Assignment 3: Data Exploration & Python

Title: Data Exploration & **Python**

Initially we read the dataset using,

```
#to read the dataset
tordata_orig = pd.read_csv("1950-2020_actual_tornadoes.csv")
tordata = pd.read_csv("1950-2020_actual_tornadoes.csv")
```

'tordata' is the dataframe name I have given. The command tordata displays a few data.

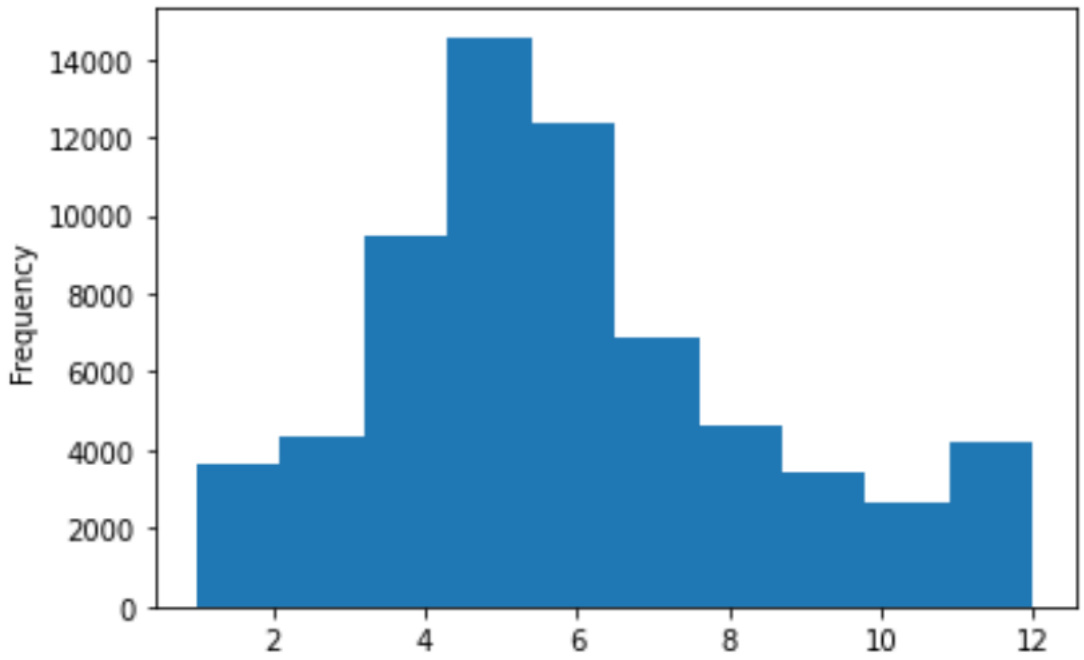
```
In [141]: tordata
Out[141]:
           om    yr  mo  dy        date      time  ...  sg   f1  f2  f3  f4  fc
0         192  1950  10   1  1950-10-01  21:00:00  ...   1   25   0   0   0   0
1         193  1950  10   9  1950-10-09  02:15:00  ...   1   47   0   0   0   0
2         195  1950  11  20  1950-11-20  02:20:00  ...   1  177   0   0   0   0
3         196  1950  11  20  1950-11-20  04:00:00  ...   1  209   0   0   0   0
4         197  1950  11  20  1950-11-20  07:30:00  ...   1  101   0   0   0   0
...       ...   ...  ..  ..         ...       ...  ...  ..  ...  ..  ..  ..  ..
66239  619522  2020   9   1  2020-09-01  18:10:00  ...   1  191   0   0   0   0
66240  619523  2020   9   3  2020-09-03  15:57:00  ...   1    3   0   0   0   0
66241  619524  2020   9   5  2020-09-05  16:17:00  ...   1   39   0   0   0   0
66242  619525  2020   9   5  2020-09-05  18:29:00  ...   1   19   0   0   0   0
66243  619526  2020   9   7  2020-09-07  15:14:00  ...   1   41   0   0   0   0

[66244 rows x 29 columns]
```

tordata.dtypes gives all the data types and tordata.mo displays all months.

```
In [142]: tordata.dtypes
Out[142]:
om          int64
yr          int64
mo          int64
dy          int64
date       object
time       object
tz          int64
st         object
stf         int64
stn         int64
mag         int64
inj         int64
fat         int64
loss      float64
closs     float64
slat      float64
slon      float64
elat      float64
elon      float64
len       float64
wid         int64
ns          int64
sn          int64
```

```
In [143]: tordata.mo
Out[143]:
0         10
1         10
2         11
3         11
4         11
         ..
66239      9
66240      9
66241      9
66242      9
66243      9
Name: mo, Length: 66244, dtype: int64
```

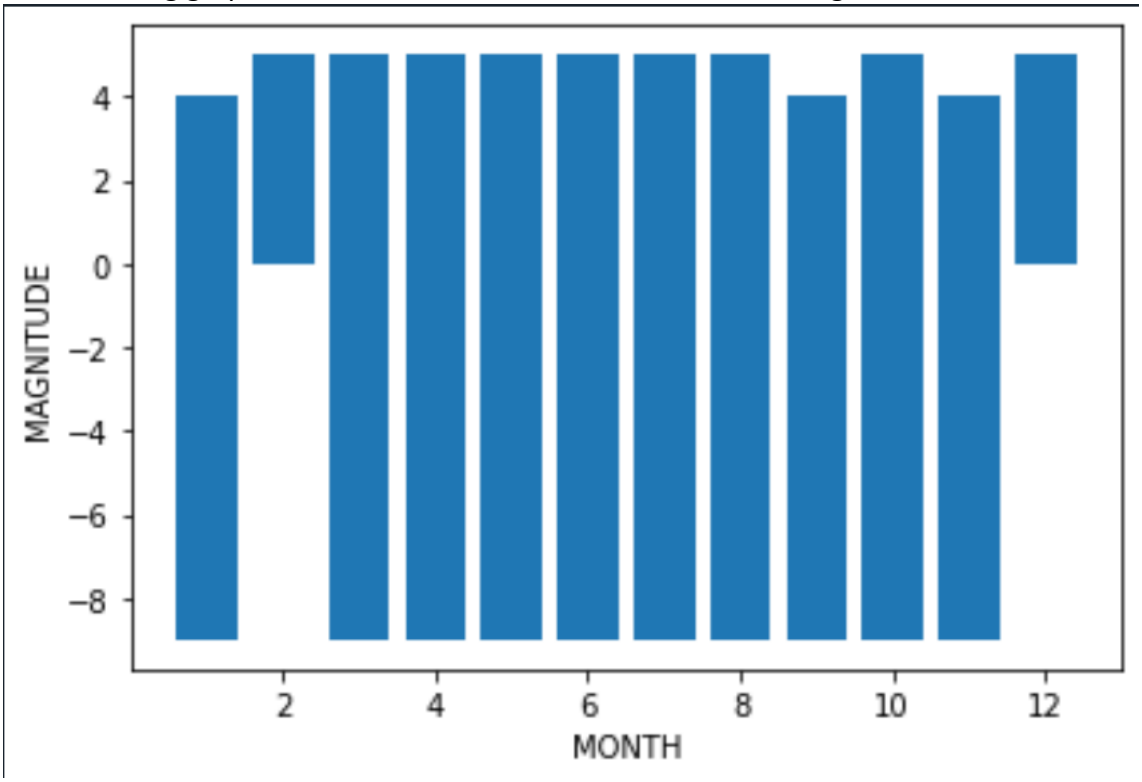
Histogram to show month,



sort_values displays the values in ascending order. Here, month is arranged.

```
In [146]: tordata.sort_values("mo")
Out[146]:
           om    yr  mo  dy        date      time  ...  sg   f1   f2  f3  f4  fc
28469       4  1990   1  19  1990-01-19  15:40:00  ...   1  455    5   0   0   0
57342  428331  2013   1  30  2013-01-30  02:32:00  ...   1  181  101   0   0   0
57343  428698  2013   1  30  2013-01-30  02:45:00  ...   1   43   21   0   0   0
57344  428696  2013   1  30  2013-01-30  02:52:00  ...   1   21    0   0   0   0
57345  432103  2013   1  30  2013-01-30  02:59:00  ...   1  147    0   0   0   0
...       ...   ...  ..  ..         ...       ...  ...  ..  ...  ...  ..  ..  ..
17750     833  1977  12   5  1977-12-05  08:40:00  ...   1  107    0   0   0   0
17751     834  1977  12   5  1977-12-05  10:20:00  ...   1   77    0   0   0   0
17752     835  1977  12   5  1977-12-05  12:30:00  ...   1  107    0   0   0   0
17754     837  1977  12   5  1977-12-05  13:42:00  ...   1   21    0   0   0   0
59158  612738  2015  12  28  2015-12-28  08:30:00  ...   1  113    0   0   0   0

[66244 rows x 29 columns]
```

The following graph shows the relation between month and magnitude,



Clearly, from the below command we get OH has the highest average injuries(inj).

```
plt.ylabel('MAGNITUDE')
plt.bar(x,y)
plt.show()

#groups two or more values for better analysis
tordata.groupby(['mo'])[['mag']].sum()
tordata.groupby(['st'])[['inj','fat']].sum()
tordata.groupby(['st'])['inj'].mean()
tordata.groupby(['st'])['fat'].mean()
tordata.groupby(['st'])[['inj','fat']].mean()

#explains how many times the state has occured
torstate=tordata.groupby(['st']).size()
print(torstate)

#bar graph to show state and number of tornados
x=tordata['st']    #workssss
y=tordata['stn']
plt.xticks(fontsize=5,rotation=90)
plt.xlabel('STATE')
plt.ylabel('NUMBER OF TORNADOES')
plt.bar(x,y)
plt.show()
```

Console 1/A

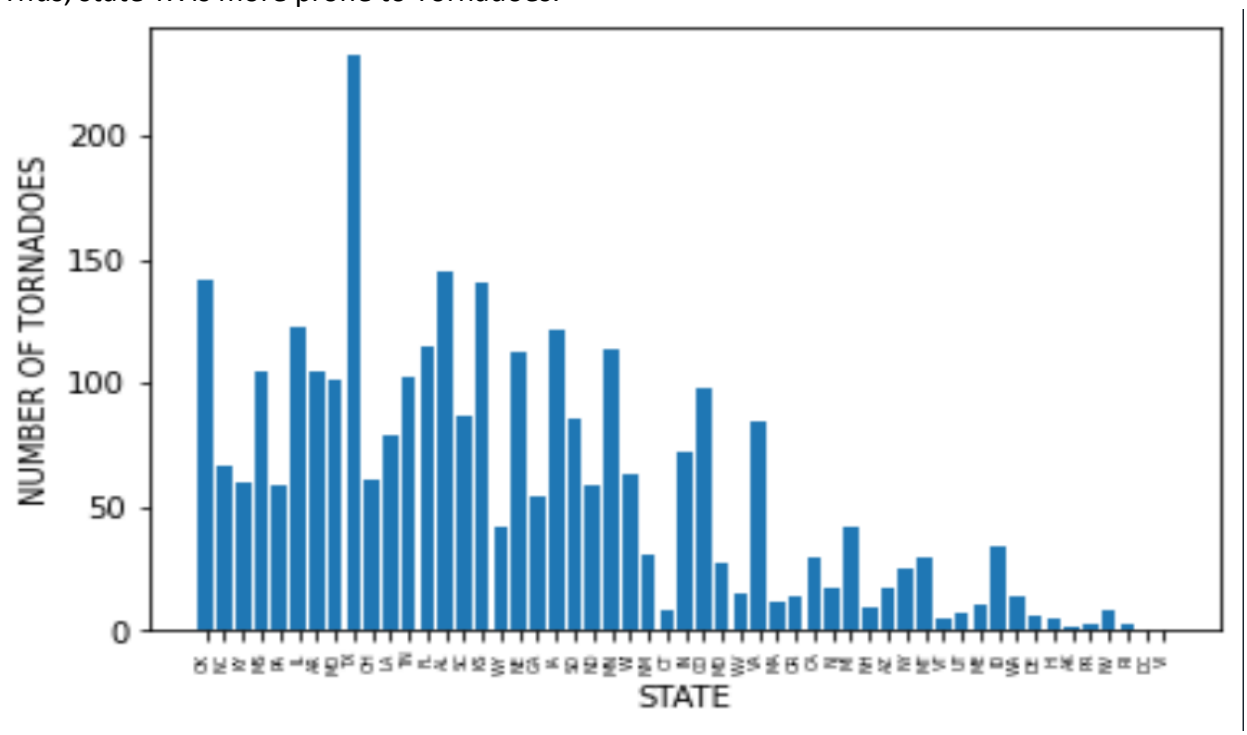
```
IN    2.979003
KS    0.618949
KY    3.197866
LA    1.548298
MA    8.820809
MD    0.839895
ME    0.149254
MI    2.844255
MN    1.006119
MO    1.692566
MS    2.686381
MT    0.074419
NC    1.684844
ND    0.228464
NE    0.429353
NH    0.319149
NJ    0.402516
NM    0.260518
NV    0.021505
NY    0.762931
OH    4.040343
OK    1.488210
OR    2.520661
PA    1.207218
PR    0.035714
RI    1.916667
```

Also, the maximum average fatalities (fat) is MA. The mean() gives the average of all values.

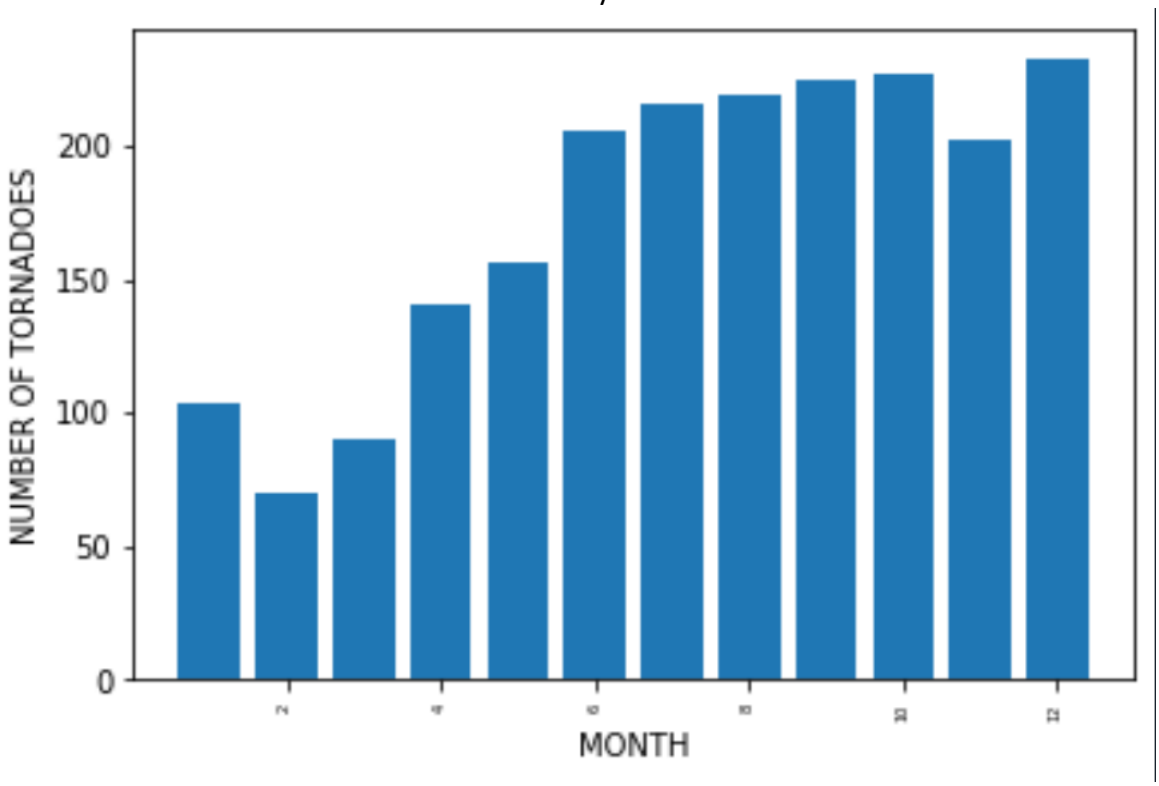
```
#groups two or more values for better analysis
tordata.groupby(['mo'])[['mag']].sum()
tordata.groupby(['st'])[['inj','fat']].sum()
tordata.groupby(['st'])['inj'].mean()
tordata.groupby(['st'])['fat'].mean()
tordata.groupby(['st'])[['inj','fat']].mean()

#explains how many times the state has occured
torstate=tordata.groupby(['st']).size()
print(torstate)

#bar graph to show state and number of tornados
x=tordata['st']    #workssss
y=tordata['stn']
plt.xticks(fontsize=5,rotation=90)
plt.xlabel('STATE')
plt.ylabel('NUMBER OF TORNADOES')
plt.bar(x,y)
plt.show()
```

```
IL    0.083940
IN    0.175853
KS    0.044721
KY    0.171678
LA    0.100736
MA    0.606936
MD    0.023622
ME    0.007463
MI    0.205128
MN    0.049975
MO    0.146577
MS    0.197834
MT    0.009302
NC    0.084278
ND    0.017478
NE    0.020527
NH    0.010638
NJ    0.006289
NM    0.006472
NV    0.000000
NY    0.066810
OH    0.193991
OK    0.108712
OR    0.049587
```

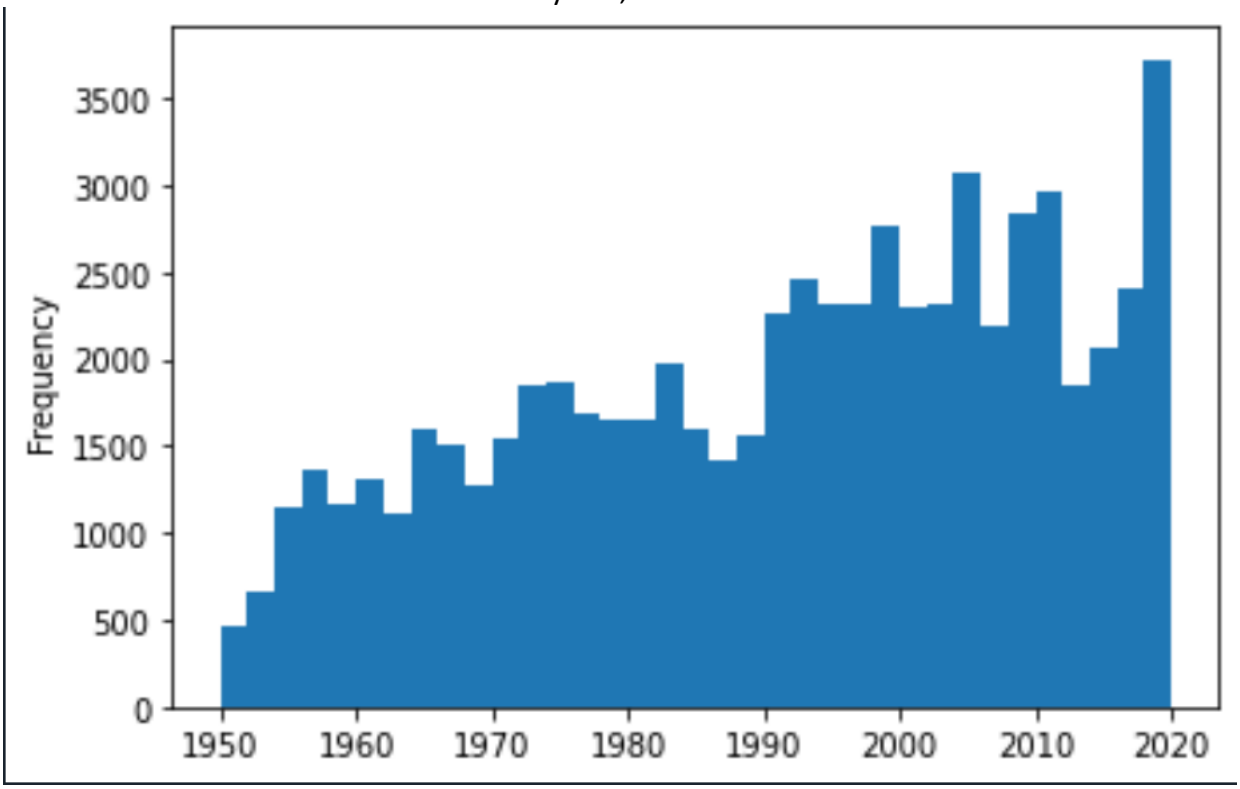
Clearly from the chart, we can see TX (13%) has more tornadoes followed by KS ,OK and FL. Thus, state TX is more prone to Tornadoes.



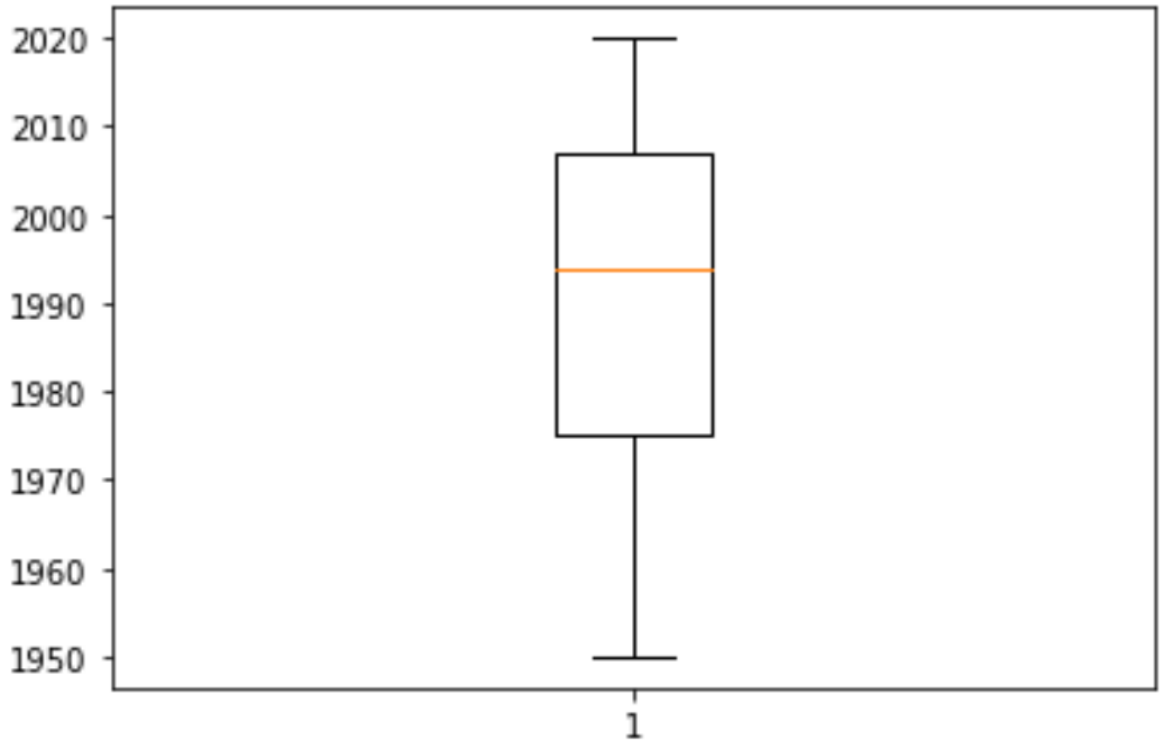
From the chart, we see December has the highest Tornadoes. We can also understand that there are more Tornadoes towards the end of year.



Looks like Tornadoes have increased over years,



Boxplot to show year,



To sum it all, the state with more Tornadoes is TX. December is more prone to Tornadoes. MA and OH have high average fat and inj respectively. Analytics is a very important feature. We were able to retrieve such useful information only because we collected to necessary information. Incase this data wasn't collected, we wouldn't have known the Tornado-prone month and State. Luckily, we have data to analyze, so we can be more aware at that time and take pre-cautions.