**Implementing K-Means**

Description

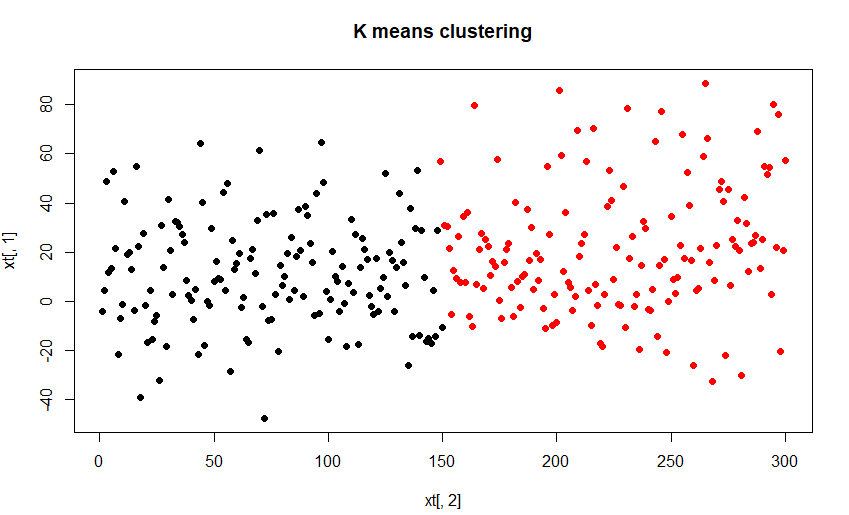
In this implementation, I am implementing the K-Means clustering in R programming. I am generating a univariate sample having samples from three gaussian distributions.

Solution

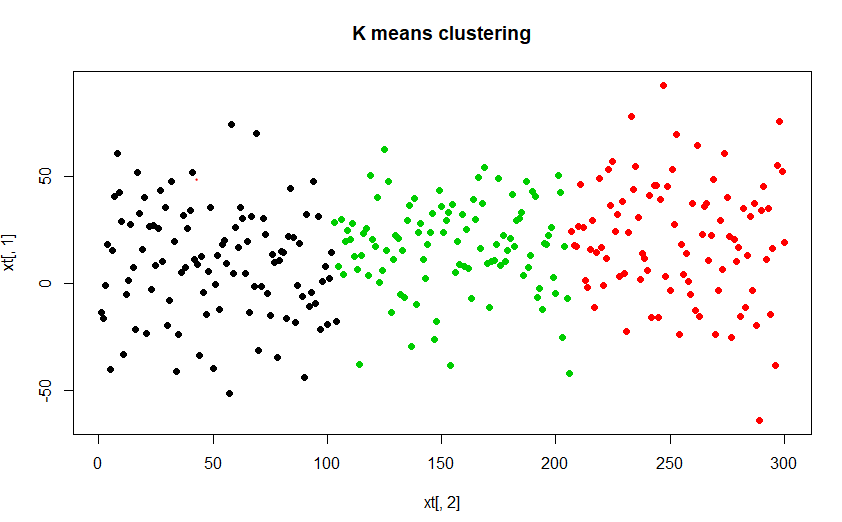
K-means clustering is a clustering algorithm which aims at clustering a sample into K partitions. I am using Euclidean distance between the points as a measure to cluster the samples. I ran the algorithm with K being 2, 3 and 4 and plotted the results. I also plotted the reconstruction error for different values of K.

