-:**Part 3**:-

PDF:- Probability density function

CDF:- Cumulative density function

PPF:- Percent point functions

**Question 3:**

Mercury makes a 2.4 lt V-6 engine, The Laser XRi, used in speedboats. The companies engineer believe that the engine delivers an average power of 220 horsepower and that the standard deviation of power delivered is 15 horsepower. A potential buyer intends to sample 100 engines(each engine to be run a single time). What is the probability that the sample mean will be less than 217 horsepower?.

**Solution**:

Here,

Population mean = 220 and,

SD = 15.

Applying central limit theorem,

N = 100 (>30)

Sample mean = population mean = 220

SD = 15/sqrt(N)

*import scipy.stats as sp*

*import numpy as num*

*#cdf(x, loc = mean, scale = std)*

*#pop\_mu = xbar (sample mean)*

*pop\_mu = 220*

*x = 217*

*pop\_sigma = 15*

*sample\_size = 100*

*std = pop\_sigma/num.sqrt(sample\_size)*

*#print("Sample std dev - ",std)*

*prob = sp.norm.cdf(x,pop\_mu,std)*

*print ("Probability of horse power less than 217-“,prob)*

O/P = 0.022750131948179195

**Question 4**:

Comcast, the computer services company, is planning to invest heavily in online television services. As part of the decision, the company wants to estimate the average no of online shows a family of four would watch per day. A random sample of n=100 families is obtained, and in this sample the average no of shows viewed per day is 6.5 and the population standard deviation is known to be 3.2. Construct a 95% confidence interval for the average no of online television shows watched by the entire population of families of four.

**Solution**:

N = 100

Sample Mean(Xbar) = 6.5

SD = 3.2

Now, we need interval estimate with 95% confidence.

95%

b

2.5% 2.5%

-Z 0 +Z

Now,

Mu1 = xbar + Z(SD/sqrt(N))

Mu2 = xbar - Z(SD/sqrt(N))

Z= PPF (with x=.025,loc = 0 and scale = 1)

import scipy.stats as st

import math as m

N = 100

xbar = 6.5

SD = 3.2

#z = st.norm.ppf(.025)

Mu1=xbar+st.norm.ppf(.025)\*(SD/m.sqrt(N))

Mu2=xbar-st.norm.ppf(.025)\*(SD/m.sqrt(N))

print (Mu2, ' to ', Mu1)

***Another approach***:-

*import scipy.stats as sp*

*import math as num*

*#ppf - opposite of cdf*

*#ppf(area\_under\_curve, loc = mean, scale = std)*

*sample\_mean = 6.5*

*#2.5% on either sides for 95% confidence interval*

*area1 = .025*

*area2 = .974*

*pop\_sigma = 3.2*

*sample\_size = 100*

*sample\_std = pop\_sigma/num.sqrt(sample\_size)*

*point1=sp.norm.ppf(area1,sample\_mean,sample\_std)*

*point2=sp.norm.ppf(area2,sample\_mean,sample\_std)*

*print (point1)*

*print (point2)*

O/P

5.872811524947182

7.121802800353621

**Question 5**:

A stock market analyst wants to estimate the average return on a certain stock. A random sample of 15 days yields an average (annualized) return of Xbar=10.37% and a standard deviation of s=3.5%. Assuming a normal population of returns, give a 95% confidence interval for the average return on this stock.

**Solution**: