Ch06_Data Exploring and Analysis

September 5, 2018

1 Chapter 6: Data Exploring Analysis

```
In [5]: import pandas as pd
        import numpy as np
        data = np.array(['O','S','S','A'])
        S1 = pd.Series(data)   # without adding index
        S2 = pd.Series(data,index=[100,101,102,103]) # with adding index
        print (S1)
        print ("\n")
        print (S2)
```

```
0    O
1    S
2    S
3    A
dtype: object


100    O
101    S
102    S
103    A
dtype: object
```

1.0.1 Create series from dictionary

```
In [6]: import pandas as pd
        import numpy as np
        data = {'X' : 0., 'Y' : 1., 'Z' : 2.}
        SERIES1 = pd.Series(data)
        print (SERIES1)
```

```
X    0.0
Y    1.0
Z    2.0
dtype: float64
```

```
In [7]: import pandas as pd
        import numpy as np
        data = {'X' : 0., 'Y' : 1., 'Z' : 2.}
        SERIES1 = pd.Series(data,index=['Y','Z','W','X'])
        print (SERIES1)
```

```
Y    1.0
Z    2.0
W    NaN
X    0.0
dtype: float64
```

```
In [9]: # Use sclara to create a series
        import pandas as pd
        import numpy as np
        Series1 = pd.Series(7, index=[0, 1, 2, 3, 4])
        print (Series1)
```

```
0    7
1    7
2    7
3    7
4    7
dtype: int64
```

1.0.2 Accessing Data from Series

```
In [18]: import pandas as pd
         Series1 = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])

         print ("Example 1:Retrieve the first element")
         print (Series1[0] )
         print ("\nExample 2:Retrieve the first three element")
         print (Series1[:3])

         print ("\nExample 3:Retrieve the last three element")
         print(Series1[-3:])

         print ("\nExample 4:Retrieve a single element")
         print (Series1['a'])

         print ("\nExample 5:Retrieve multiple elements")
         print (Series1[['a','c','d']])
```

```
Example 1:Retrieve the first element
1
```

```
Example 2:Retrieve the first three element
a    1
b    2
c    3
dtype: int64

Example 3:Retrieve the last three element
c    3
d    4
e    5
dtype: int64

Example 4:Retrieve a single element
1

Example 5:Retrieve multiple elements
a    1
c    3
d    4
dtype: int64


In [5]:

my_series1
 0     5
1     6
2     7
3     8
4     9
5    10
dtype: int64

my_series2
 [[-1.79805538  1.2064565   0.52702868  1.80635619 -0.59607098 -0.30413582
   0.0987201  -1.96081146  0.70702519 -0.14291129]
 [ 1.67241792 -0.66720531  0.51659468 -0.6948038  -0.77523975 -0.14449286
  -1.30728967 -0.27468199  0.41362055 -0.0730409 ]
 [-1.41568504  0.64875767  1.28034714 -1.25395052 -1.51666171  1.45420099
   0.78888101 -0.30570775  0.586197    0.08412997]
 [-0.28438464  0.73398081  0.37524566 -1.53615335 -0.02963768  0.64138327
  -0.29687117 -0.47331108  0.37236995  0.9637345 ]
 [ 1.7787956   1.49304139  0.16643397 -1.13733089 -0.10784825 -0.73869741
  -1.17272967  0.17071335 -0.48801218 -1.99638411]]


In [20]: import pandas as pd
         import numpy as np
```

```
        my_series1 = pd.Series([5, 6, 7, 8, 9, 10])
        print ("my_series1\n", my_series1)
        print ("\n Series Analysis\n ")
        print ("Series mean value : ", my_series1.mean())  # find mean value in a series
        print ("Series max value : ",my_series1.max())   # find max value in a series
        print ("Series min value : ",my_series1.min())   # find min value in a series
        print ("Series standred deviation value : ",my_series1.std())   # find standred deviati
```

```
my_series1
 0     5
1     6
2     7
3     8
4     9
5    10
dtype: int64

 Series Analysis

Series mean value :  7.5
Series max value :  10
Series min value :  5
Series standred deviation value :  1.8708286933869707
```

```
In [11]: my_series1.describe()
```

```
Out[11]: count     6.000000
         mean      7.500000
         std       1.870829
         min       5.000000
         25%       6.250000
         50%       7.500000
         75%       8.750000
         max      10.000000
         dtype: float64
```

```
In [17]: my_series_11 = my_series1
         print (my_series1)
         my_series_11.index = ['A', 'B', 'C', 'D', 'E', 'F']
         print (my_series_11)
         print (my_series1)
```

```
0     5
1     6
2     7
3     8
4     9
5    10
```

```
dtype: int64
A     5
B     6
C     7
D     8
E     9
F    10
dtype: int64
A     5
B     6
C     7
D     8
E     9
F    10
dtype: int64
```

```
In [21]: my_series_11 = my_series1.copy()
         print (my_series1)
         my_series_11.index = ['A', 'B', 'C', 'D', 'E', 'F']
         print (my_series_11)
         print (my_series1)
```

```
0     5
1     6
2     7
3     8
4     9
5    10
dtype: int64
A     5
B     6
C     7
D     8
E     9
F    10
dtype: int64
0     5
1     6
2     7
3     8
4     9
5    10
dtype: int64
```

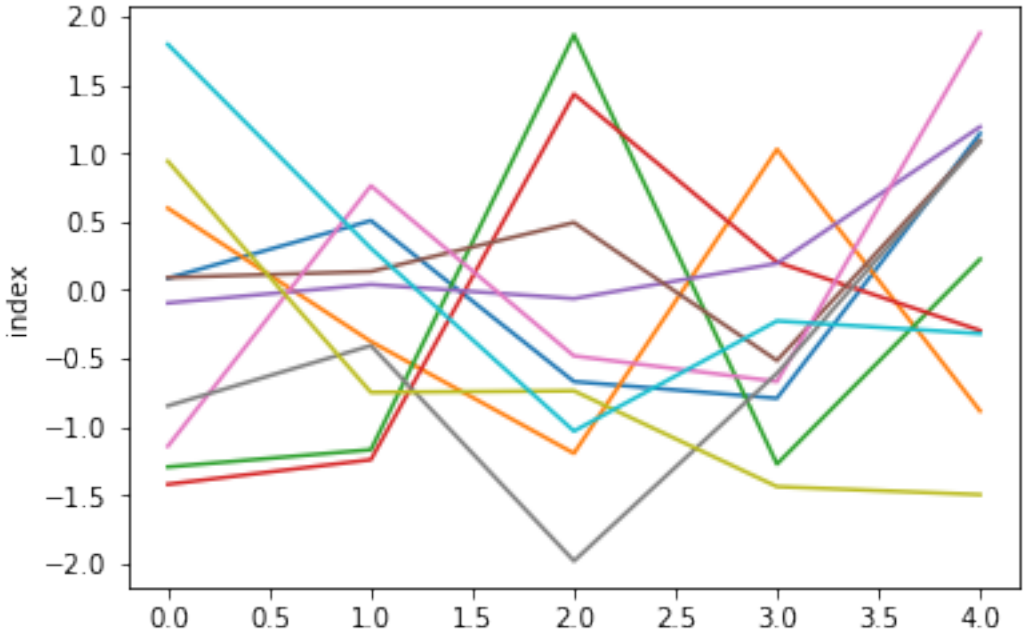
```
In [23]: 'F' in my_series_11
```

```
Out[23]: True
```

```
In [27]: temp = my_series_11 < 8
         temp

Out[27]: A     True
         B     True
         C     True
         D    False
         E    False
         F    False
         dtype: bool

In [35]: len(my_series_11)

Out[35]: 6

In [28]: temp = my_series_11[my_series_11 < 8 ] * 2
         temp

Out[28]: A    10
         B    12
         C    14
         dtype: int64

In [37]: def AddSeries(x,y):
             for i in range (len(x)):
                 print (x[i] + y[i])

In [39]: print ("Add two series\n")
         AddSeries (my_series_11, my_series1)

Add two series

10
12
14
16
18
20


In [40]: import pandas as pd
         import numpy as np
         my_series2 = np.random.randn(5, 10)
         print ("\nmy_series2\n", my_series2)


my_series2
 [[ 0.08590877  0.59702919 -1.29330859 -1.42021041 -0.09535271  0.09058623
  -1.14191133 -0.84699991  0.94028641  1.79400706]
```

```
 [ 0.50645411 -0.37674882 -1.16751734 -1.24061761  0.03981985  0.13478382
   0.76132521 -0.40671662 -0.7484758   0.30420489]
 [-0.66951224 -1.19373055  1.86446782  1.43047631 -0.06302096  0.49239499
  -0.48208329 -1.9805521  -0.73735706 -1.03152802]
 [-0.79181088  1.02769491 -1.27216885  0.20320462  0.19385809 -0.51614599
  -0.66898612 -0.60962025 -1.43724096 -0.22663712]
 [ 1.14193093 -0.8842498   0.22409272 -0.29599594  1.1917404   1.09016684
   1.87701454  1.08452103 -1.49587483 -0.31887386]]
```

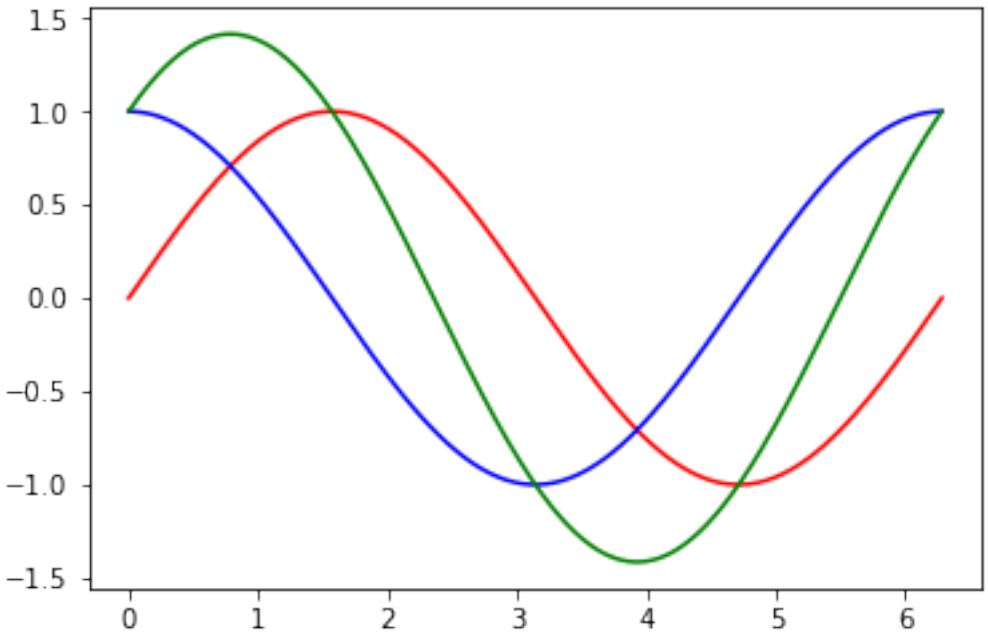
In [49]:
```
import matplotlib.pyplot as plt
plt.plot(my_series2)
plt.ylabel('index')
plt.show()
```



In [54]:
```
from numpy import *
import math
import matplotlib.pyplot as plt

t = linspace(0, 2*math.pi, 400)
a = sin(t)
b = cos(t)
c = a + b
```

In [50]:
```
plt.plot(t, a, 'r') # plotting t, a separately
plt.plot(t, b, 'b') # plotting t, b separately
```

```
plt.plot(t, c, 'g') # plotting t, c separately
plt.show()
```



1.0.3 create Data frame from lisits

```
In [19]: import pandas as pd
         data = [10,20,30,40,50]
         DF1 = pd.DataFrame(data)
         print (DF1)
```

```
    0
0  10
1  20
2  30
3  40
4  50
```

```
In [22]: import pandas as pd
         data = [['Ossama',25],['Ali',43],['Ziad',32]]
         DF1 = pd.DataFrame(data,columns=['Name','Age'])
         print (DF1)
```

```
     Name  Age
0  Ossama   25
1     Ali   43
```

```
2    Ziad   32
```

```
In [21]: import pandas as pd
         data = [['Ossama',25],['Ali',43],['Ziad',32]]
         DF1 = pd.DataFrame(data,columns=['Name','Age'],dtype=float)
         print (DF1)
```

```
     Name   Age
0  Ossama  25.0
1     Ali  43.0
2    Ziad  32.0
```

```
In [ ]: Create data frame from dictionaries
```

```
In [24]: import pandas as pd
         data = {'Name':['Omar', 'Ali', 'Mohammed', 'Ossama'],'Age':[30,25,44,4237]}
         DF1 = pd.DataFrame(data)
         print (DF1)
```

```
    Age      Name
0    30      Omar
1    25       Ali
2    44  Mohammed
3  4237    Ossama
```

```
In [26]: import pandas as pd
         data = {'Name':['Omar', 'Ali', 'Mohammed', 'Ossama'],'Age':[30,25,44,4237]}
         DF1 = pd.DataFrame(data, index=['Employee1','Employee2','Employee3','Employee4'])
         print (DF1)
```

```
            Age      Name
Employee1    30      Omar
Employee2    25       Ali
Employee3    44  Mohammed
Employee4  4237    Ossama
```

```
In [3]: import pandas as pd
        data = [{'Test1': 10, 'Test2': 20},{'Test3': 30, 'Project': 20, 'Final': 20}]
        df = pd.DataFrame(data)
        print (df)
```

```
   Final  Project  Test1  Test2  Test3
0    NaN      NaN   10.0   20.0    NaN
1   20.0     20.0    NaN    NaN   30.0
```

```
In [13]: import pandas as pd
         data = [{'Test1': 10, 'Test2': 20},{'Test1': 30, 'Test2': 20, 'Project': 20}]

         #With three column indices, values same as dictionary keys
         df1 = pd.DataFrame(data, index=['First', 'Second'], columns=['Test2', 'Project' , 'Test

         #With two column indices with one index with other name
         df2 = pd.DataFrame(data, index=['First', 'Second'], columns=['Project', 'Test_1','Test2
         print (df1)
         print ("\n")
         print (df2)

        Test2  Project  Test1
First      20      NaN     10
Second     20     20.0     30


        Project  Test_1  Test2
First       NaN     NaN     20
Second     20.0     NaN     20


In [16]: import pandas as pd

         data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
               'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa'])}

         df1 = pd.DataFrame(data)
         print (df1)

       Test1  Test2
Ahmed   70.0     56
Ali     89.0     77
Omar    55.0     82
Salwa    NaN     65


In [51]: import pandas as pd

         data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
               'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa'])}
         df1 = pd.DataFrame(data)
         print (df1['Test2'])   # Column selection
         print("\n")
         print (df1[:])   # Column selection

Ahmed    56
Ali      77
Omar     82
```

```
Salwa    65
Name: Test2, dtype: int64
```

```
       Test1  Test2
Ahmed   70.0     56
Ali     89.0     77
Omar    55.0     82
Salwa    NaN     65
```

```
In [46]: df1.iloc[:, [1,0 ]]
```

```
Out[46]:         Test2  Test1
         Ahmed     56   70.0
         Ali       77   89.0
         Omar      82   55.0
         Salwa     65    NaN
```

```
In [39]: df1[0:4:1]
```

```
Out[39]:         Test1  Test2
         Ahmed   70.0     56
         Ali     89.0     77
         Omar    55.0     82
         Salwa    NaN     65
```

```
In [66]: # add a new Column
         import pandas as pd
         data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
               'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa'])}
         df1 = pd.DataFrame(data)
         print (df1)
         df1['Project'] = pd.Series([90,83,67, 87],index=['Ali','Omar','Salwa', 'Ahmed'])
         print ("\n")
         df1['Average'] = round((df1['Test1']+df1['Test2']+df1['Project'])/3, 2)

         print (df1)
```

```
       Test1  Test2
Ahmed   70.0     56
Ali     89.0     77
Omar    55.0     82
Salwa    NaN     65
```

```
       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
```

```
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN
```

```
In [70]: import pandas as pd
         data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
               'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa'])}
         print (df1)
         df2 = df1
         print ("\n")
         print (df2)
```

```
       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN


       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN
```

```
In [71]: # Delete a column in data frame using del function
         print ("Deleting the first column using DEL function:")
         del df2['Test2']
         print (df2)

         # Delete a column in data frame  using pop function
         print ("\nDeleting another column using POP function:")
         df2.pop('Project')
         print (df2)
```

```
Deleting the first column using DEL function:
       Test1  Project  Average
Ahmed   70.0       87    71.00
Ali     89.0       90    85.33
Omar    55.0       83    73.33
Salwa    NaN       67      NaN

Deleting another column using POP function:
       Test1  Average
Ahmed   70.0    71.00
Ali     89.0    85.33
Omar    55.0    73.33
```

```
Salwa    NaN      NaN
```

```
In [72]: print (df1)
```

```
       Test1  Average
Ahmed   70.0    71.00
Ali     89.0    85.33
Omar    55.0    73.33
Salwa    NaN      NaN
```

```
In [73]: print (df2)
```

```
       Test1  Average
Ahmed   70.0    71.00
Ali     89.0    85.33
Omar    55.0    73.33
Salwa    NaN      NaN
```

```
In [83]: # add a new Column
         import pandas as pd
         data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
               'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa'])}
         df1 = pd.DataFrame(data)
         df1['Project'] = pd.Series([90,83,67, 87],index=['Ali','Omar','Salwa', 'Ahmed'])
         print ("\n")
         df1['Average'] = round((df1['Test1']+df1['Test2']+df1['Project'])/3, 2)
         print (df1)

         print ("\n")
         df2= df1.copy()   # copy df1 into df2 using copy() method
         print (df2)
         #delete columns using del and pop methods
         del df2['Test2']
         df2.pop('Project')
         print ("\n")
         print (df1)
         print ("\n")
         print (df2)
```

```
       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN
```

```
       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN
```

```
       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN
```

```
       Test1  Average
Ahmed   70.0    71.00
Ali     89.0    85.33
Omar    55.0    73.33
Salwa    NaN      NaN
```

```
In [106]: # add a new Column
          import pandas as pd
          data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
                'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa'])}
          df1 = pd.DataFrame(data)
          df1['Project'] = pd.Series([90,83,67, 87],index=['Ali','Omar','Salwa', 'Ahmed'])
          print ("\n")
          df1['Average'] = round((df1['Test1']+df1['Test2']+df1['Project'])/3, 2)
          print (df1)
          print ("\nselect  iloc function to retrieve  row number 2")
          print (df1.iloc[2])
          print ("\nslice rows")
          print (df1[2:4] )
```

```
       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN

select  iloc function to retrieve  row number 2
Test1      55.00
Test2      82.00
```

```
Project    83.00
Average    73.33
Name: Omar, dtype: float64

slice rows
       Test1  Test2  Project  Average
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN


In [108]: print (df1)

       Test1  Test2  Project  Average
Ahmed   70.0     56       87    71.00
Ali     89.0     77       90    85.33
Omar    55.0     82       83    73.33
Salwa    NaN     65       67      NaN


In [ ]: import pandas as pd
        data = {'Test1' : pd.Series([70, 55, 89], index=['Ahmed', 'Omar', 'Ali']),
              'Test2' : pd.Series([56, 82, 77, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa']),
              'Project' : pd.Series([87, 83, 90, 67], index=['Ahmed', 'Omar', 'Ali', 'Salwa']),
              'Average' : pd.Series([71, 73.33, 85.33, 66], index=['Ahmed', 'Omar', 'Ali', 'Salw

        data = pd.DataFrame(data)
        print (data)
        print("\n")
        df2 = pd.DataFrame([[80, 70, 90, 80]], columns = ['Test1','Test2','Project','Average'],
        data = data.append(df2)
        print (data)

In [138]: print (data)
          print ('\n')
          data = data.drop('Omar')
          print (data)

        Average  Project  Test1  Test2
Ahmed     71.00       87   70.0     56
Ali       85.33       90   89.0     77
Omar      73.33       83   55.0     82
Salwa     66.00       67    NaN     65
Khalid    80.00       90   80.0     70


        Average  Project  Test1  Test2
Ahmed     71.00       87   70.0     56
Ali       85.33       90   89.0     77
Salwa     66.00       67    NaN     65
```

```
Khalid    80.00       90   80.0     70
```

```
In [74]: import pandas as pd
         data = {'Age' : pd.Series([30, 25, 44, ], index=['Ahmed', 'Omar', 'Ali']),
               'Salary' : pd.Series([25000, 17000, 30000, 12000], index=['Ahmed', 'Omar', 'Ali',
               'Height' : pd.Series([160, 154, 175, 165], index=['Ahmed', 'Omar', 'Ali', 'Salwa'
               'Weight' : pd.Series([85, 70, 92, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa']),
               'Gender' : pd.Series(['Male', 'Male', 'Male', 'Female'], index=['Ahmed', 'Omar',

         data = pd.DataFrame(data)
         print (data)
         print("\n")
         df2 = pd.DataFrame([[42, 31000, 170, 80, 'Female']], columns = ['Age','Salary','Height'
                           , index=['Mona'])
         data = data.append(df2)
         print (data)
```

```
        Age  Gender  Height  Salary  Weight
Ahmed  30.0    Male     160   25000      85
Ali    44.0    Male     175   30000      92
Omar   25.0    Male     154   17000      70
Salwa   NaN  Female     165   12000      65


        Age  Gender  Height  Salary  Weight
Ahmed  30.0    Male     160   25000      85
Ali    44.0    Male     175   30000      92
Omar   25.0    Male     154   17000      70
Salwa   NaN  Female     165   12000      65
Mona   42.0  Female     170   31000      80
```

```
In [63]: data.describe()
```

```
Out[63]:              Age      Height        Salary     Weight
         count   4.000000    5.000000      5.000000   5.000000
         mean   35.250000  144.800000  23000.000000  78.400000
         std     9.215024   42.517055   8276.472679  10.968136
         min    25.000000   70.000000  12000.000000  65.000000
         25%    28.750000  154.000000  17000.000000  70.000000
         50%    36.000000  160.000000  25000.000000  80.000000
         75%    42.500000  165.000000  30000.000000  85.000000
         max    44.000000  175.000000  31000.000000  92.000000
```

```
In [64]: data.describe(include='all')
```

```
Out[64]:               Age Gender      Height        Salary     Weight
         count    4.000000      5    5.000000      5.000000   5.000000
```

```
unique        NaN      2         NaN           NaN         NaN
top           NaN   Male         NaN           NaN         NaN
freq          NaN      3         NaN           NaN         NaN
mean    35.250000    NaN  144.800000  23000.000000  78.400000
std      9.215024    NaN   42.517055   8276.472679  10.968136
min     25.000000    NaN   70.000000  12000.000000  65.000000
25%     28.750000    NaN  154.000000  17000.000000  70.000000
50%     36.000000    NaN  160.000000  25000.000000  80.000000
75%     42.500000    NaN  165.000000  30000.000000  85.000000
max     44.000000    NaN  175.000000  31000.000000  92.000000
```

In [66]:
```
data.Salary.describe()
```

Out[66]:
```
count        5.000000
mean     23000.000000
std       8276.472679
min      12000.000000
25%      17000.000000
50%      25000.000000
75%      30000.000000
max      31000.000000
Name: Salary, dtype: float64
```

In [67]:
```
data.describe(include=[np.number])
```

Out[67]:

	Age	Height	Salary	Weight
count	4.000000	5.000000	5.000000	5.000000
mean	35.250000	144.800000	23000.000000	78.400000
std	9.215024	42.517055	8276.472679	10.968136
min	25.000000	70.000000	12000.000000	65.000000
25%	28.750000	154.000000	17000.000000	70.000000
50%	36.000000	160.000000	25000.000000	80.000000
75%	42.500000	165.000000	30000.000000	85.000000
max	44.000000	175.000000	31000.000000	92.000000

In [68]:
```
data.describe(include=[np.object])
```

Out[68]:

	Gender
count	5
unique	2
top	Male
freq	3

In [70]:
```
data.describe(exclude=[np.number])
```

Out[70]:

	Gender
count	5
unique	2
top	Male
freq	3

```
In [71]: data
```

```
Out[71]:          Age  Gender  Height  Salary  Weight
         Ahmed  30.0    Male     160   25000      85
         Ali    44.0    Male     175   30000      92
         Omar   25.0    Male     154   17000      70
         Salwa   NaN  Female     165   12000      65
         Mona   42.0  Female      70   31000      80
```

```
In [75]: OptimalWeight = data['Height']- 100
         OptimalWeight
```

```
Out[75]: Ahmed    60
         Ali      75
         Omar     54
         Salwa    65
         Mona     70
         Name: Height, dtype: int64
```

```
In [93]: unOptimalCases =  data['Weight'] <= OptimalWeight
         unOptimalCases
```

```
Out[93]: Ahmed    False
         Ali      False
         Omar     False
         Salwa     True
         Mona     False
         dtype: bool
```

1.1 Create Panel

```
In [141]: np.random.randn(4, 3)
```

```
Out[141]: array([[-1.03612404, -1.15264536, -0.96478642],
                 [-0.48753308, -0.29837715,  1.55695023],
                 [-0.40013819, -0.49845239,  0.8309264 ],
                 [-0.09094099,  0.28760056, -0.65767939]])
```

```
In [143]: # creating an empty panel
          import pandas as pd
          import numpy as np

          data = np.random.rand(2,4,5)
          Paneldf = pd.Panel(data)
          print (Paneldf)
```

```
<class 'pandas.core.panel.Panel'>
Dimensions: 2 (items) x 4 (major_axis) x 5 (minor_axis)
Items axis: 0 to 1
Major_axis axis: 0 to 3
Minor_axis axis: 0 to 4
```

```
In [94]: data = {'Item1' : pd.DataFrame(np.random.randn(4, 3)),
                 'Item2' : pd.DataFrame(np.random.randn(4, 2))}
         p = pd.Panel(data)

In [95]: p

Out[95]: <class 'pandas.core.panel.Panel'>
         Dimensions: 2 (items) x 4 (major_axis) x 3 (minor_axis)
         Items axis: Item1 to Item2
         Major_axis axis: 0 to 3
         Minor_axis axis: 0 to 2

In [97]: p['Item1'].describe()

Out[97]:                0         1         2
         count  4.000000  4.000000  4.000000
         mean  -0.376501 -0.489938  0.571478
         std    0.433701  1.646804  1.287483
         min   -0.973237 -2.087780 -0.746382
         25%   -0.559222 -1.567878 -0.143273
         50%   -0.246122 -0.768469  0.370201
         75%   -0.063401  0.309471  1.084952
         max   -0.040521  1.664965  2.291892

In [104]: import pandas as pd
          data1 = {'Age' : pd.Series([30, 25, 44, ], index=['Ahmed', 'Omar', 'Ali']),
                'Salary' : pd.Series([25000, 17000, 30000, 12000], index=['Ahmed', 'Omar', 'Ali'
                'Height' : pd.Series([160, 154, 175, 165], index=['Ahmed', 'Omar', 'Ali', 'Salwa
                'Weight' : pd.Series([85, 70, 92, 65], index=['Ahmed', 'Omar', 'Ali', 'Salwa']),
                'Gender' : pd.Series(['Male', 'Male', 'Male', 'Female'], index=['Ahmed', 'Omar',

          data2 = {'Age' : pd.Series([24, 19, 33,25  ], index=['Ziad', 'Majid', 'Ayman', 'Ahlam'
                'Salary' : pd.Series([17000, 7000, 22000, 21000], index=['Ziad', 'Majid', 'Ayman
                'Height' : pd.Series([170, 175, 162, 177], index=['Ziad', 'Majid', 'Ayman', 'Ahl
                'Weight' : pd.Series([77, 84, 74, 90], index=['Ziad', 'Majid', 'Ayman', 'Ahlam']
                'Gender' : pd.Series(['Male', 'Male', 'Male', 'Female'], index=['Ziad', 'Majid',

In [105]: data = {'Group1' :data1,
                  'Group2' :data2}
          p = pd.Panel(data)

In [106]: p['Group1'].describe()

Out[106]:          Age Gender  Height   Salary  Weight
          count    3.0      4     4.0      4.0     4.0
          unique   3.0      2     4.0      4.0     4.0
          top     30.0   Male   175.0  30000.0    70.0
          freq     1.0      3     1.0      1.0     1.0

In [107]: p['Group1']['Salary'].describe()
```

```
Out[107]: count         4.0
          unique        4.0
          top       30000.0
          freq          1.0
          Name: Salary, dtype: float64
```

```
In [147]: # creating an empty panel
          import pandas as pd
          import numpy as np
          data = {'Item1' : pd.DataFrame(np.random.randn(4, 3)),
                  'Item2' : pd.DataFrame(np.random.randn(4, 2))}
          Paneldf = pd.Panel(data)
          print (Paneldf['Item1'])
          print ("\n")
          print (Paneldf['Item2'])
```

```
          0         1         2
0 -1.069595  0.835842  0.950269
1  1.063784  0.520086  1.342309
2 -2.236069  0.229717  0.752612
3  1.014550  0.903234  2.011993


          0         1   2
0 -1.126333  1.528085 NaN
1 -1.255712  0.076873 NaN
2  1.593704 -0.648342 NaN
3  0.287446  1.591275 NaN
```

```
In [149]: print (Paneldf.major_xs(1))
```

```
      Item1     Item2
0  1.063784 -1.255712
1  0.520086  0.076873
2  1.342309       NaN
```

```
In [150]: print (Paneldf.minor_xs(1))
```

```
      Item1     Item2
0  0.835842  1.528085
1  0.520086  0.076873
2  0.229717 -0.648342
3  0.903234  1.591275
```

1.2 Data anlysis

```
In [11]: import pandas as pd
         import numpy as np
```

```
Number = [1,2,3,4,5,6,7,8,9,10]
Names = ['Ali Ahmed','Mohamed Ziad','Majid Salim','Salwa Ahmed', 'Ahlam Mohamed', 'Omar
        'Khalid Yousif', 'Safa Humaid', 'Amjad Tayel']

City = ['Fujairah','Dubai','Sharjah', 'AbuDhabi','Fujairah','Dubai','Sharjah', 'AbuDhab
columns = ['Number', 'Name', 'City' ]
dataset= pd.DataFrame({'Number': Number , 'Name': Names, 'City': City}, columns = colum
Gender= pd.DataFrame({'Gender':['Male','Male','Male','Female', 'Female', 'Male', 'Femal
                                'Female', 'Male']})
Height = pd.DataFrame(np.random.randint(120,175, size=(12, 1)))
Weight = pd.DataFrame(np.random.randint(50,110, size=(12, 1)))

dataset['Gender']= Gender
dataset['Height']= Height
dataset['Weight']= Weight
dataset.set_index('Number')
```

Out[11]:

Number	Name	City	Gender	Height	Weight
1	Ali Ahmed	Fujairah	Male	155	65
2	Mohamed Ziad	Dubai	Male	165	59
3	Majid Salim	Sharjah	Male	159	82
4	Salwa Ahmed	AbuDhabi	Female	138	106
5	Ahlam Mohamed	Fujairah	Female	152	100
6	Omar Ali	Dubai	Male	145	108
7	Amna Mohammed	Sharjah	Female	151	67
8	Khalid Yousif	AbuDhabi	Male	171	96
9	Safa Humaid	Sharjah	Female	140	82
10	Amjad Tayel	Fujairah	Male	161	92

```
In [186]: print ( dataset.describe()) # Summary statistics for numerical columns
```

```
          Number      Height      Weight
count  10.00000   10.00000   10.000000
mean    5.50000  148.00000   85.500000
std     3.02765   15.37675   10.617072
min     1.00000  128.00000   71.000000
25%     3.25000  134.50000   78.000000
50%     5.50000  149.00000   84.000000
75%     7.75000  159.50000   92.000000
max    10.00000  173.00000  104.000000
```

```
In [187]: print (dataset.mean()) # Returns the mean of all columns
```

```
Number      5.5
Height    148.0
Weight     85.5
dtype: float64
```

```
In [188]: print (dataset.corr()) # Returns the correlation between columns in a DataFrame

          Number    Height    Weight
Number  1.000000  0.124105  0.174557
Height  0.124105  1.000000 -0.301503
Weight  0.174557 -0.301503  1.000000


In [189]: print (dataset.count()) # Returns the number of non-null values in each DataFrame colu

Number    10
Name      10
City      10
Gender    10
Height    10
Weight    10
dtype: int64


In [190]: print (dataset.max()) # Returns the highest value in each column

Number             10
Name      Salwa Ahmed
City          Sharjah
Gender            Male
Height             173
Weight             104
dtype: object


In [191]: print (dataset.min()) # Returns the lowest value in each column

Number                1
Name      Ahlam Mohamed
City           AbuDhabi
Gender           Female
Height              128
Weight               71
dtype: object


In [192]: print (dataset.median()) # Returns the median of each column

Number      5.5
Height    149.0
Weight     84.0
dtype: float64


In [193]: print (dataset.std()) # Returns the standard deviation of each column
```

```
Number     3.027650
Height    15.376750
Weight    10.617072
dtype: float64
```

1.2.1 Grouping

print (dataset)

```
In [3]: dataset.groupby('City')['Gender'].count()
```

```
Out[3]: City
        AbuDhabi    2
        Dubai       2
        Fujairah    3
        Sharjah     3
        Name: Gender, dtype: int64
```

```
In [4]: print (dataset.groupby('City').groups)
```

```
{'AbuDhabi': Int64Index([3, 7], dtype='int64'), 'Dubai': Int64Index([1, 5], dtype='int64'), 'Fuj
```

```
In [5]: print (dataset.groupby(['City','Gender']).groups)
```

```
{('AbuDhabi', 'Female'): Int64Index([3], dtype='int64'), ('AbuDhabi', 'Male'): Int64Index([7], d
```

```
In [7]: grouped = dataset.groupby('Gender')

        for name,group in grouped:
            print (name)
            print (group)
            print ("\n")
```

```
Female
   Number           Name      City  Gender  Height  Weight
3       4    Salwa Ahmed  AbuDhabi  Female     125      57
4       5  Ahlam Mohamed  Fujairah  Female     170      99
6       7  Amna Mohammed   Sharjah  Female     160      97
8       9    Safa Humaid   Sharjah  Female     138      70


Male
   Number           Name      City Gender  Height  Weight
0       1      Ali Ahmed  Fujairah   Male     130      72
1       2   Mohamed Ziad     Dubai   Male     129      61
2       3    Majid Salim   Sharjah   Male     153      51
```

```
5       6       Omar Ali     Dubai   Male     135      97
7       8  Khalid Yousif  AbuDhabi   Male     170      55
9      10    Amjad Tayel  Fujairah   Male     163      88
```

```
In [9]: grouped = dataset.groupby('Gender')
        print (grouped.get_group('Female'))
```

```
   Number           Name      City  Gender  Height  Weight
3       4    Salwa Ahmed  AbuDhabi  Female     125      57
4       5  Ahlam Mohamed  Fujairah  Female     170      99
6       7  Amna Mohammed   Sharjah  Female     160      97
8       9    Safa Humaid   Sharjah  Female     138      70
```

```
In [18]: # Aggregation
         grouped = dataset.groupby('Gender')
         print (grouped['Height'].agg(np.mean))
         print ("\n")
         print (grouped['Weight'].agg(np.mean))
         print ("\n")
         print (grouped.agg(np.size))
         print ("\n")
         print (grouped['Height'].agg([np.sum, np.mean, np.std]))
```

```
Gender
Female    145.250000
Male      159.333333
Name: Height, dtype: float64
```

```
Gender
Female    88.750000
Male      83.666667
Name: Weight, dtype: float64
```

```
        Number  Name  City  Height  Weight
Gender
Female       4     4     4       4       4
Male         6     6     6       6       6
```

```
        sum        mean       std
Gender
Female  581  145.250000  7.274384
```

```
Male    956  159.333333  8.891944
```

```
In [19]: ### Transformations
```

```
In [ ]: dataset = dataset.set_index(['Number'])
        print (dataset)
```

```
In [26]:
```

```
                Name      City  Gender  Height  Weight
Number
1          Ali Ahmed  Fujairah    Male     155      65
2       Mohamed Ziad     Dubai    Male     165      59
3        Majid Salim   Sharjah    Male     159      82
4        Salwa Ahmed  AbuDhabi  Female     138     106
5      Ahlam Mohamed  Fujairah  Female     152     100
6           Omar Ali     Dubai    Male     145     108
7      Amna Mohammed   Sharjah  Female     151      67
8      Khalid Yousif  AbuDhabi    Male     171      96
9        Safa Humaid   Sharjah  Female     140      82
10       Amjad Tayel  Fujairah    Male     161      92
```

```
In [28]: grouped = dataset.groupby('Gender')
         score = lambda x: (x - x.mean()) / x.std()*10
         print (grouped.transform(score))
```

```
          Height     Weight
Number
1      -4.873325  -9.911893
2       6.372810 -13.097858
3      -0.374871  -0.884990
4      -9.966479   9.730865
5       9.279136   6.346216
6     -16.119460  12.920860
7       7.904449 -12.269352
8      13.120491   6.548929
9      -7.217106  -3.807730
10      1.874356   4.424952
```

1.2.2 Filtration

```
In [30]: print (dataset.groupby('City').filter(lambda x: len(x) >= 3))
```

```
                Name      City  Gender  Height  Weight
Number
1          Ali Ahmed  Fujairah    Male     155      65
```

3	Majid Salim	Sharjah	Male	159	82
5	Ahlam Mohamed	Fujairah	Female	152	100
7	Amna Mohammed	Sharjah	Female	151	67
9	Safa Humaid	Sharjah	Female	140	82
10	Amjad Tayel	Fujairah	Male	161	92