Notes on Date & Time in Python & Pandas

1. Importance in Data Science

Date & time handling is essential for time-series analysis, trend detection, forecasting, and seasonality analysis.

Typical tasks: converting strings → datetime, extracting components, calculating differences, and formatting for reporting.

1. Key Operations

Conversion

datetime.strptime() → Convert string → Python datetime/date.

pd.to_datetime() → Convert Series → pandas datetime (datetime64[ns]).

Extracting Parts

Python: .year, .month, .day, .strftime("%A"), .strftime("%B").

Pandas: .dt.year, .dt.month_name(), .dt.day_name().

Current Date & Time

datetime.date.today() → Current date.

datetime.datetime.now() → Current date & time.

Date Arithmetic

timedelta / pd.Timedelta → Add/subtract days, weeks, hours.

Vectorized in pandas for whole columns.

Date Difference

Subtraction gives a timedelta.

Use .days or .total_seconds() for meaningful values.

Formatting

strftime() → Custom string formats (%Y-%m-%d, %d-%b-%Y, %I:%M %p).

Pandas: .dt.strftime() for Series formatting.

Summary

Convert strings → datetime (strptime, pd.to_datetime).

Extract year, month, day, names for grouping/analysis.

Use timedelta for shifting dates and calculating durations.

Format with strftime for clean reporting.

Pandas .dt accessor is essential for working with datetime columns at scale.

Common strftime Codes

Year

%Y → 2025 (full year)

%y → 25 (2-digit year)

Month

%m → 09 (month number)

%B → September (full name)

%b → Sep (short name)

Day

%d → 15 (day number)

%A → Monday (full name)

%a → Mon (short name)

Time

%H → Hour (00–23)

%I → Hour (01–12)

%M → Minute

%S → Second

%p → AM/PM

```python
date_data = '2014-11-1'
type(date_data)
```

```
str
```

```python
# Converting str to datetime.date using strptime
import datetime

converted_time = datetime.datetime.strptime(date_data, '%Y-%m-%d').date()
type(converted_time)
```

```
datetime.date
```

```python
# Extracting year
converted_time.year
```

```
2014
```

```
# Extracting month
converted_time.month
```

```
11
```

```
#Dayname
converted_time.strftime('%A')
```

```
'Saturday'
```

```
#Month name
converted_time.strftime('%b')
```

```
'Nov'
```

```
# Todays dateandtime

datetime.datetime.now()
```

```
datetime.datetime(2025, 9, 15, 19, 1, 10, 260958)
```

```
# Todays date

datetime.date.today()
```

```
datetime.date(2025, 9, 15)
```

```
# Edate # add 3 months from today

today = datetime.date.today()
add_3Months = today + datetime.timedelta(days=90)
print(add_3Months)
```

```
2025-12-14
```

```
## Datediff calculation
My_Birth_Date = datetime.date(1997,11,8)
today = datetime.date.today()
datediff = (today - My_Birth_Date).days
year = datediff // 365
month = (datediff // 365) % 30
days = (datediff % 365) % 30

print(f'{year} years. {month} months and {days} days')
```

```
27 years. 27 months and 18 days
```

```
import pandas as pd
sales_data = {
    'Product': ['Laptop', 'Laptop', 'Laptop', 'Mobile', 'Mobile', 
'Mobile',
                'Tablet', 'Tablet', 'Tablet', 'Headphones',
'Headphones', 'Headphones',
                'Camera', 'Camera', 'Camera', 'Printer', 'Printer',
'Printer',
                'Monitor', 'Monitor'],
    'Region': ['North', 'South', 'West', 'North', 'South', 'West',
               'North', 'South', 'West', 'North', 'South', 'West',
               'North', 'South', 'West', 'North', 'South', 'West',
               'North', 'South'],
    'Sales': [120000, 95000, 88000, 150000, 132000, 128000,
              67000, 72000, 65000, 40000, 37000, 35000,
              78000, 82000, 79000, 56000, 48000, 50000,
              92000, 87000],
    'Quantity': [120, 95, 88, 300, 280, 275,
                 150, 160, 140, 200, 190, 185,
                 90, 100, 95, 70, 65, 68,
                 110, 105],
    'Date': ['2024-01-15', '2024-01-15', '2024-01-15', '2024-02-10',
'2024-02-10', '2024-02-10',
             '2024-03-05', '2024-03-05', '2024-03-05', '2024-04-20',
'2024-04-20', '2024-04-20',
             '2024-05-12', '2024-05-12', '2024-05-12', '2024-06-08',
'2024-06-08', '2024-06-08',
             '2024-07-25', '2024-07-25']
}

df = pd.DataFrame(sales_data)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Product   20 non-null     object
 1   Region    20 non-null     object
 2   Sales     20 non-null     int64 
 3   Quantity  20 non-null     int64 
 4   Date      20 non-null     object
dtypes: int64(2), object(3)
memory usage: 932.0+ bytes
```

```
# Pandas convertion to date
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d',
errors='coerce')
df['Date'].dtype == 'datetime64[ns]'
```

```
True
```

```
df['Date'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 20 entries, 0 to 19
Series name: Date
Non-Null Count  Dtype
--------------  -----
20 non-null     datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 292.0 bytes
```

```python
df['Date']
```

```
0    2024-01-15
1    2024-01-15
2    2024-01-15
3    2024-02-10
4    2024-02-10
5    2024-02-10
6    2024-03-05
7    2024-03-05
8    2024-03-05
9    2024-04-20
10   2024-04-20
11   2024-04-20
12   2024-05-12
13   2024-05-12
14   2024-05-12
15   2024-06-08
16   2024-06-08
17   2024-06-08
18   2024-07-25
19   2024-07-25
Name: Date, dtype: datetime64[ns]
```

```python
## Extracting year

df['Date'].dt.year
```

```
0     2024
1     2024
2     2024
3     2024
4     2024
5     2024
6     2024
7     2024
8     2024
9     2024
10    2024
11    2024
12    2024
```

```
13    2024
14    2024
15    2024
16    2024
17    2024
18    2024
19    2024
Name: Date, dtype: int32
```

```
## Month name

df['Date'].dt.month_name()
```

```
0      January
1      January
2      January
3     February
4     February
5     February
6        March
7        March
8        March
9        April
10       April
11       April
12         May
13         May
14         May
15        June
16        June
17        June
18        July
19        July
Name: Date, dtype: object
```

```
df['Date'].dt.day_name()
```

```
0       Monday
1       Monday
2       Monday
3     Saturday
4     Saturday
5     Saturday
6      Tuesday
7      Tuesday
8      Tuesday
9     Saturday
10    Saturday
11    Saturday
12      Sunday
```

```
13      Sunday
14      Sunday
15    Saturday
16    Saturday
17    Saturday
18    Thursday
19    Thursday
Name: Date, dtype: object
```

```
# one year later
df['Date'] + pd.Timedelta(days=370)
```

```
0    2025-01-19
1    2025-01-19
2    2025-01-19
3    2025-02-14
4    2025-02-14
5    2025-02-14
6    2025-03-10
7    2025-03-10
8    2025-03-10
9    2025-04-25
10   2025-04-25
11   2025-04-25
12   2025-05-17
13   2025-05-17
14   2025-05-17
15   2025-06-13
16   2025-06-13
17   2025-06-13
18   2025-07-30
19   2025-07-30
Name: Date, dtype: datetime64[ns]
```

```
# Deduct one year from '1997-11-8'

datetime.date(1997,11,8) - datetime.timedelta(days=370)
```

```
datetime.date(1996, 11, 3)
```

```
## Date formatting

today = datetime.date.today()
today.strftime('%Y-%b-%d')
```

```
'2025-Sep-15'
```

```
today = datetime.date.today()
today.strftime('%y-%b-%d')
```

```
'25-Sep-15'
```

```
today = datetime.date.today()
today.strftime('%y/%b/%d')
```

```
'25/Sep/15'
```

```
today = datetime.date.today()
today.strftime('%y/%b/%a')
```

```
'25/Sep/Mon'
```

Use .strftime() to format datetime into strings.

Codes like %Y, %m, %d, %H, %M, %S, %A, %B give you full control over format.

```
df['Date']
```

```
0    2024-01-15
1    2024-01-15
2    2024-01-15
3    2024-02-10
4    2024-02-10
5    2024-02-10
6    2024-03-05
7    2024-03-05
8    2024-03-05
9    2024-04-20
10   2024-04-20
11   2024-04-20
12   2024-05-12
13   2024-05-12
14   2024-05-12
15   2024-06-08
16   2024-06-08
17   2024-06-08
18   2024-07-25
19   2024-07-25
Name: Date, dtype: datetime64[ns]
```

```
## Pandas date formatting

df['Date'].dt.strftime('%b')
```

```
0      Jan
1      Jan
2      Jan
3      Feb
4      Feb
5      Feb
6      Mar
7      Mar
8      Mar
```

```
9     Apr
10    Apr
11    Apr
12    May
13    May
14    May
15    Jun
16    Jun
17    Jun
18    Jul
19    Jul
Name: Date, dtype: object
```

```
df['Date'].dt.strftime('%y:%m:%d')
```

```
0     24:01:15
1     24:01:15
2     24:01:15
3     24:02:10
4     24:02:10
5     24:02:10
6     24:03:05
7     24:03:05
8     24:03:05
9     24:04:20
10    24:04:20
11    24:04:20
12    24:05:12
13    24:05:12
14    24:05:12
15    24:06:08
16    24:06:08
17    24:06:08
18    24:07:25
19    24:07:25
Name: Date, dtype: object
```

```
df['Date'].dt.strftime('%A-%b')
```

```
0       Monday-Jan
1       Monday-Jan
2       Monday-Jan
3     Saturday-Feb
4     Saturday-Feb
5     Saturday-Feb
6      Tuesday-Mar
7      Tuesday-Mar
8      Tuesday-Mar
9     Saturday-Apr
10    Saturday-Apr
```

```
11    Saturday-Apr
12      Sunday-May
13      Sunday-May
14      Sunday-May
15    Saturday-Jun
16    Saturday-Jun
17    Saturday-Jun
18    Thursday-Jul
19    Thursday-Jul
Name: Date, dtype: object
```

```
DateVaariable = '2024-11-1'
type(DateVaariable)
```

```
str
```

```
import datetime
```

```
DateVaariable = datetime.datetime.strptime(DateVaariable, '%Y-%m-
%d').date()
print(DateVaariable)
```

```
2024-11-01
```

```
type(DateVaariable)
```

```
datetime.date
```