



Python for Data Science

DataFrames

Pandas - DataFrames

```
In [4]: import numpy as np  
import pandas as pd
```

```
In [5]: from numpy.random import randn
```

```
In [6]: np.random.seed(101)
```

```
In [ ]:
```

Pandas - DataFrames

```
In [4]: import numpy as np  
import pandas as pd
```

```
In [5]: from numpy.random import randn
```

```
In [6]: np.random.seed(101)
```



To see the same data elsewhere

```
In [ ]: df = pd.DataFrame|
```

Init signature: `pd.DataFrame(data=None, index=None, columns=None, dtype=None, copy=False)`

Docstring:

Two-dimensional size-mutable, potentially heterogeneous tabular data

```
In [5]: from numpy.random import randn
```

```
In [6]: np.random.seed(101)
```

```
In [7]: df = pd.DataFrame(randn(5,4),['A','B','C','D','E'],['W','X','Y','Z'])
```

```
In [8]: df
```

Out[8]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

	W	A	T	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [9]: df['W']

Out[9]: A 2.706850
B 0.651118
C -2.018168
D 0.188695
E 0.190794
Name: W, dtype: float64

In []:

```
In [9]: df['W']
```

```
Out[9]: A    2.706850  
       B    0.651118  
       C   -2.018168  
       D    0.188695  
       E    0.190794  
       Name: W, dtype: float64
```

```
In [10]: type(df['W'])
```

```
Out[10]: pandas.core.series.Series
```

```
In [ ]:
```



```
Out[9]: A    2.788888  
        B    0.651118  
        C   -2.018168  
        D    0.188695  
        E    0.190794  
        Name: W, dtype: float64
```

```
In [10]: type(df['W'])
```

```
Out[10]: pandas.core.series.Series
```

```
In [11]: type(df)
```

```
Out[11]: pandas.core.frame.DataFrame
```

```
In [ ]:
```

```
In [11]: type(df)
```

```
Out[11]: pandas.core.frame.DataFrame
```

```
In [14]: df[['W', 'Z']]
```

```
Out[14]:
```

	W	Z
A	2.706850	0.503826
B	0.651118	0.605965
C	-2.018168	-0.589001
D	0.188695	0.955057
E	0.190794	0.683509

```
In [ ]:
```


E	0.190794	0.683509
---	----------	----------

```
In [17]: df['new'] = df['W'] + df['Y']
```

```
In [18]: df
```

```
Out[18]:
```

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

```
In [ ]: |
```

```
In [19]: df.drop('new')
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-19-6ebc454a305e> in <module>()  
----> 1 df.drop('new')  
  
C:\Users\Marcial\Anaconda\envs\py35\lib\site-packages\pandas\core\generic.py in  
drop(self, labels, axis, level, inplace, errors)  
    1875         new_axis = axis.drop(labels, level=level,  
errors=errors)  
    1876     else:  
-> 1877         new_axis = axis.drop(labels, errors=errors)  
    1878         dropped = self.reindex(**{axis_name: new_axis})  
    1879     try:  
  
C:\Users\Marcial\Anaconda\envs\py35\lib\site-packages\pandas\indexes\base.py in  
drop(self, labels, errors)  
    3049         if errors != 'ignore':  
    3050             raise ValueError('labels %s not contained in axis' %  
-> 3051                             labels[mask])
```

	W	A	T	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

In [19]: `df.drop('new')`

Axis = 0, for the drop method refers to the index value; for row it is 0 and for column it is 1.

Signature: `df.drop(labels, axis=0, level=None, inplace=False, errors='raise')`

Docstring:

Return new object with labels in requested axis removed.

```
C:\Users\Marcial\Anaconda\envs\py35\lib\site-packages\pandas\core\generic.py in
drop(self, labels, axis, level, inplace, errors)
    1875         new_axis = axis.drop(labels, level=level,
errors=errors)
```


A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

In [20]: `df.drop('new',axis=1)`

Out[20]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [23]: df

Out[23]:

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

In []:

	0.190794	1.978757	2.605967	0.683509	2.790702
--	----------	----------	----------	----------	----------

In [22]: `df.drop('new',axis=1,)`

Out[22]:

Signature: `df.drop(labels, axis=0, level=None, inplace=False, errors='raise')`

Docstring:

Return new object with labels in requested axis removed.

C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [23]: `df`

Out[23]:

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959

A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

In [22]: `df.drop('new',axis=1,inplace=True)`

Out[22]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [26]: `df.drop('E')`

Out[26]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

In []:

B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

Axis=0 for Rows and
Axis=1 for Columns

In [27]: `df.drop('E',axis=0)`

Out[27]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

In []:

Out[27]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

In [28]: df.shape

Out[28]: (5, 4)

In []:

In [28]: df.shape

Out[28]: (5, 4)

This is Columns = 1

This is Rows = 0

In [29]: df

Out[29]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

In [28]: df.shape

Out[28]: (5, 4)

In [31]: df[['Z', 'X']]

Out[31]:

	Z	X
A	0.503826	0.628133
B	0.605965	-0.319318
C	-0.589001	0.740122
D	0.955057	-0.758872
E	0.683509	1.978757

In []:

In [32]: `# ROWS`

In [33]: `df`

Out[33]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []:

Out[33]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In []: df.loc|

Type: property
String form: <property object at 0x0000015E8B161EF8>
Docstring:
Purely label-location based indexer for selection by label.

Out[33]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [34]: df.loc['A']

Out[34]: W 2.706850
X 0.628133
Y 0.907969
Z 0.503826
Name: A, dtype: float64

In []:

E	0.190794	1.978757	2.605967	0.683509
---	----------	----------	----------	----------

In [35]: `df.loc['C']`

Out[35]:

W	-2.018168
X	0.740122
Y	0.528813
Z	-0.589001

Name: C, dtype: float64

In [36]: `df.iloc[2]`

Out[36]:

W	-2.018168
X	0.740122
Y	0.528813
Z	-0.589001

Name: C, dtype: float64

In []:

```
Y      0.528813  
Z     -0.589001  
Name: C, dtype: float64
```

```
In [37]: df.loc['B', 'Y']
```

```
Out[37]: -0.84807698340363147
```

```
In [38]: df
```

```
Out[38]:
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

name: C, dtype: float64

In [37]: df.loc['B', 'Y']

Out[37]: -0.84807698340363147

In [39]: df.loc[['A', 'B'], ['W', 'Y']]

Out[39]:

	W	Y
A	2.706850	0.907969
B	0.651118	-0.848077

In []: