Mastering Python

الدرس #8

Pandas and Data Analysis تحليل البيانات

By:

Hussam Hourani

V1.0 - NOV 2019

Agenda

- What is Panada
- Panada’s Data Structure
- Panda Series
- DataFrame
- Applying functions on data
- Export to Excel and csv formats
- Reading from files
- Plotting
- DataFrame – adding/deleting Columns
- DataFrame – Group By

What is Panada

Panda is powerful Python data Analysis Library

The Library provids high-performance, easy-to-use data structures and data analysis and modeling tools that is fast and flexible

It uses “relational” or “labeled” data both easy and intuitive.

It is rounding up the capabilities of Numpy, Scipy and Matplotlab

The word pandas is an acronym which is derived from "Python and data analysis" and "panel data".

https://pandas.pydata.org/

Panada’s Data Structure

Data Structures

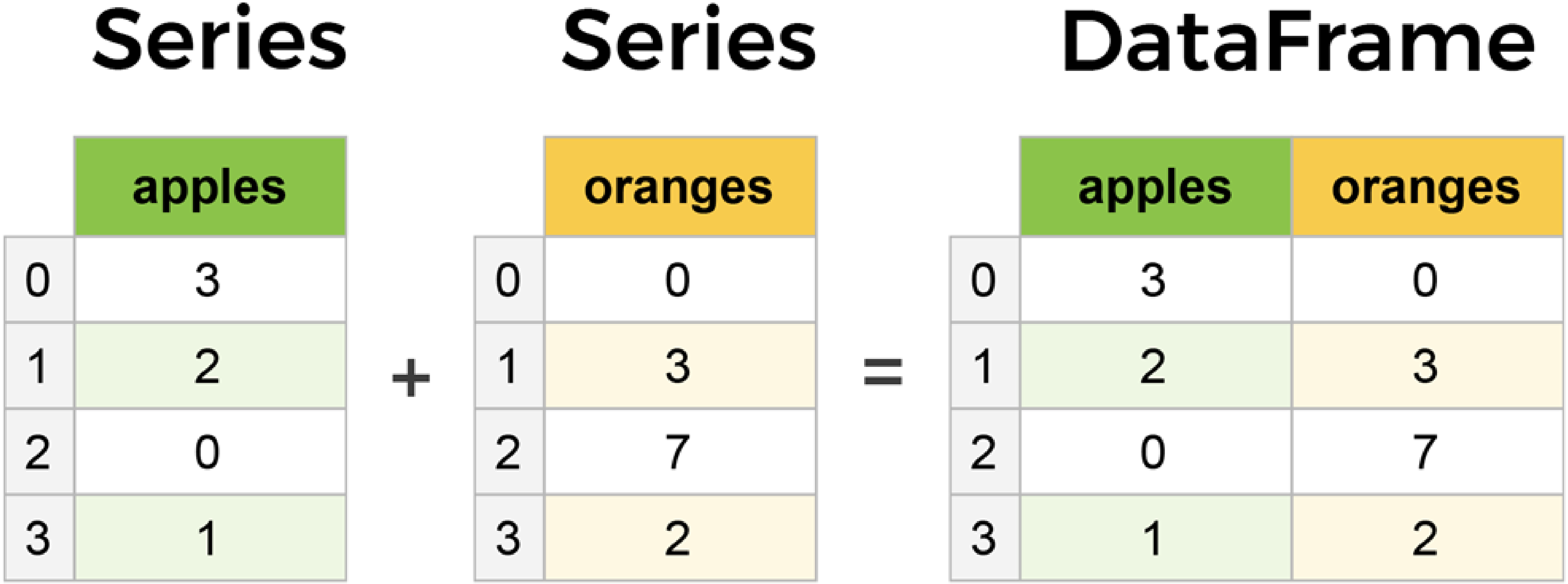
Series

- A Series is a one-dimensional labelled array-like object

DataFrame

- DataFrame is based on spreadsheets concepts

Panada’s Data Structure



Series

	apples
0	3
1	2
2	0
3	1

+

Series

	oranges
0	0
1	3
2	7
3	2

=

DataFrame

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

Pandas Series

```
import pandas as pd

data = [100, 120, 140, 180, 200, 210, 214]

s = pd.Series(data)

print(s)
```

Output

```
0     100
1     120
2     140
3     180
4     200
5     210
6     214
```

```
import pandas as pd

data = [100, 120, 140, 180, 200, 210, 214]

s = pd.Series(data,
index=['a', 'b', 'c', 'd', 'e', 'f', 'g'])

print(s)
```

Output

```
a     100
b     120
c     140
d     180
e     200
f     210
g     214
```

Pandas Series

```
import pandas as pd

data = [100, 120, 140, 180, 200, 210, 214]

s = pd.Series(data, index=['a', 'b', 'c',
'd', 'e', 'f', 'g'])

print(s)
print(s.index)
print(s.values)

print(s[0])
print(s['b'])
print(s['f'],s['g'])

print ( sum(s))
```

Output

print(s)

```
a     100
b     120
c     140
d     180
e     200
f     210
g     214
```

print(s.index)

```
Index(['a', 'b', 'c', 'd', 'e', 'f',
'g'], dtype='object')
```

print(s.values)

```
[100 120 140 180 200 210 214]
```

print(s[0])

```
100
```

print(s['b'])

```
120
```

print(s['f'],s['g'])

```
210 214
```

print (sum(s))

```
1164
```

Pandas Series

```
import pandas as pd

data = [100, 120, 140, 180, 200, 210, 214]

s = pd.Series(data)

print(s)

print( s.head())
print( s.head(2))

print( s.tail())
print( s.tail(2))
```

Output

print(s)

```
0     100
1     120
2     140
3     180
4     200
5     210
6     214
```

print(s.head())

```
0     100
1     120
2     140
3     180
4     200
```

print(s.head(2))

```
0     100
1     120
```

print(s.tail())

```
2     140
3     180
4     200
5     210
6     214
```

print(s.tail(2))

```
5     210
6     214
```

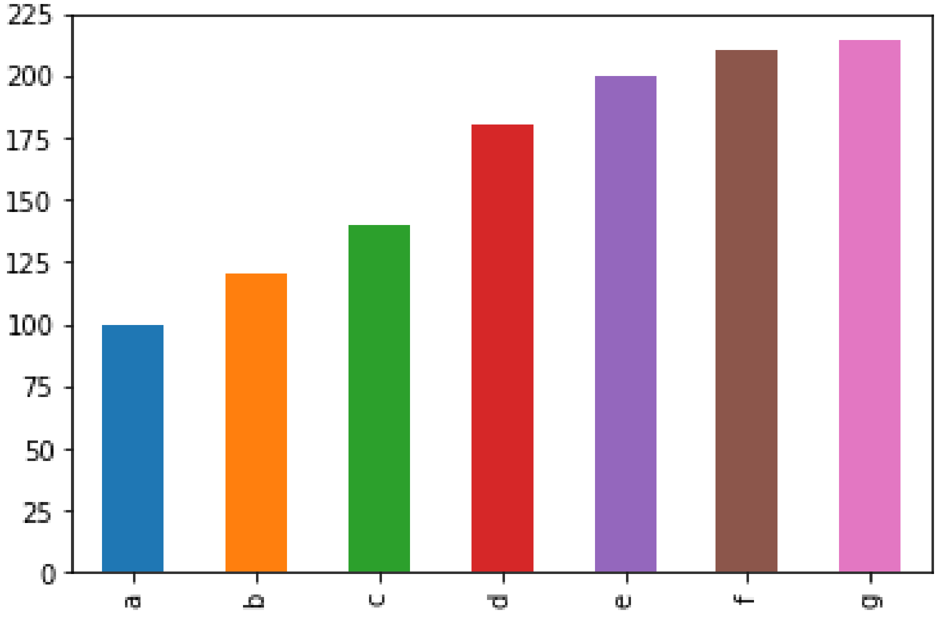
Pandas Series

```
import pandas as pd

data = [100, 120, 140, 180, 200, 210, 214]

s = pd.Series(data)

print( s.describe())

s.plot(kind="bar")
```

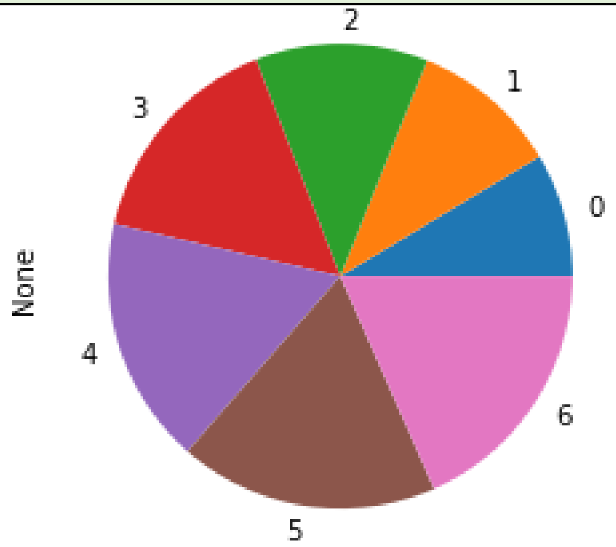
Output

```
count      7.000000
mean     166.285714
std       46.078608
min      100.000000
25%      130.000000
50%      180.000000
75%      205.000000
max      214.000000
```

Output



s.plot(kind="pie")

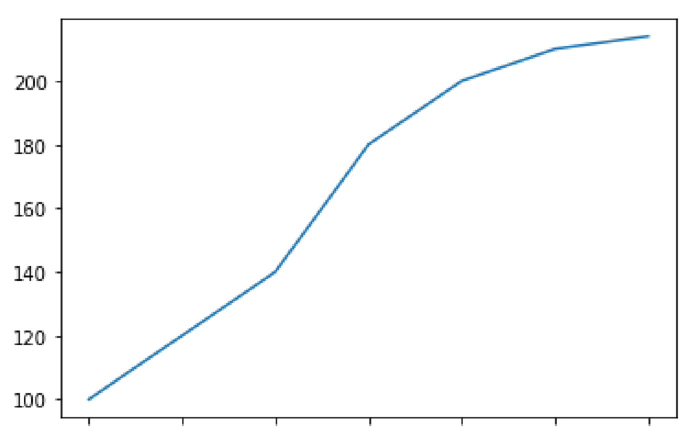


Pandas Series

```
import pandas as pd

data = [100, 120, 140, 180, 200, 210,
214]

s = pd.Series(data)

print(s)

s.index = ["a", "b", "c", "d",
"e","f","g"]

print(s)

print("mean :",s.mean())
print("std :",s.std())

print( s.agg(['sum','max']))


s.plot()
```

Output

```
0     100
1     120
2     140
3     180
4     200
5     210
6     214
dtype: int64

a     100
b     120
c     140
d     180
e     200
f     210
g     214
dtype: int64

mean : 166.28571428571428
std : 46.07860778320126

sum    1164
max     214
dtype: int64
```



Pandas Series

```
import pandas as pd
s1 = pd.Series([10, 20, 30, 40, 50, 60], index=pd.date_range('20130102', periods=6))
print(s1)
print (s1[0])
print (s1[1:3])
print (s1>20)
s1[2]=999
print (s1[2])
```

Output

print(s1)

```
2013-01-02    10
2013-01-03    20
2013-01-04    30
2013-01-05    40
2013-01-06    50
2013-01-07    60
```

print (s1[0])

```
10
```

print (s1[1:3])

```
2013-01-03    20
2013-01-04    30
```

print (s1>20)

```
2013-01-02    False
2013-01-03    False
2013-01-04    True
2013-01-05    True
2013-01-06    True
2013-01-07    True
```

print (s1[2])

```
999
```

Pandas Series

```
import pandas as pd

s = pd.Series([100, 120, 140, 180, 200, 210, 214])
print(s)
s2=s.apply( lambda x: x if x > 180 else x*10 )
print (s2)

s3= s2[s2>1000]
print (s3)
```

Output

print(s)

```
0     100
1     120
2     140
3     180
4     200
5     210
6     214
```

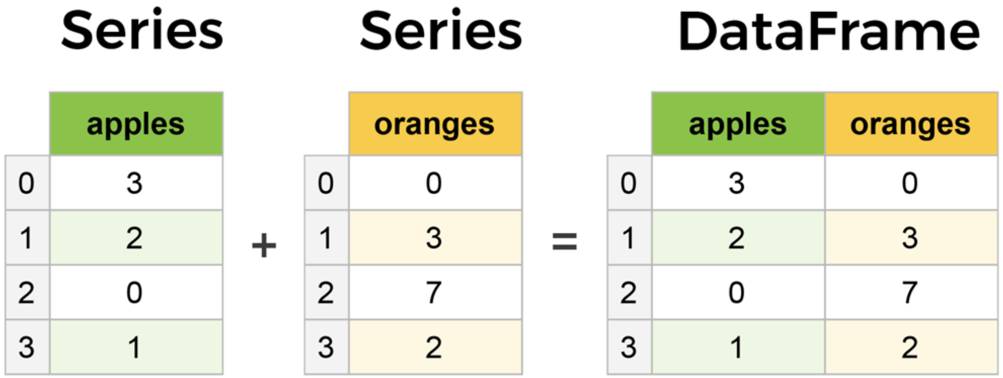
print (s2)

```
0     1000
1     1200
2     1400
3     1800
4      200
5      210
6      214
```

print (s3)

```
1     1200
2     1400
3     1800
```

Pandas DataFrame



Series

	apples
0	3
1	2
2	0
3	1

+

Series

	oranges
0	0
1	3
2	7
3	2

=

DataFrame

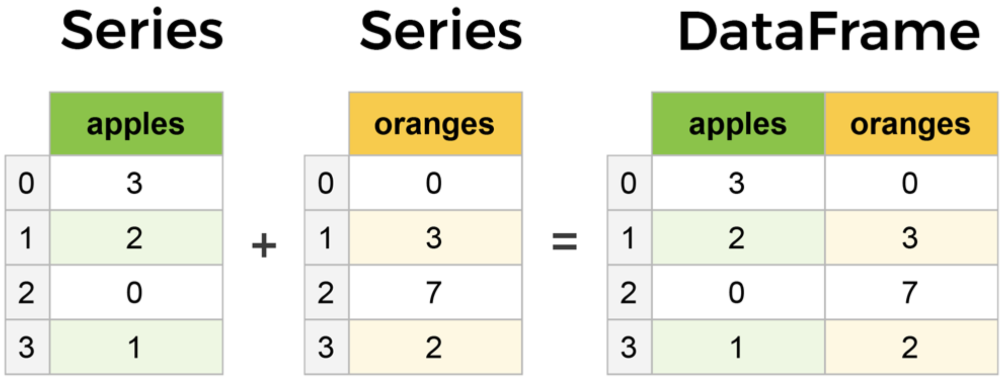
	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

```
import pandas as pd

data = {
    'apples': [3, 2, 0, 1],
    'oranges': [0, 3, 7, 2]
}
purchases = pd.DataFrame(data)
print ( purchases )
print ( purchases.describe() )
```

Output

```
   apples  oranges
0       3        0
1       2        3
2       0        7
3       1        2


         apples   oranges
count  4.000000   4.00000
mean   1.500000   3.00000
std    1.290994   2.94392
min    0.000000   0.00000
25%    0.750000   1.50000
50%    1.500000   2.50000
75%    2.250000   4.00000
max    3.000000   7.00000
```

Pandas DataFrame



Series

	apples
0	3
1	2
2	0
3	1

+

Series

	oranges
0	0
1	3
2	7
3	2

=

DataFrame

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

```
   apples  oranges
0       3        0
1       2        3
2       0        7
3       1        2
```

```
import pandas as pd

data = {
    'apples': [3, 2, 0, 1],
    'oranges': [0, 3, 7, 2]
}
purchases = pd.DataFrame(data, index=['June',
'Robert', 'Lily', 'David'])
print ( purchases )
print ( purchases.describe() )
```

Output

```
        apples  oranges
June         3        0
Robert       2        3
Lily         0        7
David        1        2
```

Pandas DataFrame

```
import pandas as pd

XYZ_web= {'Day':[1,2,3,4,5,6],
          "Visitors":[1000, 700,6000,1000,400,350],
          "Bounce_Rate":[20,20, 23,15,10,34]}

df= pd.DataFrame(XYZ_web)

print(df)

print(df.head(2))

print(df.tail(2))
```

Output

```
   Day  Visitors  Bounce_Rate
0    1      1000           20
1    2       700           20
2    3      6000           23
3    4      1000           15
4    5       400           10
5    6       350           34


   Day  Visitors  Bounce_Rate
0    1      1000           20
1    2       700           20


   Day  Visitors  Bounce_Rate
4    5       400           10
5    6       350           34
```

Pandas DataFrame - Change the Column Headers

```
import pandas as pd

df = pd.DataFrame({
"Day":[1,2,3,4],
"Visitors":[200, 100,230,300],
"Bounce_Rate":[20,45,60,10]})

print(df)

df = df.rename(columns={"Visitors":"Users"})

print(df)
```

Output

```
   Day  Visitors  Bounce_Rate
0    1       200           20
1    2       100           45
2    3       230           60
3    4       300           10

   Day  Users  Bounce_Rate
0    1    200           20
1    2    100           45
2    3    230           60
3    4    300           10
```

Pandas DataFrame - join

```
import pandas as pd
df1 = pd.DataFrame({
"Int_Rate":[2,1,2,3],
"IND_GDP":[50,45,45,67]},
index=[2001, 2002,2003,2004])

df2 = pd.DataFrame({“
Low_Tier_HPI":[50,45,67,34],
"Unemployment":[1,3,5,6]},
index=[2001, 2003,2004,2004])

joined= df1.join(df2)
print(joined)
```

Output

```
      Int_Rate  IND_GDP  Low_Tier_HPI  Unemployment
2001         2       50          50.0           1.0
2002         1       45           NaN           NaN
2003         2       45          45.0           3.0
2004         3       67          67.0           5.0
2004         3       67          34.0           6.0
```

Pandas DataFrame - Concat

```
import pandas as pd
df1 = pd.DataFrame({
"HPI":[80,90,70,60],
"Int_Rate":[2,1,2,3],
"IND_GDP":[50,45,45,67]},
index=[2001, 2002,2003,2004])

df2 = pd.DataFrame({
"HPI":[80,90,70,60],
"Int_Rate":[2,1,2,3],
"IND_GDP":[50,45,45,67]},
index=[2005, 2006,2007,2008])

concat= pd.concat([df1,df2])

print(concat)
```

Output

```
      HPI  Int_Rate  IND_GDP
2001   80         2       50
2002   90         1       45
2003   70         2       45
2004   60         3       67
2005   80         2       50
2006   90         1       45
2007   70         2       45
2008   60         3       67
```

Pandas DataFrame

```
import numpy as np
import pandas as pd

dates = pd.date_range('20190101', periods=8)

df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))

print(df.head())

print(df['P'])
```

Output

```
                   P         Q         R         S
2019-01-01  0.559359  1.467287 -0.828655  1.089501
2019-01-02 -1.440048 -0.357750 -0.477209  1.103896
2019-01-03  0.983483 -1.234996  1.365034 -1.754437
2019-01-04 -0.021533  0.786407 -0.900596  0.445841
2019-01-05  1.019841  0.193859  1.183123 -1.080805
```

```
2019-01-01    0.559359
2019-01-02   -1.440048
2019-01-03    0.983483
2019-01-04   -0.021533
2019-01-05    1.019841
2019-01-06   -1.309754
2019-01-07   -0.923993
2019-01-08   -0.694139
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df[0:1])
print(df['20190102':'20190104'])
```