Result

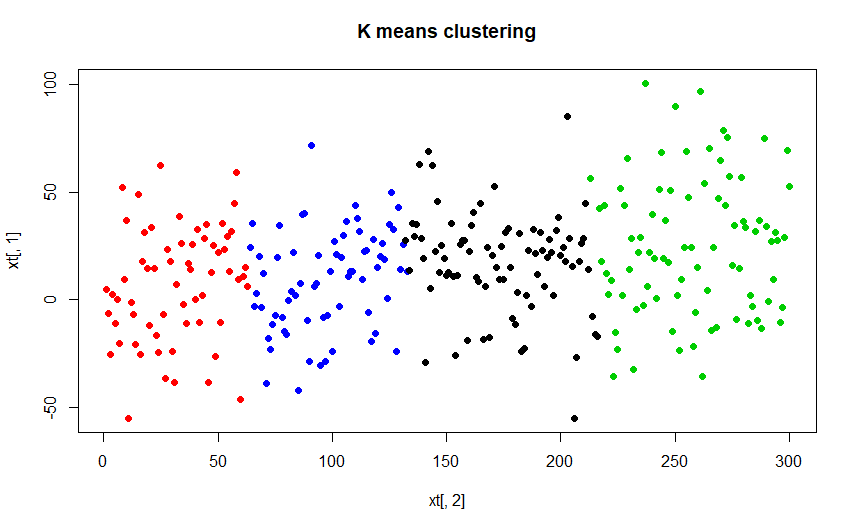
For K = 2



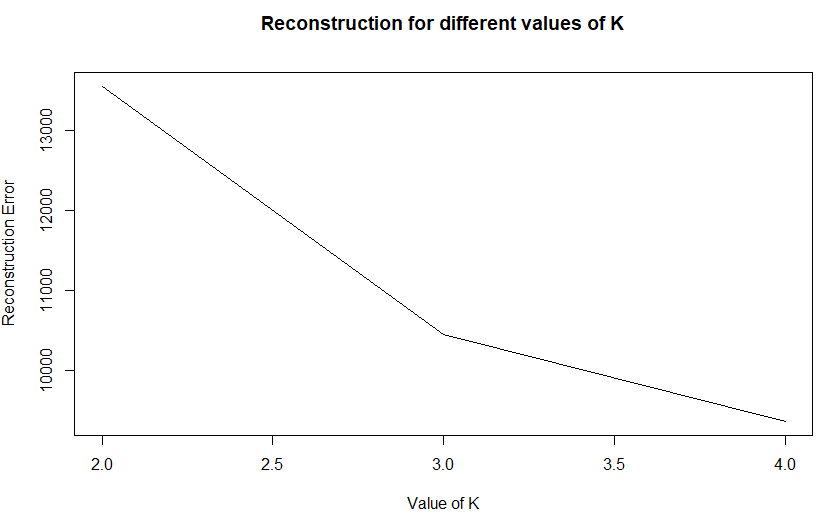
For K = 3



For K = 4



Reconstruction Error



From the reconstruction error we can see that there is a huge drop in reconstruction error when we move from k=2 to k=3, but the drop reduces from k=3 to k=4. We can see an elbow at k=3 and can pick k=3 as the number of centres in our clustering algorithm.

**Code: -**

set.seed(123)

error\_cal <- function(x) #Function for calculating reconstruction error

{

error <- 0

n <- ncol(x)

k <- n - 3

for(z in seq(1:k))

{

for(t in seq(1:nrow(x)))

{

if(x[t,n] == z)

{

error <- x[t,(z+2)] + error

}

}

}

return(error)

}

#function for eucledean distance

euc\_dist <- function(xt,x2,y2) #Function for calculating the euclidean distance

{

dist = sqrt((xt[,1]-x2)^2 + (xt[,2] - y2)^2)

return(dist)

}

#Function to calculate the center

mean\_cent\_x <- function(xt,k,t,mx) #Function for finding centers

{

count <- 0

p<- 0

for(n in seq(1:nrow(xt)))

{

if(xt[n,(k+3)] == t)

{

p <- p + xt[n,1]

count <- count + 1

}

mx[k] <- p/count

}

return(mx[k])

}

mean\_cent\_y <- function(xt,k,t,my) #Function for finding centers

{

count <- 0

q <- 0

for(n in seq(1:nrow(xt)))

{

if(xt[n,(k+3)] == t)

{

q <- q + xt[n,2]

count <- count + 1

}

my[k] <- q/count

}

return(my[k])

}

kmeans <- function(k)

{

x1 <- rnorm(100,10,25)

x2 <- rnorm(100,15,20)

x3 <- rnorm(100,20,30)

x <- c(x1,x2,x3)

y <- 1:300

xt <- data.frame(x,y) #Generating a univariate sample

mx <- sample(x,k)

my <- sample(y,k)

for(a in seq(1:4)) # Running 5 iterations to find the centers

{

for(t in seq(1:k))

{

l <- t+2

xt[,l] <- euc\_dist(xt,mx[t],my[t])

}

vec <- 3:(k+2)

for(t in seq(1:k))

{

for(n in seq(1:nrow(xt)))

{

if(xt[n,(t+2)] == apply(xt[n,vec],1,min))

{

xt[n,(k+3)] <- t

}

}

}

for(t in seq(1:k)) #Finding the mean centers using the function created above

{

x <- xt

y<- xt

mx[t] <- mean\_cent\_x(x,k,t,mx)

my[t] <- mean\_cent\_y(y,k,t,my)

}

}

plot(xt[,2], xt[,1], col=xt[,(k+3)], pch = 19 , main = "K means clustering")

return(xt)

}

x2 <- kmeans(2)

x3 <- kmeans(3)

x4 <- kmeans(4)

e2 <- error\_cal(x2)

e3 <- error\_cal(x3)

e4 <- error\_cal(x4)

e <- c(e2,e3,e4)

plot(e, type ="l")

**Visualization: -**

