|  |  |  |  |
| --- | --- | --- | --- |
| VIIT Pune (@viit_pune) | Twitter | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information Technology  **Department of**  **Artificial Intelligence and Data Science** | | |
| Student Name: Siddhesh Dilip Khairnar | | | |
| Class: SY | Division: B | | Roll No: 272028 |
| Semester: 4th | | Academic Year: 2022 - 23 | |
| Subject Name & Code: Probability and Statistics (ES22201AD) | | | |
| Title of Assignment: Continuous Probability Distributions | | | |
| Date of Performance: 16/03/2023 | | Date of Submission: 10/04/2023 | |

Aim: To calculate continuous Probability Distributions in R.

Software Requirements:

R Studio or any other editor capable of executing R Scripts.

Background Information:

# The Normal Distribution Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd.

# Usage

dnorm(x, mean = 0, sd = 1, log = FALSE)

pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE) qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE) rnorm(n, mean = 0, sd = 1)

Arguments

x, q = vector of quantiles. p = vector of probabilities.

n = number of observations. If length(n) > 1, the length is taken to be the number required.

mean = vector of means.

sd = vector of standard deviations.

log, log.p = logical; if TRUE, probabilities p are given as log(p).

lower.tail = logical; if TRUE (default), probabilities are P[X ≤ x] otherwise, P[X >

x].

# Details

If mean or sd are not specified they assume the default values of 0 and 1, respectively.

The normal distribution has density.

f(x) = 1/ (√(2 π) σ) e^-((x - μ)^2/(2 σ^2))

where μ is the mean of the distribution and σ the standard deviation.

# Value

dnorm gives the density, pnorm gives the distribution function, qnorm gives the quantile function, and rnorm generates random deviates.

7 | P a g e

The length of the result is determined by n for rnorm and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than n are recycled to the length of the result. Only the first elements of the logical arguments are used.

For sd = 0 this gives the limit as sd decreases to 0, a point mass at mu. sd < 0 is an error and returns NaN.

Code:

#this function gives height of probability distrubution #i.e (y values for corresponding x values) at each point #for given mean and standard deviation

x=seq(-20,20,by=.1) #starting point, ending point and the difference between values

print(x) y=dnorm(x,mean=5.0,sd=2.0)

plot (x,y,main="Normal Distribution",col="blue")

x=seq(-20,20, by=.1) #starting point, ending point and the difference between values

print(x) y=dnorm(x,mean=10.0,sd=5.0)

plot(x,y,main="Normal Distribution",col="blue") #pnorm() function

x=seq(-20,20,by=.1) #starting point, ending point and the difference between values

print(x)

y=pnorm(x,mean=5.0,sd=1.0)

plot(x,y,main="Normal Distribution",col="blue")

#qnorm() function is the invese of the pnorm() function

#It takes the probability value and gives output

#which corresponds to the probability value

#It is useful in finding percentiles of normal distribution

y=seq(0,1,by=0.02) #area under graph ranges from 0 to 1.

x=qnorm(y,mean=2,sd=1)

plot(x,y,main="qnorm()",col="blue")

#rnorm() function in r programming is used to

#generate a vector of random numbers which are normally distributed

y=rnorm(50) #by default mean=0 and standard deviation=1

plot(y,main="Normal Distribution",col="darkorange")

y=rnorm(50,10,1) #no. of random values, mean and then standard deviation

plot(y,main="Normal Distribution",col="darkorange")

