1 # Pandas: tutorial (day 11) 2 This notbook explain Pandas lib

In [1]:
```
!pip install pandoc
```

```
Collecting pandoc
  Using cached pandoc-2.0.1.tar.gz (29 kB)
Collecting plumbum
  Using cached plumbum-1.7.2-py2.py3-none-any.whl (117 kB)
Requirement already satisfied: ply in c:\programdata\anaconda3\lib\site-package
s (from pandoc) (3.11)
Requirement already satisfied: pywin32 in c:\programdata\anaconda3\lib\site-pac
kages (from plumbum->pandoc) (228)
Building wheels for collected packages: pandoc
  Building wheel for pandoc (setup.py): started
  Building wheel for pandoc (setup.py): finished with status 'done'
  Created wheel for pandoc: filename=pandoc-2.0.1-py3-none-any.whl size=29585 s
ha256=10caccc5a2cbdb71466b3118a943cb82cc6f2ce69241b14f5f5cb04915670242
  Stored in directory: c:\users\muyyassarhussain\appdata\local\pip\cache\wheels
\2c\e3\89\89ab2e7b2c48753cd84f54c13dce591468b3dff770cc03d672
Successfully built pandoc
Installing collected packages: plumbum, pandoc
Successfully installed pandoc-2.0.1 plumbum-1.7.2
```

1- How to install libraries

In [ ]:
```
1 # pip  install pandas
2 # pip install numpy
```

Import Libraries

In [10]:
```
import pandas as pd
import numpy as np
```

In [13]:
```
# object create
s = pd.Series([1,3,5,np.nan,7,8,9])
s
```

Out[13]:
```
0    1.0
1    3.0
2    5.0
3    NaN
4    7.0
5    8.0
6    9.0
dtype: float64
```

In [17]:
```
dates = pd.date_range("20220101", periods=20)
dates
```

Out[17]:
```
DatetimeIndex(['2022-01-01', '2022-01-02', '2022-01-03', '2022-01-04',
               '2022-01-05', '2022-01-06', '2022-01-07', '2022-01-08',
               '2022-01-09', '2022-01-10', '2022-01-11', '2022-01-12',
               '2022-01-13', '2022-01-14', '2022-01-15', '2022-01-16',
               '2022-01-17', '2022-01-18', '2022-01-19', '2022-01-20'],
              dtype='datetime64[ns]', freq='D')
```

In [30]:
```
dates = pd.date_range("20220101", periods=6)
dates
df = pd.DataFrame(np.random.randn(6, 4), index=dates, columns=list("ABCD"))
df
```

Out[30]:

	A	B	C	D
2022-01-01	-0.582858	0.003849	-1.159810	0.619239
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434
2022-01-03	-0.194256	0.862808	0.459097	0.109270
2022-01-04	0.847551	-0.158184	-0.032580	1.559263
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664

In [31]:
```
df.index
```

Out[31]:
```
DatetimeIndex(['2022-01-01', '2022-01-02', '2022-01-03', '2022-01-04',
               '2022-01-05', '2022-01-06'],
              dtype='datetime64[ns]', freq='D')
```

In [32]:
```
df.to_numpy
```

Out[32]:
```
<bound method DataFrame.to_numpy of                    A         B         C
D
2022-01-01 -0.582858  0.003849 -1.159810  0.619239
2022-01-02 -0.391342 -2.084955  0.968443 -0.852434
2022-01-03 -0.194256  0.862808  0.459097  0.109270
2022-01-04  0.847551 -0.158184 -0.032580  1.559263
2022-01-05 -1.022700 -0.939458  0.785812 -0.076403
2022-01-06 -0.194729 -2.234721 -0.197630 -0.754664>
```

In [33]:
```
df.describe
```

Out[33]:
```
<bound method NDFrame.describe of                    A         B         C
D
2022-01-01 -0.582858  0.003849 -1.159810  0.619239
2022-01-02 -0.391342 -2.084955  0.968443 -0.852434
2022-01-03 -0.194256  0.862808  0.459097  0.109270
2022-01-04  0.847551 -0.158184 -0.032580  1.559263
2022-01-05 -1.022700 -0.939458  0.785812 -0.076403
2022-01-06 -0.194729 -2.234721 -0.197630 -0.754664>
```

Sorting

In [37]:
```
df.sort_values(by="B" , ascending=True)
```

Out[37]:

	A	B	C	D
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403
2022-01-04	0.847551	-0.158184	-0.032580	1.559263
2022-01-01	-0.582858	0.003849	-1.159810	0.619239
2022-01-03	-0.194256	0.862808	0.459097	0.109270

