**30. Normal Distribution**

**Aim:**

To write an R program to generate and visualize a normal distribution, and calculate its probability density and cumulative distribution values.

1. **Using formula**
2. **Using dnorm,pnorm,qnorm,rnorm**

**Algorithm:**

**Step 1:** Start and define the aim of studying Normal Distribution using R.

**Step 2:** Import or generate dataset and check for missing values/outliers.

**Step 3:** Compute descriptive statistics (mean, median, standard deviation and variance).

**Step 4:** Visualize data using histogram, density plot, and Q–Q plot.

**Step 5:** Perform normality test (e.g., Shapiro–Wilk) to check if data follows Normal distribution.

**Step 6:** Use Normal distribution functions (dnorm, pnorm, qnorm, rnorm) to calculate probabilities, densities, quantiles, and simulate values.

**Step 7:** Interpret the results and conclude whether the data fits the Normal distribution.

**Step 8:**  End the program

1. **Normal Distribution by Using formula**

**Program :**

x <- c(126, 176, 158, 180, 186, 166, 166, 164, 178, 170,

189, 195, 172, 187, 190, 186, 185, 168, 179, 178,

182, 179, 170, 175, 186, 159, 161, 178, 175, 185,

175, 162, 173, 172, 177, 175, 172, 177, 180)

mean\_x <- mean(x)

sd\_x <- sd(x)

print(paste("Mean:", mean\_x))

print(paste("Standard Deviation:", sd\_x))

b <- (x - mean\_x) \* (x - mean\_x) / (2 \* sd\_x \* sd\_x

k1 <- exp(-b)

k2 <- 1 / (sqrt(2 \* pi) \* sd\_x)

k3 <- k2 \* k1

plot(x, type = "o", col = "blue", main = "Original Data Plot", ylab = "Values", xlab = "Index")

plot(x, k3, type = "o", col = "red", main = "Normal Distribution", ylab = "Density", xlab = "x values")

y <- dnorm(x, mean = mean\_x, sd = sd\_x)

plot(x, y, type = "o", col = "green", main = "Normal Density from dnorm()", ylab = "Density", xlab = "x values")

1. **Normal Distribution by Using dnorm,pnorm,qnorm,rnorm**

**Program:**

getwd()

setwd("C:/Users/MCA/Documents")

getwd()

wine <- read.csv("winequality-red.csv", sep = ";")

print(wine)

x <- wine$fixed.acidity

print(head(x, 10))

print(mean(x))

print(sd(x))

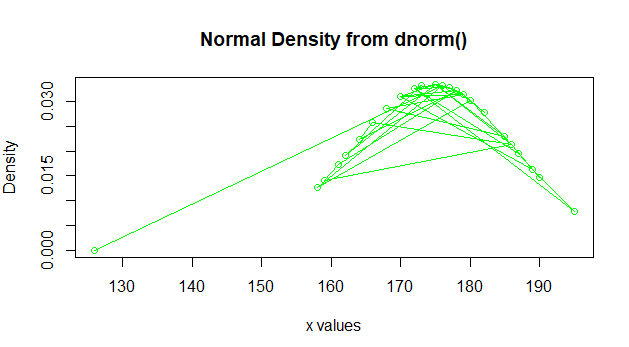
y <- dnorm(x, mean = mean(x), sd = sd(x))

