20. Pandas – DataFrame - Date and time

Contents

20. Pandas – DataFrame - Date and time

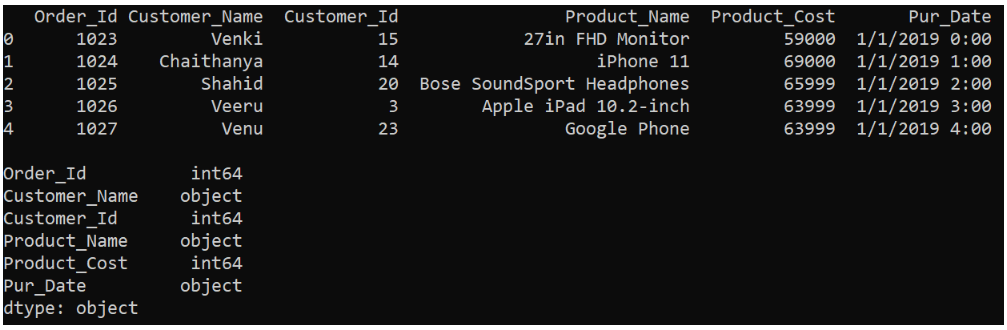
1. Date data type

- ✓ Whenever we load csv file, if that file contains any column having date values then by default pandas will consider that column as object
- ✓ We can provide parse_dates = ['name of the column1', 'name of the column2'] then pandas considered those columns as datetime data type

Program Name Creating DataFrames
demo1.py
File name sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv')

print(df.head())
print()
print(df.dtypes)
```

Output

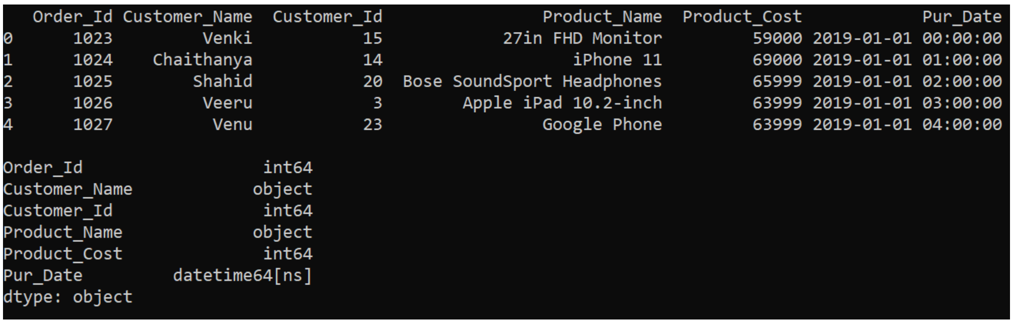


```
   Order_Id Customer_Name  Customer_Id                Product_Name  Product_Cost       Pur_Date
0      1023         Venki           15            27in FHD Monitor         59000  1/1/2019 0:00
1      1024    Chaithanya           14                   iPhone 11         69000  1/1/2019 1:00
2      1025        Shahid           20  Bose SoundSport Headphones         65999  1/1/2019 2:00
3      1026         Veeru            3        Apple iPad 10.2-inch         63999  1/1/2019 3:00
4      1027          Venu           23                Google Phone         63999  1/1/2019 4:00

Order_Id          int64
Customer_Name    object
Customer_Id       int64
Product_Name     object
Product_Cost      int64
Pur_Date         object
dtype: object
```

Program Loading csv file by using parse_dates paramter
Name demo2.py
File name sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

print(df.head())
print()
print(df.dtypes)
```

Output



```
   Order_Id Customer_Name  Customer_Id                Product_Name  Product_Cost            Pur_Date
0      1023         Venki           15            27in FHD Monitor         59000 2019-01-01 00:00:00
1      1024    Chaithanya           14                   iPhone 11         69000 2019-01-01 01:00:00
2      1025        Shahid           20  Bose SoundSport Headphones         65999 2019-01-01 02:00:00
3      1026         Veeru            3        Apple iPad 10.2-inch         63999 2019-01-01 03:00:00
4      1027          Venu           23                Google Phone         63999 2019-01-01 04:00:00

Order_Id                  int64
Customer_Name            object
Customer_Id               int64
Product_Name             object
Product_Cost              int64
Pur_Date         datetime64[ns]
dtype: object
```

2. Converting object data type into date data type

- ✓ We can convert from object data type into datetime data type explicitly.
- ✓ By using,
    - ○ to_datetime(p) function
    - ○ astype(p) method in Series

2.1. to_datetime(p)

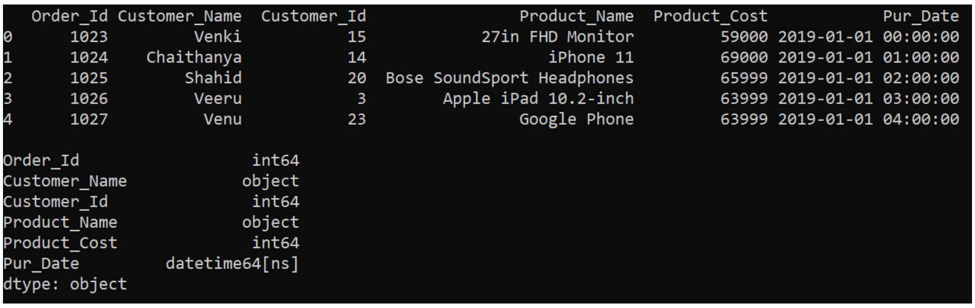
- ✓ to_datetime(p) is predefined function in pandas
- ✓ This function we should access by using pandas library.
- ✓ This function convert from object data type into date data type.

Program Name: Loading csv file and converting Pur_Date column into Date format demo3.py

File name: sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv')

df['Pur_Date'] = pd.to_datetime(df['Pur_Date'])

print(df.head())
print()
print(df.dtypes)
```

Output



```
   Order_Id Customer_Name  Customer_Id                Product_Name  Product_Cost            Pur_Date
0      1023         Venki           15            27in FHD Monitor         59000 2019-01-01 00:00:00
1      1024    Chaithanya           14                   iPhone 11         69000 2019-01-01 01:00:00
2      1025        Shahid           20  Bose SoundSport Headphones         65999 2019-01-01 02:00:00
3      1026         Veeru            3        Apple iPad 10.2-inch         63999 2019-01-01 03:00:00
4      1027          Venu           23                Google Phone         63999 2019-01-01 04:00:00

Order_Id                  int64
Customer_Name            object
Customer_Id               int64
Product_Name             object
Product_Cost              int64
Pur_Date         datetime64[ns]
dtype: object
```

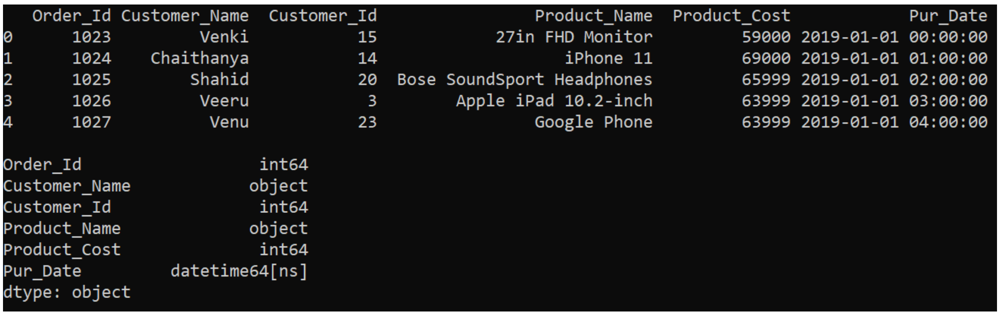
2.2. astype (p)

- ✓ astype(p) is predefined method Series class
- ✓ This method we should access by using Series object only
- ✓ This method convert from object data type into datetime data type.

Program Loading csv file and converting Pur_Date column into Date format
Name demo4.py
File name sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv')

df['Pur_Date'] = df['Pur_Date'].astype('datetime64[ns]')

print(df.head())
print()
print(df.dtypes)
```

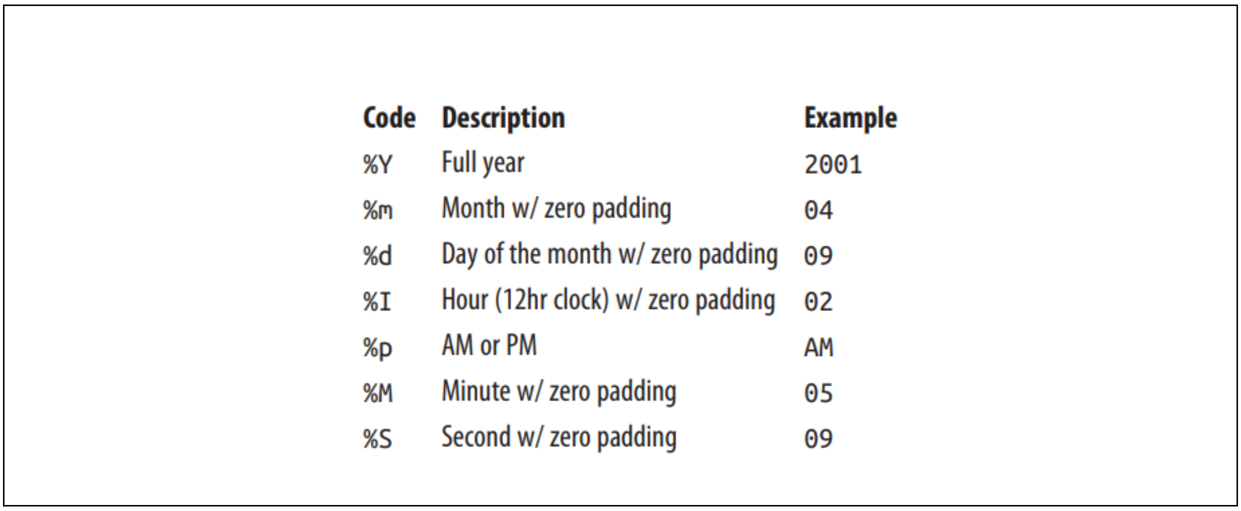
Output



```
   Order_Id Customer_Name  Customer_Id                Product_Name  Product_Cost            Pur_Date
0      1023         Venki           15             27in FHD Monitor         59000 2019-01-01 00:00:00
1      1024    Chaithanya           14                   iPhone 11         69000 2019-01-01 01:00:00
2      1025        Shahid           20  Bose SoundSport Headphones         65999 2019-01-01 02:00:00
3      1026         Veeru            3        Apple iPad 10.2-inch         63999 2019-01-01 03:00:00
4      1027          Venu           23                Google Phone         63999 2019-01-01 04:00:00

Order_Id                  int64
Customer_Name            object
Customer_Id               int64
Product_Name             object
Product_Cost              int64
Pur_Date         datetime64[ns]
dtype: object
```

3. Format parameter

- ✓ We can represent the date formats in different ways,
    - ○ March 23rd, 2015 as "03-23-15" or "3|23|2015" and etc
- ✓ So, we can use the format parameter to specify the exact format of the string.



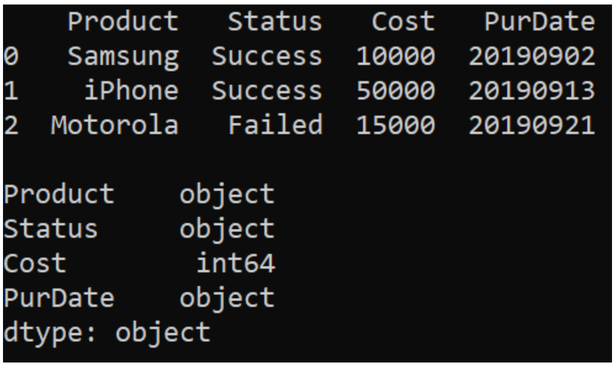
Code	Description	Example
%Y	Full year	2001
%m	Month w/ zero padding	04
%d	Day of the month w/ zero padding	09
%I	Hour (12hr clock) w/ zero padding	02
%p	AM or PM	AM
%M	Minute w/ zero padding	05
%S	Second w/ zero padding	09

Program Name

Creating a DataFrame
demo5.py

```
import pandas as pd

data = {
        'Product': ['Samsung', 'iPhone', 'Motorola'],
        'Status': ['Success', 'Success', 'Failed'],
        'Cost': [10000, 50000, 15000],
        'PurDate': ['20190902','20190913','20190921'],
}

df = pd.DataFrame(data)

print(df.head())
print()
print(df.dtypes)
```

Output



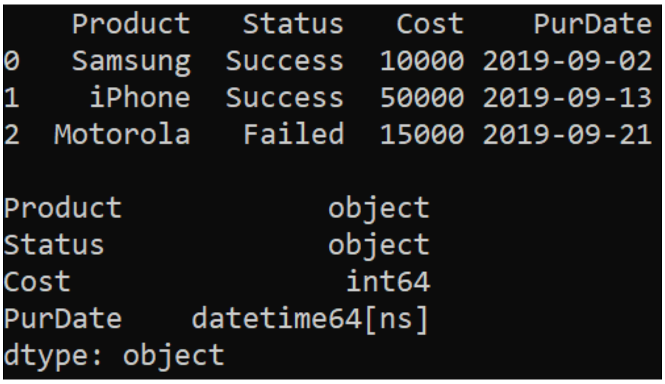
```
    Product   Status   Cost   PurDate
0   Samsung  Success  10000  20190902
1    iPhone  Success  50000  20190913
2  Motorola   Failed  15000  20190921

Product     object
Status      object
Cost         int64
PurDate     object
dtype: object
```

Program Name

Converting Date with specific format
demo6.py

```
import pandas as pd

data = {
        'Product': ['Samsung', 'iPhone', 'Motorola'],
        'Status': ['Success', 'Success', 'Failed'],
        'Cost': [10000, 50000, 15000],
        'PurDate': ['20190902','20190913','20190921'],
}

df = pd.DataFrame(data)
df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df.dtypes)
```

Output



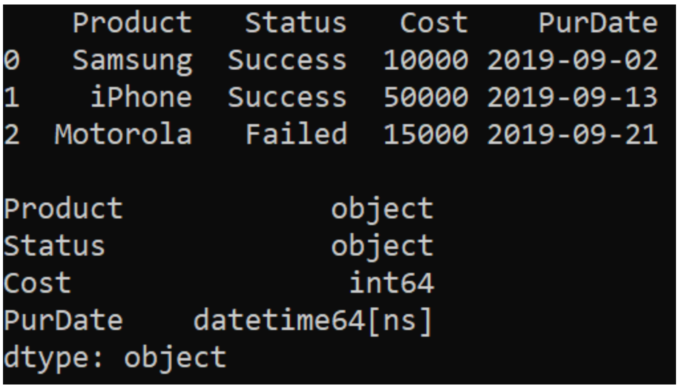
```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2019-09-02
1    iPhone  Success  50000 2019-09-13
2  Motorola   Failed  15000 2019-09-21

Product              object
Status               object
Cost                  int64
PurDate      datetime64[ns]
dtype: object
```

Program Name

Converting Date with specific format
demo7.py

```
import pandas as pd


data = {
	"Product": ["Samsung", "iPhone", "Motorola"],
	"Status": ["Success", "Success", "Failed"],
	"Cost": [10000, 50000, 15000],
	"PurDate": ['02092019','13092019','21092019'],
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'], format = '%d%m%Y')

print(df.head())
print()
print(df.dtypes)
```

Output



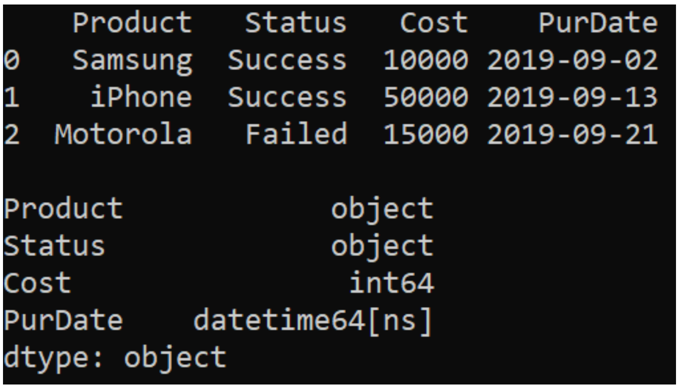
```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2019-09-02
1    iPhone  Success  50000 2019-09-13
2  Motorola   Failed  15000 2019-09-21

Product            object
Status             object
Cost                int64
PurDate    datetime64[ns]
dtype: object
```

Program Name

Converting Date with specific format
demo8.py

```
import pandas as pd

data = {
	"Product": ["Samsung", "iPhone", "Motorola"],
	"Status": ["Success", "Success", "Failed"],
	"Cost": [10000, 50000, 15000],
	"PurDate": ['02Sep2019','13Sep2019','21Sep2019'],
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df.dtypes)
```

Output



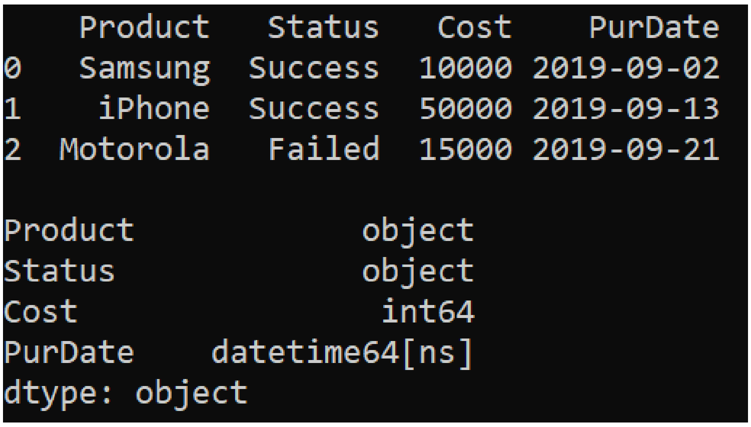
```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2019-09-02
1    iPhone  Success  50000 2019-09-13
2  Motorola   Failed  15000 2019-09-21

Product              object
Status               object
Cost                  int64
PurDate      datetime64[ns]
dtype: object
```

Program Name

Converting Date with specific format
demo9.py

```
import pandas as pd

data = {
	"Product": ["Samsung", "iPhone", "Motorola"],
	"Status": ["Success", "Success", "Failed"],
	"Cost": [10000, 50000, 15000],
	"PurDate": ['02Sep2019','13Sep2019','21Sep2019'],
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'], format = '%d%b%Y')

print(df.head())
print()
print(df.dtypes)
```

Output



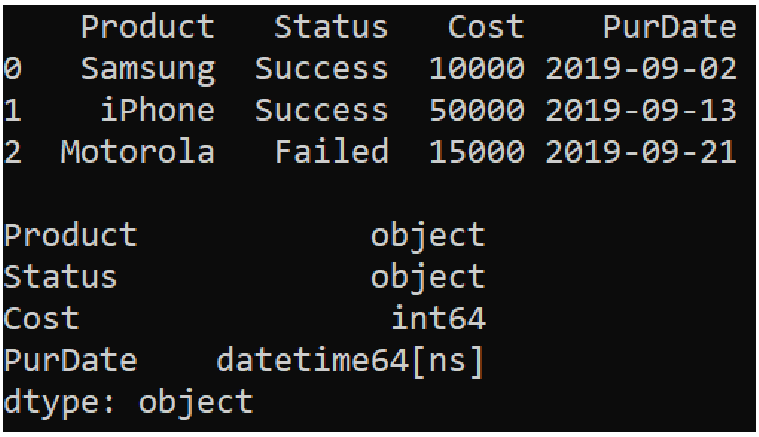
```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2019-09-02
1    iPhone  Success  50000 2019-09-13
2  Motorola   Failed  15000 2019-09-21

Product              object
Status               object
Cost                  int64
PurDate      datetime64[ns]
dtype: object
```

Program Name

Converting Date with specific format
demo10.py

```
import pandas as pd

data = {
        "Product": ["Samsung", "iPhone", "Motorola"],
        "Status": ["Success", "Success", "Failed"],
        "Cost": [10000, 50000, 15000],
        "PurDate": ['02-Sep-2019','13-Sep-2019','21-Sep-2019'],
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df.dtypes)
```

Output



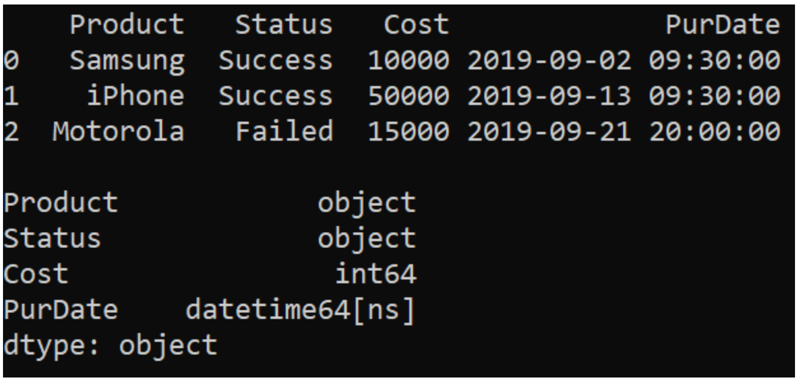
```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2019-09-02
1    iPhone  Success  50000 2019-09-13
2  Motorola   Failed  15000 2019-09-21

Product                  object
Status                   object
Cost                      int64
PurDate          datetime64[ns]
dtype: object
```

Program Name

Converting Date with specific format
demo11.py

```
import pandas as pd

data = {
        "Product": ["Samsung", "iPhone", "Motorola"],
        "Status": ["Success", "Success", "Failed"],
        "Cost": [10000, 50000, 15000],
        "PurDate":
['20190902093000','20190913093000','20190921200000'],
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df.dtypes)
```

Output



```
    Product   Status   Cost             PurDate
0   Samsung  Success  10000 2019-09-02 09:30:00
1    iPhone  Success  50000 2019-09-13 09:30:00
2  Motorola   Failed  15000 2019-09-21 20:00:00

Product            object
Status             object
Cost                int64
PurDate    datetime64[ns]
dtype: object
```

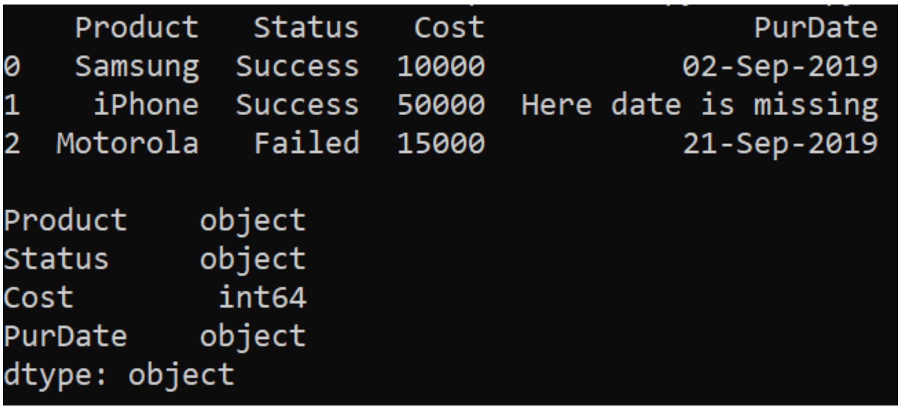
4. NaT values

- ✓ Date column may contains missing values in pandas DataFrame.
- ✓ If Date column contains missing values then while converting into Date data type then we will get an error.
- ✓ By using errors = "coerce" keyword argument we can solve this problem.
- ✓ This argument converts Date column missing values into NaT (Not a Time) values.
  - ○ Coerce errors i.e. convert un parse able date into **NaT** (Not a Time)

Program Name: Creating DataFrame
demo12.py

```
import pandas as pd

data = {
	"Product": ["Samsung", "iPhone", "Motorola"],
	"Status": ["Success", "Success", "Failed"],
	"Cost": [10000, 50000, 15000],
	"PurDate": ['02-Sep-2019','Here date is missing','21-Sep-2019']
}

df = pd.DataFrame(data)

print(df.head())
print()
print(df.dtypes)
```

Output



```
    Product   Status   Cost              PurDate
0   Samsung  Success  10000          02-Sep-2019
1    iPhone  Success  50000  Here date is missing
2  Motorola   Failed  15000          21-Sep-2019

Product     object
Status      object
Cost         int64
PurDate     object
dtype: object
```

Program Name

Converting Date with specific format: Error

demo13.py

```
import pandas as pd

data = {
	"Product": ["Samsung", "iPhone", "Motorola"],
	"Status": ["Success", "Success", "Failed"],
	"Cost": [10000, 50000, 15000],
	"PurDate": ['02-Sep-2019','Here date is missing','21-Sep-2019']
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df.dtypes)
```

Output

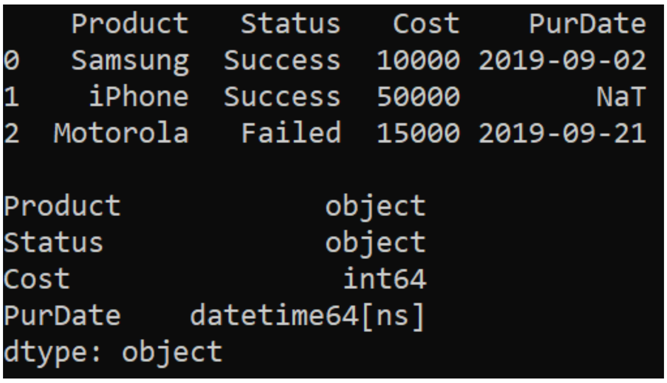
```
TypeError: invalid string coercion to datetime for "Here date is missing" at position 1
```

Program Name

to_datetime(p) function
demo14.py

```
import pandas as pd

data = {
	"Product": ["Samsung", "iPhone", "Motorola"],
	"Status": ["Success", "Success", "Failed"],
	"Cost": [10000, 50000, 15000],
	"PurDate": ['02-Sep-2019','Here date is missing','21-Sep-2019']
}

df = pd.DataFrame(data)

df['PurDate'] = pd.to_datetime(df['PurDate'], errors="coerce")

print(df.head())
print()
print(df.dtypes)
```

Output



```
    Product   Status   Cost   PurDate
0   Samsung  Success  10000 2019-09-02
1    iPhone  Success  50000        NaT
2  Motorola   Failed  15000 2019-09-21

Product            object
Status             object
Cost                int64
PurDate    datetime64[ns]
dtype: object
```

5. Selecting from start to end date values

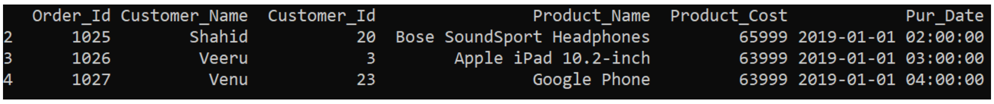
- ✓ Based on requirement we can select specific dates, like
    - ○ Start date to end date

Program Name: Selecting Dataframe in between the dates
demo15.py

File Name: sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates=['Pur_Date'])

start = df['Pur_Date'] > '2019-1-1 01:00:00'
end = df['Pur_Date'] < '2019-1-1 05:00:00'

result = df[start & end]

print(result)
```

Output



```
   Order_Id Customer_Name  Customer_Id                Product_Name  Product_Cost            Pur_Date
2      1025        Shahid           20  Bose SoundSport Headphones         65999 2019-01-01 02:00:00
3      1026         Veeru            3        Apple iPad 10.2-inch         63999 2019-01-01 03:00:00
4      1027          Venu           23                Google Phone         63999 2019-01-01 04:00:00
```

6. Access specific dates like last 20 days or 2 months or 2 years records

- ✓ Based on requirement we can get last 10 days, 20 days, 40 days, 1 month, 2 months, 3 months, 1 year and 2 years date data as well.
- ✓ We can also sort the dataframe by using sort_values()

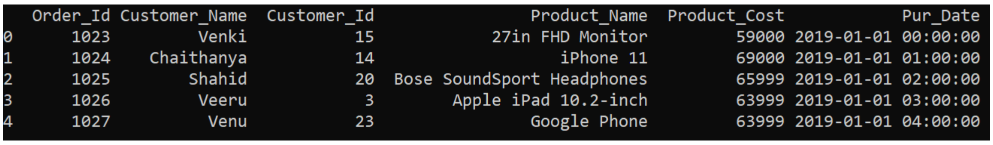
Program Creating DataFrame

Name demo16.py

File name sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

print(df.head())
```

Output



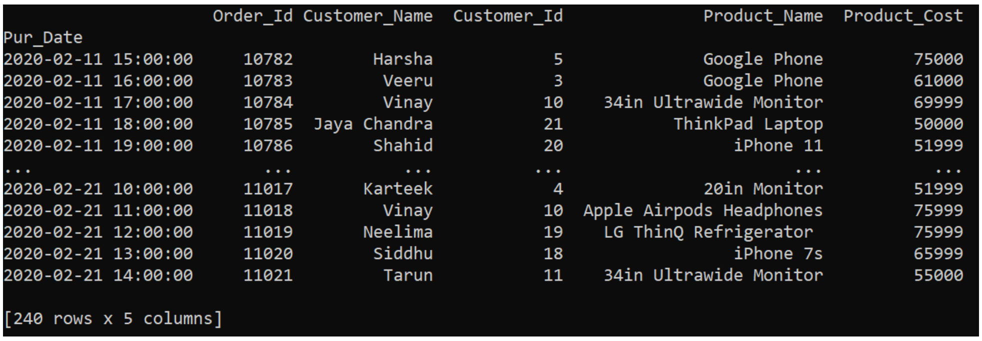
```
   Order_Id Customer_Name  Customer_Id                Product_Name  Product_Cost            Pur_Date
0      1023         Venki           15            27in FHD Monitor         59000 2019-01-01 00:00:00
1      1024    Chaithanya           14                   iPhone 11         69000 2019-01-01 01:00:00
2      1025        Shahid           20  Bose SoundSport Headphones         65999 2019-01-01 02:00:00
3      1026         Veeru            3        Apple iPad 10.2-inch         63999 2019-01-01 03:00:00
4      1027          Venu           23                Google Phone         63999 2019-01-01 04:00:00
```

Program Name   Accessing last 10 days records
demo17.py

File name   sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

new_df = df.set_index("Pur_Date")
days_10 = new_df.last("10D")

print(days_10)
```

Output



```
                     Order_Id Customer_Name  Customer_Id              Product_Name  Product_Cost
Pur_Date
2020-02-11 15:00:00     10782        Harsha            5              Google Phone         75000
2020-02-11 16:00:00     10783         Veeru            3              Google Phone         61000
2020-02-11 17:00:00     10784         Vinay           10    34in Ultrawide Monitor         69999
2020-02-11 18:00:00     10785  Jaya Chandra           21           ThinkPad Laptop         50000
2020-02-11 19:00:00     10786        Shahid           20                 iPhone 11         51999
...                       ...           ...          ...                       ...           ...
2020-02-21 10:00:00     11017       Karteek            4              20in Monitor         51999
2020-02-21 11:00:00     11018         Vinay           10  Apple Airpods Headphones         75999
2020-02-21 12:00:00     11019       Neelima           19    LG ThinQ Refrigerator          75999
2020-02-21 13:00:00     11020       Siddhu            18                 iPhone 7s         65999
2020-02-21 14:00:00     11021         Tarun           11    34in Ultrawide Monitor         55000

[240 rows x 5 columns]
```

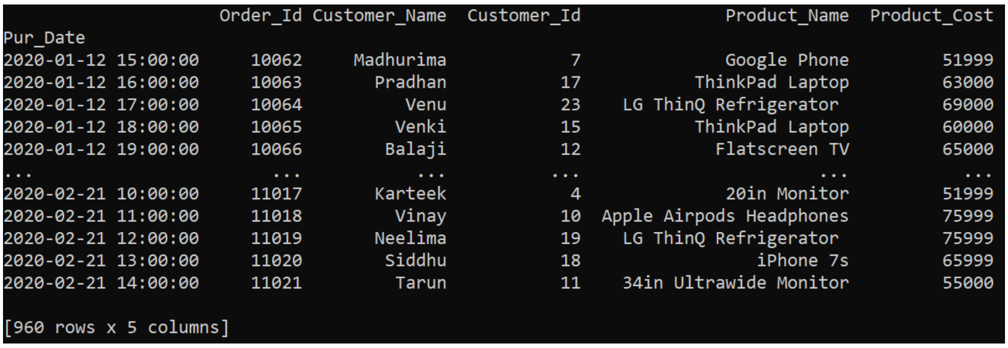
Program Name: Accessing last 40 days records

demo18.py

File name: sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

new_df = df.set_index("Pur_Date")
days_40 = new_df.last("40D")

print(days_40)
```

Output



```
                     Order_Id Customer_Name  Customer_Id             Product_Name  Product_Cost
Pur_Date
2020-01-12 15:00:00     10062     Madhurima            7             Google Phone         51999
2020-01-12 16:00:00     10063       Pradhan           17          ThinkPad Laptop         63000
2020-01-12 17:00:00     10064          Venu           23     LG ThinQ Refrigerator        69000
2020-01-12 18:00:00     10065         Venki           15          ThinkPad Laptop         60000
2020-01-12 19:00:00     10066        Balaji           12            Flatscreen TV         65000
...                       ...           ...          ...                      ...           ...
2020-02-21 10:00:00     11017       Karteek            4             20in Monitor         51999
2020-02-21 11:00:00     11018         Vinay           10   Apple Airpods Headphones       75999
2020-02-21 12:00:00     11019       Neelima           19     LG ThinQ Refrigerator        75999
2020-02-21 13:00:00     11020        Siddhu           18                iPhone 7s         65999
2020-02-21 14:00:00     11021         Tarun           11     34in Ultrawide Monitor       55000

[960 rows x 5 columns]
```

