

DateTime

November 20, 2016

```
In [1]: import pandas as pd
```

```
In [2]: ufo = pd.read_csv('http://bit.ly/uforeports')
```

```
In [3]: ufo.head(2)
```

```
Out[3]:
```

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00

```
In [4]: ufo.dtypes # time column is object => string
```

```
Out[4]: City                object
Colors Reported            object
Shape Reported             object
State                     object
Time                      object
dtype: object
```

```
In [6]: # meh => not the best methodology to do this
ufo.Time.str.slice(-5, -3).astype(int).head(3)
```

```
Out[6]: 0    22
1    20
2    14
Name: Time, dtype: int64
```

```
In [7]: ufo['Time'] = pd.to_datetime(ufo.Time)
```

```
In [8]: ufo.head(4)
```

```
Out[8]:
```

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	1930-06-01 22:00:00
1	Willingboro	NaN	OTHER	NJ	1930-06-30 20:00:00
2	Holyoke	NaN	OVAL	CO	1931-02-15 14:00:00
3	Abilene	NaN	DISK	KS	1931-06-01 13:00:00

```
In [9]: # Time is now datetime format
ufo.dtypes
```

```
Out [9]: City                object
        Colors Reported      object
        Shape Reported       object
        State                 object
        Time                  datetime64[ns]
        dtype: object
```

```
In [14]: ufo.Time.dt.hour.head(5)
```

```
Out [14]: 0    22
          1    20
          2    14
          3    13
          4    19
          Name: Time, dtype: int64
```

```
In [13]: ufo.Time.dt.weekday_name.head(5)
```

```
Out [13]: 0    Sunday
          1    Monday
          2    Sunday
          3    Monday
          4    Tuesday
          Name: Time, dtype: object
```

```
In [12]: ufo.Time.dt.dayofyear.head(5)
```

```
Out [12]: 0    152
          1    181
          2     46
          3    152
          4    108
          Name: Time, dtype: int64
```

```
In [20]: ts = pd.to_datetime('1/1/1999') # outputs Timestamp => assigned to ts
        ms = pd.to_datetime('12/30/2000')
```

```
In [21]: ufo.loc[ufo.Time >= ms, :] # => give me dates later than ms => 12/30/2000
```

```
Out [21]:
```

	City	Colors Reported	Shape Reported	State	
18225	Simi Valley	NaN	FORMATION	CA	2000-12-30 10:0
18226	San Francisco	NaN	FORMATION	CA	2000-12-30 22:0
18227	San Francisco	NaN	TRIANGLE	CA	2000-12-30 22:0
18228	Kingsville	NaN	LIGHT	TX	2000-12-31 04:0
18229	Chicago	NaN	DISK	IL	2000-12-31 11:4
18230	Pismo Beach	NaN	OVAL	CA	2000-12-31 20:0
18231	Pismo Beach	NaN	OVAL	CA	2000-12-31 20:0
18232	Lodi	NaN	NaN	WI	2000-12-31 20:3
18233	Anchorage	RED	VARIOUS	AK	2000-12-31 21:0

18234	Capitola	NaN	TRIANGLE	CA	2000-12-31 22:0
18235	Fountain Hills	NaN	NaN	AZ	2000-12-31 23:0
18236	Grant Park	NaN	TRIANGLE	IL	2000-12-31 23:0
18237	Spirit Lake	NaN	DISK	IA	2000-12-31 23:0
18238	Eagle River	NaN	NaN	WI	2000-12-31 23:4
18239	Eagle River	RED	LIGHT	WI	2000-12-31 23:4
18240	Ybor	NaN	OVAL	FL	2000-12-31 23:5

```
In [22]: # math operations with datetime
ufo.Time.max() # gives latest timestamp in series
```

```
Out[22]: Timestamp('2000-12-31 23:59:00')
```

```
In [23]: # max - min
ufo.Time.max() - ufo.Time.min()
```

```
Out[23]: Timedelta('25781 days 01:59:00')
```

```
In [24]: # max - min => .days pulls out number of days => this is AWESOME!!!
(ufo.Time.max() - ufo.Time.min()).days
```

```
Out[24]: 25781
```

```
In [25]: %matplotlib inline
```

```
/opt/conda/lib/python3.5/site-packages/matplotlib/font_manager.py:273: UserWarning:
  warnings.warn('Matplotlib is building the font cache using fc-list. This may take
/opt/conda/lib/python3.5/site-packages/matplotlib/font_manager.py:273: UserWarning:
  warnings.warn('Matplotlib is building the font cache using fc-list. This may take
```

```
In [26]: ufo['Year'] = ufo.Time.dt.year
```

```
In [27]: ufo.head()
```

```
Out[27]:
```

	City	Colors	Reported	Shape	Reported	State	\
0	Ithaca	NaN	NaN	TRIANGLE	NaN	NY	
1	Willingboro	NaN	NaN	OTHER	NaN	NJ	
2	Holyoke	NaN	NaN	OVAL	NaN	CO	
3	Abilene	NaN	NaN	DISK	NaN	KS	
4	New York Worlds Fair	NaN	NaN	LIGHT	NaN	NY	

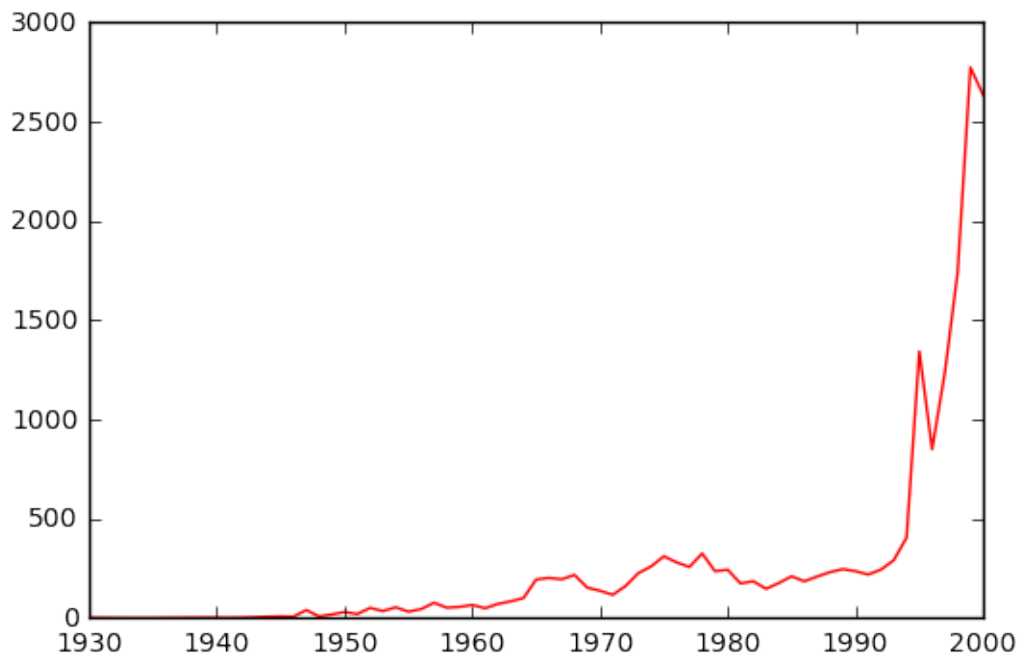
	Time	Year
0	1930-06-01 22:00:00	1930
1	1930-06-30 20:00:00	1930
2	1931-02-15 14:00:00	1931
3	1931-06-01 13:00:00	1931
4	1933-04-18 19:00:00	1933

```
In [30]: ufo.Year.value_counts().sort_index().head()
```

```
Out[30]: 1930    2
         1931    2
         1933    1
         1934    1
         1935    1
         Name: Year, dtype: int64
```

```
In [34]: # line plot by default of sitings by year, c = colour
         ufo.Year.value_counts().sort_index().plot(c='red')
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8486a516a0>
```



```
In [ ]:
```