Correlation&Covariance | Untitled | Untitled | Untitled | Untitled | Untitled | Untitled | Untitled | Untitled

Zeppelin

Correlation&Covaria...

default

FINISHED

```
%python
from pandas import Series, DataFrame
import numpy as np, pandas as pd
df = DataFrame([[1.4,np.nan],[7.1,-4.5],
               [np.nan,np.nan],[0.75,-1.3]],
               index=['a','b','c','d'],
               columns=['one','two'])
df
df.sum()
df.sum(axis=1)
df.mean(axis=1,skipna=False)
```

```
      one  two
a  1.40  NaN
b  7.10 -4.5
c   NaN  NaN
d  0.75 -1.3
one    9.25
two   -5.80
dtype: float64
a    1.40
b    2.60
c     NaN
d   -0.55
dtype: float64
a      NaN
b    1.300
c      NaN
d   -0.275
dtype: float64
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:24:22 PM. (outdated)

FINISHED

```
%python
df.idxmax()
df.describe()
```

```
one    b
two    d
dtype: object
            one       two
count  3.000000  2.000000
mean   3.083333 -2.900000
std    3.493685  2.262742
min    0.750000 -4.500000
25%    1.075000 -3.700000
50%    1.400000 -2.900000
75%    4.250000 -2.100000
max    7.100000 -1.300000
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:24:14 PM.

Zeppelin

Correlation&Covaria...

default

FINISHED

```
%python
obj = Series(['a','a','b','c'] * 4)
obj
obj.describe()
```

```
0     a
1     a
2     b
3     c
4     a
5     a
6     b
7     c
8     a
9     a
10    b
11    c
12    a
13    a
14    b
15    c
dtype: object
count      16
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:24:17 PM.

FINISHED

```
%pyspark
from pandas_datareader import data as wb
all_data = {}
for ticker in ['AAPL','IBM','MSFT','GOOG']:
  all_data[ticker] = wb.get_data_yahoo(ticker)
price = DataFrame({tic: data['Adj Close']
    for tic, data in all_data.items()})
volume = DataFrame({tic: data['Volume']
    for tic, data in all_data.items()})
returns = price.pct_change()
returns.tail()
```

```
                AAPL      GOOG       IBM      MSFT
Date
2017-02-15  0.003629 -0.001792  0.008605 -0.000619
2017-02-16 -0.001181  0.006325 -0.001376 -0.000155
2017-02-17  0.002734  0.004744 -0.004189  0.001550
2017-02-21  0.007221  0.004335 -0.002269 -0.002012
2017-02-22  0.002999 -0.001082  0.004937 -0.002016
```

Took 1 sec. Last updated by anonymous at February 23 2017, 7:21:13 PM.

FINISHED

```
%pyspark
returns.MSFT.corr(returns.IBM)
```

```
0.49515377802280924
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:21:43 PM.

FINISHED

```
%pyspark
returns.MSFT.cov(returns.IBM)
```

```
8.5977652563835441e-05
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:22:03 PM.

FINISHED

```
%pyspark
returns.corr()
```

```
          AAPL      GOOG       IBM      MSFT
AAPL  1.000000  0.409541  0.381549  0.388972
GOOG  0.409541  1.000000  0.402872  0.470820
IBM   0.381549  0.402872  1.000000  0.495154
MSFT  0.388972  0.470820  0.495154  1.000000
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:22:18 PM.

FINISHED

```
%pyspark
returns.cov()
```

```
          AAPL      GOOG       IBM      MSFT
AAPL  0.000270  0.000105  0.000075  0.000093
GOOG  0.000105  0.000244  0.000075  0.000107
IBM   0.000075  0.000075  0.000144  0.000086
MSFT  0.000093  0.000107  0.000086  0.000210
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:22:35 PM.

FINISHED

```
%pyspark
returns.corrwith(returns.IBM)
```

```
AAPL    0.381549
GOOG    0.402872
IBM     1.000000
MSFT    0.495154
dtype: float64
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:22:57 PM.

FINISHED

```
%pyspark
returns.corrwith(volume)
```

```
AAPL   -0.074323
GOOG   -0.009670
IBM    -0.194432
MSFT   -0.091017
dtype: float64
```

Took 0 sec. Last updated by anonymous at February 23 2017, 7:23:08 PM.

READY