**Binomial Distribution in R**

**N= 20 trials, p =1/6=probability of success**

**>help(dbinom)**

**>?dbinom**

**dbinom command is used to find values for the probability density function of x ,of f(x).** **This function gives the probability density distribution at each point.**

**# x=3**

**>dbinom(x=3,size=20,p=1/6)**

**[1] 0.2378866**

**>dbinom(x=0:3,size = 20,prob=1/6)**

**[1] 0.02608405 0.10433621 0.19823881 0.23788657**

**>sum(dbinom(x=0:3,size = 20,prob = 1/6))**

**[1] 0.5665456**

**pbinom command gives us values for the probability distribution function x, f(x).** **This function gives the cumulative probability of an event. It is a single value representing the probability.**

**#p(x<=3)**

**> pbinom(q=3,size=20,prob = 1/6,lower.tail = T)**

**[1] 0.5665456**

**rbinom command to take a random sample from a binomial distribution.** **This function generates required number of random values of given probability from a given sample.**

**Find 8 random values from a sample of 150 with probability of 0.4.**

**x <- rbinom(8,150,.4)**

**> print(x)**

**[1] 56 50 65 45 54 55 65 63**

**qbinom command to find the quantiles for a binomial distribution : This function takes the probability value and gives a number whose cumulative value matches the probability value.**

**How many heads will have a probability of 0.25 will come out when a coin**

**# is tossed 51 times.**

**y <- qbinom(0.25,51,1/2)**

**> y**

**[1] 23**

**Poison Distribution in R**

**>help(dpois)**

**?dpois**

**dpois() : This function is used for illustration of Poisson density in an R plot. The function dpois() calculates the probability of a random variable that is available within a certain range.**

**Syntax:**

**dpois(k, lambda, log)**

where,

K: number of successful events happened in an interval

\lambda: mean per interval

log: If TRUE then the function returns probability in form of log

ppois()

This function is used for the illustration of cumulative probability function in an R plot. The function ppois() calculates the probability of a random variable that will be equal to or less than a number.

syntax:

ppois(q, lambda, lower.tail, log)

K: number of successful events happened in an interval

\lambda: mean per interval

lower.tail: If TRUE then left tail is considered otherwise if the FALSE right tail is considered

log: If TRUE then the function returns probability in form of log

rpois()

The function rpois() is used for generating random numbers from a given Poisson’s distribution.

q: number of randon numbers needed

\lambda: mean per interval

qpois()

The function qpois() is used for generating quantile of a given Poisson’s distribution. In probability, quantiles are marked points that divide the graph of a probability distribution into intervals (continuous ) which have equal probabilities.

Syntax:

qpois(q, lambda, lower.tail, log)

K: number of successful events happened in an interval

lambda: mean per interval

lower.tail: If TRUE then left tail is considered otherwise if the FALSE right tail is considered

log: If TRUE then the function returns probability in form of log

# P(X=4)

> dpois(x=2,lambda=5)

[1] 0.08422434

> dpois(x=0,lambda=5)

[1] 0.006737947

> dpois(x=0:5,lambda = 5)

[1] 0.006737947 0.033689735 0.084224337 0.140373896 0.175467370 0.175467370

# P(X<=4)

> sum(dpois(x=0:4,lambda = 5))

[1] 0.4404933

# P(X<=4)

ppois(q=4,lambda=5,lower.tail = T)

[1] 0.4404933

#P(X>=12)

> ppois(q=12,lambda=5,lower.tail = F)

[1] 0.002018852

# rpois command : to take a random sample from a poison distribution.

rpois(10,lambda=5)

[1] 7 3 4 4 5 5 5 7 6 5

qpois(p=0.3,lambda = 7,lower.tail = T)

**Normal Distribution in R**

help(pnorm)

> ?pnorm

Mean =75, Standard Deviation = 5

pnorm(q=70,mean=75,sd=5,lower.tail = T)

[1] 0.1586553

pnorm(q=85,mean=75,sd=5,lower.tail = F)

[1] 0.02275013

> P(Z>=1)

pnorm(q=1,mean=0,sd=1,lower.tail = F)

[1] 0.1586553

Find Q1

qnorm(p=0.25,mean=75,sd=5,lower.tail = T)

[1] 71.62755

>x<-seq(from=55,to=95,by=0.25)

> x

> dens<-dnorm(x,mean=75,sd=5)

>plot(x,dens)

>plot(x,dens,type = "l")

>plot(x,dens,type = "l",main = "x~normal:mean=75,sd=5",xlab="x",ylab = "probability density",las=1)

> abline(v=75)

>random<-rnorm(n=40,mean=75,sd=5)

> random

>hist(random)

**Monte Carlo Simulation in R**

1. **Generating Random numbers**

Here we simulate standard Normal random numbers with mean 0 and standard deviation 1.

## Simulate standard Normal random numbers

x <- rnorm(10)

> x

We can modify the default parameters to simulate numbers with mean 20 and standard deviation 2.

x <- rnorm(10, 20, 2)

> x

>summary(x)

If you wanted to know what was the probability of a random Normal variable of being less than, say, 2, you could use the pnorm() function to do that calculation.

> pnorm(2)

**2. Setting the random number seed**

When simulating any random numbers it is essential to set the random number seed. Setting the random number seed with set.seed() ensures reproducibility of the sequence of random numbers.

For example, I can generate 5 Normal random numbers with rnorm().

> set.seed(1)

> rnorm(5)

[1] -0.6264538 0.1836433 -0.8356286 1.5952808 0.3295078

Note that if I call rnorm() again I will of course get a different set of 5 random numbers.

> rnorm(5)

If I want to reproduce the original set of random numbers, I can just reset the seed with set.seed().

> set.seed(1)

> rnorm(5) ## Same as before

[1] -0.6264538 0.1836433 -0.8356286 1.5952808 0.3295078

1. **Random Sampling**

The sample() function draws randomly from a specified set of (scalar) objects allowing you to sample from arbitrary distributions of numbers.

>set.seed(1)

> sample(1:10, 4)

[1] 9 4 7 1

>sample(letters, 5)

[1] "r" "s" "a" "u" "w"

>sample(1:10)

[1] 10 6 9 2 1 5 8 4 3 7

sample(1:10)

[1] 10 6 9 2 1 5 8 4 3 7

> sample(1:10)

[1] 5 10 2 8 6 1 4 3 9 7

>sample(1:10, replace = TRUE)

[1] 3 6 10 10 6 4 4 10 9 7

Here’s how you can sample rows from a data frame.

>library(datasets)

> data(airquality)

> head(airquality)

Ozone Solar.R Wind Temp Month Day

1 41 190 7.4 67 5 1

2 36 118 8.0 72 5 2

3 12 149 12.6 74 5 3

4 18 313 11.5 62 5 4

5 NA NA 14.3 56 5 5

6 28 NA 14.9 66 5 6

>set.seed(20)

> idx <- seq\_len(nrow(airquality))

> samp <- sample(idx, 6)

> airquality[samp, ]

Ozone Solar.R Wind Temp Month Day

107 NA 64 11.5 79 8 15

120 76 203 9.7 97 8 28

130 20 252 10.9 80 9 7

98 66 NA 4.6 87 8 6

29 45 252 14.9 81 5 29

45 NA 332 13.8 80 6 14

**Example:**

**>coin=c("Head","Tail")**

**> set.seed(1)**

**> y=sample(coin,6,replace = T)**

**>y**

**[1] "Head" "Tail" "Head" "Head" "Tail" "Head"**

**>length(y[y=="Head"])**

**>a=replicate(100,length(sample(coin,6,replace = T)[y=="Head"]))**

**> a**

**>mean(a)**