In [38]:
```
df["D"]
```

Out[38]:
```
2022-01-01    0.619239
2022-01-02   -0.852434
2022-01-03    0.109270
2022-01-04    1.559263
2022-01-05   -0.076403
2022-01-06   -0.754664
Freq: D, Name: D, dtype: float64
```

In [40]:
```
# Row wise selection
df[0:2]
```

Out[40]:

	A	B	C	D
2022-01-01	-0.582858	0.003849	-1.159810	0.619239
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434

In [43]:
```
df.loc[dates[2]]
```

Out[43]:
```
A   -0.194256
B    0.862808
C    0.459097
D    0.109270
Name: 2022-01-03 00:00:00, dtype: float64
```

In [45]:
```
df.loc[:,["A", "B"]]
```

Out[45]:

	A	B
2022-01-01	-0.582858	0.003849
2022-01-02	-0.391342	-2.084955
2022-01-03	-0.194256	0.862808
2022-01-04	0.847551	-0.158184
2022-01-05	-1.022700	-0.939458
2022-01-06	-0.194729	-2.234721

In [52]:
```
df.loc[dates[5]]
```

Out[52]:
```
A   -0.194729
B   -2.234721
C   -0.197630
D   -0.754664
Name: 2022-01-06 00:00:00, dtype: float64
```

In [53]:
```
df.loc[:,["A", "B"]]
```

Out[53]:

	A	B
2022-01-01	-0.582858	0.003849
2022-01-02	-0.391342	-2.084955
2022-01-03	-0.194256	0.862808
2022-01-04	0.847551	-0.158184
2022-01-05	-1.022700	-0.939458
2022-01-06	-0.194729	-2.234721

In [55]:
```
df.loc[dates[5]]
```

Out[55]:
```
A   -0.194729
B   -2.234721
C   -0.197630
D   -0.754664
Name: 2022-01-06 00:00:00, dtype: float64
```

In [56]:
```
df.loc["20220102":"20220104",["A", "B"]]
```

Out[56]:

	A	B
2022-01-02	-0.391342	-2.084955
2022-01-03	-0.194256	0.862808
2022-01-04	0.847551	-0.158184

In [59]:
```
df.loc["20220102":"20220104",["A", "B","C"]]
```

Out[59]:

	A	B	C
2022-01-02	-0.391342	-2.084955	0.968443
2022-01-03	-0.194256	0.862808	0.459097
2022-01-04	0.847551	-0.158184	-0.032580

In [60]:
```
df.loc["20220102",["A", "B","C"]]
```

Out[60]:
```
A   -0.391342
B   -2.084955
C    0.968443
Name: 2022-01-02 00:00:00, dtype: float64
```

The codes can be accessed using below link

https://stackoverflow.com/questions/53903889/log behind-pandas-loc-selecting-values (https://stackoverflow.com/questions/53903889/log behind-pandas-loc-selecting-values)

In [63]:
```
df.at[dates[0],"A"]
```

Out[63]:
```
-0.5828584608597441
```

In [64]:
```
df.at[dates[5],"A"]
```

Out[64]:
```
-0.194728905546206
```

In [65]:
```
df.iloc333[3]
```

Out[65]:
```
A    0.847551
B   -0.158184
C   -0.032580
D    1.559263
Name: 2022-01-04 00:00:00, dtype: float64
```

In [69]:
```
df.iloc[0:2, 3:4]
```

Out[69]:

	D
2022-01-01	0.619239
2022-01-02	-0.852434

In [70]:
```
df.iloc[:, 0:2]
```

Out[70]:

	A	B
2022-01-01	-0.582858	0.003849
2022-01-02	-0.391342	-2.084955
2022-01-03	-0.194256	0.862808
2022-01-04	0.847551	-0.158184
2022-01-05	-1.022700	-0.939458
2022-01-06	-0.194729	-2.234721

Bolian

In [71]:
```
df [df["A"] > 0]
```

Out[71]:

	A	B	C	D
2022-01-04	0.847551	-0.158184	-0.03258	1.559263

In [72]:
```
df [df["B"] > 0]
```

Out[72]:

	A	B	C	D
2022-01-01	-0.582858	0.003849	-1.159810	0.619239
2022-01-03	-0.194256	0.862808	0.459097	0.109270

In [73]:
```
df [df["B"] > 1]
```

Out[73]:

	A	B	C	D

In [74]:
```
df [df > 0]
```

Out[74]:

	A	B	C	D
2022-01-01	NaN	0.003849	NaN	0.619239
2022-01-02	NaN	NaN	0.968443	NaN
2022-01-03	NaN	0.862808	0.459097	0.109270
2022-01-04	0.847551	NaN	NaN	1.559263
2022-01-05	NaN	NaN	0.785812	NaN
2022-01-06	NaN	NaN	NaN	NaN

In [79]:
```
df2 =df.copy()
```

In [85]:
```
df2["E"] = [ "one", "one" , "two", "three", "four","one"]
```

In [81]:
```
df2.shape
```

Out[81]:
```
(6, 4)
```

In [83]:
```
df2
```

Out[83]:

	A	B	C	D	E
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	one
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	one
2022-01-03	-0.194256	0.862808	0.459097	0.109270	two
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	three
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	four
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	three

In [89]:
```
df2 ["new"]=[1.2,2.2,2.3,3.3,3.4,3.5]
df2
```

Out[89]:

	A	B	C	D	E	mean	new
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	one	1.2	1.2
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	one	2.2	2.2
2022-01-03	-0.194256	0.862808	0.459097	0.109270	two	2.3	2.3
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	three	3.3	3.3
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	four	3.4	3.4
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	one	3.5	3.5

In [91]:
```
df2["mean"]
```

Out[91]:
```
2022-01-01    1.2
2022-01-02    2.2
2022-01-03    2.3
2022-01-04    3.3
2022-01-05    3.4
2022-01-06    3.5
Freq: D, Name: mean, dtype: float64
```

In [92]:
```
del df2["mean"]
df2
```

Out[92]:

	A	B	C	D	E	new
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	one	1.2
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	one	2.2
2022-01-03	-0.194256	0.862808	0.459097	0.109270	two	2.3
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	three	3.3
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	four	3.4
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	one	3.5

In [93]:
```
df2 = df2.drop(columns="E")
df2
```

Out[93]:

	A	B	C	D	new
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	1.2
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	2.2
2022-01-03	-0.194256	0.862808	0.459097	0.109270	2.3
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	3.3
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	3.4
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	3.5

In [94]:
```
df2.mean()
```

Out[94]:
```
A     -0.256389
B     -0.758444
C      0.137222
D      0.100712
new    2.650000
dtype: float64
```

In [96]:
```
df2["mean"] = df2.mean(axis=1)
df2
```

Out[96]:

	A	B	C	D	new	mean
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	1.2	0.016084
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	2.2	-0.032058
2022-01-03	-0.194256	0.862808	0.459097	0.109270	2.3	0.707384
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	3.3	1.103210
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	3.4	0.429450
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	3.5	0.023651

In [97]:
```
df2["std"] = df2.std(axis=1)
df2
```

Out[97]:

	A	B	C	D	new	mean	std
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	1.2	0.016084	0.837504
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	2.2	-0.032058	1.483211
2022-01-03	-0.194256	0.862808	0.459097	0.109270	2.3	0.707384	0.870968
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	3.3	1.103210	1.263358
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	3.4	0.429450	1.624173
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	3.5	0.023651	1.891437

In [98]:
```
df2.describe()
```

Out[98]:

	A	B	C	D	new	mean	std
count	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000
mean	-0.256389	-0.758444	0.137222	0.100712	2.650000	0.374620	1.328442
std	0.622635	1.228141	0.779545	0.901980	0.909395	0.460803	0.420217
min	-1.022700	-2.234721	-1.159810	-0.852434	1.200000	-0.032058	0.837504
25%	-0.534979	-1.798581	-0.156367	-0.585098	2.225000	0.017976	0.969065
50%	-0.293036	-0.548821	0.213258	0.016434	2.800000	0.226551	1.373284
75%	-0.194374	-0.036659	0.704133	0.491747	3.375000	0.637900	1.588933
max	0.847551	0.862808	0.968443	1.559263	3.500000	1.103210	1.891437

In [99]:
```
df2
```

Out[99]:

	A	B	C	D	new	mean	std
2022-01-01	-0.582858	0.003849	-1.159810	0.619239	1.2	0.016084	0.837504
2022-01-02	-0.391342	-2.084955	0.968443	-0.852434	2.2	-0.032058	1.483211
2022-01-03	-0.194256	0.862808	0.459097	0.109270	2.3	0.707384	0.870968
2022-01-04	0.847551	-0.158184	-0.032580	1.559263	3.3	1.103210	1.263358
2022-01-05	-1.022700	-0.939458	0.785812	-0.076403	3.4	0.429450	1.624173
2022-01-06	-0.194729	-2.234721	-0.197630	-0.754664	3.5	0.023651	1.891437

In [104]:
```
#iloc
#loc

df2.loc['2022-01-02','D']
```

Out[104]:
```
-0.8524338690180518
```

In [100]:
```
df2.index
```

Out[100]:
```
DatetimeIndex(['2022-01-01', '2022-01-02', '2022-01-03', '2022-01-04',
               '2022-01-05', '2022-01-06'],
              dtype='datetime64[ns]', freq='D')
```

In [ ]:
```

```