1. Step of data exploration and preparation.

* **Variable identification**

***Data types of cyber crime database.***



* **Cmd:**

print(cc.columns)

* **Output:**

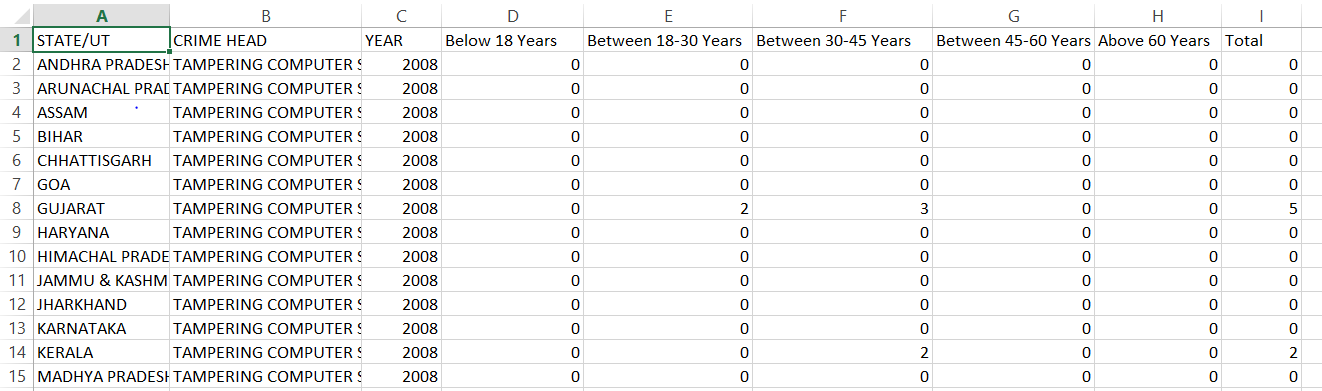
Index(['STATE/UT', 'CRIME HEAD', 'YEAR', 'Below 18 Years',

'Between 18-30 Years', 'Between 30-45 Years', 'Between 45-60 Years',

'Above 60 Years', 'Total'],

dtype='object')

***Display total rows and columns.***



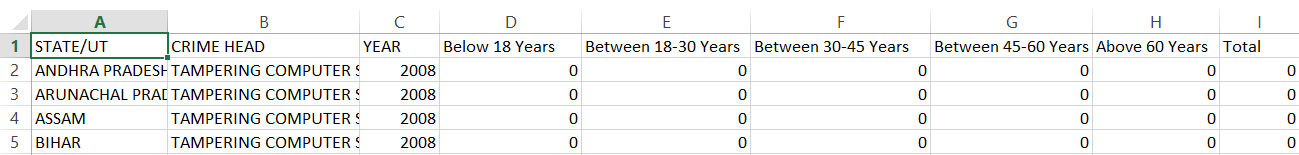
* **Cmd:**

print(cc.shape)

* **Output:**

(4180, 9)

***Display the dataset first five rows***



* **Cmd:**

print(cc.head())

* **Output:**

STATE/UT Total

0 ANDHRA PRADESH ... 0

1 ARUNACHAL PRADESH ... 0

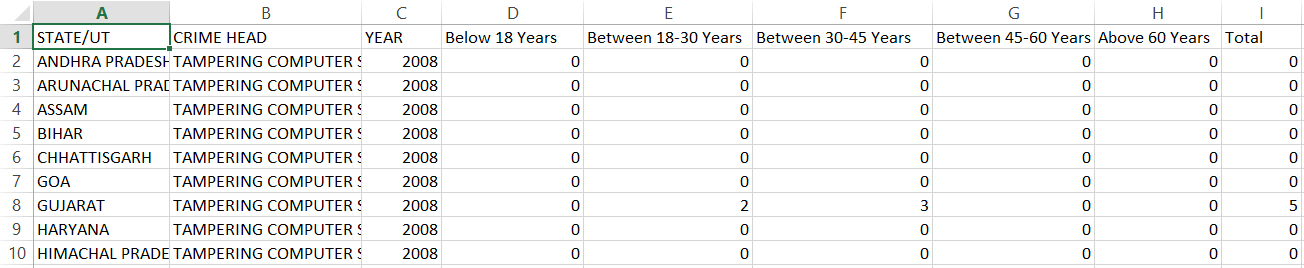
2 ASSAM ... 0

3 BIHAR ... 0

4 CHHATTISGARH ... 0

[5 rows x 9 columns]

***Display the whole dataset records***



* **Cmd:**

print(cc)

* **Output:**

STATE/UT ... Total

0 ANDHRA PRADESH ... 0

1 ARUNACHAL PRADESH ... 0

2 ASSAM ... 0

3 BIHAR ... 0

4 CHHATTISGARH ... 0

5 GOA ... 0

6 GUJARAT ... 5

7 HARYANA ... 0

8 HIMACHAL PRADESH ... 0

9 JAMMU & KASHMIR ... 0

10 JHARKHAND ... 0

11 KARNATAKA ... 0

12 KERALA ... 2

13 MADHYA PRADESH ... 0

14 MAHARASHTRA ... 4

15 MANIPUR ... 0

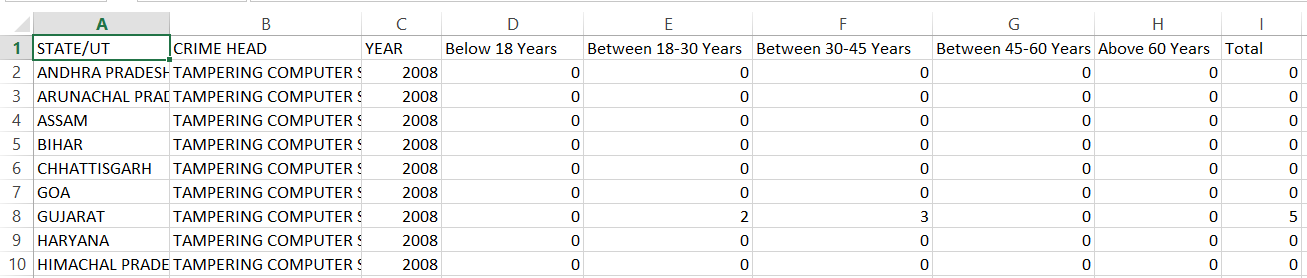
16 MEGHALAYA ... 0

17 MIZORAM ... 0

18 NAGALAND ... 0

19 ORISSA ... 0

20 PUNJAB ... 4

 ***Display the dataset information***

* **Cmd:**

print(cc.info())

* **Output:**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 4180 entries, 0 to 4179

Data columns (total 9 columns):

STATE/UT 4180 non-null object

CRIME HEAD 4180 non-null object

YEAR 4180 non-null int64

Below 18 Years 4180 non-null int64

Between 18-30 Years 4180 non-null int64

Between 30-45 Years 4180 non-null int64

Between 45-60 Years 4180 non-null int64

Above 60 Years 4180 non-null int64

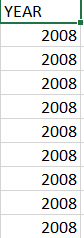
Total 4180 non-null int64

dtypes: int64(7), object(2)

memory usage: 261.3+ KB

None

***Display the dataset single column value***



* **Cmd:**

print(cc.YEAR)

* **Output:**

0 2008

1 2008

2 2008

3 2008

4 2008

5 2008

6 2008

7 2008

8 2008

9 2008

...

4177 2012

4178 2012

4179 2012

Name: YEAR, Length: 4180, dtype: int64

* **Univariate Analysis:-**

Display the dataset mean, median, standard deviation, minimum,maximum, Q1,Q2,Q3

* **Cmd:**

print(cc.describe())

* **Output:**

|  |
| --- |
| YEAR Below 18 Years Above 60 Years Total |
| count 4180.000000 4180.000000 4180.000000 4180.000000 |
| mean 2010.000000 0.202392 0.047368 8.349043 |
| std 1.414383 2.056906 0.404160 58.039943 |
| min 2008.000000 0.000000 0.000000 0.000000 |
| 25% 2009.000000 0.000000 0.000000 0.000000 |
| 50% 2010.000000 0.000000 0.000000 0.000000 |
| 75% 2011.000000 0.000000 0.000000 0.000000 |
| max 2012.000000 65.000000 8.000000 1522.000000 |

Group By:

countries = cc.groupby('Below 18 Years').sum()

print(countries)

output:

YEAR Between 18-30 Years ... Above 60 Years Total

Below 18 Years ...

0.0 8112256 4942 ... 56 10416

1.0 90496 1504 ... 28 3253

2.0 58303 1343 ... 19 2687

3.0 28148 453 ... 6 886

4.0 28144 1079 ... 3 1805

5.0 6031 283 ... 0 469

6.0 6029 332 ... 1 593

7.0 2010 40 ... 0 64

8.0 2011 260 ... 1 487

9.0 8042 445 ... 1 669

10.0 12060 884 ... 9 1432

11.0 2012 31 ... 0 54

12.0 6033 287 ... 16 633

13.0 6035 201 ... 5 344

14.0 6034 616 ... 4 982

15.0 2012 81 ... 0 151

16.0 8042 1214 ... 29 2426

21.0 4024 752 ... 4 1203

23.0 2011 686 ... 5 1161

25.0 2011 695 ... 5 1184

37.0 4024 555 ... 0 992

65.0 4024 1836 ... 6 3008

[22 rows x 6 columns]

Sort the values and get the first 10

countries=cc.sort\_values(by = 'STATE/UT',ascending=True)[1:11]

print(countries)

output:

STATE/UT ... Total

2271 A & N ISLANDS ... 0

2917 A & N ISLANDS ... 0

143 A & N ISLANDS ... 0

3981 A & N ISLANDS ... 0

1093 A & N ISLANDS ... 0

1359 A & N ISLANDS ... 0

333 A & N ISLANDS ... 0

3145 A & N ISLANDS ... 0

1435 A & N ISLANDS ... 0

2233 A & N ISLANDS ... 0

[10 rows x 9 columns]

Lower case

cc["STATE/UT"]=cc["STATE/UT"].str.lower()

print(cc)

output:

STATE/UT ... Total

0 andhra pradesh ... 0

1 arunachal pradesh ... 0

2 assam ... 0

3 bihar ... 0

4 chhattisgarh ... 0

5 goa ... 0

6 gujarat ... 5

7 haryana ... 0

8 himachal pradesh ... 0

9 jammu & kashmir ... 0

10 jharkhand ... 0

11 karnataka ... 0

12 kerala ... 2

13 madhya pradesh ... 0

14 maharashtra ... 4

15 manipur ... 0

16 meghalaya ... 0

17 mizoram ... 0

18 nagaland ... 0

19 orissa ... 0

20 punjab ... 4

21 rajasthan ... 1

22 sikkim ... 0

23 tamil nadu ... 0

24 tripura ... 0

Upper case

cc["STATE/UT"]=cc["STATE/UT"].str.upper()

print(cc)

output:

STATE/UT ... Total

0 ANDHRA PRADESH ... 0

1 ARUNACHAL PRADESH ... 0

2 ASSAM ... 0

3 BIHAR ... 0

4 CHHATTISGARH ... 0

5 GOA ... 0

6 GUJARAT ... 5

7 HARYANA ... 0

8 HIMACHAL PRADESH ... 0

9 JAMMU & KASHMIR ... 0

10 JHARKHAND ... 0

11 KARNATAKA ... 0

12 KERALA ... 2

13 MADHYA PRADESH ... 0

14 MAHARASHTRA ... 4

15 MANIPUR ... 0

Replace missing values with a number

df['Below 18 Years'].fillna(0, inplace=True)