	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.

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:

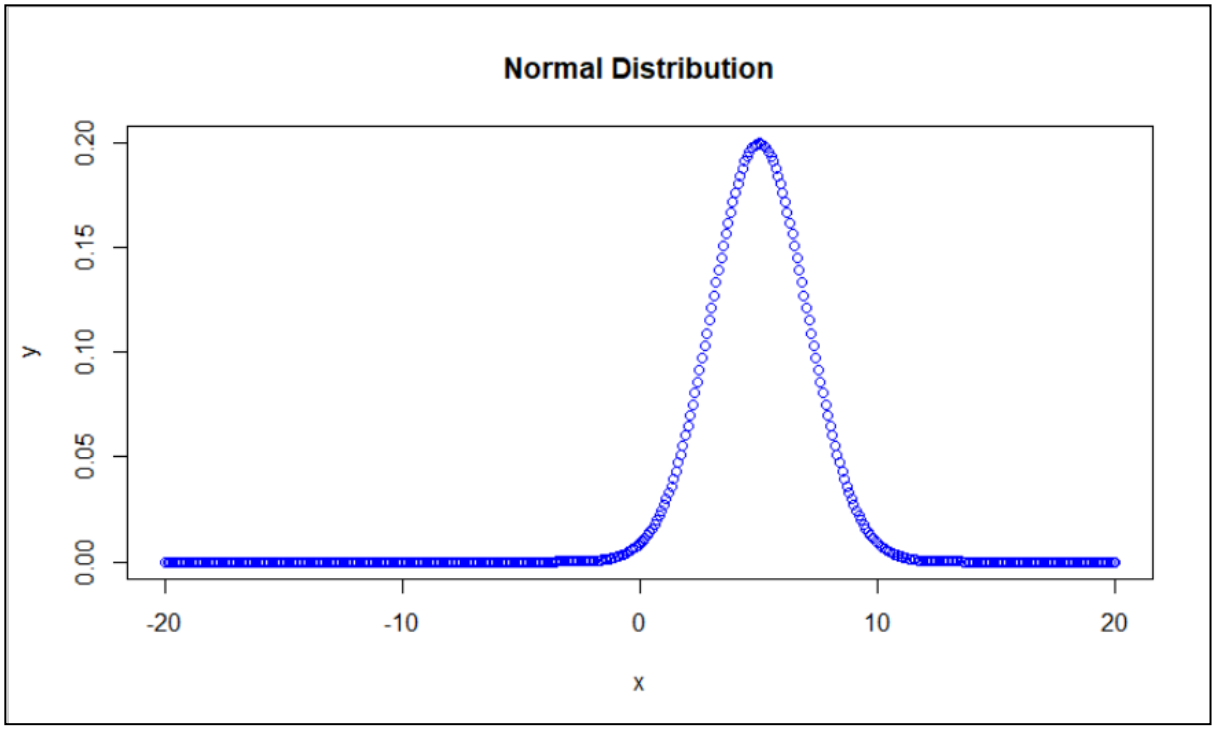
```
> x=seq(-20,20,by=.1)     #starting point, ending point and the difference between values
> print(x)
  [1] -20.0 -19.9 -19.8 -19.7 -19.6 -19.5 -19.4
  [8] -19.3 -19.2 -19.1 -19.0 -18.9 -18.8 -18.7
 [15] -18.6 -18.5 -18.4 -18.3 -18.2 -18.1 -18.0
 [22] -17.9 -17.8 -17.7 -17.6 -17.5 -17.4 -17.3
 [29] -17.2 -17.1 -17.0 -16.9 -16.8 -16.7 -16.6
 [36] -16.5 -16.4 -16.3 -16.2 -16.1 -16.0 -15.9
 [43] -15.8 -15.7 -15.6 -15.5 -15.4 -15.3 -15.2
 [50] -15.1 -15.0 -14.9 -14.8 -14.7 -14.6 -14.5
 [57] -14.4 -14.3 -14.2 -14.1 -14.0 -13.9 -13.8
 [64] -13.7 -13.6 -13.5 -13.4 -13.3 -13.2 -13.1
 [71] -13.0 -12.9 -12.8 -12.7 -12.6 -12.5 -12.4
 [78] -12.3 -12.2 -12.1 -12.0 -11.9 -11.8 -11.7
 [85] -11.6 -11.5 -11.4 -11.3 -11.2 -11.1 -11.0
 [92] -10.9 -10.8 -10.7 -10.6 -10.5 -10.4 -10.3
 [99] -10.2 -10.1 -10.0  -9.9  -9.8  -9.7  -9.6
[106]  -9.5  -9.4  -9.3  -9.2  -9.1  -9.0  -8.9
[113]  -8.8  -8.7  -8.6  -8.5  -8.4  -8.3  -8.2
[120]  -8.1  -8.0  -7.9  -7.8  -7.7  -7.6  -7.5
[127]  -7.4  -7.3  -7.2  -7.1  -7.0  -6.9  -6.8
[134]  -6.7  -6.6  -6.5  -6.4  -6.3  -6.2  -6.1
[141]  -6.0  -5.9  -5.8  -5.7  -5.6  -5.5  -5.4
[148]  -5.3  -5.2  -5.1  -5.0  -4.9  -4.8  -4.7
[155]  -4.6  -4.5  -4.4  -4.3  -4.2  -4.1  -4.0
[162]  -3.9  -3.8  -3.7  -3.6  -3.5  -3.4  -3.3
[169]  -3.2  -3.1  -3.0  -2.9  -2.8  -2.7  -2.6
[176]  -2.5  -2.4  -2.3  -2.2  -2.1  -2.0  -1.9
[183]  -1.8  -1.7  -1.6  -1.5  -1.4  -1.3  -1.2
[190]  -1.1  -1.0  -0.9  -0.8  -0.7  -0.6  -0.5
[197]  -0.4  -0.3  -0.2  -0.1   0.0   0.1   0.2
[204]   0.3   0.4   0.5   0.6   0.7   0.8   0.9
[211]   1.0   1.1   1.2   1.3   1.4   1.5   1.6
[218]   1.7   1.8   1.9   2.0   2.1   2.2   2.3
[225]   2.4   2.5   2.6   2.7   2.8   2.9   3.0
[232]   3.1   3.2   3.3   3.4   3.5   3.6   3.7
[239]   3.8   3.9   4.0   4.1   4.2   4.3   4.4
[246]   4.5   4.6   4.7   4.8   4.9   5.0   5.1
[253]   5.2   5.3   5.4   5.5   5.6   5.7   5.8
[260]   5.9   6.0   6.1   6.2   6.3   6.4   6.5
