How to handle Date Time?

Data: TaxiFare.csv

Your column is like this

tf - DataFrame

Index	unique_id	amount	date_time_of_pickup
0	26:21.0	4.50	2009-06-15 17:26:21 UTC
1	52:16.0	16.90	2010-01-05 16:52:16 UTC
2	35:00.0	5.70	2011-08-18 00:35:00 UTC
3	30:42.0	7.70	2012-04-21 04:30:42 UTC
4	51:00.0	5.30	2010-03-09 07:51:00 UTC

And, column is an ‘object’!

```
In [3]: tf.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   unique_id             50000 non-null  object
 1   amount                50000 non-null  float64
 2   date_time_of_pickup   50000 non-null  object
 3   longitude_of_pickup   50000 non-null  float64
 4   latitude_of_pickup    50000 non-null  float64
 5   longitude_of_dropoff  50000 non-null  float64
 6   latitude_of_dropoff   50000 non-null  float64
 7   no_of_passenger       50000 non-null  int64
dtypes: float64(5), int64(1), object(2)
memory usage: 3.1+ MB
```

Describe and Head

```
In [4]: tf.date_time_of_pickup.describe()
Out[4]:
count                          50000
unique                         49555
top       2011-09-03 01:30:00 UTC
freq                               3
Name: date_time_of_pickup, dtype: object
```

```
In [5]: tf.date_time_of_pickup.head()
Out[5]:
0    2009-06-15 17:26:21 UTC
1    2010-01-05 16:52:16 UTC
2    2011-08-18 00:35:00 UTC
3    2012-04-21 04:30:42 UTC
4    2010-03-09 07:51:00 UTC
Name: date_time_of_pickup, dtype: object
```

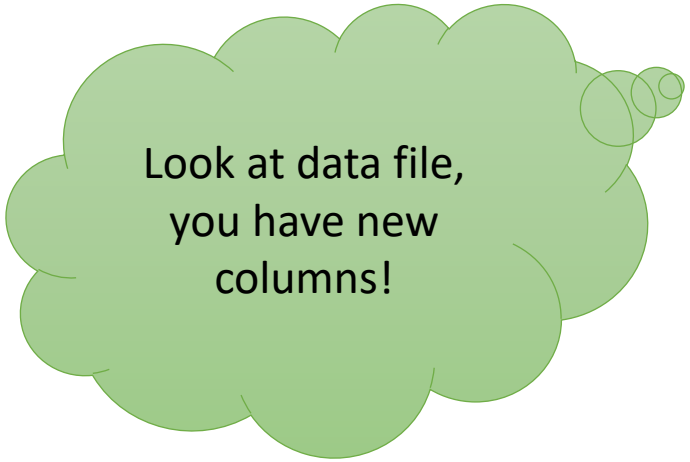
```
In [7]: tf.date_time_of_pickup = pd.to_datetime(tf.date_time_of_pickup)

In [8]: tf.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   unique_id             50000 non-null  object
 1   amount                50000 non-null  float64
 2   date_time_of_pickup   50000 non-null  datetime64[ns, UTC]
 3   longitude_of_pickup   50000 non-null  float64
 4   latitude_of_pickup    50000 non-null  float64
 5   longitude_of_dropoff  50000 non-null  float64
 6   latitude_of_dropoff   50000 non-null  float64
 7   no_of_passenger       50000 non-null  int64
dtypes: datetime64[ns, UTC](1), float64(5), int64(1), object(1)
memory usage: 3.1+ MB

In [9]: tf.date_time_of_pickup.head()
Out[9]:
0   2009-06-15 17:26:21+00:00
1   2010-01-05 16:52:16+00:00
2   2011-08-18 00:35:00+00:00
3   2012-04-21 04:30:42+00:00
4   2010-03-09 07:51:00+00:00
Name: date_time_of_pickup, dtype: datetime64[ns, UTC]
```

First, convert into 'datetime' data type

Check through info and head

Now extract year, month, day and hour

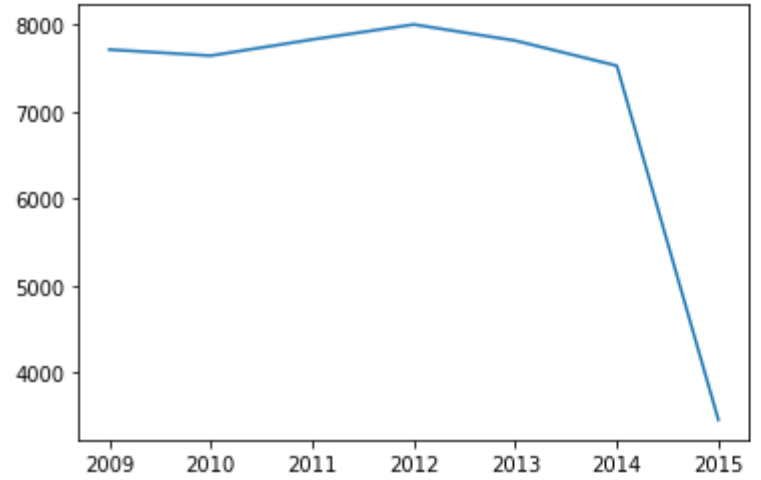
```
In [11]: tf['year'] = tf["date_time_of_pickup"].dt.year

In [12]: tf['month'] = tf["date_time_of_pickup"].dt.month_name()

In [13]: tf['day'] = tf["date_time_of_pickup"].dt.day_name()

In [14]: tf['hour'] = tf["date_time_of_pickup"].dt.hour #24hours time
```



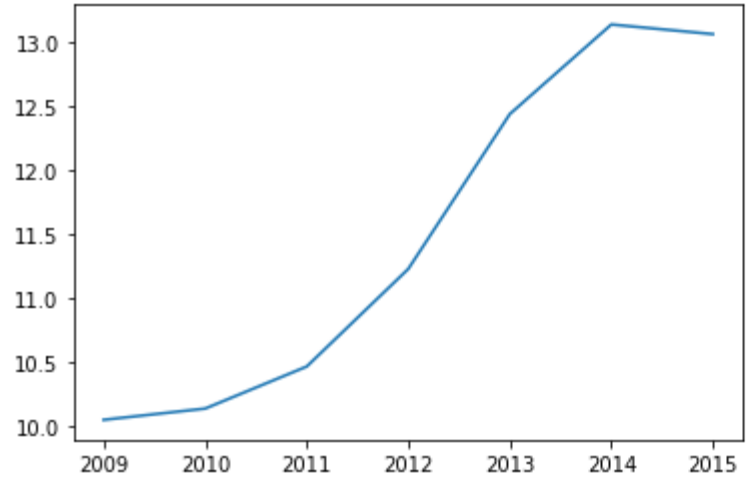
year	month	day	hour
2009	June	Monday	17
2010	January	Tuesday	16
2011	August	Thursday	0
2012	April	Saturday	4
2010	March	Tuesday	7
2011	January	Thursday	9
2012	November	Tuesday	20
2012	January	Wednesday	17
2012	December	Monday	13
2009	September	Wednesday	1
2012	April	Sunday	7
2012	December	Monday	11

Now you can do your regular EDA!


```
In [16]: plt.plot(tf.groupby('year')['amount'].count())
Out[16]: [<matplotlib.lines.Line2D at 0x2a2e95fbf10>]
```




```
In [17]: plt.plot(tf.groupby('year')['amount'].mean())
Out[17]: [<matplotlib.lines.Line2D at 0x2a2e8d1f7c0>]
```



HAPPINESS IS ...

... learning new skills