Handle the Date and Time formating uses Various Method

- Extract the month
- Ectract the year
- extract the day
- Perform a various type of method in date and time data set

In [1]:
```
# import the operating system
import os
os.getcwdb()
```

Out[1]:
```
b'C:\\Users\\ap983\\Desktop\\Pandey  IMp\\COMPLETE PYTHON PROJECT\\5. ORDER A
ND TIME PROJECT'
```

In [2]:
```
## import all libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
## read csv file
data = pd.read_csv('C:/Users/ap983/Desktop/Pandey  IMp/COMPLETE PYTHON PROJECT/
```

In [4]:
```
# top five rocord in the dataset
data.head()
```

Out[4]:

	date	msg
0	2013-12-15 00:50:00	ищу на сегодня мужика 37
1	2014-04-29 23:40:00	ПАРЕНЬ БИ ИЩЕТ ДРУГА СЕЙЧАС!! СМС ММС 0955532826
2	2012-12-30 00:21:00	Днепр.м 43 позн.с д/ж *.о 067.16.34.576
3	2014-11-28 00:31:00	КИЕВ ИЩУ Д/Ж ДО 45 МНЕ СЕЙЧАС СКУЧНО 093 629 9...
4	2013-10-26 23:11:00	Зая я тебя никогда не обижу люблю тебя!) Даше

In [5]:
```
# read second csv file
data1 = pd.read_csv('C:/Users/ap983/Desktop/Pandey  IMp/COMPLETE PYTHON PROJECT
```

In [6]:
```
data1.head()
```

Out[6]:

	date	product_id	city_id	orders
0	2019-12-10	5628	25	3
1	2018-08-15	3646	14	157
2	2018-10-23	1859	25	1
3	2019-08-17	7292	25	1
4	2019-01-06	4344	25	3

In [7]:
```
# check the last five record
data.tail()
```

Out[7]:

	date	msg
995	2012-03-16 00:50:00	ПАРЕНЬ СДЕЛАЕТ МАССАЖ ЖЕНЩИНАМ -066-877-32-44
996	2014-01-23 23:14:00	сельский п 23 ищу девушку для отношений
997	2012-10-15 23:37:00	Д+Д ДЛЯ серьезных отношений. Мой номер 093-156...
998	2012-06-21 23:34:00	7 ДНЕПР М.34 ПОЗ.С Д/Ж ДЛЯ ВСТРЕЧ.Т.098 809 15 14
999	2014-06-19 23:25:00	Парень поласкает девушке... т.0662035584

In [8]:
```
## check the last five record
data1.tail()
```

Out[8]:

	date	product_id	city_id	orders
995	2018-10-08	255	13	1
996	2018-12-06	5521	7	1
997	2019-05-07	487	26	14
998	2019-03-03	1503	21	2
999	2019-10-15	6371	7	22

In [9]:
```
## check the shape
data.shape
```

Out[9]:
```
(1000, 2)
```

In [10]:
```
## check the shape of the second data
data1.shape
```

Out[10]:
```
(1000, 4)
```

In [11]:
```
## check the columns name
data.columns
```

Out[11]:
```
Index(['date', 'msg'], dtype='object')
```

In [12]:
```
## check the columns name
data1.columns
```

Out[12]:
```
Index(['date', 'product_id', 'city_id', 'orders'], dtype='object')
```

In [13]:
```
## check the data information
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   date    1000 non-null   object
 1   msg     1000 non-null   object
dtypes: object(2)
memory usage: 15.8+ KB
```

In [14]:
```
## check the second data info
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   date        1000 non-null   object
 1   product_id  1000 non-null   int64 
 2   city_id     1000 non-null   int64 
 3   orders      1000 non-null   int64 
dtypes: int64(3), object(1)
memory usage: 31.4+ KB
```

In [15]:
```
# change the date columns object to date
data1['date'] = pd.to_datetime(data1['date'])
```

In [16]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   date        1000 non-null   datetime64[ns]
 1   product_id  1000 non-null   int64         
 2   city_id     1000 non-null   int64         
 3   orders      1000 non-null   int64         
dtypes: datetime64[ns](1), int64(3)
memory usage: 31.4 KB
```

In [17]:
```
data1.head()
```

Out[17]:

	date	product_id	city_id	orders
0	2019-12-10	5628	25	3
1	2018-08-15	3646	14	157
2	2018-10-23	1859	25	1
3	2019-08-17	7292	25	1
4	2019-01-06	4344	25	3

In [18]:
```
# extract the year
data1['year'] = data1['date'].dt.year
```

In [19]:
```
data1.head()
```

Out[19]:

	date	product_id	city_id	orders	year
0	2019-12-10	5628	25	3	2019
1	2018-08-15	3646	14	157	2018
2	2018-10-23	1859	25	1	2018
3	2019-08-17	7292	25	1	2019
4	2019-01-06	4344	25	3	2019

In [20]:
```
## extract the month
data1['Months'] = data1['date'].dt.month
```

In [21]:
```
data1.head()
```

Out[21]:

	date	product_id	city_id	orders	year	Months
0	2019-12-10	5628	25	3	2019	12
1	2018-08-15	3646	14	157	2018	8
2	2018-10-23	1859	25	1	2018	10
3	2019-08-17	7292	25	1	2019	8
4	2019-01-06	4344	25	3	2019	1

In [22]:
```
data1['Months_name'] = data1['date'].dt.month_name()
```

In [23]:
```
data1.head()
```

Out[23]:

	date	product_id	city_id	orders	year	Months	Months_name
0	2019-12-10	5628	25	3	2019	12	December
1	2018-08-15	3646	14	157	2018	8	August
2	2018-10-23	1859	25	1	2018	10	October
3	2019-08-17	7292	25	1	2019	8	August
4	2019-01-06	4344	25	3	2019	1	January

In [24]:
```
# extract the days
data1['date_day'] = data1['date'].dt.day
```

In [25]:
```
data1.head()
```

Out[25]:

	date	product_id	city_id	orders	year	Months	Months_name	date_day
0	2019-12-10	5628	25	3	2019	12	December	10
1	2018-08-15	3646	14	157	2018	8	August	15
2	2018-10-23	1859	25	1	2018	10	October	23
3	2019-08-17	7292	25	1	2019	8	August	17
4	2019-01-06	4344	25	3	2019	1	January	6

In [26]:
```
# check tha week
data1['date_dow'] = data1['date'].dt.dayofweek
data1.head()
```

Out[26]:

	date	product_id	city_id	orders	year	Months	Months_name	date_day	date_dow
0	2019-12-10	5628	25	3	2019	12	December	10	1
1	2018-08-15	3646	14	157	2018	8	August	15	2
2	2018-10-23	1859	25	1	2018	10	October	23	1
3	2019-08-17	7292	25	1	2019	8	August	17	5
4	2019-01-06	4344	25	3	2019	1	January	6	6

In [27]:
```
# extract the number of week
data1['date_week'] = data1['date'].dt.weekofyear
data1.head(10)
```

```
C:\Users\ap983\AppData\Local\Temp\ipykernel_5248\3505791224.py:2: FutureWarni
ng: Series.dt.weekofyear and Series.dt.week have been deprecated. Please use
Series.dt.isocalendar().week instead.
  data1['date_week'] = data1['date'].dt.weekofyear
```

Out[27]:

	date	product_id	city_id	orders	year	Months	Months_name	date_day	date_dow	date_we
0	2019-12-10	5628	25	3	2019	12	December	10	1	
1	2018-08-15	3646	14	157	2018	8	August	15	2	
2	2018-10-23	1859	25	1	2018	10	October	23	1	
3	2019-08-17	7292	25	1	2019	8	August	17	5	
4	2019-01-06	4344	25	3	2019	1	January	6	6	
5	2018-08-23	1811	25	4	2018	8	August	23	3	
6	2018-11-21	1282	26	1	2018	11	November	21	2	
7	2019-03-27	5022	2	41	2019	3	March	27	2	
8	2019-06-29	3699	3	15	2019	6	June	29	5	
9	2018-08-30	4373	11	3	2018	8	August	30	3	

In [28]:
```
# print the days name
data1['days_name']=data1['date'].dt.day_name()
data1.head(10)
```

Out[28]:

	date	product_id	city_id	orders	year	Months	Months_name	date_day	date_dow	date_we
0	2019-12-10	5628	25	3	2019	12	December	10	1	
1	2018-08-15	3646	14	157	2018	8	August	15	2	
2	2018-10-23	1859	25	1	2018	10	October	23	1	
3	2019-08-17	7292	25	1	2019	8	August	17	5	
4	2019-01-06	4344	25	3	2019	1	January	6	6	
5	2018-08-23	1811	25	4	2018	8	August	23	3	
6	2018-11-21	1282	26	1	2018	11	November	21	2	
7	2019-03-27	5022	2	41	2019	3	March	27	2	
8	2019-06-29	3699	3	15	2019	6	June	29	5	
9	2018-08-30	4373	11	3	2018	8	August	30	3	

In [29]:
```
data1['date_dow'].value_counts()
```

Out[29]:
```
4    177
5    156
2    155
0    133
6    128
3    127
1    124
Name: date_dow, dtype: int64
```

In [30]:
```
data1['days_name'].value_counts()
```

Out[30]:
```
Friday       177
Saturday     156
Wednesday    155
Monday       133
Sunday       128
Thursday     127
Tuesday      124
Name: days_name, dtype: int64
```

In [31]:
```
data1.columns
```

Out[31]:
```
Index(['date', 'product_id', 'city_id', 'orders', 'year', 'Months',
       'Months_name', 'date_day', 'date_dow', 'date_week', 'days_name'],
      dtype='object')
```

In [32]:
```
data1.drop(columns = ['product_id','city_id','orders'],inplace = True)
```

In [33]:
```
data1.head(10)
```

Out[33]:

	date	year	Months	Months_name	date_day	date_dow	date_week	days_name
0	2019-12-10	2019	12	December	10	1	50	Tuesday
1	2018-08-15	2018	8	August	15	2	33	Wednesday
2	2018-10-23	2018	10	October	23	1	43	Tuesday
3	2019-08-17	2019	8	August	17	5	33	Saturday
4	2019-01-06	2019	1	January	6	6	1	Sunday
5	2018-08-23	2018	8	August	23	3	34	Thursday
6	2018-11-21	2018	11	November	21	2	47	Wednesday
7	2019-03-27	2019	3	March	27	2	13	Wednesday
8	2019-06-29	2019	6	June	29	5	26	Saturday
9	2018-08-30	2018	8	August	30	3	35	Thursday

In [34]:
```
np.where
```

Out[34]:
```
<function numpy.where>
```

In [35]:
```
data1['date_is_weekend'] = np.where(data1['days_name'].isin(['Sunday', 'Saturda
```

In [36]:
```
data1.head(10)
```

Out[36]:

	date	year	Months	Months_name	date_day	date_dow	date_week	days_name	date_is_wee
0	2019-12-10	2019	12	December	10	1	50	Tuesday	
1	2018-08-15	2018	8	August	15	2	33	Wednesday	
2	2018-10-23	2018	10	October	23	1	43	Tuesday	
3	2019-08-17	2019	8	August	17	5	33	Saturday	
4	2019-01-06	2019	1	January	6	6	1	Sunday	
5	2018-08-23	2018	8	August	23	3	34	Thursday	
6	2018-11-21	2018	11	November	21	2	47	Wednesday	
7	2019-03-27	2019	3	March	27	2	13	Wednesday	
8	2019-06-29	2019	6	June	29	5	26	Saturday	
9	2018-08-30	2018	8	August	30	3	35	Thursday	

In [37]:
```
# extract quarter
data1['quarter'] = data1['date'].dt.quarter
```

In [38]:
```
data1.head()
```

Out[38]:

	date	year	Months	Months_name	date_day	date_dow	date_week	days_name	date_is_wee
0	2019-12-10	2019	12	December	10	1	50	Tuesday	
1	2018-08-15	2018	8	August	15	2	33	Wednesday	
2	2018-10-23	2018	10	October	23	1	43	Tuesday	
3	2019-08-17	2019	8	August	17	5	33	Saturday	
4	2019-01-06	2019	1	January	6	6	1	Sunday	

In [39]:
```
# Extract Semesterwise

data1['half_yearly'] = np.where(data1['quarter'].isin([1,2]),1,2)
data1.head()
```

Out[39]:

	date	year	Months	Months_name	date_day	date_dow	date_week	days_name	date_is_wee
0	2019-12-10	2019	12	December	10	1	50	Tuesday	
1	2018-08-15	2018	8	August	15	2	33	Wednesday	
2	2018-10-23	2018	10	October	23	1	43	Tuesday	
3	2019-08-17	2019	8	August	17	5	33	Saturday	
4	2019-01-06	2019	1	January	6	6	1	Sunday	

In [40]:
```
import datetime
today = datetime.datetime.today()
today
```

Out[40]:
```
datetime.datetime(2023, 6, 20, 12, 33, 49, 357884)
```

In [41]:
```
today - data1['date']
```

Out[41]:
```
0     1288 days 12:33:49.357884
1     1770 days 12:33:49.357884
2     1701 days 12:33:49.357884
3     1403 days 12:33:49.357884
4     1626 days 12:33:49.357884
                  ...
995   1716 days 12:33:49.357884
996   1657 days 12:33:49.357884
997   1505 days 12:33:49.357884
998   1570 days 12:33:49.357884
999   1344 days 12:33:49.357884
Name: date, Length: 1000, dtype: timedelta64[ns]
```

In [42]:
```
(today - data1['date']).dt.days
```

Out[42]:
```
0      1288
1      1770
2      1701
3      1403
4      1626
       ...
995    1716
996    1657
997    1505
998    1570
999    1344
Name: date, Length: 1000, dtype: int64
```

In [43]:
```
data.head()
```

Out[43]:

	date	msg
0	2013-12-15 00:50:00	ищу на сегодня мужика 37
1	2014-04-29 23:40:00	ПАРЕНЬ БИ ИЩЕТ ДРУГА СЕЙЧАС!! СМС ММС 0955532826
2	2012-12-30 00:21:00	Днепр.м 43 позн.с д/ж *.о 067.16.34.576
3	2014-11-28 00:31:00	КИЕВ ИЩУ Д/Ж ДО 45 МНЕ СЕЙЧАС СКУЧНО 093 629 9...
4	2013-10-26 23:11:00	Зая я тебя никогда не обижу люблю тебя!) Даше

In [44]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   date    1000 non-null   object
 1   msg     1000 non-null   object
dtypes: object(2)
memory usage: 15.8+ KB
```

In [45]:
```
data['date'] = pd.to_datetime(data['date'])
```

In [46]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype         
---  ------  --------------  -----         
 0   date    1000 non-null   datetime64[ns]
 1   msg     1000 non-null   object        
dtypes: datetime64[ns](1), object(1)
memory usage: 15.8+ KB
```

In [47]:
```
import datetime as dt
```

In [48]:
```
#extract the time  hour,min,sec
data['hour'] = data['date'].dt.hour
data['min'] = data['date'].dt.minute
data['sec'] = data['date'].dt.second
```

In [49]:
```
data.head()
```

Out[49]:

	date	msg	hour	min	sec
0	2013-12-15 00:50:00	ищу на сегодня мужика 37	0	50	0
1	2014-04-29 23:40:00	ПАРЕНЬ БИ ИЩЕТ ДРУГА СЕЙЧАС!! СМС ММС 0955532826	23	40	0
2	2012-12-30 00:21:00	Днепр.м 43 позн.с д/ж *.o 067.16.34.576	0	21	0
3	2014-11-28 00:31:00	КИЕВ ИЩУ Д/Ж ДО 45 МНЕ СЕЙЧАС СКУЧНО 093 629 9...	0	31	0
4	2013-10-26 23:11:00	Зая я тебя никогда не обижу люблю тебя!) Даше	23	11	0

In [50]:
```
data['time'] = data['date'].dt.time
```

In [51]:
```
data.head()
```

Out[51]:

	date	msg	hour	min	sec	time
0	2013-12-15 00:50:00	ищу на сегодня мужика 37	0	50	0	00:50:00
1	2014-04-29 23:40:00	ПАРЕНЬ БИ ИЩЕТ ДРУГА СЕЙЧАС!! СМС ММС 0955532826	23	40	0	23:40:00
2	2012-12-30 00:21:00	Днепр.м 43 позн.с д/ж *.o 067.16.34.576	0	21	0	00:21:00
3	2014-11-28 00:31:00	КИЕВ ИЩУ Д/Ж ДО 45 МНЕ СЕЙЧАС СКУЧНО 093 629 9...	0	31	0	00:31:00
4	2013-10-26 23:11:00	Зая я тебя никогда не обижу люблю тебя!) Даше	23	11	0	23:11:00

In [52]:
```
today
```

Out[52]:
```
datetime.datetime(2023, 6, 20, 12, 33, 49, 357884)
```

In [53]:
```
today - data['date']
```

Out[53]:
```
0     3474 days 11:43:49.357884
1     3338 days 12:53:49.357884
2     3824 days 12:12:49.357884
3     3126 days 12:02:49.357884
4     3523 days 13:22:49.357884
                  ...
995   4113 days 11:43:49.357884
996   3434 days 13:19:49.357884
997   3899 days 12:56:49.357884
998   4015 days 12:59:49.357884
999   3287 days 13:08:49.357884
Name: date, Length: 1000, dtype: timedelta64[ns]
```

In [54]:
```
# in second
(today - data['date'])/np.timedelta64(1,'s')
```

Out[54]:
```
0      3.001958e+08
1      2.884496e+08
2      3.304376e+08
3      2.701298e+08
4      3.044354e+08
           ...
995    3.554054e+08
996    2.967456e+08
997    3.369202e+08
998    3.469428e+08
999    2.840441e+08
Name: date, Length: 1000, dtype: float64
```

In [55]:
```
# in minute
(today - data['date'])/np.timedelta64(1,'m')
```

Out[55]:
```
0      5.003264e+06
1      4.807494e+06
2      5.507293e+06
3      4.502163e+06
4      5.073923e+06
           ...
995    5.923424e+06
996    4.945760e+06
997    5.615337e+06
998    5.782380e+06
999    4.734069e+06
Name: date, Length: 1000, dtype: float64
```

In [56]:
```
# in hours
(today - data['date'])/np.timedelta64(1,'h')
```

Out[56]:
```
0      83387.730377
1      80124.897044
2      91788.213711
3      75036.047044
4      84565.380377
           ...
995    98723.730377
996    82429.330377
997    93588.947044
998    96372.997044
999    78901.147044
Name: date, Length: 1000, dtype: float64
```

this is all about to handle date and time format