# Simulate customer email interaction dataset

set.seed(123)

n <- 500 # Number of customers

# Creating dataset

customer\_data <- data.frame(

email\_open\_rate = rbeta(n, shape1 = 2, shape2 = 5), # Beta distribution (0-1)

clicks\_per\_email = rpois(n, lambda = 3), # Poisson distribution

time\_spent = rgamma(n, shape = 2, scale = 2) # Gamma distribution

)

# Scale the data (important for distance-based clustering)

scaled\_data <- scale(customer\_data)

# Function to calculate clustering metrics (Silhouette & WSS)

calculate\_metrics <- function(data, clusters) {

wss <- sum(kmeans(data, centers = length(unique(clusters)))$withinss)

return(list(WSS = wss))

}

# ✅ 1. K-Means Clustering

wss <- sapply(1:10, function(k) { kmeans(scaled\_data, centers = k)$tot.withinss })

plot(1:10, wss, type = "b", main = "Elbow Method for K-means", xlab = "Number of Clusters", ylab = "WSS")

k <- 3 # Choosing k=3 from the elbow method

kmeans\_model <- kmeans(scaled\_data, centers = k)

kmeans\_metrics <- calculate\_metrics(scaled\_data, kmeans\_model$cluster)

# ✅ 2. Hierarchical Clustering

hclust\_model <- hclust(dist(scaled\_data), method = "ward.D2")

hclust\_clusters <- cutree(hclust\_model, k = k)

hclust\_metrics <- calculate\_metrics(scaled\_data, hclust\_clusters)

# ✅ 3. DBSCAN Clustering (Using Manual Calculation Instead of Library)

eps <- 0.5 # Estimated epsilon value

minPts <- 5 # Minimum points per cluster

dbscan\_clusters <- rep(0, n) # Initialize cluster assignment

for (i in 1:n) {

neighbors <- which(rowSums((scaled\_data - scaled\_data[i,])^2) < eps^2)

if (length(neighbors) >= minPts) {

dbscan\_clusters[i] <- i # Assign cluster number

}

}

dbscan\_metrics <- calculate\_metrics(scaled\_data, dbscan\_clusters)

# ✅ Compare Algorithm Performance

performance\_df <- data.frame(

Algorithm = c("K-Means", "Hierarchical", "DBSCAN"),

WSS = c(kmeans\_metrics$WSS, hclust\_metrics$WSS, dbscan\_metrics$WSS)

)

print(performance\_df)

# ✅ Visualizing Clusters using PCA (Without External Library)

pca <- prcomp(scaled\_data)

plot(pca$x[,1], pca$x[,2], col = kmeans\_model$cluster, pch = 19,

main = "K-Means Clustering", xlab = "PC1", ylab = "PC2")