

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

```
In [24]: df=pd.read_csv('titanic.csv',usecols=['Cabin','Ticket','Survived'],r  
df.head()
```

Out[24]:

	Survived	Ticket	Cabin
0	0	A/5 21171	NaN
1	1	PC 17599	C85
2	1	STON/O2. 3101282	NaN
3	1	113803	C123
4	0	373450	NaN

```
In [25]: df['Number']=[5,3,6,3,'A']  
df.head()
```

Out[25]:

	Survived	Ticket	Cabin	Number
0	0	A/5 21171	NaN	5
1	1	PC 17599	C85	3
2	1	STON/O2. 3101282	NaN	6
3	1	113803	C123	3
4	0	373450	NaN	A

```
In [28]: ## detaching categorical values from Number column  
df['Num_col']=pd.to_numeric(df['Number'],errors='coerce',downcast='float')  
df['Cat_col']=np.where(df['Num_col'].isnull(),df['Number'],np.nan) #VAL  
df
```

Out[28]:

	Survived	Ticket	Cabin	Number	Num_col	Cat_col
0	0	A/5 21171	NaN	5	5.0	NaN
1	1	PC 17599	C85	3	3.0	NaN
2	1	STON/O2. 3101282	NaN	6	6.0	NaN
3	1	113803	C123	3	3.0	NaN
4	0	373450	NaN	A	NaN	A

Date time handing

```
In [29]: date=pd.read_csv('orders.csv')  
time=pd.read_csv('messages.csv')
```

```
In [30]: time
```

	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!) Даще
...
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

1000 rows × 2 columns

```
In [32]: date.head()
```

Out[32]:

	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 [33]:

```
date.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 [34]:

```
time.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 [35]:

```
## converting datetime object into datetime object
```

In [36]:

```
date['date']=pd.to_datetime(date['date'])
time['date']=pd.to_datetime(time['date'])
```

In [37]:

```
date.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 [38]: `time.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
```

Extracting year from data

In [39]: `date['Year']=date['date'].dt.year
date.head()`

Out[39]:

	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

Extracting month

In [40]: `date['Month']=date['date'].dt.month
date.head()`

Out[40]:

	date	product_id	city_id	orders	Year	Month
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 [41]: `date['Year'].nunique()`

Out[41]: 2

Extracting month name

In [42]: `date['Month Name']=date['date'].dt.month_name()
date.head()`

	date	product_id	city_id	orders	Year	Month	Month 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

Extracting days of the month

In [47]: `date['Tarik of Month']=date['date'].dt.day
date.head()`

Out[47]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no	v
0	2019-12-10	5628	25	3	2019	12	December	10	50	
1	2018-08-15	3646	14	157	2018	8	August	15	33	
2	2018-10-23	1859	25	1	2018	10	October	23	43	
3	2019-08-17	7292	25	1	2019	8	August	17	33	
4	2019-01-06	4344	25	3	2019	1	January	6	1	

◀ ▶

Extracting week of the month

In [44]:

```
date['Week no']=date['date'].dt.week
date.head()
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_3252\618125611.py:1: FutureWarning: Series.dt.weekofyear and Series.dt.week have been deprecated. Please use Series.dt.isocalendar().week instead.

```
date['Week no']=date['date'].dt.week
```

Out[44]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no
0	2019-12-10	5628	25	3	2019	12	December	10	50
1	2018-08-15	3646	14	157	2018	8	August	15	33
2	2018-10-23	1859	25	1	2018	10	October	23	43
3	2019-08-17	7292	25	1	2019	8	August	17	33
4	2019-01-06	4344	25	3	2019	1	January	6	1

◀ ▶

Extracting day of the week

```
In [45]: date['day of week']=date['date'].dt.dayofweek  
date.head()
```

Out[45]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no	v
0	2019-12-10	5628	25	3	2019	12	December	10	50	
1	2018-08-15	3646	14	157	2018	8	August	15	33	
2	2018-10-23	1859	25	1	2018	10	October	23	43	
3	2019-08-17	7292	25	1	2019	8	August	17	33	
4	2019-01-06	4344	25	3	2019	1	January	6	1	

Extracting day name

```
In [48]: date['day name']=date['date'].dt.day_name()  
date.head()
```

Out[48]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no	v
0	2019-12-10	5628	25	3	2019	12	December	10	50	
1	2018-08-15	3646	14	157	2018	8	August	15	33	
2	2018-10-23	1859	25	1	2018	10	October	23	43	
3	2019-08-17	7292	25	1	2019	8	August	17	33	
4	2019-01-06	4344	25	3	2019	1	January	6	1	

```
In [50]: date.drop('Tarik of Month',inplace=True,axis=1)
```

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

Out[51]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no	v
0	2019-12-10	5628	25	3	2019	12	December	10	50	
1	2018-08-15	3646	14	157	2018	8	August	15	33	
2	2018-10-23	1859	25	1	2018	10	October	23	43	
3	2019-08-17	7292	25	1	2019	8	August	17	33	
4	2019-01-06	4344	25	3	2019	1	January	6	1	



Binarizing dayname

```
In [52]: date['Weekend_0/1']=np.where(date['day name'].isin(['Saturday','Sunday']),1,0)
date.head()
```

Out[52]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no	v
0	2019-12-10	5628	25	3	2019	12	December	10	50	
1	2018-08-15	3646	14	157	2018	8	August	15	33	
2	2018-10-23	1859	25	1	2018	10	October	23	43	
3	2019-08-17	7292	25	1	2019	8	August	17	33	
4	2019-01-06	4344	25	3	2019	1	January	6	1	



Extracting Quarter of the year

```
In [54]: date['Quarter']=date['date'].dt.quarter  
date.head()
```

Out[54]:

	date	product_id	city_id	orders	Year	Month	Month Name	Days of Month	Week no	v
0	2019-12-10	5628	25	3	2019	12	December	10	50	
1	2018-08-15	3646	14	157	2018	8	August	15	33	
2	2018-10-23	1859	25	1	2018	10	October	23	43	
3	2019-08-17	7292	25	1	2019	8	August	17	33	
4	2019-01-06	4344	25	3	2019	1	January	6	1	

◀ ▶

Extracting Hour min and sec from data

```
In [55]: time.hear()
```

Out[55]:

	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 ИШУ 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!) Даще

```
In [56]: time.drop('msg',axis=1,inplace=True)
```

```
In [57]: time['Year']=time['date'].dt.year  
time.head()
```

Out[57]:

	date	Year
0	2013-12-15 00:50:00	2013
1	2014-04-29 23:40:00	2014
2	2012-12-30 00:21:00	2012
3	2014-11-28 00:31:00	2014
4	2013-10-26 23:11:00	2013

In [58]:

```
time['Hour']=time['date'].dt.hour  
time.head()
```

Out[58]:

	date	Year	Hour
0	2013-12-15 00:50:00	2013	0
1	2014-04-29 23:40:00	2014	23
2	2012-12-30 00:21:00	2012	0
3	2014-11-28 00:31:00	2014	0
4	2013-10-26 23:11:00	2013	23

In [60]:

```
time['Min']=time['date'].dt.minute  
time.head(2)
```

Out[60]:

	date	Year	Hour	Min
0	2013-12-15 00:50:00	2013	0	50
1	2014-04-29 23:40:00	2014	23	40

In [61]:

```
time['sec']=time['date'].dt.second  
time.head(2)
```

Out[61]:

	date	Year	Hour	Min	sec
0	2013-12-15 00:50:00	2013	0	50	0
1	2014-04-29 23:40:00	2014	23	40	0

In []: