ASSIGNMENT NO.1

Name-: More Pankaj Sampat

Roll No-: 054

Branch-: TE Comp

Subject-: COMPUTATIONAL STATISTICS

College-: PVG'S COET Pune.

COURSE-: AIML HONOUR Course

Problem Statement-: Compute Estimators of the main statistical measures like Mean, Variance, Standard Deviation, Covariance, Correlation and Standard error with respect to any example. Display graphically the distribution of samples.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(5)
x = np.random.randint(10,70,10)
y = np.random.randint(20,40,10)
x.sort()
y.sort()
```

```
x
```

```
array([18, 19, 24, 26, 45, 46, 48, 49, 57, 64])
```

```
y
```

```
array([27, 27, 32, 32, 33, 35, 36, 36, 36, 37])
```

```
def calc_mean(dataset):
    '''
    Def : Mean is defined as the arithmetic average of
    a population.

    Formula : (sum of obesrvations)/(No. of observations)
    '''
    return dataset.sum()/len(dataset)
```

```
def calc_variance(dataset,mean):
    '''
    Def : Variance is the degree of variation/spread
    in the dataset.
    Formula : 1) Σ((X - X_mean)^2) / n
    '''
    squared_diff = np.square(dataset-mean)
    return calc_mean(squared_diff)
```

```
def calc_SD(variance):
    '''
 Def : 1) Standard deviation is the amount of deviation
 of points around the mean.
 2) Variation but in terms of the actual dataset.
 Formula : √(variance)
    '''

    return np.sqrt(variance)
```

```
def calc_covariance(dataset1,dataset2):
    '''
 Def : Covariance measures the relationship trend
 between two sets of data.
 Formula : 1) Σ((X - X_mean)*(Y - Y_mean)) / n
    '''

    mean1 = calc_mean(dataset1)
    mean2 = calc_mean(dataset2)
    return np.sum(np.multiply(dataset1-mean1,dataset2-mean2))/len(dataset1)
```

```
def calc_correlation(dataset1,dataset2):
    '''
 Def : Covariance measures the relationship trend
 between two sets of data.
 Formula : 1) Σ((X - X_mean)*(Y - Y_mean)) / √(Σ(X - X_mean)^2*Σ(Y - Y_mean)^2)
 '''
    mean1 = calc_mean(dataset1)
    mean2 = calc_mean(dataset2)
    num = np.sum(np.multiply(dataset1-mean1,dataset2-mean2))
    de = np.multiply(np.sum(np.square(dataset1-mean1)),np.sum(np.square(dataset2-mean2)
    return num/np.sqrt(de)
```

```
def calc_SE(dataset,sd):
    '''
    Def : The standard error is a statistical term that
    easures the accuracy with which a sample
    distribution represents a population by using
    standard deviation.
    Formula : Standard_deviation / √(n)
    '''
    return sd/np.sqrt(len(dataset))
```

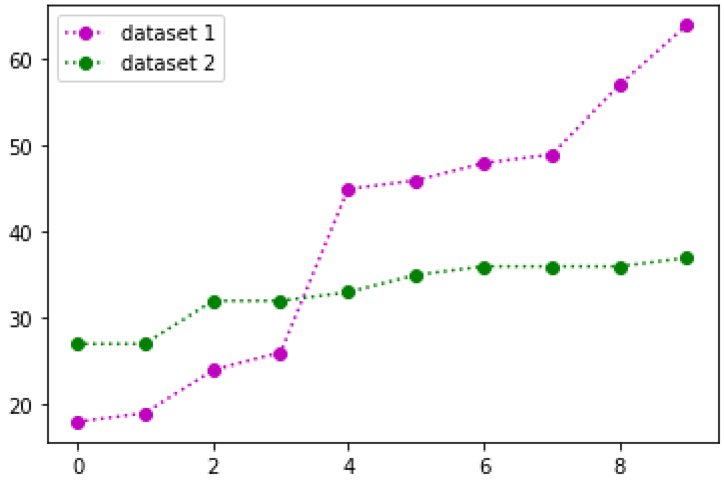
```
mean = calc_mean(x)
mean2 = calc_mean(y)
variance = calc_variance(x,mean)
S_D = calc_SD(variance)
covariance = calc_covariance(x,y)
correlation = calc_correlation(x,y)
S_E = calc_SE(x,S_D)

print(mean,mean2,variance,S_D,covariance,correlation,S_E)
```

```
39.6 33.1 244.64000000000001 15.640971836813723 49.84 0.9164339069491503 4.946109582
```

Dataset

```
plt.plot(x,"mo:",label="dataset 1")
plt.plot(y,"go:",label = "dataset 2")
plt.legend(loc="upper left")
```

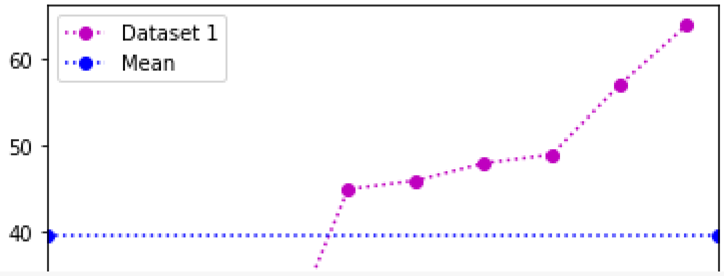
```
<matplotlib.legend.Legend at 0x7f5ec41bd650>
```



Mean

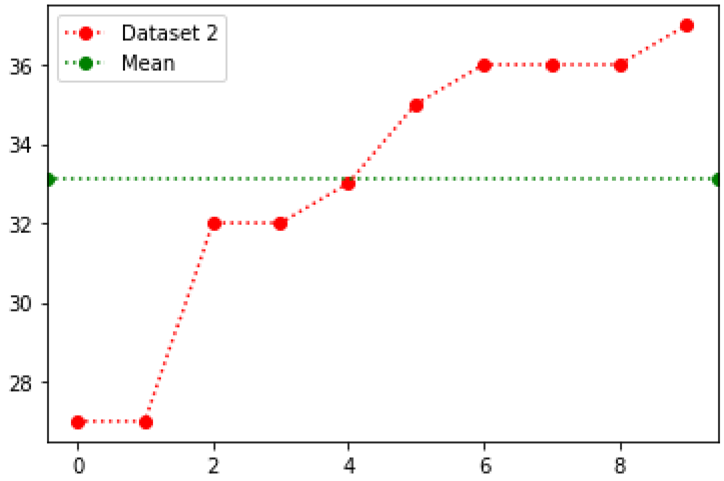
```
plt.plot(x,"mo:",label="Dataset 1")
plt.axhline(mean,color='b',marker= 'o', linestyle=':',label="Mean")
plt.legend(loc="upper left")
```

```
<matplotlib.legend.Legend at 0x7f5ec41db950>
```



```
plt.plot(y,"ro:",label="Dataset 2")
plt.axhline(mean2,color='g',marker= 'o', linestyle=':',label="Mean")
plt.legend(loc="upper left")
```

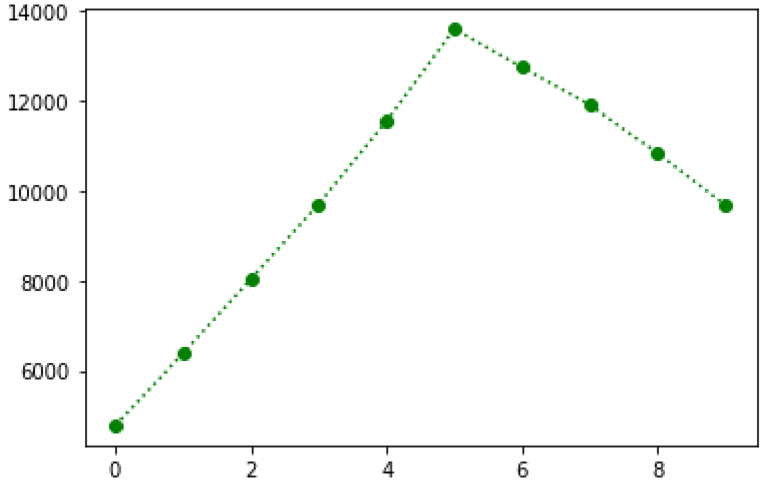
```
<matplotlib.legend.Legend at 0x7f5ec4119810>
```



Corelation

```
corr = np.correlate(x, y, "same")
plt.plot(list(corr),"go:",label = "Correlation")
```

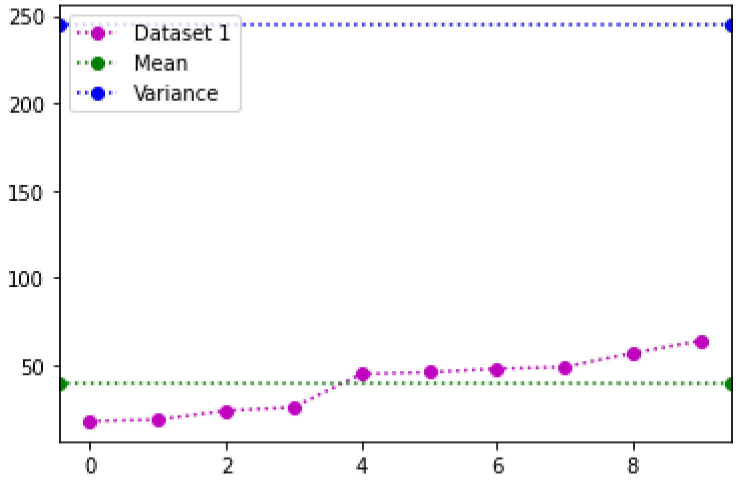
```
[<matplotlib.lines.Line2D at 0x7f5ec13a0a90>]
```



Variance

```
plt.plot(x,"mo:",label="Dataset 1")
plt.axhline(mean,color='g',marker= 'o', linestyle=':',label="Mean")
plt.axhline(variance,color='b',marker= 'o', linestyle=':',label="Variance")
plt.legend(loc="upper left")
```

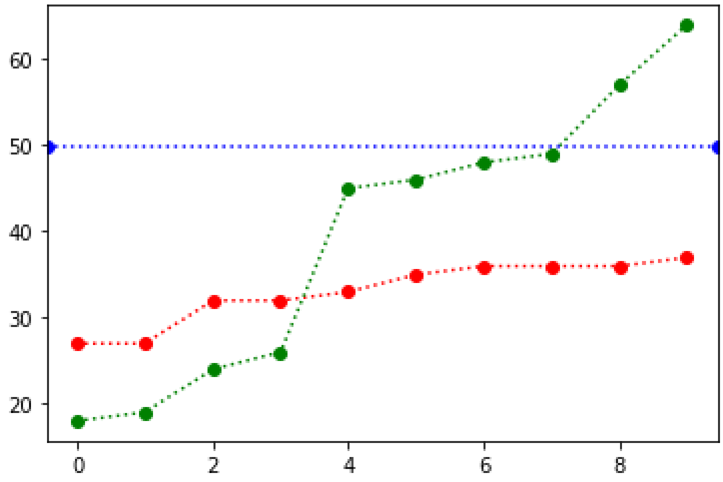
```
<matplotlib.legend.Legend at 0x7f5ec133e810>
```



Covariance

```
plt.plot(y,"ro:",label="Dataset 2")
plt.plot(x,"go:",label="Daraset 1")
plt.axhline(covariance,color='b',marker= 'o', linestyle=':',label="covariance")
```

```
<matplotlib.lines.Line2D at 0x7f5ec13583d0>
```


60
50
40
30
20
0
2
4
6
8