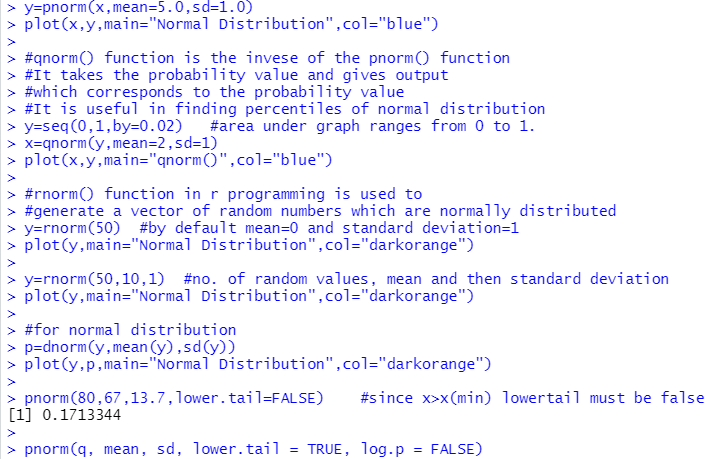
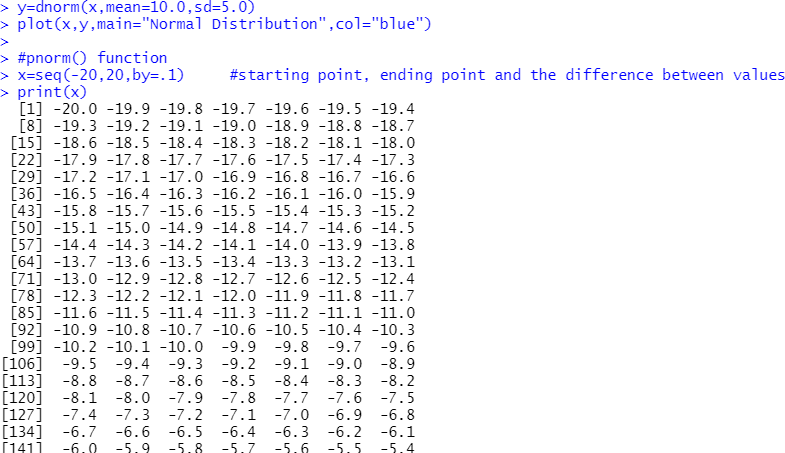
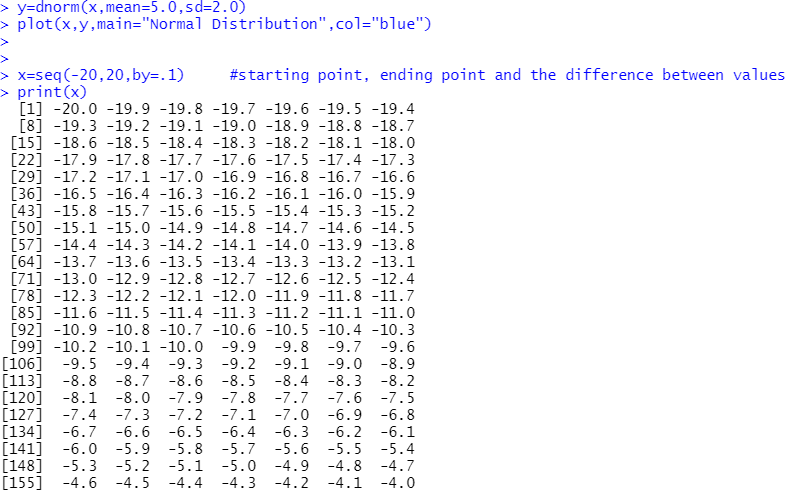
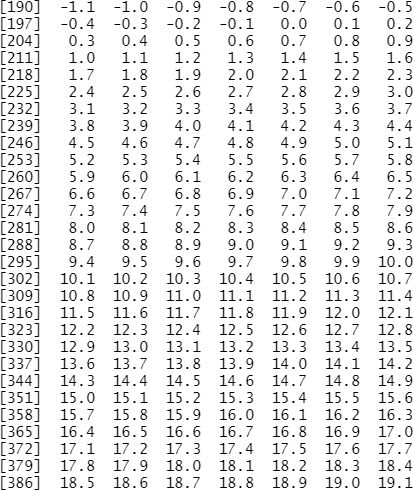
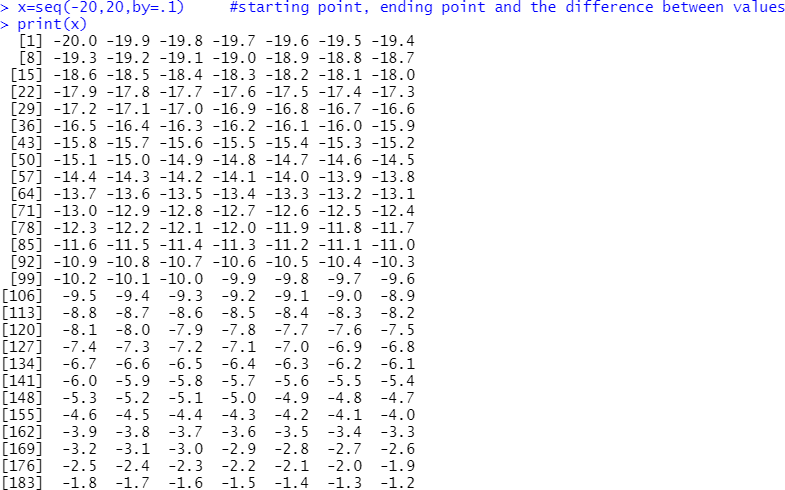
#for normal distribution

p=dnorm(y,mean(y),sd(y))

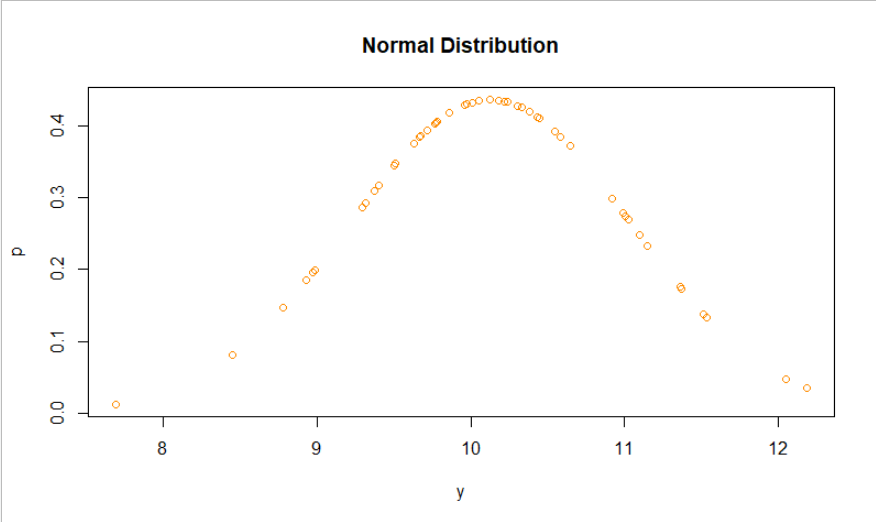
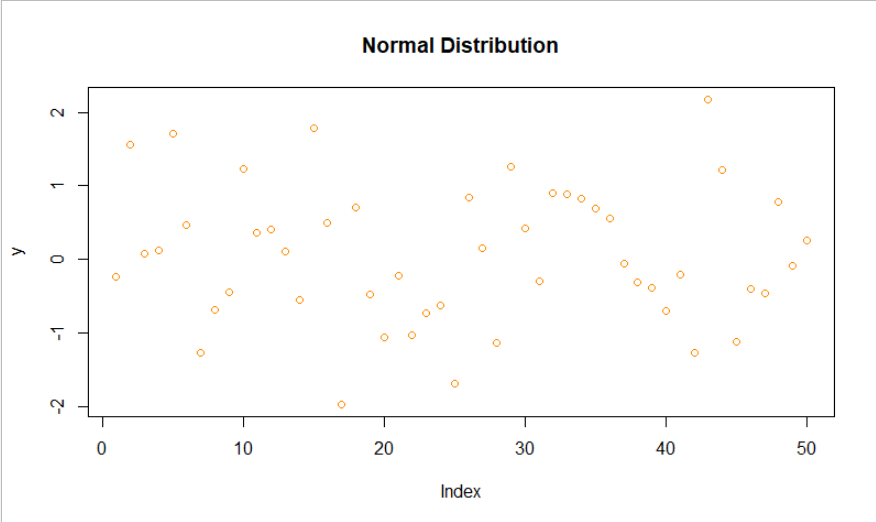
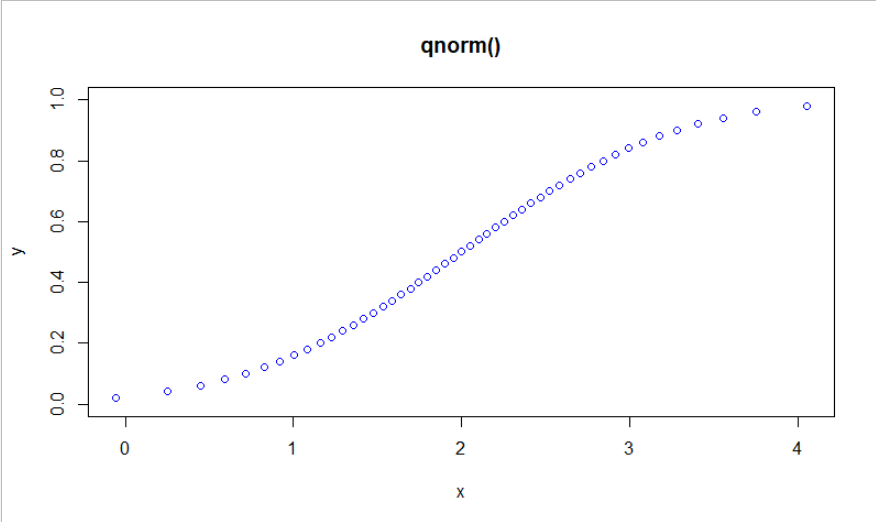
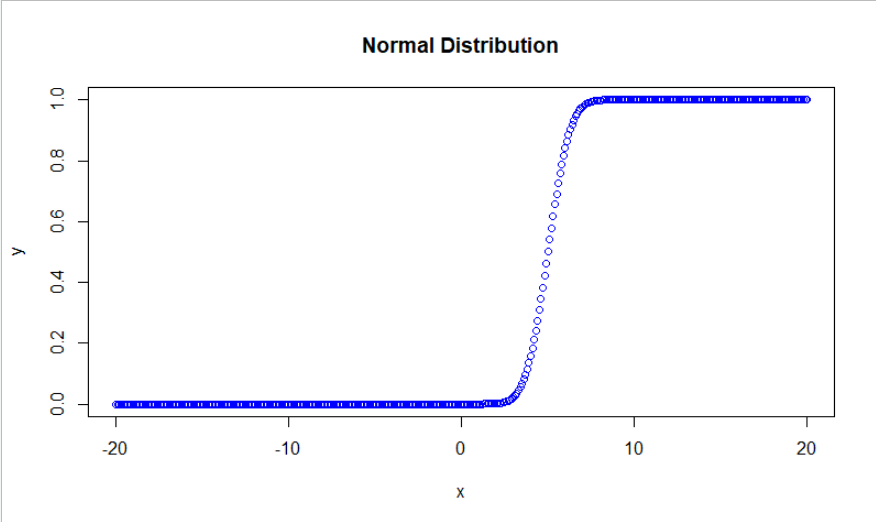
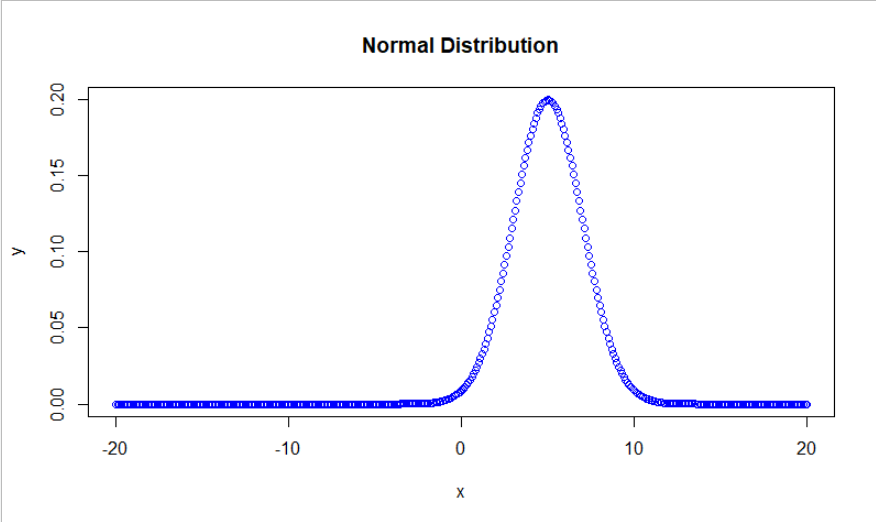
plot(y,p,main="Normal Distribution",col="darkorange")

pnorm(80,67,13.7,lower.tail=FALSE) #since x>x(min) lowertail must be false

Result:



Visualization:



Conclusion: Hence in this assignment we’ve learned different functions To Generate Normal Distribution in R which are dnorm, qnorm, pnorm, rnorm.