Create a pandas series from a dictionary of values and ndarray

In [3]:
```
import pandas as pd
import numpy as np
myArr = np.array([23,1,9,80,48,67])
s = pd.Series(myArr)
print(s)
#create dictionary
dictionary = {'A':10, 'B':20,'C':30}

#create a series from dictionary
series = pd.Series(dictionary)

print(series)
```

```
0    23
1     1
2     9
3    80
4    48
5    67
dtype: int32
A    10
B    20
C    30
dtype: int64
```

#2 **Given a Series, print all the element that are above the 75th percentile

In [4]:
```
#import pandas and numpy
import pandas as pd
import numpy as np
s = pd.Series(np.array([1,3,4,5,6,8,8]))
print(s)

res = s.quantile(q=0.75)
#the element that are above the 75th precentile

print("The element that are above the 75 percentile:-")
print(s[s>res])
```

```
0    1
1    3
2    4
3    5
4    6
5    8
6    8
dtype: int32
The element that are above the 75 percentile:-
5    8
6    8
dtype: int32
```

```
import pandas as pd
dic= { "itemcat":["car","Mobile Phone" "Washing Machine", "TV"],
       "itemname" : ["ford", "Hitachi", "Symphony", "LG"],
      "expenditure":[700000, 50000, 20000, 50000]}
quartSales = pd.DataFrame.from_dict(dic,orient='index')
print(quartSales)
qs= quartSales.groupby("itemcat")

print("Result after Filterings Dataframe")
print(qs["itemcat", "expenditure"].sum())
```

	0	1	2	3
itemcat	car	Mobile PhoneWashing Machine	TV	None
itemname	ford	Hitachi	Symphony	LG
expenditure	700000	50000	20000	50000

```
Result after Filtering Dataframe
                  expenditure
itemcat
Ac                      50000
Aircoller               12000
Washing Machine         14000
car                   7000000
```

Create a dataframe based on ecommerce data and generate descriptive statistics(mean, meridian, mode, quartile and variance).

In [27]:
```
import pandas as pd
sales = { "invoice":[1001,1002,1003,1004,1005,1006,1007],
          "ProductName" : ["Led", "AC", "deodrant", "jeans", "Books", "Shoes","Jacket" ],
         "Quantity": [2,1,3,5,8,5,4],
         "Price": [65000, 55000, 1000, 2500, 950, 4000, 2200]}
df = pd.DataFrame(sales)
print(df["Price"].describe().round(2))
```

```
count       7.00
mean    18664.29
std     28403.36
min       950.00
25%      1600.00
50%      2500.00
75%     29500.00
max     65000.00
Name: Price, dtype: float64
```

```
import pandas as pd
dic = {"Class":["I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX", "X", "XI", "XII"],
       "Pass-Percentage": [100,100,100,98,99,98,97,100,100,99,96.5,100]}
result = pd.DataFrame(dic)
print(result)
print(result.dtypes)
print("shape of the Dataframe:")
print(result.shape)
```

```
   Class  Pass-Percentage
0      I            100.0
1     II            100.0
2    III            100.0
3     IV             98.0
4      V             99.0
5     VI             98.0
6    VII             97.0
7   VIII            100.0
8     IX            100.0
9      X             99.0
10    XI             96.5
11   XII            100.0
Class              object
Pass-Percentage   float64
dtype: object
shape of the Dataframe:
(12, 2)
```

#7 Find the sum of each column, or find the column with the lowest mean.

In [36]:
```
import pandas as pd
profit = {"TCS": {"Qtr1":2500, "Qtr2": 2000, "Qtr3" : 3000, "Qtr4":2500},
          "WIPRO": {"Qtr1":2800, "Qtr2": 2400, "Qtr3":3600, "Qtr4":2500},
          "L&T": {"Qtr1":2100, "Qtr2": 5700, "Qtr3":3500, "Qtr4":2100} }
fd = pd.DataFrame(profit)
print(df)
print()
print("Column wise sum in DataFrame is: ")
#print mean value of each column
print()
print("Column wise mean value are: ")
print(df.mean(axis=0))
print()
print("Column with minimum mean value is ")
print(df.mean(axis=0).idxmin())
```

```
   invoice ProductName  Quantity  Price
0     1001         Led         2  65000
1     1002          AC         1  55000
2     1003    deodrant         3   1000
3     1004       jeans         5   2500
4     1005       Books         8    950
5     1006       Shoes         5   4000
6     1007      Jacket         4   2200

Column wise sum in DataFrame is:

Column wise mean value are:
invoice      1004.000000
Quantity        4.000000
Price       18664.285714
dtype: float64

Column with minimum mean value is
Quantity
```

#8 Locate the 3 largest values in a DataFrame

[42]:
```
import pandas as pd
dic = {"Name": ["Rohit", "Mohit", "Deepak", "Aditya", "Akshita", "Yash"],
       "MarksInIp"  : [ 85,45,92,89,88,94]}
marks = pd.DataFrame(dic)
#Find the 3 largest value for marks in IP Column
print(marks.nlargest(3,["MarksInIp"]))
```

```
     Name  MarksInIp
5    Yash         94
2  Deepak         92
3  Aditya         89
```

Subtract the mean of a row from each element of the row in a DataFrame.

In [47]:
```
import pandas as pd
import pandas as pd
profit = {"TCS": {"Qtr1":2500, "Qtr2": 2000, "Qtr3" : 3000, "Qtr4":2500},
          "WIPRO": {"Qtr1":2800, "Qtr2": 2400, "Qtr3":3600, "Qtr4":2500},
          "L&T": {"Qtr1":2100, "Qtr2": 5700, "Qtr3":3500, "Qtr4":2100} }
df = pd.DataFrame(profit)
print(df)
print()
print('Mean of each row is:')
print(df.mean(axis=1))
print()
print("Dataframe after subtracting mean value of each row of that row is:")
print(df.sub(df.mean(axis=1), axis = 0))
```

```
       TCS  WIPRO   L&T
Qtr1  2500   2800  2100
Qtr2  2000   2400  5700
Qtr3  3000   3600  3500
Qtr4  2500   2500  2100

Mean of each row is:
Qtr1    2466.666667
Qtr2    3366.666667
Qtr3    3366.666667
Qtr4    2366.666667
dtype: float64

Dataframe after subtracting mean value of each row of that row is:
              TCS       WIPRO          L&T
Qtr1    33.333333  333.333333  -366.666667
Qtr2 -1366.666667 -966.666667  2333.333333
Qtr3  -366.666667  233.333333   133.333333
Qtr4   133.333333  133.333333  -266.666667
```

#10 Replace all negative value in a DataFrame with a 0

In [49]:
```
import pandas as pd
dic = {"data1" : [-5,-2,5,8,9,-6],
       "data2" : [2,4,10,15,-5,-8]
      }

df = pd.DataFrame(dic)
print(df)
print()
print("DataFrame after replacing negative values with 0 :")
df[df<0]=0
print(df)
```

```
   data1  data2
0     -5      2
1     -2      4
2      5     10
3      8     15
4      9     -5
5     -6     -8

DataFrame after replacing negative values with 0 :
   data1  data2
0      0      2
1      0      4
2      5     10
3      8     15
4      9      0
5      0      0
```

#11 Replace all missing values in a dataframe with 999

[50]:
```
import pandas as pd
import numpy as np
empData = {"empid" : [101,102,103,104,105,106],
           "eName": ["sachin","Yash", "Prashi", np.nan, "Ansh", "Chetna"],
          "Dob" : ["12-02-2005", "15-10-2004", "11-08-2003","1-02-2005","22-05-2005","13-07-2002"]}
df = pd.DataFrame(empData)
print(df)
df = df.fillna({"eName" : 999, "Dob": 999})
print()
print(df)
```

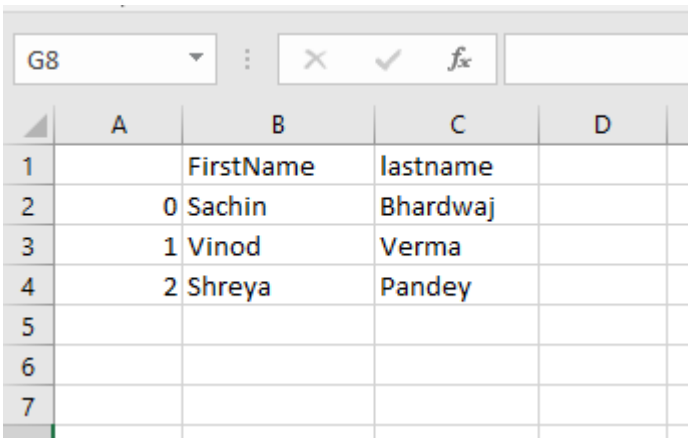
```
   empid   eName         Dob
0    101  sachin  12-02-2005
1    102    Yash  15-10-2004
2    103  Prashi  11-08-2003
3    104     NaN   1-02-2005
4    105    Ansh  22-05-2005
5    106  Chetna  13-07-2002

   empid   eName         Dob
0    101  sachin  12-02-2005
1    102    Yash  15-10-2004
2    103  Prashi  11-08-2003
3    104     999   1-02-2005
4    105    Ansh  22-05-2005
5    106  Chetna  13-07-2002
```

#12 importing and exporting data between pandas and CSV file

In [51]:
```
import pandas as pd
df = pd.read_csv("G:\ip csv file .csv")
print(df)
```

```
   Unnamed: 0 FirstName LastName
0           0    Mayank    Kumar
1           1    Prashi   Sharma
2           2      Yash    Gupta
3           3   Shubham      Rao
```

In [55]:
```
import pandas as pd
list = [{"FirstName" : "Sachin", "lastname" : "Bhardwaj"},
        {"FirstName" : "Vinod", "lastname" : "Verma"},
        {"FirstName" : "Shreya", "lastname" : "Pandey"}
       ]
df1 = pd.DataFrame(list)
#saving the datagframe
df1.to_csv("G:\ip csv file .csv")
```



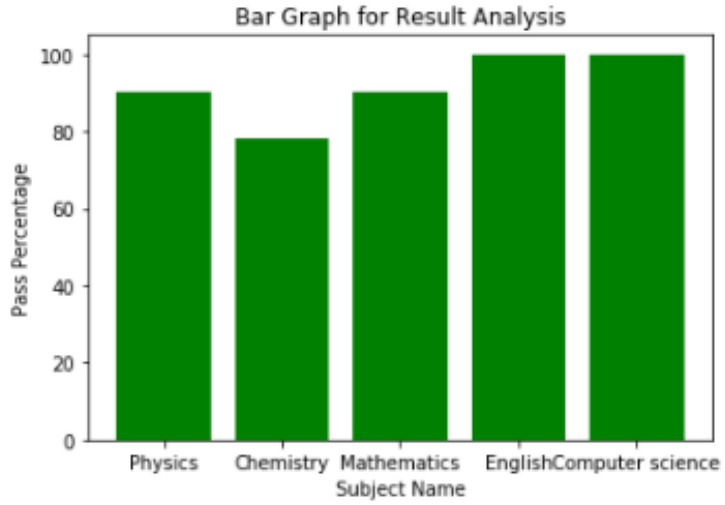
G8

	A	B	C	D
1		FirstName	lastname	
2	0	Sachin	Bhardwaj	
3	1	Vinod	Verma	
4	2	Shreya	Pandey	
5				
6				
7				

#13 importing and exporting data between pandas and Mysql database.

In [59]:
```
import pandas as pd
import mysql.connector

cnx = mysql.connector.connect(user='scott', password='password',
                              host='localhost')
```

]:
```
import matplotlib.pyplot as plt
Subject = ["Physics", "Chemistry", "Mathematics", "English", "Computer science"]
Percentage = [90,78,90,100,100]
plt.bar(Subject,Percentage, align = "center", color="green")
plt.xlabel("Subject Name")
plt.ylabel("Pass Percentage")
plt.title("Bar Graph for Result Analysis")
plt.show()
```



#15 For the data frame created above, analyze and plot appropriate charts with title and legend

In [77]:
```
import matplotlib.pyplot as plt
import numpy as np
s = ['1st', '2nd', '3rd']
per_sc = [95,89,77]
per_com = [90,93,75]
per_hum = [97,92,89]
x = np.arange(len(s))
plt.bar(x,per_sc,label="Science", width = 0.25, color = "green")
plt.bar(x+.25, per_com, label="Commerce", width = 0.25, color="red")
plt.bar(x+.50, per_hum, label="Humanities", width =0.25, color="gold")
plt.xticks(x,s)
plt.xlabel("Position")
plt.ylabel("Percentage")
plt.title("Bar graph for result analysis")
plt.legend()
plt.show()
```