[267]   6.6   6.7   6.8   6.9   7.0   7.1   7.2
[274]   7.3   7.4   7.5   7.6   7.7   7.8   7.9
[281]   8.0   8.1   8.2   8.3   8.4   8.5   8.6
[288]   8.7   8.8   8.9   9.0   9.1   9.2   9.3
[295]   9.4   9.5   9.6   9.7   9.8   9.9  10.0
[302]  10.1  10.2  10.3  10.4  10.5  10.6  10.7
[309]  10.8  10.9  11.0  11.1  11.2  11.3  11.4
[316]  11.5  11.6  11.7  11.8  11.9  12.0  12.1
[323]  12.2  12.3  12.4  12.5  12.6  12.7  12.8
[330]  12.9  13.0  13.1  13.2  13.3  13.4  13.5
[337]  13.6  13.7  13.8  13.9  14.0  14.1  14.2
[344]  14.3  14.4  14.5  14.6  14.7  14.8  14.9
[351]  15.0  15.1  15.2  15.3  15.4  15.5  15.6
[358]  15.7  15.8  15.9  16.0  16.1  16.2  16.3
[365]  16.4  16.5  16.6  16.7  16.8  16.9  17.0
[372]  17.1  17.2  17.3  17.4  17.5  17.6  17.7
[379]  17.8  17.9  18.0  18.1  18.2  18.3  18.4
[386]  18.5  18.6  18.7  18.8  18.9  19.0  19.1
```

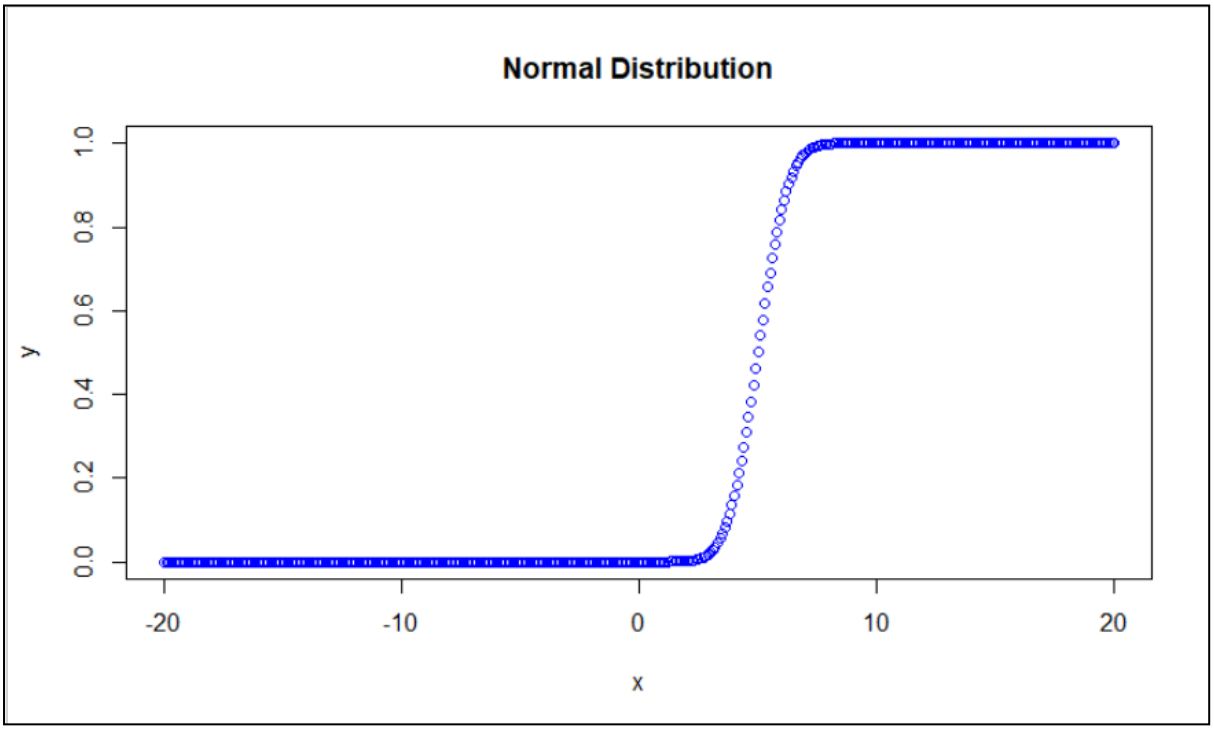
```
> y=dnorm(x,mean=5.0,sd=2.0)
> plot(x,y,main="Normal Distribution",col="blue")
>
>
> x=seq(-20,20,by=.1)     #starting point, ending point and the difference between values
> print(x)
  [1] -20.0 -19.9 -19.8 -19.7 -19.6 -19.5 -19.4
  [8] -19.3 -19.2 -19.1 -19.0 -18.9 -18.8 -18.7
 [15] -18.6 -18.5 -18.4 -18.3 -18.2 -18.1 -18.0
 [22] -17.9 -17.8 -17.7 -17.6 -17.5 -17.4 -17.3
 [29] -17.2 -17.1 -17.0 -16.9 -16.8 -16.7 -16.6
 [36] -16.5 -16.4 -16.3 -16.2 -16.1 -16.0 -15.9
 [43] -15.8 -15.7 -15.6 -15.5 -15.4 -15.3 -15.2
 [50] -15.1 -15.0 -14.9 -14.8 -14.7 -14.6 -14.5
 [57] -14.4 -14.3 -14.2 -14.1 -14.0 -13.9 -13.8
 [64] -13.7 -13.6 -13.5 -13.4 -13.3 -13.2 -13.1
 [71] -13.0 -12.9 -12.8 -12.7 -12.6 -12.5 -12.4
 [78] -12.3 -12.2 -12.1 -12.0 -11.9 -11.8 -11.7
 [85] -11.6 -11.5 -11.4 -11.3 -11.2 -11.1 -11.0
 [92] -10.9 -10.8 -10.7 -10.6 -10.5 -10.4 -10.3
 [99] -10.2 -10.1 -10.0  -9.9  -9.8  -9.7  -9.6
[106]  -9.5  -9.4  -9.3  -9.2  -9.1  -9.0  -8.9
[113]  -8.8  -8.7  -8.6  -8.5  -8.4  -8.3  -8.2
[120]  -8.1  -8.0  -7.9  -7.8  -7.7  -7.6  -7.5
[127]  -7.4  -7.3  -7.2  -7.1  -7.0  -6.9  -6.8
[134]  -6.7  -6.6  -6.5  -6.4  -6.3  -6.2  -6.1
[141]  -6.0  -5.9  -5.8  -5.7  -5.6  -5.5  -5.4
[148]  -5.3  -5.2  -5.1  -5.0  -4.9  -4.8  -4.7
[155]  -4.6  -4.5  -4.4  -4.3  -4.2  -4.1  -4.0
```

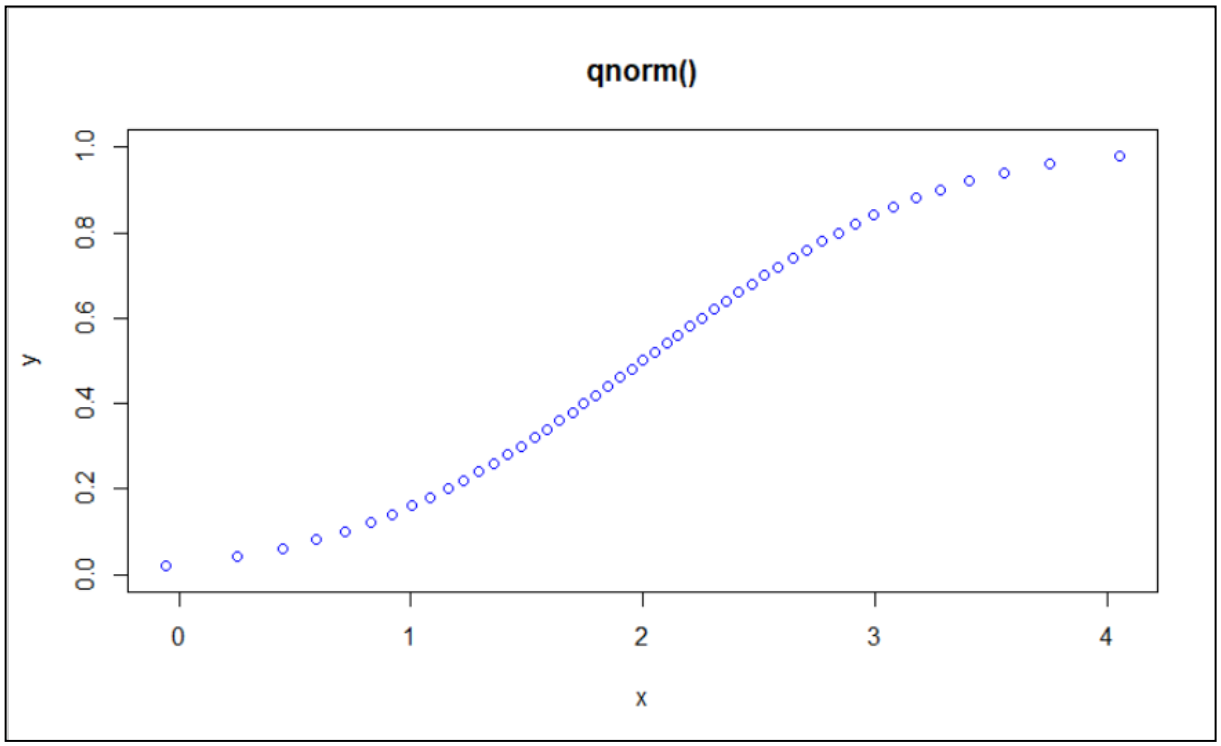
```
> y=dnorm(x,mean=10.0,sd=5.0)
> plot(x,y,main="Normal Distribution",col="blue")
>
> #pnorm() function
> x=seq(-20,20,by=.1)     #starting point, ending point and the difference between values
> print(x)
  [1] -20.0 -19.9 -19.8 -19.7 -19.6 -19.5 -19.4
  [8] -19.3 -19.2 -19.1 -19.0 -18.9 -18.8 -18.7
 [15] -18.6 -18.5 -18.4 -18.3 -18.2 -18.1 -18.0
 [22] -17.9 -17.8 -17.7 -17.6 -17.5 -17.4 -17.3
 [29] -17.2 -17.1 -17.0 -16.9 -16.8 -16.7 -16.6
 [36] -16.5 -16.4 -16.3 -16.2 -16.1 -16.0 -15.9
 [43] -15.8 -15.7 -15.6 -15.5 -15.4 -15.3 -15.2
 [50] -15.1 -15.0 -14.9 -14.8 -14.7 -14.6 -14.5
 [57] -14.4 -14.3 -14.2 -14.1 -14.0 -13.9 -13.8
 [64] -13.7 -13.6 -13.5 -13.4 -13.3 -13.2 -13.1
 [71] -13.0 -12.9 -12.8 -12.7 -12.6 -12.5 -12.4
 [78] -12.3 -12.2 -12.1 -12.0 -11.9 -11.8 -11.7
 [85] -11.6 -11.5 -11.4 -11.3 -11.2 -11.1 -11.0
 [92] -10.9 -10.8 -10.7 -10.6 -10.5 -10.4 -10.3
 [99] -10.2 -10.1 -10.0  -9.9  -9.8  -9.7  -9.6
[106]  -9.5  -9.4  -9.3  -9.2  -9.1  -9.0  -8.9
[113]  -8.8  -8.7  -8.6  -8.5  -8.4  -8.3  -8.2
[120]  -8.1  -8.0  -7.9  -7.8  -7.7  -7.6  -7.5
[127]  -7.4  -7.3  -7.2  -7.1  -7.0  -6.9  -6.8
[134]  -6.7  -6.6  -6.5  -6.4  -6.3  -6.2  -6.1
[141]  -6.0  -5.9  -5.8  -5.7  -5.6  -5.5  -5.4
```

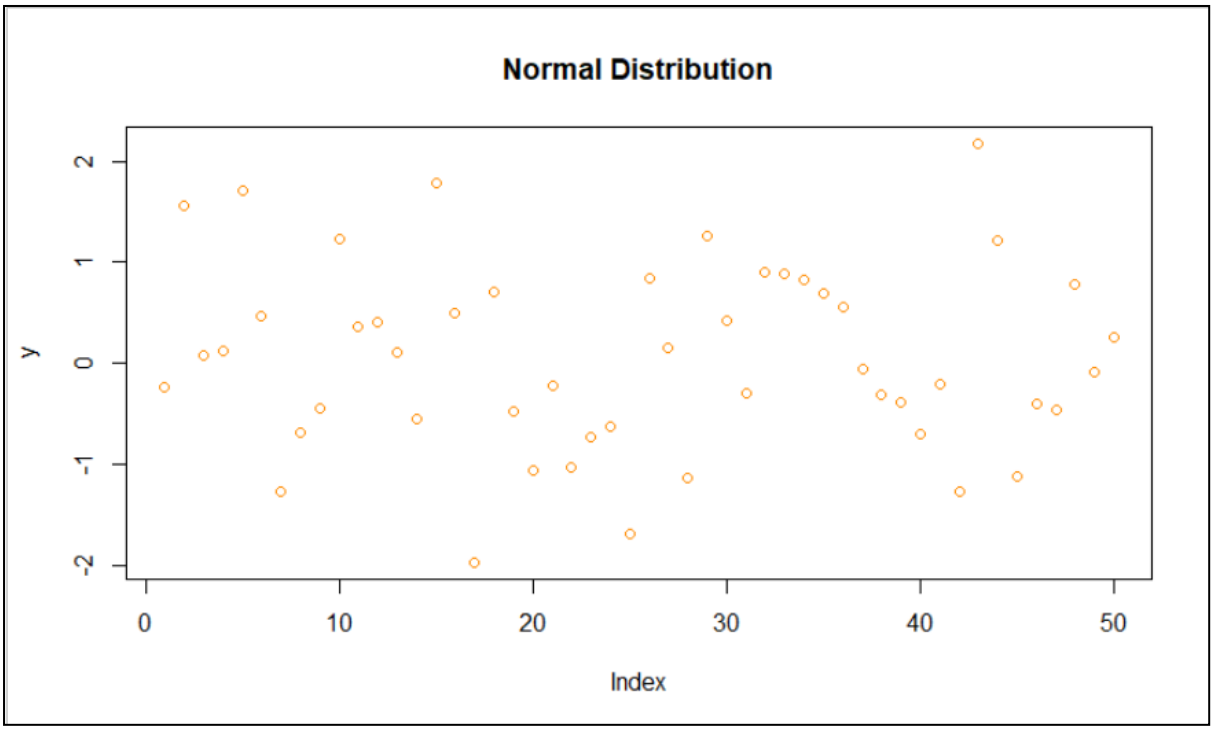
```
> y=pnorm(x,mean=5.0,sd=1.0)
> plot(x,y,main="Normal Distribution",col="blue")
>
> #qnorm() function is the invese of the pnorm() function
> #It takes the probability value and gives output
> #which corresponds to the probability value
> #It is useful in finding percentiles of normal distribution
> 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
[1] 0.1713344
>
> pnorm(q, mean, sd, lower.tail = TRUE, log.p = FALSE)
```

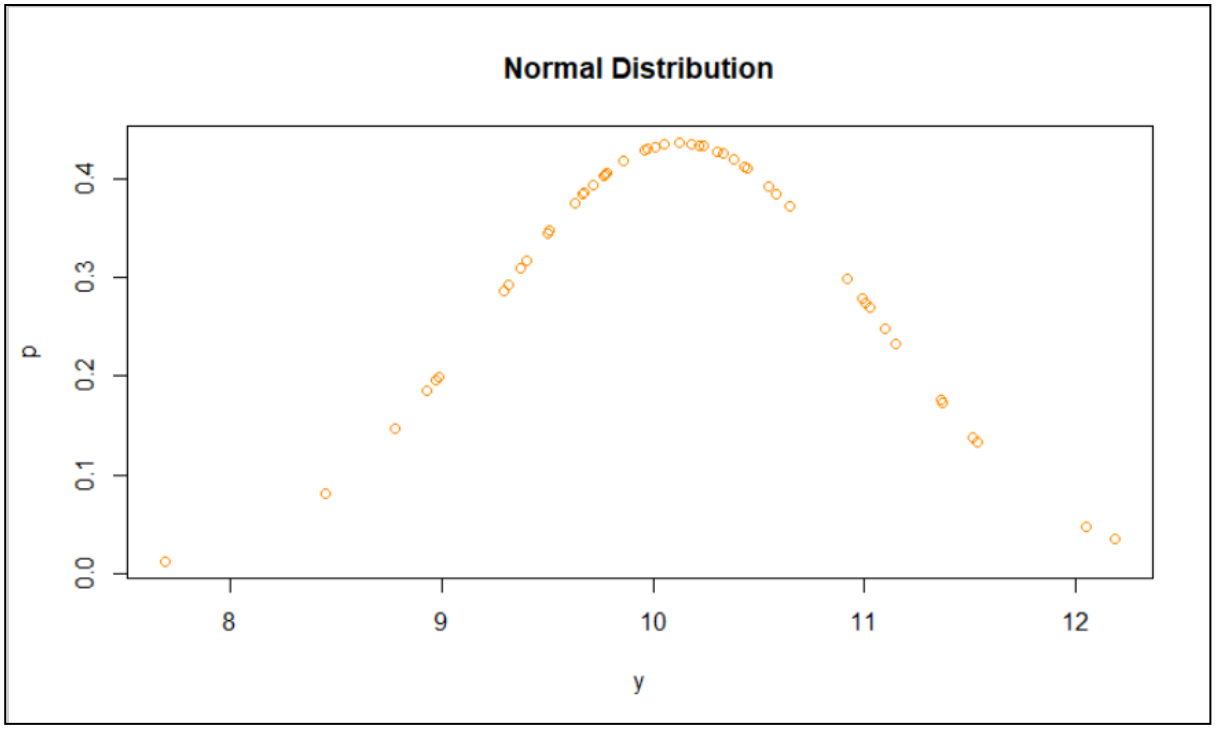
Visualization:




Normal Distribution
y
0.0
0.2
0.4
0.6
0.8
1.0
-20
-10
0
10
20
x


qnorm()
y
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
x





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