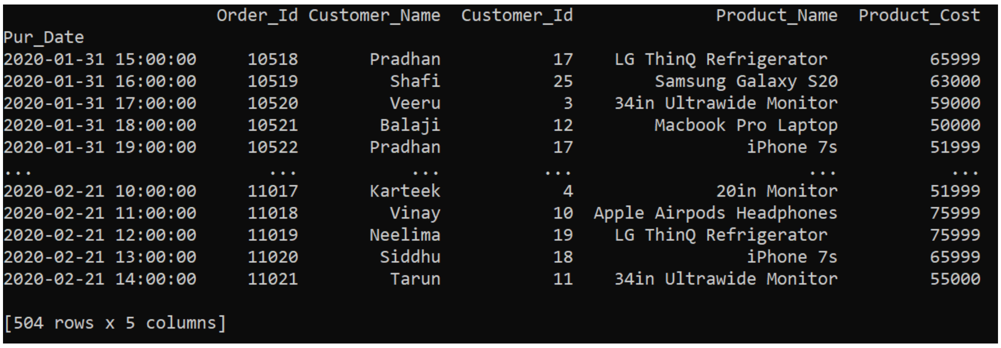
Program Name Accessing last 1 month records

File name demo19.py

sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

new_df = df.set_index("Pur_Date")
month_1 = new_df.last("1M")

print(month_1)
```

Output



```
                     Order_Id Customer_Name  Customer_Id              Product_Name  Product_Cost
Pur_Date
2020-01-31 15:00:00     10518       Pradhan           17    LG ThinQ Refrigerator          65999
2020-01-31 16:00:00     10519         Shafi           25        Samsung Galaxy S20         63000
2020-01-31 17:00:00     10520         Veeru            3    34in Ultrawide Monitor         59000
2020-01-31 18:00:00     10521        Balaji           12        Macbook Pro Laptop         50000
2020-01-31 19:00:00     10522       Pradhan           17                 iPhone 7s         51999
...                       ...           ...          ...                       ...           ...
2020-02-21 10:00:00     11017       Karteek            4              20in Monitor         51999
2020-02-21 11:00:00     11018         Vinay           10  Apple Airpods Headphones         75999
2020-02-21 12:00:00     11019       Neelima           19    LG ThinQ Refrigerator          75999
2020-02-21 13:00:00     11020        Siddhu           18                 iPhone 7s         65999
2020-02-21 14:00:00     11021         Tarun           11    34in Ultrawide Monitor         55000

[504 rows x 5 columns]
```

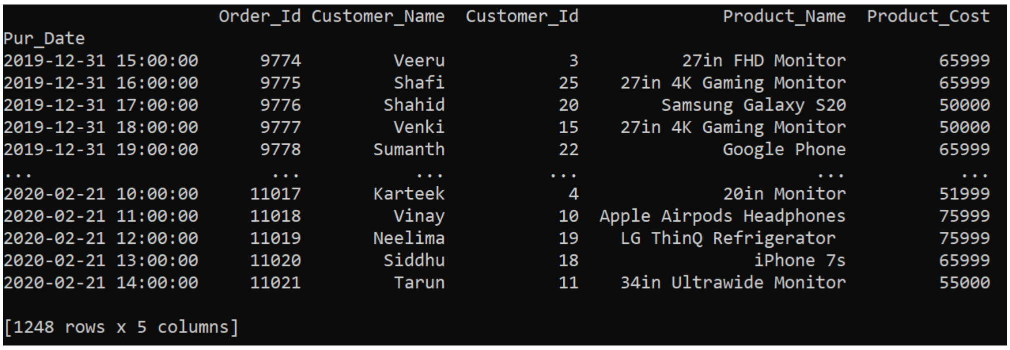
Program Name Accessing last 1 year records

File name demo20.py

sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

new_df = df.set_index("Pur_Date")
year_1 = new_df.last("1Y")

print(year_1)
```

Output



```
                     Order_Id Customer_Name  Customer_Id            Product_Name  Product_Cost
Pur_Date
2019-12-31 15:00:00      9774         Veeru            3         27in FHD Monitor         65999
2019-12-31 16:00:00      9775         Shafi           25   27in 4K Gaming Monitor         65999
2019-12-31 17:00:00      9776        Shahid           20       Samsung Galaxy S20         50000
2019-12-31 18:00:00      9777         Venki           15   27in 4K Gaming Monitor         50000
2019-12-31 19:00:00      9778       Sumanth           22             Google Phone         65999
...                       ...           ...          ...                      ...           ...
2020-02-21 10:00:00     11017       Karteek            4             20in Monitor         51999
2020-02-21 11:00:00     11018         Vinay           10  Apple Airpods Headphones         75999
2020-02-21 12:00:00     11019       Neelima           19    LG ThinQ Refrigerator          75999
2020-02-21 13:00:00     11020       Siddhu            18                iPhone 7s         65999
2020-02-21 14:00:00     11021         Tarun           11   34in Ultrawide Monitor         55000

[1248 rows x 5 columns]
```

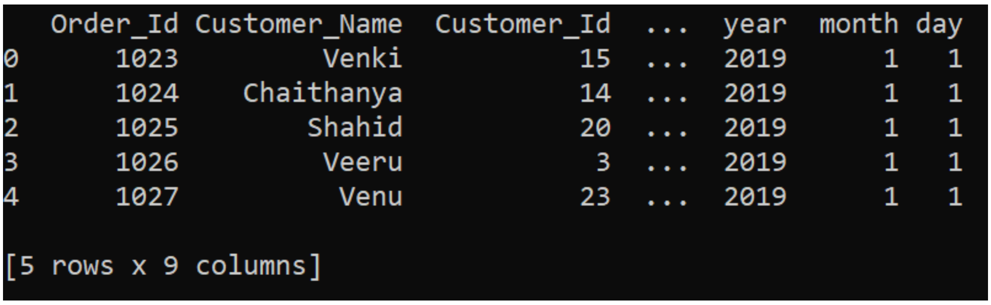
7. Extract year, month, day from Date column

- ✓ Based on requirement we can get year, month, day, hour, minute from Date column.
- ✓ Sometimes it can be useful to break up a column of dates into components.

Program Breaking up the Date column value into multiple features
Name demo21.py
File name sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

df['year'] = df['Pur_Date'].dt.year
df['month'] = df['Pur_Date'].dt.month
df['day'] = df['Pur_Date'].dt.day

print(df.head())
```

Output

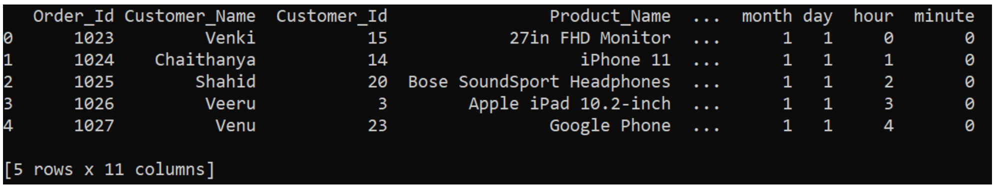


```
   Order_Id Customer_Name  Customer_Id  ...  year  month  day
0      1023         Venki           15  ...  2019      1    1
1      1024    Chaithanya           14  ...  2019      1    1
2      1025        Shahid           20  ...  2019      1    1
3      1026         Veeru            3  ...  2019      1    1
4      1027          Venu           23  ...  2019      1    1

[5 rows x 9 columns]
```

Program Name Breaking up the Date column value into multiple features
demo22.py
File name sales7_dates.csv

```
import pandas as pd

df = pd.read_csv('sales7_dates.csv', parse_dates = ['Pur_Date'])

df['year'] = df['Pur_Date'].dt.year
df['month'] = df['Pur_Date'].dt.month
df['day'] = df['Pur_Date'].dt.day
df['hour'] = df['Pur_Date'].dt.hour
df['minute'] = df['Pur_Date'].dt.minute

print(df.head())
```

Output



```
   Order_Id Customer_Name  Customer_Id                Product_Name  ...  month day  hour  minute
0      1023         Venki           15            27in FHD Monitor  ...      1   1     0       0
1      1024    Chaithanya           14                   iPhone 11  ...      1   1     1       0
2      1025        Shahid           20  Bose SoundSport Headphones  ...      1   1     2       0
3      1026         Veeru            3        Apple iPad 10.2-inch  ...      1   1     3       0
4      1027          Venu           23                Google Phone  ...      1   1     4       0

[5 rows x 11 columns]
```

8. Encoding Days of the Week

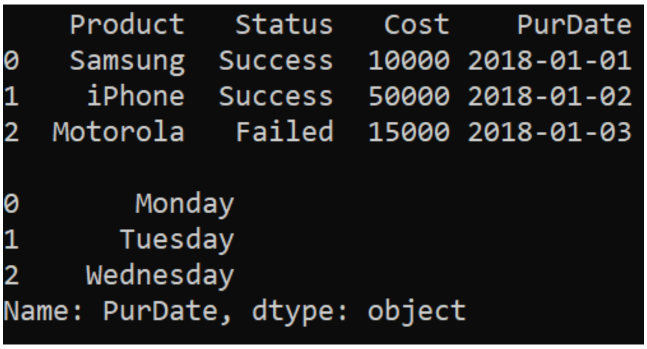
- ✓ We can get the day of the week for each date by using pandas
  - ○ Knowing the days names will helpful to understand the business flow, like we can compare total sales on specific day for the past three years.

Program Name Encoding Days of the Week

demo23.py

```
import pandas as pd

data = {
        'Product': ['Samsung', 'iPhone', 'Motorola'],
        'Status': ['Success', 'Success', 'Failed'],
        'Cost': [10000, 50000, 15000],
        'PurDate': ['2018-01-01','2018-01-02','2018-01-03'],
}

df = pd.DataFrame(data)
df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df["PurDate"].dt.day_name())
```

Output



```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2018-01-01
1    iPhone  Success  50000 2018-01-02
2  Motorola   Failed  15000 2018-01-03

0       Monday
1      Tuesday
2    Wednesday
Name: PurDate, dtype: object
```

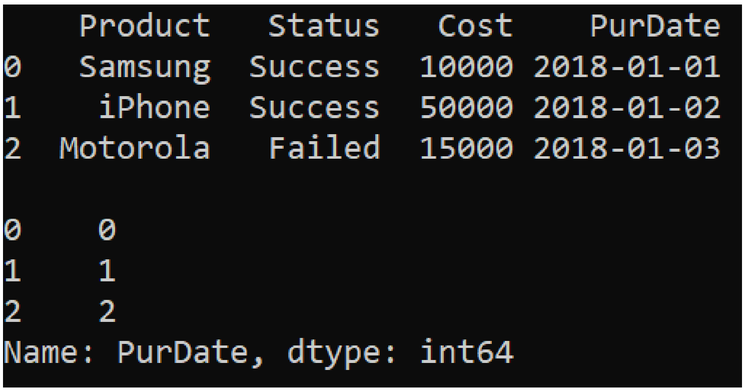
Note

- ✓ The day of the week with Monday = 0, …. Sunday = 6

Program Encoding Days of the Week
Name demo24.py

```
import pandas as pd

data = {
        'Product': ['Samsung', 'iPhone', 'Motorola'],
        'Status': ['Success', 'Success', 'Failed'],
        'Cost': [10000, 50000, 15000],
        'PurDate': ['2018-01-01','2018-01-02','2018-01-03'],
}

df = pd.DataFrame(data)
df['PurDate'] = pd.to_datetime(df['PurDate'])

print(df.head())
print()
print(df['PurDate'].dt.weekday)
```

Output



```
    Product   Status   Cost    PurDate
0   Samsung  Success  10000 2018-01-01
1    iPhone  Success  50000 2018-01-02
2  Motorola   Failed  15000 2018-01-03

0    0
1    1
2    2
Name: PurDate, dtype: int64
```