pandas-introduction

October 1, 2023

1 ——————-Pandas——————

It supports two data structures:

1. Series
2. Dataframe

1. Series

Create an Empty Series

[6]:
```python
import pandas as pd
ls = pd.Series()
print(ls)
```

```
Series([], dtype: float64)
```

```
C:\Users\ABDULLAH KHAN\AppData\Local\Temp\ipykernel_18452\2248554822.py:2:
FutureWarning: The default dtype for empty Series will be 'object' instead of
'float64' in a future version. Specify a dtype explicitly to silence this
warning.
  ls = pd.Series()
```

Create a Series from ndarray

[7]:
```python
import pandas as pd
import numpy as np
data = np.array(['a','b','c','d'])
ls = pd.Series(data)
print(ls)
```

```
0    a
1    b
2    c
3    d
dtype: object
```

Create a Pandas Series from a list:

[1]:
```python
import pandas as pd
a = [6, 7, 2]
```

```
ls = pd.Series(a)
print(ls)
```

```
0    6
1    7
2    2
dtype: int64
```

Create a Pandas Series from a dictionary:

[2]:
```
import pandas as pd
calories = {"day1": 420,
            "day2": 380,
            "day3": 390
           }
df = pd.Series(calories)
print(df)
```

```
day1    420
day2    380
day3    390
dtype: int64
```

1.0.1 Create DataFrame

A pandas DataFrame can be created using various inputs like –

Lists

dict

Series

Numpy ndarrays

Another DataFrame

Create an Empty DataFrame

[9]:
```
import pandas as pd
df = pd.DataFrame()
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

Create a DataFrame from Lists

[10]:
```
import pandas as pd
a = [6, 7, 2]
ls = pd.DataFrame(a)
```

```
print(ls)
```

```
   0
0  6
1  7
2  2
```

Create a DataFrame from Dict of ndarrays / Lists

```
[13]: import pandas as pd
data = {
        'Name':['Tom', 'Jack', 'Steve', 'Ricky'],
        'Age':[28,34,29,42]
       }
df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
print(df)
```

```
        Name  Age
rank1    Tom   28
rank2   Jack   34
rank3  Steve   29
rank4  Ricky   42
```

Create a DataFrame from List of Dicts

```
[19]: import pandas as pd
data = [
    {'a': 1, 'b': 2},
    {'a': 5, 'b': 10, 'c': 20}
    ]
df = pd.DataFrame(data)
print(df.loc[0])#Pandas use the loc attribute to return one or more specified␣
 ↪row(s).
```

```
a    1.0
b    2.0
c    NaN
Name: 0, dtype: float64
```

1.0.2 Pandas - Descriptive Statistics

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

d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
   'Lee','David','Gasper','Betina','Andres']),
   'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.
 ↪65])
}

#Create a DataFrame
df = pd.DataFrame(d)
print(df)
print("------------------**sum**---------------------")
print(df.sum())
print("------------------**mean**---------------------")
print(df.mean())
print("------------------**mode**---------------------")
print(df.mode())
print("------------------**median**---------------------")
print(df.median())
print("------------------**std**---------------------")
print(df.std())
print("------------------**describe**---------------------")
print(df.describe())
print("------------------**max**---------------------")
print(df.max())
print("------------------**min**---------------------")
print(df.min())
```

```
      Name  Age  Rating
0      Tom   25    4.23
1    James   26    3.24
2    Ricky   25    3.98
3      Vin   23    2.56
4    Steve   30    3.20
5    Smith   29    4.60
6     Jack   23    3.80
7      Lee   34    3.78
8    David   40    2.98
9   Gasper   30    4.80
10  Betina   51    4.10
11  Andres   46    3.65
------------------**sum**---------------------
Name      TomJamesRickyVinSteveSmithJackLeeDavidGasperBe…
Age                                                    382
Rating                                               44.92
dtype: object
------------------**mean**---------------------
Age       31.833333
Rating     3.743333
dtype: float64
------------------**mode**---------------------
      Name   Age  Rating
0   Andres  23.0    2.56
```

```
1   Betina  25.0    2.98
2    David  30.0    3.20
3   Gasper   NaN    3.24
4     Jack   NaN    3.65
5    James   NaN    3.78
6      Lee   NaN    3.80
7    Ricky   NaN    3.98
8    Smith   NaN    4.10
9    Steve   NaN    4.23
10     Tom   NaN    4.60
11     Vin   NaN    4.80
------------------**median**---------------------
Age       29.50
Rating     3.79
dtype: float64
------------------**std**---------------------
Age       9.232682
Rating    0.661628
dtype: float64
------------------**describe**---------------------
             Age     Rating
count  12.000000  12.000000
mean   31.833333   3.743333
std     9.232682   0.661628
min    23.000000   2.560000
25%    25.000000   3.230000
50%    29.500000   3.790000
75%    35.500000   4.132500
max    51.000000   4.800000
------------------**max**---------------------
Name      Vin
Age        51
Rating    4.8
dtype: object
------------------**min**---------------------
Name      Andres
Age           23
Rating      2.56
dtype: object

C:\Users\ABDULLAH KHAN\AppData\Local\Temp\ipykernel_18452\2901324558.py:16:
FutureWarning: The default value of numeric_only in DataFrame.mean is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  print(df.mean())
C:\Users\ABDULLAH KHAN\AppData\Local\Temp\ipykernel_18452\2901324558.py:20:
FutureWarning: The default value of numeric_only in DataFrame.median is
```

```
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  print(df.median())
C:\Users\ABDULLAH KHAN\AppData\Local\Temp\ipykernel_18452\2901324558.py:22:
FutureWarning: The default value of numeric_only in DataFrame.std is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  print(df.std())
```

[ ]: