Pandas

```
import pandas as pd
```

```
headers = ['name', 'age', 'weight']
values = ['salma', 21, 75]
myser = pd.Series(data = values, index = headers)
```

```
myser
```

```
name      salma
age          21
weight       75
dtype: object
```

```
myser['name']
```

```
'salma'
```

```
my_sec_ser = pd.Series(values)
my_sec_ser
```

```
0    salma
1       21
2       75
dtype: object
```

```
pd.Series([str, print])
```

```
0                <class 'str'>
1    <built-in function print>
dtype: object
```

```
sem_1 = pd.Series(data = [25, 46], index = ['Math', 'Physics'])
sem_2 = pd.Series(data = [29, 37, 27], index = ['Algebra', 'Cs', 'Physics'])
```

```
all_sem = sem_1 + sem_2
all_sem
```

```
Algebra      NaN
Cs           NaN
Math         NaN
Physics     73.0
dtype: float64
```

```
df1 = pd.DataFrame (0, index = ['X', 'Y', 'Z'], columns = ['C1', 'C2'])
df1
```

	C1	C2
X	0	0
Y	0	0
Z	0	0

Next steps: Generate code with `df1` View recommended plots

```
df1 = pd.DataFrame (index = ['X', 'Y', 'Z'], columns = ['C1', 'C2'])
df1
```

	C1	C2
X	NaN	NaN
Y	NaN	NaN
Z	NaN	NaN

```
df1.reset_index()
```

	index	C1	C2
0	X	NaN	NaN
1	Y	NaN	NaN
2	Z	NaN	NaN

```
import numpy as np
from numpy.random import randn
np.random.seed(100212251)
```

```
dff = pd.DataFrame (randn(3,3), index = ['X', 'Y', 'Z'], columns = ['C1', 'C2', 'C3'])
```

```
dff
```

	C1	C2	C3
X	-1.834470	-0.751930	-0.193889
Y	-0.189548	-1.563833	1.359748
Z	1.127761	-0.797036	0.792673

Next steps: Generate code with dff | View recommended plots

```
dff['C1']
```

```
X    1.056924
Y   -1.033658
Z   -0.712264
Name: C1, dtype: float64
```

```
dff[['C1', 'C3']]
```

	C1	C3
X	1.056924	-1.821241
Y	-1.033658	1.574977
Z	-0.712264	2.051359

```
dff['C4'] = dff['C1'] + dff['C2']
dff
```

	C1	C2	C3	C4
X	1.056924	1.254387	-1.821241	2.311311
Y	-1.033658	-1.620970	1.574977	-2.654628
Z	-0.712264	-0.491356	2.051359	-1.203619

Next steps: Generate code with `dff` View recommended plots

```
dff.loc['Y']
```

```
C1   -1.033658
C2   -1.620970
C3    1.574977
C4   -2.654628
Name: Y, dtype: float64
```

```
dff.iloc[2]
```

```
C1   -0.712264
C2   -0.491356
C3    2.051359
C4   -1.203619
Name: Z, dtype: float64
```

```
dff.loc[['X', 'Z']]
```

	C1	C2	C3	C4
X	-1.834470	-0.751930	-0.193889	-2.586400
Z	1.127761	-0.797036	0.792673	0.330726

```
dff.loc[['X', 'Z'], ['C2', 'C4']]
```

	C2	C4
X	-0.751930	-2.586400
Z	-0.797036	0.330726

```
dff.loc[['X'], ['C3']]
```

	C3
X	-0.193889

```
print(dff.describe())
```

```
             C1        C2        C3        C4
count  3.000000  3.000000  3.000000  3.000000
mean  -0.298752 -1.037600  0.652844 -1.336352
std    1.484132  0.456289  0.786200  1.502611
min   -1.834470 -1.563833 -0.193889 -2.586400
25%   -1.012009 -1.180434  0.299392 -2.169891
50%   -0.189548 -0.797036  0.792673 -1.753381
75%    0.469107 -0.774483  1.076211 -0.711328
max    1.127761 -0.751930  1.359748  0.330726
```

```
dff.drop('C4', axis = 1)
```

	C1	C2	C3
X	-1.834470	-0.751930	-0.193889
Y	-0.189548	-1.563833	1.359748
Z	1.127761	-0.797036	0.792673

```
dff
```

	C1	C2	C3	C4
X	-1.834470	-0.751930	-0.193889	-2.586400
Y	-0.189548	-1.563833	1.359748	-1.753381
Z	1.127761	-0.797036	0.792673	0.330726

Next steps: Generate code with dff View recommended plots

```
dff.drop('Y', axis = 0, inplace = True)
dff.drop('C4', axis = 1, inplace = True)
dff.drop('X', axis = 0, inplace = True)
```

```
dff
```

	C1	C2	C3
Z	1.127761	-0.797036	0.792673

```
dff.head(n = 2)
```

	C1	C2	C3	C4
X	1.056924	1.254387	-1.821241	2.311311
Y	-1.033658	-1.620970	1.574977	-2.654628
Z	-0.712264	-0.491356	2.051359	-1.203619

Next steps: Generate code with `dff` View recommended plots

```
dff<=0
```

	C1	C2	C3	C4
X	False	False	True	False
Y	True	True	False	True
Z	True	True	False	True

```
dff_2 = dff[dff<=0]
dff_2
```

	C1	C2	C3	C4
X	NaN	NaN	-1.821241	NaN
Y	-1.033658	-1.620970	NaN	-2.654628
Z	-0.712264	-0.491356	NaN	-1.203619

Next steps: Generate code with `dff_2` View recommended plots

```
dff_2= dff[(dff['C3'] <= 0) & (dff['C1'] >= 0.5)].loc['X']
dff_2
```

```
C1    1.056924
C2    1.254387
C3   -1.821241
C4    2.311311
Name: X, dtype: float64
```

```
df = pd.DataFrame([
    [15, 32, 101],
    [np.nan, 46, 103],
    [np.nan, 52, np.nan]
], index = ['A', 'B', 'C'], columns = ['C1', 'C2', 'C3'])
```

```
df
```

	C1	C2	C3
A	15.0	32	101.0
B	NaN	46	103.0
C	NaN	52	NaN

Next steps: Generate code with df   View recommended plots

```
df.dropna()
```

	C1	C2	C3
A	15.0	32	101.0

```
df
```

	C1	C2	C3
A	15.0	32	101.0
B	NaN	46	103.0
C	NaN	52	NaN

Next steps: Generate code with df   View recommended plots

```
df.dropna(inplace = True)
```

```
df
```

	C1	C2	C3
A	15.0	32	101.0

```
df.dropna(axis = 0)
```

	C1	C2	C3
A	15.0	32	101.0

```
df.dropna(axis = 1)
```

	C2
A	32
B	46
C	52

```
df
```

	C1	C2	C3
A	15.0	32	101.0
B	NaN	46	103.0
C	NaN	52	NaN

Next steps: Generate code with df View recommended plots

```
df.dropna(axis = 1, thresh = 2)
```

	C2	C3
A	32	101.0
B	46	103.0
C	52	NaN

```
df.fillna('AAA')
```

	C1	C2	C3
A	15.0	32	101.0
B	AAA	46	103.0
C	AAA	52	AAA

```
df.fillna(df['C2'].sum())
```

	C1	C2	C3
A	15.0	32	101.0
B	130.0	46	103.0
C	130.0	52	130.0

```
df.fillna(df['C2'].mean())
```

	C1	C2	C3
A	15.000000	32	101.000000
B	43.333333	46	103.000000
C	43.333333	52	43.333333

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.