Pandas Practice

Import the Libraries

- **Install libraries through pip (Python Packages Manager)**
- pip install numpy
- pip install pandas

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

data = pd.Series([3, 5, 34, 90, 1]) # Series is like column in excel sheet
data
```

Out[ ]:
```
0     3
1     5
2    34
3    90
4     1
dtype: int64
```

In [ ]:
```
list_dates = pd.date_range('20220105', periods= 5)
list_dates
```

Out[ ]:
```
DatetimeIndex(['2022-01-05', '2022-01-06', '2022-01-07', '2022-01-08',
               '2022-01-09'],
              dtype='datetime64[ns]', freq='D')
```

In [ ]:
```
# pass list_dates as in index,
df = pd.DataFrame(np.random.randn(5, 4), index = list_dates,  columns = list('ABDC'))
df
```

Out[ ]:

	A	B	D	C
2022-01-05	-0.862948	0.746623	-0.887232	0.669863
2022-01-06	-1.584317	-1.373865	0.263347	1.623824
2022-01-07	-0.715966	1.063780	0.359892	-0.618176
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306
2022-01-09	-1.327070	0.988475	0.425598	-1.055742

In [ ]:
```
# it will show the first two heads or row
df.head(2)
```

Out[ ]:

	A	B	D	C
2022-01-05	-0.862948	0.746623	-0.887232	0.669863
2022-01-06	-1.584317	-1.373865	0.263347	1.623824

In [ ]:

```
# it will show last two tail
df.tail(2)
```

Out[ ]:

	A	B	D	C
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306
2022-01-09	-1.327070	0.988475	0.425598	-1.055742

Change DateFrame into numpy 2-D array

In [ ]:
```
df.to_numpy()
```

Out[ ]:
```
array([[-0.86294836,  0.74662315, -0.88723155,  0.66986251],
       [-1.58431714, -1.37386494,  0.26334703,  1.62382359],
       [-0.71596567,  1.06378022,  0.35989171, -0.61817573],
       [-0.36165429,  2.12528779, -1.64929777, -1.10230578],
       [-1.32707003,  0.98847478,  0.42559758, -1.05574221]])
```

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

Out[ ]:
```
DatetimeIndex(['2022-01-05', '2022-01-06', '2022-01-07', '2022-01-08',
               '2022-01-09'],
              dtype='datetime64[ns]', freq='D')
```

In [ ]:
```
# Describe() show the statistical information about data
df.describe()
```

Out[ ]:

	A	B	D	C
count	5.000000	5.000000	5.000000	5.000000
mean	-0.970391	0.710060	-0.297539	-0.096508
std	0.487586	1.279608	0.927998	1.199037
min	-1.584317	-1.373865	-1.649298	-1.102306
25%	-1.327070	0.746623	-0.887232	-1.055742
50%	-0.862948	0.988475	0.263347	-0.618176
75%	-0.715966	1.063780	0.359892	0.669863
max	-0.361654	2.125288	0.425598	1.623824

In [ ]:
```
# sort indexes
df.sort_index(ascending = False)
```

Out[ ]:

	A	B	D	C
2022-01-09	-1.327070	0.988475	0.425598	-1.055742
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306
2022-01-07	-0.715966	1.063780	0.359892	-0.618176
2022-01-06	-1.584317	-1.373865	0.263347	1.623824
2022-01-05	-0.862948	0.746623	-0.887232	0.669863

In [ ]:
```
# sort by value
df.sort_values(by='A', ascending=True)
```

Out[ ]:

	A	B	D	C
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306
2022-01-07	-0.715966	1.063780	0.359892	-0.618176
2022-01-05	-0.862948	0.746623	-0.887232	0.669863
2022-01-09	-1.327070	0.988475	0.425598	-1.055742
2022-01-06	-1.584317	-1.373865	0.263347	1.623824

In [ ]:
```
df['B']
```

Out[ ]:
```
2022-01-05    0.746623
2022-01-06   -1.373865
2022-01-07    1.063780
2022-01-08    2.125288
2022-01-09    0.988475
Freq: D, Name: B, dtype: float64
```

In [ ]:
```
# from 06 to 08
df.loc[ '20220106' :  '20220108', ['A', 'B', 'C'] ]
```

Out[ ]:

	A	B	C
2022-01-06	-1.584317	-1.373865	1.623824
2022-01-07	-0.715966	1.063780	-0.618176
2022-01-08	-0.361654	2.125288	-1.102306

In [ ]:
```
# from 06 and 08
df.loc[ ['20220106' ,  '20220108'], ['A', 'B', 'C'] ]
```

Out[ ]:

	A	B	C
2022-01-06	-1.584317	-1.373865	1.623824
2022-01-08	-0.361654	2.125288	-1.102306

In [ ]:
```
list_dates
```

Out[ ]:
```
DatetimeIndex(['2022-01-05', '2022-01-06', '2022-01-07', '2022-01-08',
               '2022-01-09'],
              dtype='datetime64[ns]', freq='D')
```

In [ ]:
```
df.at[list_dates[0], 'A']
```

Out[ ]:
```
-0.8629483622771852
```

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

Out[ ]:

	A	B

	A	B
2022-01-05	-0.862948	0.746623
2022-01-06	-1.584317	-1.373865
2022-01-07	-0.715966	1.063780

In [ ]:
```
df.iloc[ : , : 1]
```

Out[ ]:

	A
2022-01-05	-0.862948
2022-01-06	-1.584317
2022-01-07	-0.715966
2022-01-08	-0.361654
2022-01-09	-1.327070

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

Out[ ]:

	A	B	D	C
2022-01-05	-0.862948	0.746623	-0.887232	0.669863
2022-01-06	-1.584317	-1.373865	0.263347	1.623824
2022-01-07	-0.715966	1.063780	0.359892	-0.618176
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306
2022-01-09	-1.327070	0.988475	0.425598	-1.055742

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

Out[ ]:

	A	B	D	C
2022-01-05	NaN	0.746623	NaN	0.669863
2022-01-06	NaN	NaN	0.263347	1.623824
2022-01-07	NaN	1.063780	0.359892	NaN
2022-01-08	NaN	2.125288	NaN	NaN
2022-01-09	NaN	0.988475	0.425598	NaN

In [ ]:
```
df[df['B'] > 0 ]
```

Out[ ]:

	A	B	D	C	E
2022-01-05	-0.862948	0.746623	-0.887232	0.669863	Khan
2022-01-07	-0.715966	1.063780	0.359892	-0.618176	Ujala
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306	Rehman

	A	B	D	C	E
2022-01-09	-1.327070	0.988475	0.425598	-1.055742	Baba

In [ ]:
```
# add new column
df2['E'] = ['Khan', 'Jawad', 'Ujala', 'Rehman', 'Baba']
df2
```

Out[ ]:

	A	B	D	C	E
2022-01-05	-0.862948	0.746623	-0.887232	0.669863	Khan
2022-01-06	-1.584317	-1.373865	0.263347	1.623824	Jawad
2022-01-07	-0.715966	1.063780	0.359892	-0.618176	Ujala
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306	Rehman
2022-01-09	-1.327070	0.988475	0.425598	-1.055742	Baba

In [ ]:
```

```

Out[ ]:

	A	B	D	C	E	mean
2022-01-05	-0.862948	0.746623	-0.887232	0.669863	Khan	-0.083424
2022-01-06	-1.584317	-1.373865	0.263347	1.623824	Jawad	-0.267753
2022-01-07	-0.715966	1.063780	0.359892	-0.618176	Ujala	0.022383
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306	Rehman	-0.246993
2022-01-09	-1.327070	0.988475	0.425598	-1.055742	Baba	-0.242185

Assignment

- ### Find Mean of all values of column and store it into mean

In [ ]:
```
# find mean of the row and store it into mean column
df2['mean'] = df2.iloc[ :  , : 5].mean(axis=1)
```

In [ ]:
```
df2
```

Out[ ]:

	A	B	D	C	E	mean
2022-01-05	-0.862948	0.746623	-0.887232	0.669863	Khan	-0.083424
2022-01-06	-1.584317	-1.373865	0.263347	1.623824	Jawad	-0.267753
2022-01-07	-0.715966	1.063780	0.359892	-0.618176	Ujala	0.022383
2022-01-08	-0.361654	2.125288	-1.649298	-1.102306	Rehman	-0.246993
2022-01-09	-1.327070	0.988475	0.425598	-1.055742	Baba	-0.242185

In [ ]:
```

```