**Python function to get mean and standard deviation for gaussian list and worst values of subset lists**

In [55]:

**import** **random** **as** **rd**

*# Functiont to find mean of input list*

**def** mean(list):

acc=0

**for** i **in** list:

acc = acc+i

m=acc/len(list)

**return** m

*# Functiont to find standard deviation of input list*

**def** sDeviation(list):

acc=0

m=mean(list)

**for** i **in** list:

acc = acc +((m-i)\*\*2)

sd=(acc/(len(list)-1))\*\*(1/2)

**return** sd

*# Defining the function with optional value of 200*

**def** getGaussianMeanSD(m,sd,length=200):

**if** length%**2** == 1:

length = length-1

*# Getting the subsets*

myList = [rd.gauss(m,sd) **for** \_ **in** range(length)]

listFirstHalf = myList[:int(len(myList)/2)]

listSecondHalf = myList[int(len(myList)/2):]

listEvenPosition = myList[0::2]

*# Computing the mean and standard deviation for myList*

m\_myList = mean(myList)

sd\_myList = standardDeviation(myList)

*# Computing the worst mean for subsets*

m\_listFirstHalf = mean(listFirstHalf)

m\_listSecondHalf = mean(listSecondHalf)

m\_listEvenPosition = mean(listEvenPosition)

m\_worst= max([m\_listFirstHalf,m\_listSecondHalf,m\_listEvenPosition],key=**lambda** x:abs(x-m))

*# Computing the worst standard deviation for subsets*

sd\_listFirstHalf = sDeviation(listFirstHalf)

sd\_listSecondHalf = sDeviation(listSecondHalf)

sd\_listEvenPosition = sDeviation(listEvenPosition)

sd\_worst= max([sd\_listFirstHalf,sd\_listSecondHalf,sd\_listEvenPosition],key=**lambda** x:abs(x-m))

**return** (m\_myList,sd\_myList),(m\_worst,sd\_worst)

*#Real world data for our analysis*

lifeExpectancyList=[60,74,73,82,51,75,76,71,82,81,71,75,79,70,78,71,80,74,57,67]

m=mean(lifeExpectancyList)

sd=sDeviation(lifeExpectancyList)

print("Mean & SD for input data => (",m, sd,")")

**for** length **in** [20, 20\*\*2, 20\*\*3,20\*\*4, 20\*\*5]:

print("here the length of list is ",length)

(my\_mean,my\_sd),(m\_worst,sd\_worst)=getGaussianMeanSD(m, sd, length)

print("Mean & SD for entire list =>",my\_mean, my\_sd)

print("Worst Mean & SD for subset",m\_worst,sd\_worst)

Mean & SD for input data => ( 72.35 8.31153288794362 )

here the length of list is 20

Mean & SD for entire list => 71.06084821805725 5.416544458631902

Worst Mean & SD for subset 70.14170291421195 4.352404744485318

here the length of list is 400

Mean & SD for entire list => 72.75383047754194 7.86694213360691

Worst Mean & SD for subset 73.43954124364849 7.45889366974273

here the length of list is 8000

Mean & SD for entire list => 72.33041950557833 8.347081150825227

Worst Mean & SD for subset 72.31523815498207 8.318294745615205

here the length of list is 160000

Mean & SD for entire list => 72.34539798759798 8.329878797220784

Worst Mean & SD for subset 72.31432741332078 8.315404884620119

here the length of list is 3200000

Mean & SD for entire list => 72.35390528282333 8.308840387193552

Worst Mean & SD for subset 72.35733903344509 8.308533316517703

**Below trend was found when using the real world data:**

Life expectancy data for 20 countries is used as the real world data for this analysis. It is observed that the accuracy of mean and standard deviation values increases as we increase the length of list. The same is observed for 20, 400, 8000, 16000 & 3200000.