Python Programming Basics

Testing Working with Dates and Time in Python

#### Contents

- 1. Introduction
- 2. datetime library Functions
- 3. Merge Three Different Columns Into a Date in Python
- 4. Format a Vector With Inconsistent Date Formats

#### Introduction

- Python has a range of date manipulation functions in datetime library that allow us to work with dates and time.
- Working on dates and time can be tedious when the data come with date values in different format.
- The datetime library of Python, converts a variety of character date formats into
  Python dates. Once converted to dates, the following functions will return
  information about dates: second, minute, hour, month, year
- Pandas also has a Timestamp function. The inbuilt function offers a nice way to make easy parsing in dates and times.

#### Base Package Functions

datetime.strptime() converts dates entered as strings into numeric dates.

datetime.strptime(x, "%Y-%m-%d")

x is a string object to be converted

"%Y-%m-%d" is the format (in which the date appears within the string) composed of codes such as:

Day	day as a number (01-31)		
	abbreviated weekday (Mon)		
	full weekday name (Monday)	%A	
Month	abbreviated month (Jan)		
	full month name (January)		
	month as a number (01-12)	%m	
Year	2-digit year (16)		
	4-digit year (2016)	%Y	

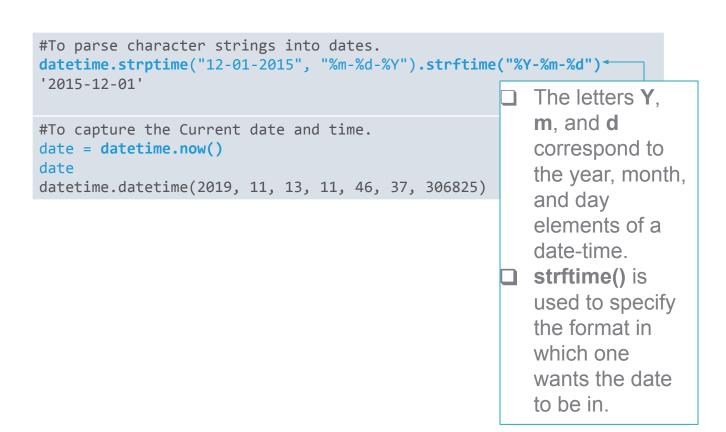
```
# Formatting a date
from datetime import datetime
x = '5 jan 2010'
ndate = datetime.strftime(datetime.strptime(x, '%d %b %Y'),'%d %b %Y')
ndate
'05 Jan 2010'
                  In Python we need to specify the format of
type(ndate)
                  the input date.
str
ndate2 = datetime.strptime(x, '%d %b %Y') ←
                                            datetime.strptime() converts x
type(ndate2)
                                            to a datetime object
datetime.datetime
# Using strftime argument to extract parts of date
datetime.strptime(ndate, '%d %b %Y').strftime('%Y%B')
'2010January'
                            Format codes can also be used to extract
                             parts of dates using strftime ()
```

```
#To extract Day of the week.
datetime.strptime(ndate, '%d %b %Y').strftime('%A')
'Tuesday'

#To extract Month of the Year.
datetime.strptime(ndate, '%d %b %Y').strftime('%B')
'January'
```

Apart from datetime library, pandas also have functions that deal with timestamps.

```
#To extract Quarter no.
import pandas as pd
pd.Timestamp(ndate).quarter
1
```



```
#To extract the hour component from the date object.
date.hour
11

#To extract the minute component from the date object.
date.minute
46

#To extract the second component from the date object.
date.second
37
```

```
# Capture current date
from datetime import date
                             date.today() returns the
today = date.today()
                             your system's current date
today
datetime.date(2019, 11, 13)
# Using operators with dates
d1 = datetime.date(datetime.strptime("20101201",'%Y%m%d'))
d2 = datetime.date(datetime.strptime("10/7/04",'%m/%d/%y'))
d1
d2
datetime.date(2010, 12, 1)
datetime.date(2004, 10, 7)
d1-d2
datetime.timedelta(2246)
                            Different operators can be
                            used with date objects
```

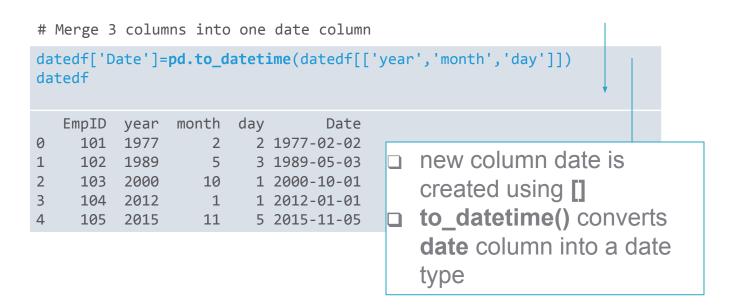
# Merge Three Different Columns Into a Date in Python

# My Dataframe

```
d = \{'EmpID' : [101, 102, 103, 104, 105],
    'year' : [1977,1989,2000,2012,2015],
    'month' : [2,5,10,1,11],
    'day' : [2,3,1,1,5]}
datedf = pd.DataFrame(d)
datedf
   EmpID year
                month
                       day
                              Data: Employee ID (EmplD) and
              2 2
5 3
10 1
1 1
    101
        1977
    102 1989
                              joining date (split into 3 columns:
    103 2000
                              year month & day)
    104 2012
                  11
    105 2015
```

# Merge Three Different Columns Into a Date in Python

We are having 3 separate columns as year, month, and day in our dataframe datedf.



#### Format a Vector With Inconsistent Date Formats

Converting dates entered as strings into numeric dates in Python is a little tricky if the date information is not represented consistently. Let's see how to deal with this kind of situation.

Note that Pandas function to\_datetime() is capable of handling such discrepancies as long as order of the date elements is consistent.

#### Quick Recap

In this session, we learnt how to deal with dates and time using base package functions in Python & pandas Timestamp, how to merge 3 different columns into one date column and how to format a vector with inconsistent dates. Here is a quick recap:

