Pandas: tuturial (Day-11)

This notebook expains how to use pandas to explore data using Python. We will walk through with the basics at first and then move on to more advanced concepts.

Installation Dependencies

In [ ]:

```
# pip install pandas
# pip install numpy
```

Import Libraries

In [ ]:

```
import numpy as np
import pandas as pd
```

In [ ]:

```
# Create object
s = pd.Series([1,3,5,np.nan,7,8,4])
s
```

Out[ ]:

```
0    1.0
1    3.0
2    5.0
3    NaN
4    7.0
5    8.0
6    4.0
dtype: float64
```

In [ ]:

```
# Create date object
dates = pd.date_range("20220119", periods=7)
dates
```

Out[ ]:

```
DatetimeIndex(['2022-01-19', '2022-01-20', '2022-01-21', '2022-01-22',
               '2022-01-23', '2022-01-24', '2022-01-25'],
              dtype='datetime64[ns]', freq='D')
```

In [ ]:

```
# Convert dates object into dataframe
df = pd.DataFrame(np.random.randn(7,4), index=dates, columns=list("ABCD"))
df
```

Out[ ]:

	A	B	C	D
2022-01-19	0.552871	0.747349	0.236371	-0.079941
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-24	0.090992	0.127961	-0.517612	-0.371983
2022-01-25	0.220869	-1.001742	0.024885	1.498282

In [ ]:

```
# Create dataframe from dictionary
df2 = pd.DataFrame(
    {
        "A": 1.2,
        "B": pd.Timestamp("20220119"),
        "C": pd.Series(1, index=list(range(4)), dtype="float32"),
        "D": np.array([3] * 4, dtype="int32"),
        "E": pd.Categorical(["test", "train", "test", "train"]),
        "F": "foo"
    }
)

df2
```

Out[ ]:

	A	B	C	D	E	F
0	1.2	2022-01-19	1.0	3	test	foo
1	1.2	2022-01-19	1.0	3	train	foo
2	1.2	2022-01-19	1.0	3	test	foo
3	1.2	2022-01-19	1.0	3	train	foo

In [ ]:

```
# Find data types of all columns in the dataframe
df2.dtypes
```

Out[ ]:

```
A           float64
B    datetime64[ns]
C           float32
D             int32
E          category
F            object
dtype: object
```

In [ ]:

```
# View data
df.head(2)
```

Out[ ]:

	A	B	C	D
2022-01-19	0.552871	0.747349	0.236371	-0.079941
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330

In [ ]:

```
df.tail(3)
```

Out[ ]:

	A	B	C	D
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-24	0.090992	0.127961	-0.517612	-0.371983
2022-01-25	0.220869	-1.001742	0.024885	1.498282

In [ ]:

```
# Find rows in the dataframe
df.index
```

Out[ ]:

```
DatetimeIndex(['2022-01-19', '2022-01-20', '2022-01-21', '2022-01-22',
               '2022-01-23', '2022-01-24', '2022-01-25'],
              dtype='datetime64[ns]', freq='D')
```

In [ ]:

```
df2.index
```

Out[ ]:

```
Int64Index([0, 1, 2, 3], dtype='int64')
```

In [ ]:

```
# Convert dataframe into numpy array
df.to_numpy()
```

Out[ ]:

```
array([[ 0.55287128,  0.74734903,  0.2363708 , -0.07994084],
       [-0.44959817, -0.19848314,  1.60275316, -0.66732998],
       [-0.58688547, -0.83409388, -2.19586949,  0.19552881],
       [-0.46505327, -0.45848132, -0.50946407,  0.71550931],
       [-1.06061766,  0.73847179,  0.84305659,  1.84909011],
       [ 0.0909924 ,  0.1279612 , -0.51761188, -0.37198255],
       [ 0.22086876, -1.00174172,  0.02488546,  1.49828169]])
```

In [ ]:

```
# Find statistical values
df.describe()
```

Out[ ]:

	A	B	C	D
count	7.000000	7.000000	7.000000	7.000000
mean	-0.242489	-0.125574	-0.073697	0.448451
std	0.553606	0.702218	1.200510	0.948376
min	-1.060618	-1.001742	-2.195869	-0.667330
25%	-0.525969	-0.646288	-0.513538	-0.225962
50%	-0.449598	-0.198483	0.024885	0.195529
75%	0.155931	0.433216	0.539714	1.106895
max	0.552871	0.747349	1.602753	1.849090

In [ ]:

```
# Transpose dataframe
df.T
```

Out[ ]:

	2022-01-19	2022-01-20	2022-01-21	2022-01-22	2022-01-23	2022-01-24	2022-01-25
A	0.552871	-0.449598	-0.586885	-0.465053	-1.060618	0.090992	0.220869
B	0.747349	-0.198483	-0.834094	-0.458481	0.738472	0.127961	-1.001742
C	0.236371	1.602753	-2.195869	-0.509464	0.843057	-0.517612	0.024885
D	-0.079941	-0.667330	0.195529	0.715509	1.849090	-0.371983	1.498282

In [ ]:

```
# Sort elements in the dataframe (row wise)
df.sort_index(axis=0, ascending=False)
```

Out[ ]:

	A	B	C	D
2022-01-25	0.220869	-1.001742	0.024885	1.498282
2022-01-24	0.090992	0.127961	-0.517612	-0.371983
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330
2022-01-19	0.552871	0.747349	0.236371	-0.079941

In [ ]:

```
# Sort dataframe values by a specific column
df.sort_values(by="C")
```

Out[ ]:

	A	B	C	D
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529
2022-01-24	0.090992	0.127961	-0.517612	-0.371983
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509
2022-01-25	0.220869	-1.001742	0.024885	1.498282
2022-01-19	0.552871	0.747349	0.236371	-0.079941
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330

In [ ]:

```
# Selection of dataframe
df["B"]
```

Out[ ]:

```
2022-01-19    0.747349
2022-01-20   -0.198483
2022-01-21   -0.834094
2022-01-22   -0.458481
2022-01-23    0.738472
2022-01-24    0.127961
2022-01-25   -1.001742
Freq: D, Name: B, dtype: float64
```

In [ ]:

```
# Row wise slicing
df[2:6]
```

Out[ ]:

	A	B	C	D
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-24	0.090992	0.127961	-0.517612	-0.371983

In [ ]:

```
# Selection with labels
df.loc[dates[0]]
```

Out[ ]:

```
A    0.552871
B    0.747349
C    0.236371
D   -0.079941
Name: 2022-01-19 00:00:00, dtype: float64
```

In [ ]:

```
# Select multiple columns
df.loc[:, ["B", "C"]]
```

Out[ ]:

	B	C

2022-01-19	0.747349	0.236371
2022-01-20	-0.198483	1.602753
2022-01-21	-0.834094	-2.195869
2022-01-22	-0.458481	-0.509464
2022-01-23	0.738472	0.843057
2022-01-24	0.127961	-0.517612
2022-01-25	-1.001742	0.024885

In [ ]:

```
# Slice through rows with specific columns
df.loc["20220120":"20220124", ["A", "B", "C"]]
```

Out[ ]:

	A	B	C
2022-01-20	-0.449598	-0.198483	1.602753
2022-01-21	-0.586885	-0.834094	-2.195869
2022-01-22	-0.465053	-0.458481	-0.509464
2022-01-23	-1.060618	0.738472	0.843057
2022-01-24	0.090992	0.127961	-0.517612

In [ ]:

```
# Extract value from specific row
df.loc["20220124", ["A", "B", "C"]]
```

Out[ ]:

```
A    0.090992
B    0.127961
C   -0.517612
Name: 2022-01-24 00:00:00, dtype: float64
```

In [ ]:

```
# Slicing through iloc
df.iloc[3:10]
```

Out[ ]:

	A	B	C	D
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-24	0.090992	0.127961	-0.517612	-0.371983
2022-01-25	0.220869	-1.001742	0.024885	1.498282

In [ ]:

```
# iloc slicing on specify columns
df.iloc[:, 0:2]
```

Out[ ]:

	A	B
2022-01-19	0.552871	0.747349
2022-01-20	-0.449598	-0.198483
2022-01-21	-0.586885	-0.834094
2022-01-22	-0.465053	-0.458481
2022-01-23	-1.060618	0.738472
2022-01-24	0.090992	0.127961
2022-01-25	0.220869	-1.001742

In [ ]:

```
# Boolean operations on dataframe
df[df["B"] > 0.2]
```

Out[ ]:

	A	B	C	D
2022-01-19	0.552871	0.747349	0.236371	-0.079941
2022-01-23	-1.060618	0.738472	0.843057	1.849090

In [ ]:

```
df[df > 0]
```

Out[ ]:

	A	B	C	D
2022-01-19	0.552871	0.747349	0.236371	NaN
2022-01-20	NaN	NaN	1.602753	NaN
2022-01-21	NaN	NaN	NaN	0.195529
2022-01-22	NaN	NaN	NaN	0.715509
2022-01-23	NaN	0.738472	0.843057	1.849090
2022-01-24	0.090992	0.127961	NaN	NaN
2022-01-25	0.220869	NaN	0.024885	1.498282

In [ ]:

```
# Copy dataframe
df3 = df.copy()
```

In [ ]:

```
# Add new column in df3
df3["E"] = ["one", "two", "three", "four", "five", "six", "seven"]
df3
```

Out[ ]:

	A	B	C	D	E
2022-01-19	0.552871	0.747349	0.236371	-0.079941	one
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330	two
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529	three
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509	four
2022-01-23	-1.060618	0.738472	0.843057	1.849090	five
2022-01-24	0.090992	0.127961	-0.517612	-0.371983	six
2022-01-25	0.220869	-1.001742	0.024885	1.498282	seven

In [ ]:

```
# Update df3 (remove column E)
df3 = df3.loc[:, :"D"]
df3
```

Out[ ]:

	A	B	C	D
2022-01-19	0.552871	0.747349	0.236371	-0.079941
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509
2022-01-23	-1.060618	0.738472	0.843057	1.849090
2022-01-24	0.090992	0.127961	-0.517612	-0.371983
2022-01-25	0.220869	-1.001742	0.024885	1.498282

In [ ]:

```
# Calculate row wise mean
df3["mean"] = df3.mean(axis=1)
df3
```

Out[ ]:

	A	B	C	D	mean
2022-01-19	0.552871	0.747349	0.236371	-0.079941	0.364163
2022-01-20	-0.449598	-0.198483	1.602753	-0.667330	0.071835
2022-01-21	-0.586885	-0.834094	-2.195869	0.195529	-0.855330
2022-01-22	-0.465053	-0.458481	-0.509464	0.715509	-0.179372
2022-01-23	-1.060618	0.738472	0.843057	1.849090	0.592500
2022-01-24	0.090992	0.127961	-0.517612	-0.371983	-0.167660
2022-01-25	0.220869	-1.001742	0.024885	1.498282	0.185574