**11) Write R program to find the following Distribution with the sample data and visualize the linear regression graphically.**

**A) Normal Distribution**

**B) Binomial Distribution**

# R - Normal Distribution

**a) Normal Distribution**

In a random collection of data from independent sources, it is generally observed that the distribution of data is normal. Which means, on plotting a graph with the value of the variable in the horizontal axis and the count of the values in the vertical axis we get a bell shape curve. The center of the curve represents the mean of the data set. In the graph, fifty percent of values lie to the left of the mean and the other fifty percent lie to the right of the graph. This is referred as normal distribution in statistics.

**R has four in built functions to generate normal distribution**

dnorm(x, mean, sd)

pnorm(x, mean, sd)

qnorm(p, mean, sd)

rnorm(n, mean, sd)

Following is the description of the parameters used in above functions −

x is a vector of numbers.

p is a vector of probabilities.

n is number of observations(sample size).

mean is the mean value of the sample data. It's default value is zero.

sd is the standard deviation. It's default value is 1.

**Dnorm**

## This function gives height of the probability distribution at each point for a given mean and standard deviation.

# Create a sequence of numbers between -10 and 10 incrementing by 0.1.

x <- seq(-10, 10, by = .1)

# Choose the mean as 2.5 and standard deviation as 0.5.

y <- dnorm(x, mean = 2.5, sd = 0.5)

# Give the chart file a name.

png(file = "dnorm.png")

plot(x,y)

graphics.off()

# Save the file.

dev.off()

**Pnorm**

This function gives the probability of a normally distributed random number to be less that the value of a given number. It is also called "Cumulative Distribution Function".

# Create a sequence of numbers between -10 and 10 incrementing by 0.2.

x <- seq(-10,10,by = .2)

# Choose the mean as 2.5 and standard deviation as 2.

y <- pnorm(x, mean = 2.5, sd = 2)

# Give the chart file a name.

png(file = "pnorm.png")

# Plot the graph.

plot(x,y)

# Save the file.

dev.off()

**Qnorm**

This function takes the probability value and gives a number whose cumulative value matches the probability value.

# Create a sequence of probability values incrementing by 0.02.

x <- seq(0, 1, by = 0.02)

# Choose the mean as 2 and standard deviation as 3.

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

# Give the chart file a name.

png(file = "qnorm.png")

# Plot the graph.

plot(x,y)

# Save the file.

dev.off()

**rnorm**

This function is used to generate random numbers whose distribution is normal. It takes the sample size as input and generates that many random numbers. We draw a histogram to show the distribution of the generated numbers.

# Create a sample of 50 numbers which are normally distributed.

y <- rnorm(50)

# Give the chart file a name.

png(file = "rnorm.png")

# Plot the histogram for this sample.

hist(y, main = "Normal DIstribution")

# Save the file.

dev.off()

# R - Binomial Distribution

**b) Binomial Distribution**

The binomial distribution model deals with finding the probability of success of an event which has only two possible outcomes in a series of experiments. For example, tossing of a coin always gives a head or a tail. The probability of finding exactly 3 heads in tossing a coin repeatedly for 10 times is estimated during the binomial distribution.

R has four in-built functions to generate binomial distribution. They are described below.

dbinom(x, size, prob)

pbinom(x, size, prob)

qbinom(p, size, prob)

rbinom(n, size, prob)

Following is the description of the parameters used in above functions −

x is a vector of numbers.

p is a vector of probabilities.

n is number of observations(sample size).

size is the number of trials.

prob is the probability of success of each trial.

**Dbinom**

This function gives the probability density distribution at each point.

# Create a sample of 50 numbers which are incremented by 1.

x <- seq(0,50,by = 1)

# Create the binomial distribution.

y <- dbinom(x,50,0.5)

# Give the chart file a name.

png(file = "dbinom.png")

plot(x,y)

dev.off()

**Pbinom**

This function gives the cumulative probability of an event. It is a single value representing the probability.

# Probability of getting 26 or less heads from a 51 tosses of a coin.

x <- pbinom(26,51,0.5)

print(x)

**qbinom**

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.

x <- qbinom(0.25,51,1/2)

print(x)

**rbinom**

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)

**12) Write R program to do the following tests with the sample data and visualize the results graphically.**

**χ2-test**

**t-test**

**F-test**

**χ2-test**

Chi-Square test is a statistical method to determine if two categorical variables have a significant correlation between them. Both those variables should be from same population and they should be categorical like − Yes/No, Male/Female, Red/Green etc.

For example, we can build a data set with observations on people's ice-cream buying pattern and try to correlate the gender of a person with the flavor of the ice-cream they prefer. If a correlation is found we can plan for appropriate stock of flavors by knowing the number of gender of people visiting.

## Syntax

The function used for performing chi-Square test is chisq.test().

The basic syntax for creating a chi-square test in R is −

**chisq.test(data)**

**data**is the data in form of a table containing the count value of the variables in the observation.

## Example

We will take the Cars93 data in the "MASS" library which represents the sales of different models of car in the year 1993.

library("MASS")

print(str(Cars93))

# Loading the Mass library.

# Creating a data frame from the main data set.

car\_data<- data.frame(Cars93$AirBags, Cars93$Type)

# Creating a table with the needed variables.

car\_data = table(Cars93$AirBags, Cars93$Type)

print(car\_data)

# Performing the Chi-Square test.

print(chisq.test(car\_data))

**Conclusion**

The above result shows the dataset has many Factor variables which can be considered as categorical variables. For our model we will consider the variables "AirBags" and "Type". Here we aim to find out any significant correlation between the types of car sold and the type of Air bags it has. If correlation is observed we can estimate which types of cars can sell better with what types of air bags.

**b) t-test**

x  <- c(0.593, 0.142, 0.329, 0.691, 0.231, 0.793, 0.519, 0.392, 0.418)

t.test(x, alternative="greater", mu=0.3)

**c) F-test**

install.packages("randomForest")

# Load the party package. It will automatically load other

# required packages.

library(party)

# Print some records from data set readingSkills.

print(head(readingSkills))

# Load the party package. It will automatically load other

# required packages.

library(party)

library(randomForest)

# Create the forest.

output.forest <- randomForest(nativeSpeaker ~ age + shoeSize + score,

                              data = readingSkills)

# View the forest results.

print(output.forest)