mlrbxhrvp

March 19, 2025

[6]:

```
import pandas as pd
```

```
my name is rahul naidu
```

[9]:

```
lst = [[1,2,3,4]]
print(lst)
```

```
[[1, 2, 3, 4], [1, 2, 3, 4]]
```

[10]:

```
series = pd.Series(lst)
print(series)
print(type(series))
```

```
0    [1, 2, 3, 4]
1    [1, 2, 3, 4]
dtype: object
<class 'pandas.core.series.Series'>
```

[11]:

```
empty = pd.Series([])
empty
```

[11]:

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

[15]:

```
a=pd.Series(['p','q','r','s','t'],index = [10,11,12,13,14],name='alphabets')
a
```

[15]:

```
10    p
11    q
12    r
13    s
14    t
Name: alphabets, dtype: object
```

[17]:

```
scalar_series = pd.Series(0.5)
scalar_series
```

[17]:

```
0    0.5
dtype: float64
```

[18]:

```
scalar_series = pd.Series(0.5,index = [1,2,3])
scalar_series
```

[18]:

```
1    0.5
2    0.5
3    0.5
dtype: float64
```

[20]:

```
dict_series = pd.Series({'p':1,'q':2,'r':3,'s':4})
dict_series
```

[20]:

```
p    1
q    2
r    3
s    4
dtype: int64
```

[21]:

```
dict_series[0]
```

```
C:\Users\Rahul\AppData\Local\Temp\ipykernel_10620\1198263229.py:1:
FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a
future version, integer keys will always be treated as labels (consistent with
DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  dict_series[0]
```

[21]:

```
np.int64(1)
```

[23]:

```
dict_series[0:3]
```

[23]:

```
p    1
q    2
r    3
dtype: int64
```

[24]:

```
max(dict_series)
```

[24]:

```
4
```

[25]:

```
dict_series = pd.Series({'p':[1,5,6],'q':[2,6,7],'r':[3,7,8]})
dict_series
```

[25]:

```
p    [1, 5, 6]
q    [2, 6, 7]
r    [3, 7, 8]
dtype: object
```

[ ]:

```
dict_series[0][1]
```

[28]:

```
df = pd.DataFrame()
print(df)
```

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

[29]:

```
lst = [1,2,3,4,5]
df=pd.DataFrame(lst)
df
```

[29]:

```
   0
0  1
1  2
2  3
3  4
4  5
```

[31]:

```
lst = [[1,2,3,4,5],[11,12,13,14,15]]
df = pd.DataFrame(lst)
df
```

[31]:

```
    0   1   2   3   4
0   1   2   3   4   5
1  11  12  13  14  15
```

[32]:

```
a=[{'a':5,'b':7,'c':9,'d':2},
   {'a':3,'b':4,'c':7,'d':45}]
df=pd.DataFrame(a)
#Dictionary keys represent the column names
df
```

[32]:

```
   a  b  c   d
0  5  7  9   2
1  3  4  7  45
```

[33]:

```
b={'RollNo':pd.Series([1,2,3,4,5]),
   'Maths':pd.Series([11,1,21,3,44]),
   'Science':pd.Series([21,22,23,24,25])}
df=pd.DataFrame(b)
df
```

[33]:

```
   RollNo  Maths  Science
0       1     11       21
1       2      1       22
2       3     21       23
3       4      3       24
```

```
4        5      44        25
```

[34]:

```
df = pd.read_csv("./Salary_data.csv")
# csv = comma separated values
df
```

[34]:

```
    YearsExperience    Salary
0               1.1   39343.0
1               1.3   46205.0
2               1.5   37731.0
3               2.0   43525.0
4               2.2   39891.0
5               2.9   56642.0
6               3.0   60150.0
7               3.2   54445.0
8               3.2   64445.0
9               3.7   57189.0
10              3.9   63218.0
11              4.0   55794.0
12              4.0   56957.0
13              4.1   57081.0
14              4.5   61111.0
15              4.9   67938.0
16              5.1   66029.0
17              5.3   83088.0
18              5.9   81363.0
19              6.0   93940.0
20              6.8   91738.0
21              7.1   98273.0
22              7.9  101302.0
23              8.2  113812.0
24              8.7  109431.0
25              9.0  105582.0
26              9.5  116969.0
27              9.6  112635.0
28             10.3  122391.0
29             10.5  121872.0
```

[35]:

```
df.columns
```

[35]:

```
Index(['YearsExperience', 'Salary'], dtype='object')
```

[36]:

```
df.shape
```

[36]:

```
(30, 2)
```

[37]:

```
df.size
```

```
[37]: 60
```

```
[38]: df.head()
```

```
[38]:    YearsExperience   Salary
      0              1.1  39343.0
      1              1.3  46205.0
      2              1.5  37731.0
      3              2.0  43525.0
      4              2.2  39891.0
```

```
[39]: df.head(10)
```

```
[39]:    YearsExperience   Salary
      0              1.1  39343.0
      1              1.3  46205.0
      2              1.5  37731.0
      3              2.0  43525.0
      4              2.2  39891.0
      5              2.9  56642.0
      6              3.0  60150.0
      7              3.2  54445.0
      8              3.2  64445.0
      9              3.7  57189.0
```

```
[40]: df.head(-5)
```

```
[40]:     YearsExperience    Salary
      0               1.1   39343.0
      1               1.3   46205.0
      2               1.5   37731.0
      3               2.0   43525.0
      4               2.2   39891.0
      5               2.9   56642.0
      6               3.0   60150.0
      7               3.2   54445.0
      8               3.2   64445.0
      9               3.7   57189.0
      10              3.9   63218.0
      11              4.0   55794.0
      12              4.0   56957.0
      13              4.1   57081.0
      14              4.5   61111.0
      15              4.9   67938.0
      16              5.1   66029.0
      17              5.3   83088.0
      18              5.9   81363.0
```

```
19                6.0   93940.0
20                6.8   91738.0
21                7.1   98273.0
22                7.9  101302.0
23                8.2  113812.0
24                8.7  109431.0
```

```
[41]: df.tail()
```

```
[41]:     YearsExperience    Salary
      25              9.0  105582.0
      26              9.5  116969.0
      27              9.6  112635.0
      28             10.3  122391.0
      29             10.5  121872.0
```

```
[42]: df.tail(-5)
```

```
[42]:     YearsExperience    Salary
      5               2.9   56642.0
      6               3.0   60150.0
      7               3.2   54445.0
      8               3.2   64445.0
      9               3.7   57189.0
      10              3.9   63218.0
      11              4.0   55794.0
      12              4.0   56957.0
      13              4.1   57081.0
      14              4.5   61111.0
      15              4.9   67938.0
      16              5.1   66029.0
      17              5.3   83088.0
      18              5.9   81363.0
      19              6.0   93940.0
      20              6.8   91738.0
      21              7.1   98273.0
      22              7.9  101302.0
      23              8.2  113812.0
      24              8.7  109431.0
      25              9.0  105582.0
      26              9.5  116969.0
      27              9.6  112635.0
      28             10.3  122391.0
      29             10.5  121872.0
```

```
[43]: df.describe()
```

[43]:

```
       YearsExperience         Salary
count        30.000000      30.000000
mean          5.313333   76003.000000
std           2.837888   27414.429785
min           1.100000   37731.000000
25%           3.200000   56720.750000
50%           4.700000   65237.000000
75%           7.700000  100544.750000
max          10.500000  122391.000000
```

[45]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 612.0 bytes
```

[46]:

```
df2 = pd.read_csv("./rating_final.csv")
df2.head()
```

[46]:

```
  userID  placeID  rating  food_rating  service_rating
0  U1077   135085       2            2               2
1  U1077   135038       2            2               1
2  U1077   132825       2            2               2
3  U1077   135060       1            2               2
4  U1068   135104       1            1               2
```

[48]:

```
df2.shape
```

[48]:

```
(1161, 5)
```

[49]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1161 entries, 0 to 1160
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   userID          1161 non-null   object
 1   placeID         1161 non-null   int64
 2   rating          1161 non-null   int64
 3   food_rating     1161 non-null   int64
 4   service_rating  1161 non-null   int64
```

```
dtypes: int64(4), object(1)
memory usage: 45.5+ KB
```

[50]:

```
df2.describe()
```

[50]:

```
             placeID       rating  food_rating  service_rating
count    1161.000000  1161.000000  1161.000000     1161.000000
mean   134192.041344     1.199828     1.215332        1.090439
std      1100.916275     0.773282     0.792294        0.790844
min    132560.000000     0.000000     0.000000        0.000000
25%    132856.000000     1.000000     1.000000        0.000000
50%    135030.000000     1.000000     1.000000        1.000000
75%    135059.000000     2.000000     2.000000        2.000000
max    135109.000000     2.000000     2.000000        2.000000
```

[ ]: