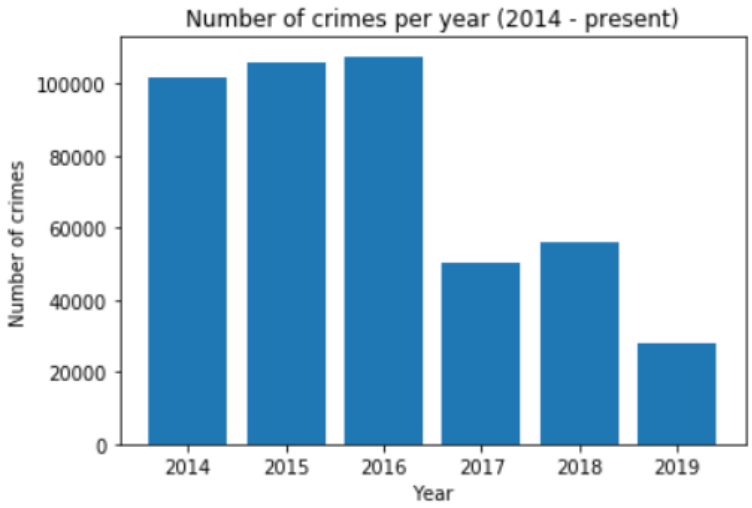
Number of crimes per year
2014: 101750 crimes
2015: 105656 crimes
2016: 107593 crimes
2017: 50159 crimes
2018: 55758 crimes
2019: 27904 crimes

```
crime_statistics['year']=crime_statistics['Reported Date'].dt.year
crime = {'Offence Count':{'Number of crimes':'sum'}}
no_of_crimes = crime_statistics.groupby('year').agg(crime).reset_index()
no_of_crimes.columns = no_of_crimes.columns.droplevel(0)
no_of_crimes.rename(columns = {'':'Year'}, inplace = True)
no_of_crimes
```

Out[22]:

	Year	Number of crimes
0	2014	101750.0
1	2015	105656.0
2	2016	107593.0
3	2017	50159.0
4	2018	55758.0
5	2019	27904.0

```
plt.bar(no_of_crimes['Year'], no_of_crimes['Number of crimes'])
plt.xlabel('Year')
plt.ylabel('Number of crimes')
plt.title('Number of crimes per year (2014 - present)')
plt.show()
```



The graph represents the number of reported crimes in the suburbs of South Australia in a year. In the years 2014 to 2016, the number of crimes per year is slightly above 100 000. But since 2017, the number of crimes has decreased significantly to about 50 000 per year. This could be due to many factors such as tighter security measures or more educated people, thus less crimes committed. 2019 has the lowest recorded number of crimes per year. However, the data shown for 2019 is not yet completed as the last reported crime in the statistics report was in March, that's why the number of crimes committed in 2019 is so low compared to the other years.

B2. Investigating the total number of crimes in different suburbs

1. Compute the total number of crimes in each suburb and plot a histogram of the total number of crimes in different suburbs

Total number of crimes in each suburb.

```
sub = {'Offence Count':{'Total number of crimes':'sum'}}
suburb = crime_statistics.groupby('Suburb - Incident').agg(sub).reset_index()
suburb.columns = suburb.columns.droplevel(0)
suburb.rename(columns = {'':'Suburbs'}, inplace = True)
suburb
```

The first five records in the data frame suburb:

	Suburbs	Total number of crimes
0	ABERFOYLE PARK	1280.0
1	ADDRESS UNKNOWN	84.0
2	ADELAIDE	24598.0
3	ADELAIDE AIRPORT	665.0
4	AGERY	5.0

...

Modify the data so that it can be plotted onto a histogram.

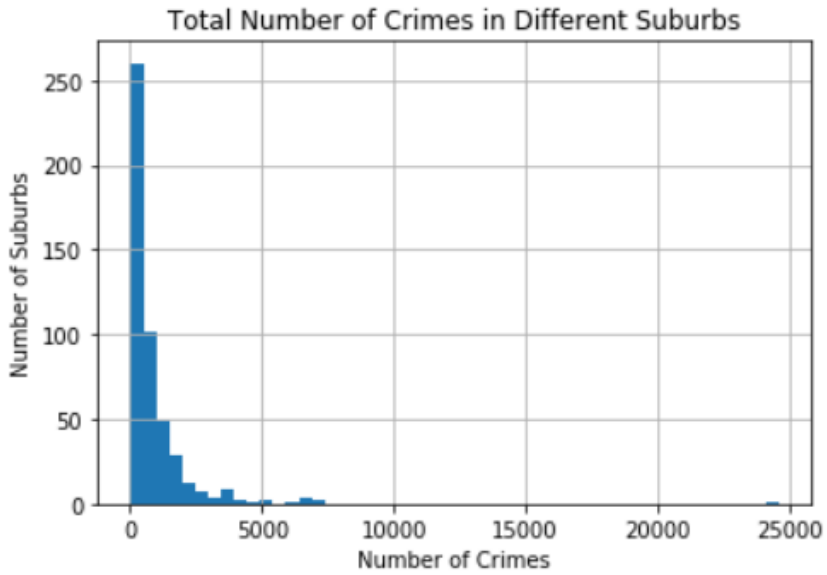
```
sub2 = {'Suburbs':{'Number of Suburbs':'count'}}
suburb2 = suburb.groupby('Total number of crimes').agg(sub2).reset_index()
suburb2.columns = suburb2.columns.droplevel(0)
suburb2.rename(columns = {'':'Number of Crimes'},inplace = True)
suburb2
```

The first five records in the data frame suburb2:

	Number of Crimes	Number of Suburbs
0	1.0	248
1	2.0	125
2	3.0	88
3	4.0	72
4	5.0	44

...

```
suburb2['Number of Crimes'].hist(bins=50)
plt.xlabel('Number of Crimes')
plt.ylabel('Number of Suburbs')
plt.title('Total Number of Crimes in Different Suburbs')
plt.show()
```



2. Consider the shape of the histogram, what can you tell? Compare the mean and median values of the plotted histogram.

```
suburb2['Number of Crimes'].mean()
```

```
Out[29]: 841.1143451143452
```

```
suburb2['Number of Crimes'].median()
```

```
Out[30]: 433.0
```

The number of crimes are quite low in most of the suburbs according to the histogram. It's rare for the suburb to have over 5000 crimes. However, there seems to be an outlier with the number of crimes at around 24000. This causes the mean to increase. That's why the mean is so much higher than the median value.

3. In which suburbs the total number of crimes are greater than 5000? Plot the total number of crimes in the suburbs with the highest number of crimes (greater than 5000) using a bar chart.

Suburbs which have total number of crimes greater than 5000:

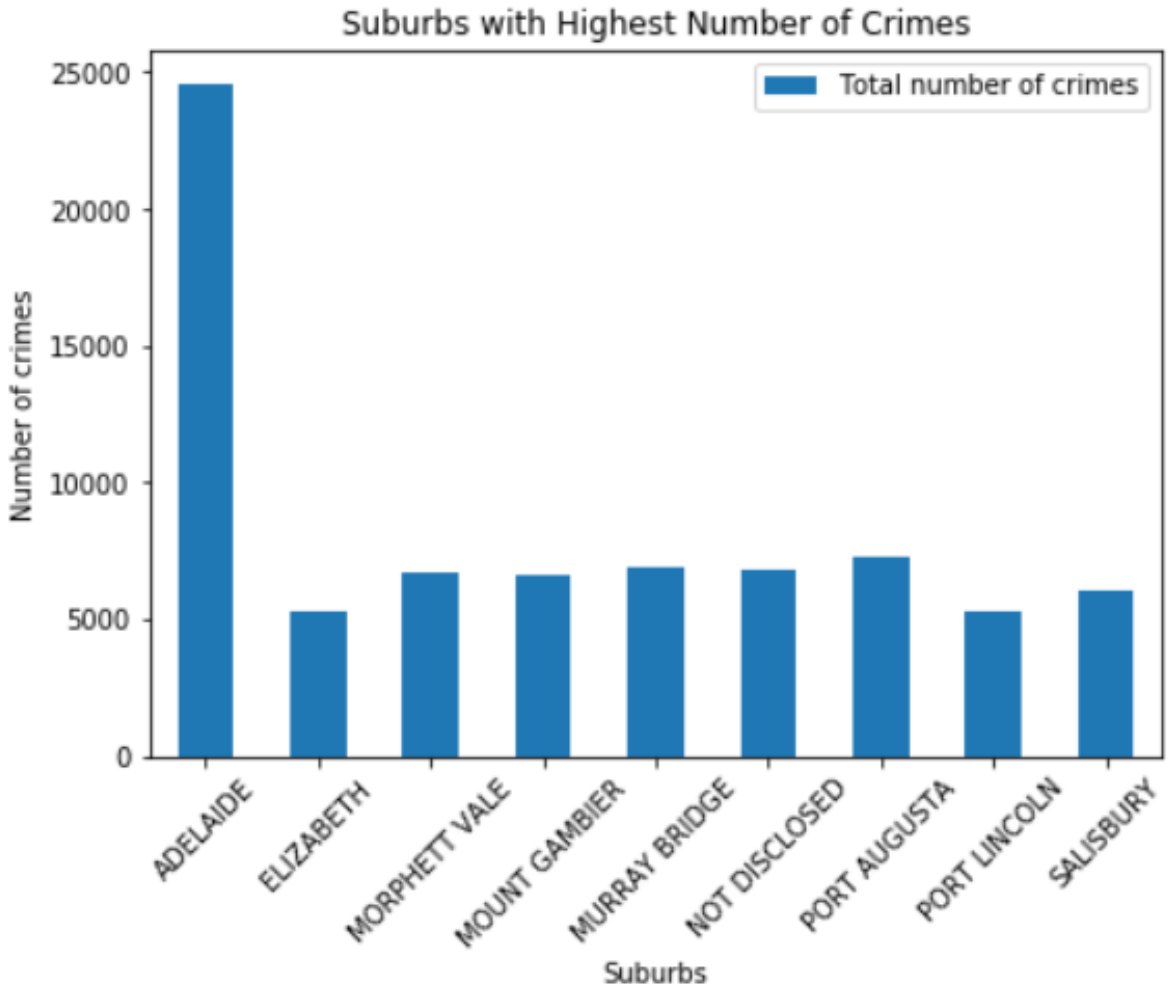
```
high_crime_sub = suburb[suburb['Total number of crimes'] > 5000]
high_crime_sub
```

Out[26]:

	Suburbs	Total number of crimes
2	ADELAIDE	24598.0
382	ELIZABETH	5270.0
879	MORPHETT VALE	6679.0
895	MOUNT GAMBIER	6592.0
930	MURRAY BRIDGE	6928.0
994	NOT DISCLOSED	6772.0
1126	PORT AUGUSTA	7298.0
1139	PORT LINCOLN	5241.0
1235	SALISBURY	6046.0

Bar chart for suburbs with highest number of crimes:

```
bar = high_crime_sub.plot.bar(figsize=(7,5))
bar.set_xticklabels(high_crime_sub['Suburbs'], rotation=45)
plt.xlabel('Suburbs')
plt.ylabel('Number of crimes')
plt.title('Suburbs with Highest Number of Crimes')
plt.show()
```



B3. Daily number of crimes

1. For each suburb, calculate the number of days that at least 15 crimes have occurred per day. (Note: your answer should contain all suburbs in the dataset together with a value showing the number of days that at least 15 crimes have happened)

Suburbs which have at least one day where the daily number of crimes are more than 15:

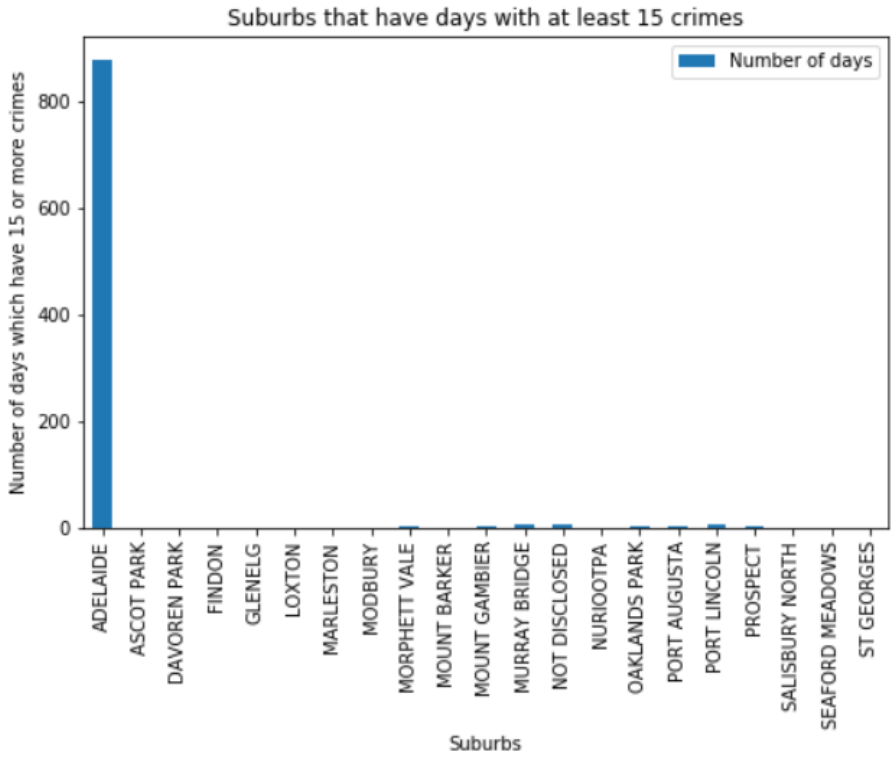
```
crime_count = {'Offence Count':'sum'}
crime_day = crime_statistics.groupby(['Suburb - Incident','Reported Date']).agg(crime_count)
crime_day = crime_day[crime_day['Offence Count']>=15]
crime_day = crime_day.groupby('Suburb - Incident').count()
crime_day = crime_day.reset_index()
crime_day.rename(columns = {'Offence Count':'Number of days'}, inplace=True)
crime_day
```

Out[41]:

	Suburb - Incident	Number of days
0	ADELAIDE	877
1	ASCOT PARK	1
2	DAVOREN PARK	1
3	FINDON	1
4	GLENELG	1
5	LOXTON	1
6	MARLESTON	1
7	MODBURY	1
8	MORPHETT VALE	3
9	MOUNT BARKER	1
10	MOUNT GAMBIER	3
11	MURRAY BRIDGE	5
12	NOT DISCLOSED	5
13	NURIOOTPA	1
14	OAKLANDS PARK	3
15	PORT AUGUSTA	4
16	PORT LINCOLN	5
17	PROSPECT	2
18	SALISBURY NORTH	1
19	SEAFORD MEADOWS	1
20	ST GEORGES	1

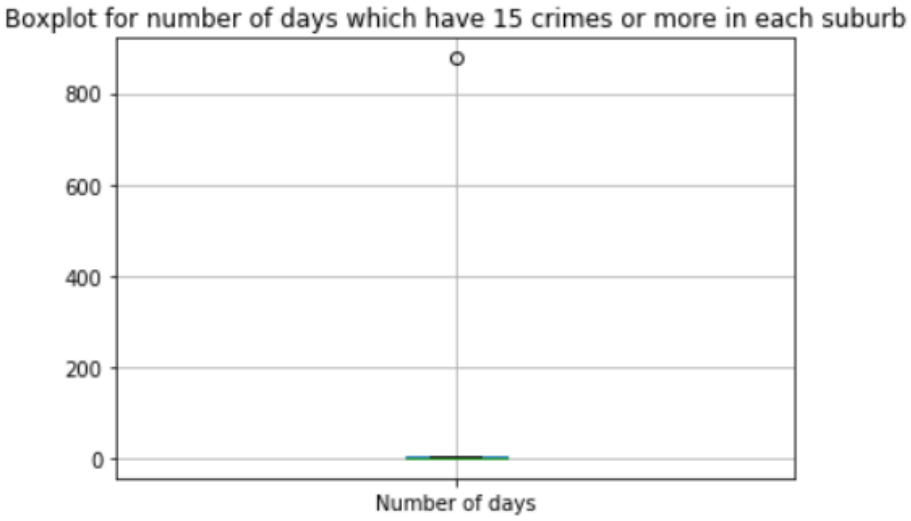
2. Now which suburbs do have at least one day where the daily number of crimes are more than 15. Plot the number of days that at least 15 crimes have occurred for the suburbs you found in this step (step 2) using a bar graph.

```
bar_day = crime_day.plot.bar(figsize=(8,5))
bar_day.set_xticklabels(crime_day['Suburb - Incident'], rotation=90)
plt.xlabel('Suburbs')
plt.ylabel('Number of days which have 15 or more crimes')
plt.title('Suburbs that have days with at least 15 crimes')
plt.show()
```



3. Use an appropriate graph to visualize and detect outliers (extreme values) on the data from step 2 and remove them. Then, plot the data again using a bar graph.

```
crime_day.boxplot(column = 'Number of days')
plt.title('Boxplot for number of days which have 15 crimes or more in each suburb')
plt.show()
```

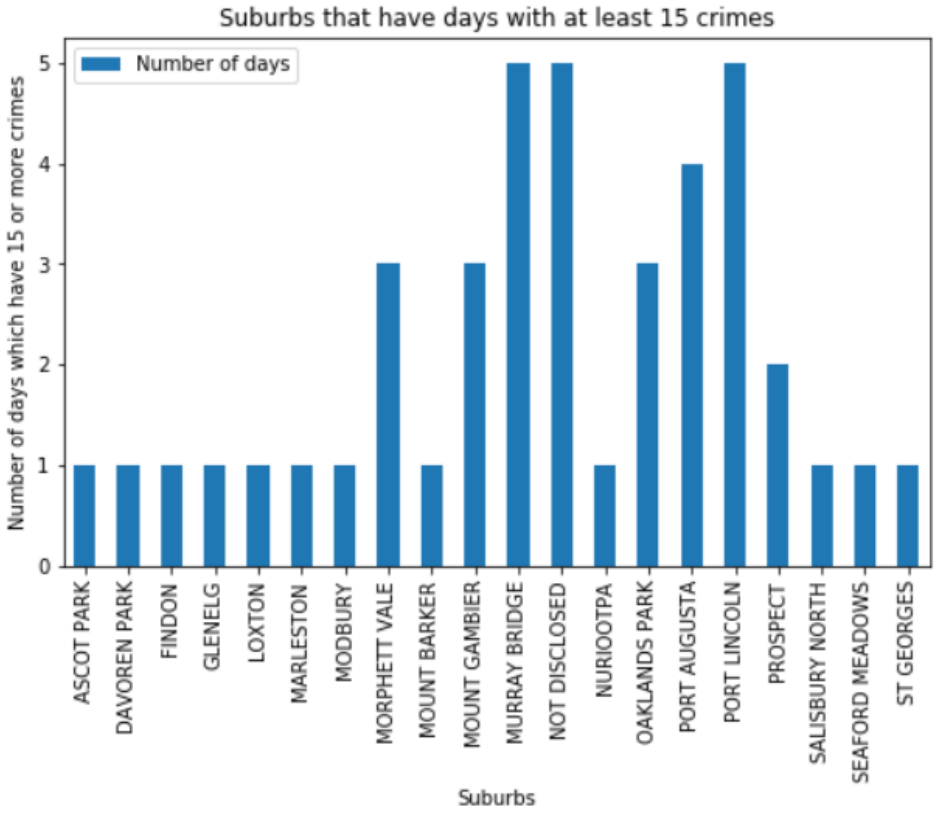


Only outlier is the suburb which has more than 800 days with at least 15 crimes.

Filter to remove outlier then plot bar graph:

```
crime_day2 = crime_day[crime_day['Number of days']<10]

bar_day2 = crime_day2.plot.bar(figsize=(8,5))
bar_day2.set_xticklabels(crime_day2['Suburb - Incident'], rotation=90)
plt.xlabel('Suburbs')
plt.ylabel('Number of days which have 15 or more crimes')
plt.title('Suburbs that have days with at least 15 crimes')
plt.show()
```



4. Compare the bar graphs in step 2 and 3. Which bar graph is easier to interpret? Why?

Bar graph in step 3 is easier to interpret. The majority of the data can be seen here in the bar graph in step 3. whereas in the bar graph from step 2, the outlier was so large that you can hardly see the rest of the data. In the bar graph in step 3, you can distinctly see the difference between the number of days which have at least 15 crimes for each suburb, which makes it easier for comparison.