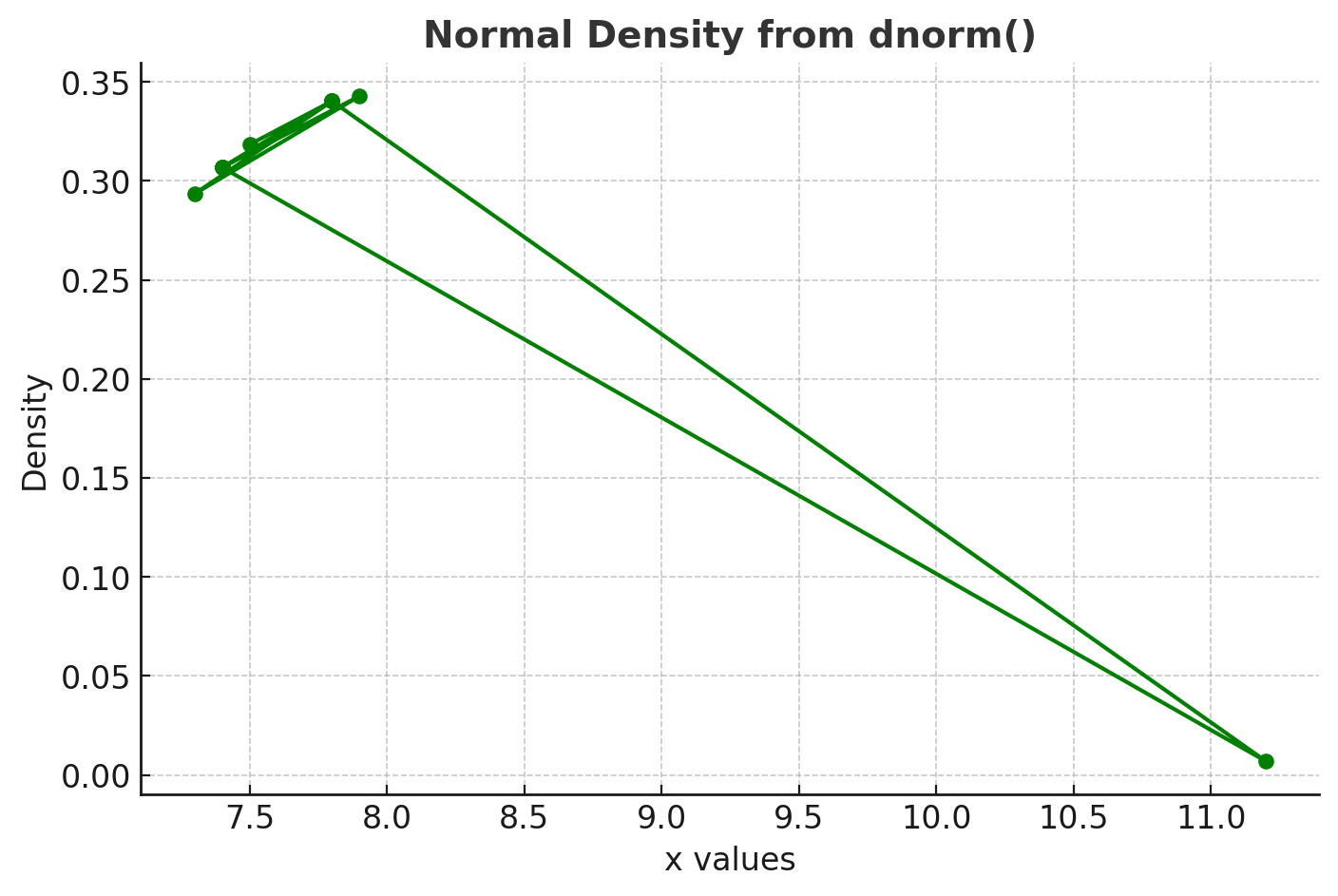
plot(x, y, type = "l", col = "darkred",

main = "Normal Distribution of Wine$Alcohol",

xlab = "Alcohol", ylab = "Density")

**OUTPUT:**

****

****

**RESULT:**

Thus, our program has been successfully saved and executed.

**32. Analysis of Variance using R**

**Aim:**

To write an R program to perform and interpret Analysis of Variance, in order to compare means across multiple groups and assess the significance of differences in a data science context

1. **Without using Built in function**
2. **Using Built in function**

**Algorithm:**

**Step 1 :** Start the Program

**Step 2:** To understand the concept of Analysis of Variance (ANOVA) in statistical data analysis.

**Step 3:** To manually compute sum of squares, degrees of freedom, and F-ratio.

**Step 4:** To implement ANOVA manually without using built-in functions in R.

**Step 5:** To calculate sum of squares, degrees of freedom, mean squares, and F-statistic manually.

**Step 6:** To learn how ANOVA is derived from basic mathematical formulas.

**Step 7:** To perform ANOVA using built-in R functions like aov() and summary().

**Step 8:** To validate the manual results by comparing with built-in function output.

**Step 9:** To apply ANOVA as a statistical technique for data science and decision-making tasks.

**Step 10:** End the program.

1. **Without using Built in function**

**Program:**

x1 <- c(8, 10, 7, 14, 11)

x2 <- c(7, 5, 10, 9, 9)

x3 <- c(12, 9, 13, 12, 14)

sum\_x1 <- sum(x1)

sum\_x2 <- sum(x2)

sum\_x3 <- sum(x3)

sum\_sq\_x1 <- sum(x1^2)

sum\_sq\_x2 <- sum(x2^2)

sum\_sq\_x3 <- sum(x3^2)

print(paste("Sum of x1:", sum\_x1, "Sum of x2:", sum\_x2, "Sum of x3:", sum\_x3))

print(paste("Sum of squares x1:", sum\_sq\_x1, "Sum of squares x2:", sum\_sq\_x2, "Sum of squares x3:", sum\_sq\_x3))

sum\_all <- sum\_x1 + sum\_x2 + sum\_x3

print(paste("Total sum:", sum\_all))

n1 <- length(x1)

n2 <- length(x2)

n3 <- length(x3)

N <- n1 + n2 + n3

ss\_treatment <- (sum\_x1^2 / n1) + (sum\_x2^2 / n2) + (sum\_x3^2 / n3) - (sum\_all^2 / N)

ss\_total <- sum\_sq\_x1 + sum\_sq\_x2 + sum\_sq\_x3 - (sum\_all^2 / N)

ss\_error <- ss\_total - ss\_treatment

df\_treatment <- 3 - 1

df\_error <- N - 3

ms\_treatment <- ss\_treatment / df\_treatment

ms\_error <- ss\_error / df\_error

F\_value <- ms\_treatment / ms\_error

cat("SS Treatment:", ss\_treatment, "\n")

cat("SS Error:", ss\_error, "\n")

cat("SS Total:", ss\_total, "\n")

cat("Degrees of Freedom (Treatment):", df\_treatment, "\n")

cat("Degrees of Freedom (Error):", df\_error, "\n")

cat("Mean Square Treatment:", ms\_treatment, "\n")

cat("Mean Square Error:", ms\_error, "\n")

cat("F value:", F\_value, "\n")

1. **Using Built in function**

**Program:**

x1 <- c(8, 10, 7, 14, 11)

x2 <- c(7, 5, 10, 9, 9)

x3 <- c(12, 9, 13, 12, 14)

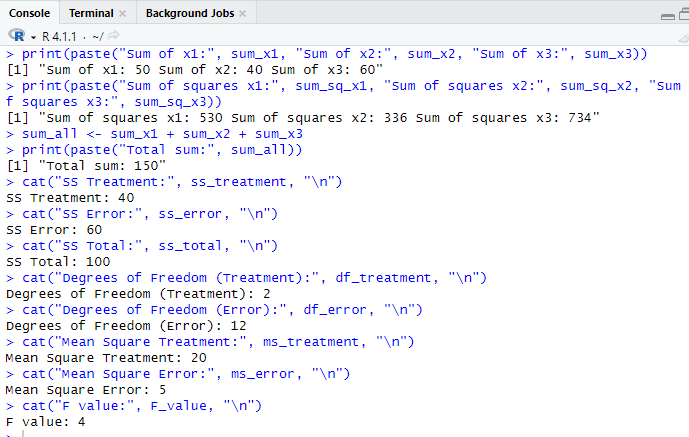
values <- c(x1, x2, x3)

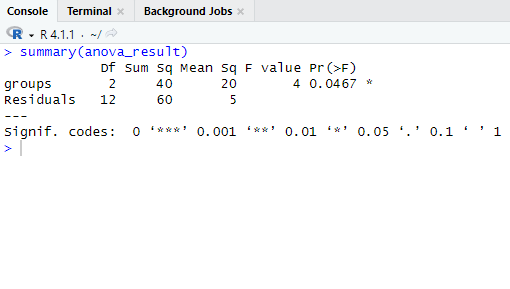
groups <- factor(rep(c("x1", "x2", "x3"), each = 5))

anova\_result <- aov(values ~ groups)

summary(anova\_result)

**OUTPUT:**

****

****

**RESULT:**

Thus, our program has been successfully saved and executed.