Output

print(df.head())

```
                   P         Q         R         S
2019-01-01 -0.716518  0.999751 -1.167948  1.402634
2019-01-02  0.895367  2.112606 -1.492215  0.696113
2019-01-03 -0.443518  0.537104 -0.548634 -0.715293
2019-01-04 -0.146201 -1.204160 -0.646642  0.373661
2019-01-05  0.761034 -0.537594 -1.062357  0.889081
```

print(df[0:1])

```
                   P         Q         R         S
2019-01-01 -0.716518  0.999751 -1.167948  1.402634
```

print(df['20190102':'20190104'])

```
                   P         Q         R         S
2019-01-02  0.895367  2.112606 -1.492215  0.696113
2019-01-03 -0.443518  0.537104 -0.548634 -0.715293
2019-01-04 -0.146201 -1.204160 -0.646642  0.373661
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df[['P','Q']])
```

Output

print(df.head())

```
                   P         Q         R         S
2019-01-01  0.194923 -0.370216  0.222318 -0.610820
2019-01-02 -0.273129  0.852155 -0.139830  2.472619
2019-01-03  1.080802 -0.389328 -0.736021  1.817618
2019-01-04  0.301922 -1.369782 -0.237636  1.604091
2019-01-05  1.239508 -0.133633  0.717355  0.494644
```

print(df[['P','Q']])

```
                   P         Q
2019-01-01  0.194923 -0.370216
2019-01-02 -0.273129  0.852155
2019-01-03  1.080802 -0.389328
2019-01-04  0.301922 -1.369782
2019-01-05  1.239508 -0.133633
2019-01-06  1.444381 -0.589104
2019-01-07 -0.033460  0.386202
2019-01-08 -0.003405 -0.15476
```

Pandas DataFrame-Show label slicing

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df.loc['20190102':'20190104', ['P', 'Q']])
```

Output

```
print(df.head())
```

```
                   P         Q         R         S
2019-01-01  1.490785  0.080617  1.065316 -0.737873
2019-01-02  2.169922  0.351161 -1.405404 -0.096362
2019-01-03  1.357195 -0.741807 -0.899061 -0.073822
2019-01-04  0.388382  0.571622  0.884952 -1.113477
2019-01-05 -1.341752 -1.751120  0.849083 -0.691668
```

```
print(df.loc['20190102':'20190104', ['P', 'Q']])
```

```
                   P         Q
2019-01-02  2.169922  0.351161
2019-01-03  1.357195 -0.741807
2019-01-04  0.388382  0.571622
```

Pandas DataFrame-Slice columns explicitly

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df.iloc[:, 1:3])
```

print(df.head())

```
                   P         Q         R         S
2019-01-01 -0.168042  0.606405 -1.945616 -0.186862
2019-01-02  1.004644 -0.359166  0.044469 -0.206094
2019-01-03 -1.322607  0.575005 -0.372156 -0.105822
2019-01-04  0.363599  1.218763 -0.180398 -1.245851
2019-01-05 -1.048407 -0.682869 -1.222956  0.236697
```

Output

print(df.iloc[:, 1:3])

```
                   Q         R
2019-01-01  0.606405 -1.945616
2019-01-02 -0.359166  0.044469
2019-01-03  0.575005 -0.372156
2019-01-04  1.218763 -0.180398
2019-01-05 -0.682869 -1.222956
2019-01-06 -0.992171  0.052238
2019-01-07 -0.663789  0.285577
2019-01-08  0.471866 -1.161482
```

Pandas DataFrame-Slice Data

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df.iloc[0, 1])
print(df.iloc[1:3, :])
```

Output

```
print(df.head())
```

```
                   P         Q         R         S
2019-01-01  0.292920  0.540170 -0.594273 -0.886492
2019-01-02 -1.067475  0.034921 -0.098853  0.278866
2019-01-03  0.335760 -1.270671 -1.406635  1.865649
2019-01-04 -0.499807  0.442014 -1.133024 -0.770023
2019-01-05  1.848311  0.933322  0.000183  1.474276
```

```
print(df.iloc[0, 1])
```

```
0.5401701494910425
```

```
print(df.iloc[1:3, :])
```

```
                   P         Q         R         S
2019-01-02 -1.067475  0.034921 -0.098853  0.278866
2019-01-03  0.335760 -1.270671 -1.406635  1.865649
```

Pandas DataFrame-Slice Data

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df.iloc[[1, 2, 4], [0, 2]])
```

`print(df.head())`

Output

```
                   P         Q         R         S
2019-01-01 -0.837851 -0.124148  0.321961  0.638967
2019-01-02  0.766274  0.423321  0.124436 -0.150456
2019-01-03  0.665798 -0.552513 -0.863151 -2.080163
2019-01-04  0.158248 -1.321945 -0.110740  1.550404
2019-01-05 -1.433151 -1.039704 -0.177634 -0.002547
```

`print(df.iloc[[1, 2, 4], [0, 2]])`

```
                   P         R
2019-01-02  0.766274  0.124436
2019-01-03  0.665798 -0.863151
2019-01-05 -1.433151 -0.177634
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df[df > 0])
```

Output

print(df.head())

```
                   P         Q         R         S
2019-01-01  0.578782  0.055686  1.341621  0.764564
2019-01-02  0.544701  1.546507 -2.601742  0.598602
2019-01-03  1.409990 -0.024957 -0.416438  0.755334
2019-01-04 -0.493287  0.012002 -0.234159  1.385892
2019-01-05 -0.671253  0.347624 -0.861024  0.521264
```

print(df[df > 0])

```
                   P         Q         R         S
2019-01-01  0.578782  0.055686  1.341621  0.764564
2019-01-02  0.544701  1.546507       NaN  0.598602
2019-01-03  1.409990       NaN       NaN  0.755334
2019-01-04       NaN  0.012002       NaN  1.385892
2019-01-05       NaN  0.347624       NaN  0.521264
2019-01-06  0.891481  1.661300       NaN       NaN
2019-01-07       NaN  0.764096  0.322414       NaN
2019-01-08  1.146346       NaN       NaN       NaN
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
print(df[df.P > 0])
```

```
print(df.head())
```

Output

```
                   P         Q         R         S
2019-01-01 -0.161983  0.863285 -1.035204  0.037662
2019-01-02 -2.632035  1.377897 -1.184486  0.787570
2019-01-03 -2.479254 -0.617841 -0.946430 -0.352664
2019-01-04  0.200430 -0.026002  0.444338  0.843492
2019-01-05  1.215763 -0.580857  0.603784  0.478216
```

```
print(df[df.P > 0])
```

```
                   P         Q         R         S
2019-01-04  0.200430 -0.026002  0.444338  0.843492
2019-01-05  1.215763 -0.580857  0.603784  0.478216
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
df['P'] = [100,200,300,100,250,200,600,700]
print(df)
```

print(df.head())

```
                   P         Q         R         S
2019-01-01 -0.446656  0.094487 -1.653418 -1.016321
2019-01-02  0.613081 -0.942631 -1.586801  0.725686
2019-01-03 -1.363058 -0.077061 -0.371137 -1.014987
2019-01-04 -1.314356  0.107385  0.149934 -0.081091
2019-01-05 -0.595413  0.712464  0.998410  0.267980
```

Output

print(df)

```
              P         Q         R         S
2019-01-01  100  0.094487 -1.653418 -1.016321
2019-01-02  200 -0.942631 -1.586801  0.725686
2019-01-03  300 -0.077061 -0.371137 -1.014987
2019-01-04  100  0.107385  0.149934 -0.081091
2019-01-05  250  0.712464  0.998410  0.267980
2019-01-06  200 -2.098965 -0.150667 -1.206213
2019-01-07  600 -0.378645  1.310790  0.592624
2019-01-08  700  0.665129 -0.782738  0.167839
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
df['P'] = [100,200,300,100,250,200,600,700]
print(df[df['P'].isin([200, 700])])
```

`print(df.head())`

Output

```
                   P         Q         R         S
2019-01-01 -0.028442  2.219554 -1.681576  1.134103
2019-01-02  1.332084  1.571340  0.170071 -1.029884
2019-01-03 -0.053359 -0.764579  0.754551 -1.959716
2019-01-04  1.646547 -1.127585 -0.696043 -0.461747
2019-01-05  0.546074  2.485169 -0.023573  0.765066
```

`print(df[df['P'].isin([200, 700])])`

```
              P         Q         R         S
2019-01-02  200  1.571340  0.170071 -1.029884
2019-01-06  200 -0.487141 -2.330181 -0.949034
2019-01-08  700 -1.122614 -0.303397 -0.560787
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
df.at[dates[0], 'P'] = 0.0
df.iat[0, 2] = 999.0
df.loc[:, 'S'] = np.array([5] * len(df))
print(df)
```

Output

print(df.head())

```
                   P         Q         R         S
2019-01-01 -0.565303 -0.249959  0.500103  1.406026
2019-01-02 -0.978005  1.982600  0.480038  1.195780
2019-01-03 -0.724407  0.872757 -0.420258 -0.746150
2019-01-04 -1.450666 -0.696588  1.704557  0.344955
2019-01-05 -0.416627  0.487301 -0.452892  0.817004
```

print(df)

```
                   P         Q           R  S
2019-01-01  0.000000 -0.249959  999.000000  5
2019-01-02 -0.978005  1.982600    0.480038  5
2019-01-03 -0.724407  0.872757   -0.420258  5
2019-01-04 -1.450666 -0.696588    1.704557  5
2019-01-05 -0.416627  0.487301   -0.452892  5
2019-01-06  0.312700  1.924013   -0.430044  5
2019-01-07 -0.620329 -0.168422   -1.403376  5
2019-01-08  2.392160  1.004386   -0.851100  5
```

Pandas DataFrame

```
import numpy as np
import pandas as pd
dates = pd.date_range('20190101', periods=8)
df = pd.DataFrame(np.random.randn(8, 4), index=dates, columns=list('PQRS'))
print(df.head())
df2 = df.copy()
df2[df2 > 0] = -df2
print(df2)
```

Output

print(df.head())

```
                   P         Q         R         S
2019-01-01  1.121135  0.215492 -0.194989  1.023495
2019-01-02 -0.494460 -0.918972  0.380034  1.792180
2019-01-03  1.189559  0.976354 -0.845745  1.918448
2019-01-04 -0.300845 -1.300496  0.407470 -0.976821
2019-01-05 -1.237620  0.751060 -0.363478  1.434655
```

print(df2)

```
                   P         Q         R         S
2019-01-01 -1.121135 -0.215492 -0.194989 -1.023495
2019-01-02 -0.494460 -0.918972 -0.380034 -1.792180
2019-01-03 -1.189559 -0.976354 -0.845745 -1.918448
2019-01-04 -0.300845 -1.300496 -0.407470 -0.976821
2019-01-05 -1.237620 -0.751060 -0.363478 -1.434655
2019-01-06 -0.734849 -1.296497 -1.509392 -1.254806
2019-01-07 -1.108732 -2.377586 -0.467828 -1.558607
2019-01-08 -0.148871 -0.727270 -0.177168 -0.117059
```

DataFrame

```
import pandas as pd

dict = {"country": ["Brazil", "Russia",
     "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New
         Dehli", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597,
              1.221],
        "population": [200.4, 143.5, 1252, 1357,
              52.98] }


my_data = pd.DataFrame(dict)

print("mean :",my_data.mean())

print(my_data.describe())

my_data.index = ["BR", "RU", "IN", "CH", "SA"]
print(my_data)
```

Output

```
mean : area            7.944
population    601.176
dtype: float64
             area   population
count    5.000000     5.000000
mean     7.944000   601.176000
std      6.200557   645.261454
min      1.221000    52.980000
25%      3.286000   143.500000
50%      8.516000   200.400000
75%      9.597000  1252.000000
max     17.100000  1357.000000

      area    capital        country  population
BR   8.516   Brasilia         Brazil      200.40
RU  17.100     Moscow         Russia      143.50
IN   3.286  New Dehli          India     1252.00
CH   9.597    Beijing          China     1357.00
SA   1.221   Pretoria   South Africa       52.98
```

DataFrame

```
import pandas as pd

d = {'one':[1,2,3,4,5],
     'two':[2,2,2,2,2],
     'letter':['a','a','b','b','c']}


df = pd.DataFrame(d)

for index, row in df.iterrows() :
    print(row['two']+3, row['letter'])
```

Output

```
5 a
5 a
5 b
5 b
5 c
```

DataFrame

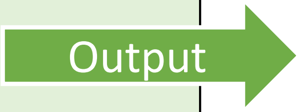
```
import pandas as pd

names = ['Bob','Jessica','Mary','John','Mel']
births = [968, 155, 77, 578, 973]

BabyDataSet = list(zip(names,births))

print (BabyDataSet)

df = pd.DataFrame(data = BabyDataSet,
columns=['Names', 'Births'])
print(df)

df= df.sort_values(['Births'], ascending=False)
print( df)

print( df.head(1) )


print ( "Max Birth:" , df['Births'].max() )
print ( “Sum Birth:" , df['Births'].sum() )
```

Output

```
[('Bob', 968), ('Jessica', 155), ('Mary', 77),
('John', 578), ('Mel', 973)]

     Names  Births
0      Bob     968
1  Jessica     155
2     Mary      77
3     John     578
4  Mel      973

     Names  Births
4      Mel     973
0      Bob     968
3     John     578
1  Jessica     155
2  Mary      77

  Names  Births
4  Mel      973

Max Birth: 973
Sum Birth: 2751
```

Applying functions on data

```
import pandas as pd
import numpy as np


d = {'one':[1,2,3,4,5],
     'two':[2,2,2,2,2],
     'letter':['a','a','b','b','c']}

df = pd.DataFrame(d)

print(df)

print( df['one'].apply( np.sqrt ) )

print( df['letter'].map(lambda x : 'map_' + x) )
```

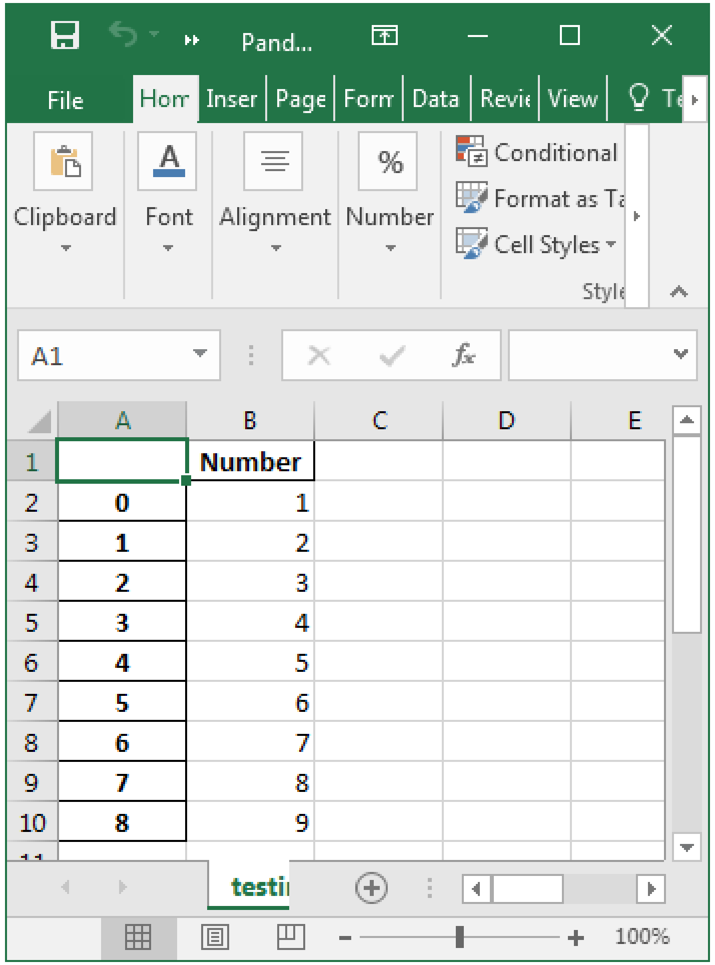

Output

```
 letter  one  two
0      a    1    2
1      a    2    2
2      b    3    2
3      b    4    2
4     c    5    2

0    1.000000
1    1.414214
2    1.732051
3    2.000000
4    2.236068
Name: one, dtype: float64

0    map_a
1    map_a
2    map_b
3    map_b
4    map_c
Name: letter, dtype: object
```

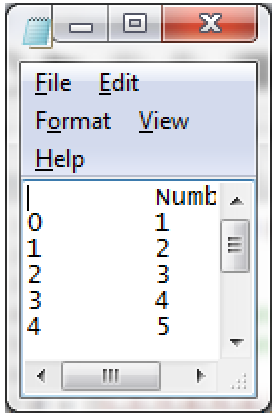
Export to Excel and csv formats

```
import pandas as pd

d = [1,2,3,4,5,6,7,8,9]
df = pd.DataFrame(d, columns = ['Number'])

df.to_excel('PandaTest.xlsx', sheet_name =
'testing', index = True)

df.to_csv('myDataFrame.csv', sep='\t')
```

Output



	A	B	C	D	E
1		Number			
2	0	1			
3	1	2			
4	2	3			
5	3	4			
6	4	5			
7	5	6			
8	6	7			
9	7	8			
10	8	9			



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

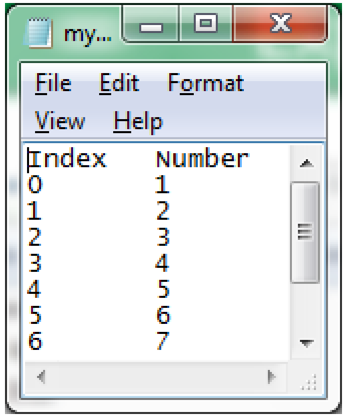
Reading from files

```
import pandas as pd

df =
pd.read_csv('myDataFrame.csv',sep='\t',index_col=0)

print(df)
```

Output

```
          Number
Index
0              1
1              2
2              3
3              4
4              5
5              6
6              7
7              8
8              9
```

input

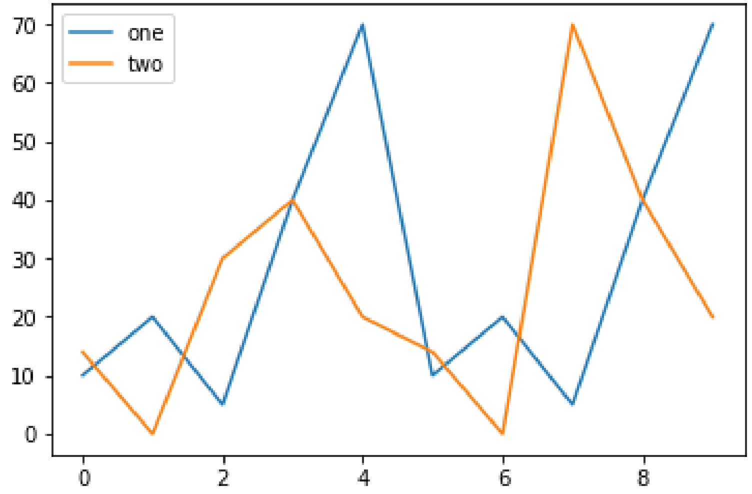


```
Index    Number
0        1
1        2
2        3
3        4
4        5
5        6
6        7
```

Plotting

```
import pandas as pd

d = {'one':[10,20,5,40,70,10,20,5,40,70],
     'two':[14,0,30,40,20,14,0,70,40,20]
     }

df = pd.DataFrame(d,columns=["one", "two"])

df.plot()
```

Output

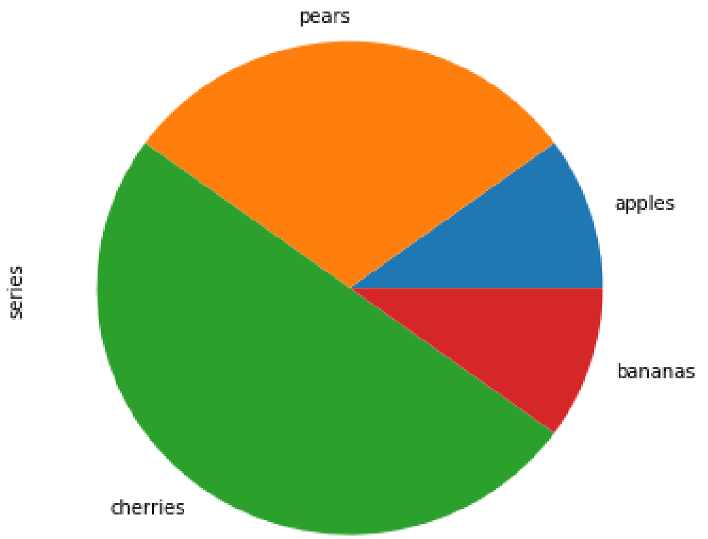


Plotting

```
import pandas as pd
fruits = ['apples', 'pears', 'cherries',
'bananas']
series = pd.Series([10, 30, 50, 10],
                   index=fruits,
                   name='series')
print(series)
series.plot.pie(figsize=(6, 6))
```

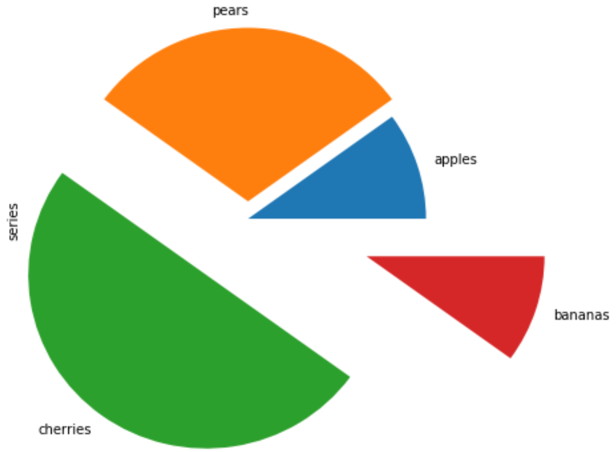
Output

```
apples      10
pears       30
cherries    50
bananas     10
```

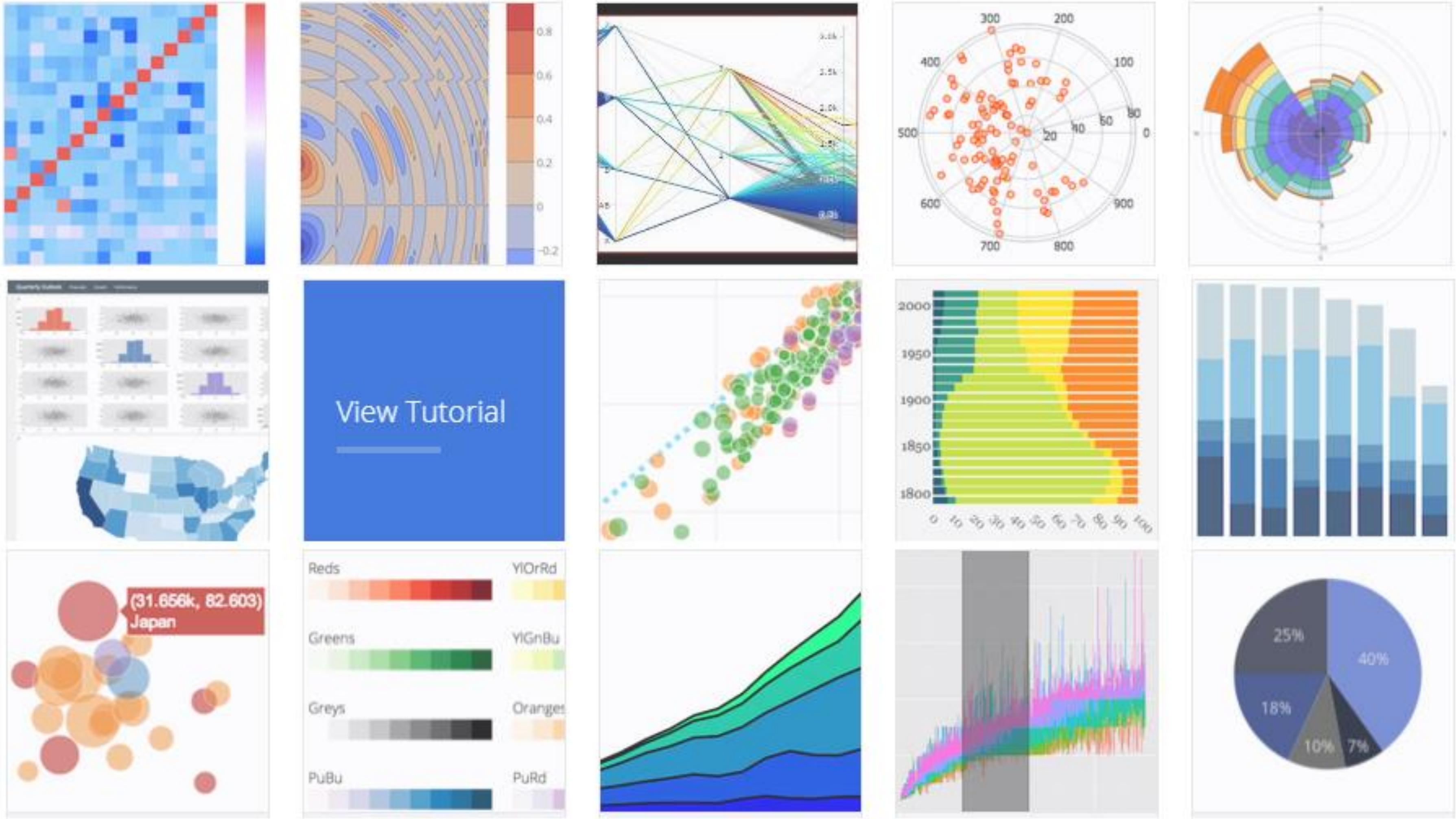


```
import pandas as pd
fruits = ['apples', 'pears', 'cherries',
'bananas']
series = pd.Series([10, 30, 50, 10],
                   index=fruits,
                   name='series')
explode = [0, 0.10, 0.40, 0.7]
series.plot.pie(figsize=(6, 6),
                explode=explode)
```

Output

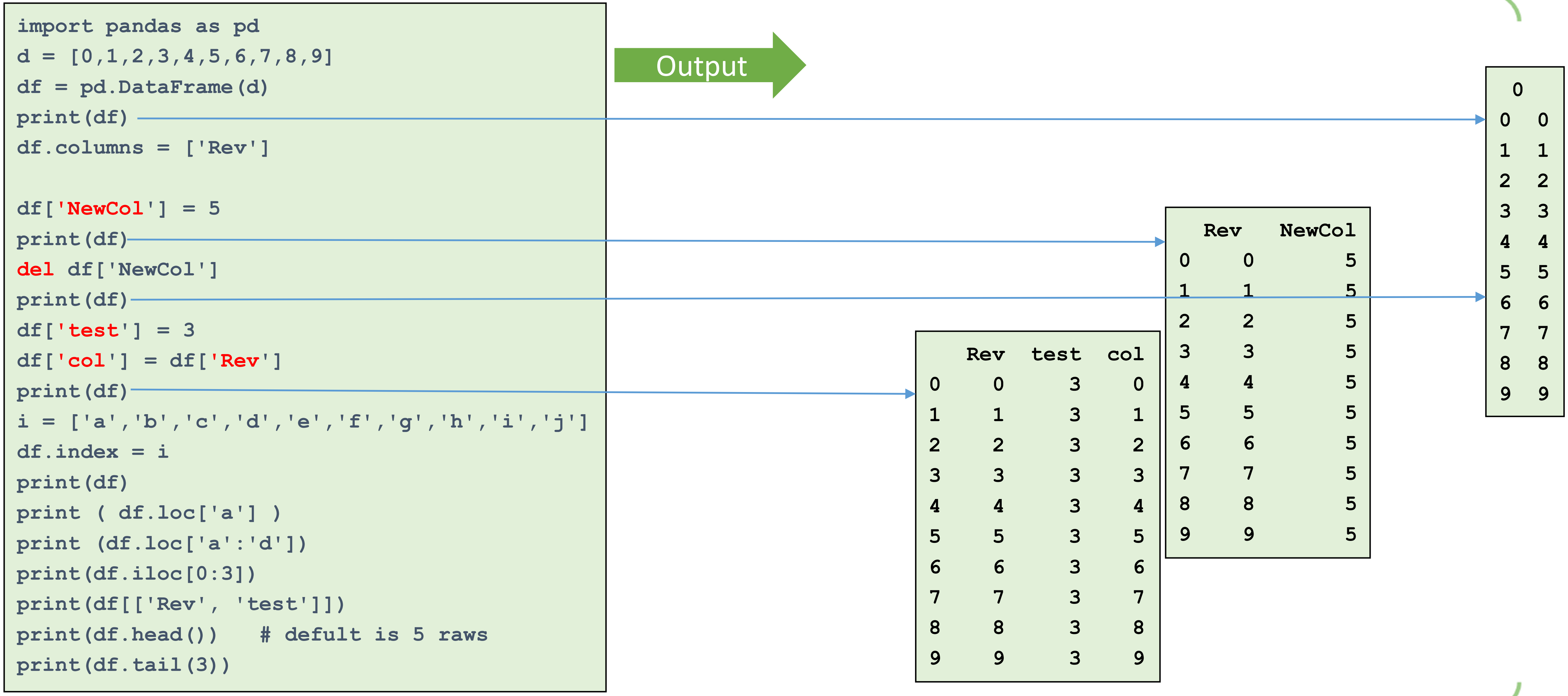


And many moreData plotting



https://plot.ly/pandas/

DataFrame – adding/deleting Columns



```
import pandas as pd
d = [0,1,2,3,4,5,6,7,8,9]
df = pd.DataFrame(d)
print(df)
df.columns = ['Rev']

df['NewCol'] = 5
print(df)
del df['NewCol']
print(df)
df['test'] = 3
df['col'] = df['Rev']
print(df)
i = ['a','b','c','d','e','f','g','h','i','j']
df.index = i
print(df)
print ( df.loc['a'] )
print (df.loc['a':'d'])
print(df.iloc[0:3])
print(df[['Rev', 'test']])
print(df.head())    # defult is 5 raws
print(df.tail(3))
```

Output

```
   0
0  0
1  1
2  2
3  3
4  4
5  5
6  6
7  7
8  8
9  9
```

```
   Rev  NewCol
0    0       5
1    1       5
2    2       5
3    3       5
4    4       5
5    5       5
6    6       5
7    7       5
8    8       5
9    9       5
```

```
   Rev  test  col
0    0     3    0
1    1     3    1
2    2     3    2
3    3     3    3
4    4     3    4
5    5     3    5
6    6     3    6
7    7     3    7
8    8     3    8
9    9     3    9
```

DataFrame – Group By

```
import pandas as pd

d = {'one':[1,1,1,1,1],
     'two':[2,2,2,2,2],
     'letter':['a','a','b','b','c']}

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

gdf = df.groupby('letter')

# Apply sum function
print (gdf.sum())

letterone =
df.groupby(['letter','one']).sum()
print( letterone )
```

```
  letter  one  two
0      a    1    2
1      a    1    2
2      b    1    2
3      b    1    2
4      c    1    2
```

```
        one  two
letter
a         2    4
b         2    4
c         1    2
```

```
            two
letter one
a       1     4
b       1     4
c       1     2
```

http://nbviewer.jupyter.org/urls/bitbucket.org/hrojas/learn-pandas/raw/master/lessons/06%20-%20Lesson.ipynb

Master in Software Engineering

Hussam Hourani has over 25 years of Organizations Transformation, VROs, PMO, Large Scale and Enterprise Programs Global Delivery, Leadership, Business Development and Management Consulting. His client experience is wide ranging across many sectors but focuses on Performance Enhancement, Transformation, Enterprise Program Management, Artificial Intelligence and Data Science.