Date: 01/Feb/2024	Gradient Descent Optimization
EXPERIMENT – 03	

AIM: To perform Gradient Descent Optimization

SOFTWARE REQUIRED: RStudio

R CODE:

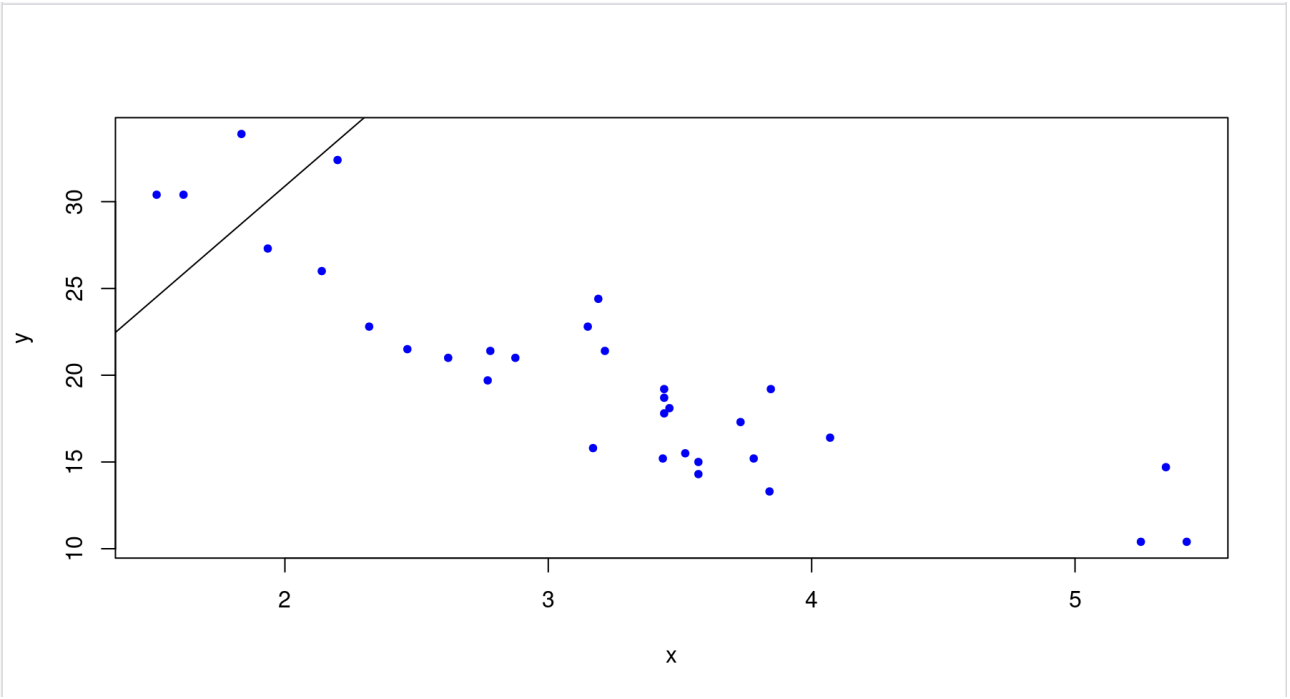
```
rm(list = ls ())
data <- mtcars
GRADIENT.DESCENT <- function (y, x, alpha, conv_threshold, n,
max_iter) {
  plot (x, y, col = "blue", pch = 20)
  m <- runif(1, 0, 1)
  c <- runif(1, 0, 1)
  yhat <- m * x + c
  MSE <- sum((y - yhat) ^ 2) / n
  converged = F
  iterations = 0
  while(converged == F) {
    m_new <- m - alpha * ((1 / n) * (sum((yhat - y) * x)))
    c_new <- c - alpha * ((1 / n) * (sum (yhat - y)))
    m <- m_new
    c <- c_new
    yhat <- m * x + c
    MSE_new <- sum((y - yhat) ^ 2) / n
    if(MSE - MSE_new <= conv_threshold){
        abline(c, m)
        converged = T
         return(paste("Optimal intercept:", c, "Optimal slope:", m,
"No of iterations:", iterations, "MSE:", MSE_new))
    }
    iterations = iterations + 1
    if(iterations >= max_iter) {
      abline(c , m)
      converged = T
        return(paste("Optimal intercept:", c, "Optimal slope:", m,
"No of iterations:", iterations, "MSE:", MSE_new))
    }
  }
}

GRADIENT.DESCENT(data$mpg, data$wt, 0.25, 0.001, length(data$mpg),
2500)
slr <- lm(mpg ~ wt, data = mtcars)
slr$coef
mpg_p <- predict (slr)
sqerr <- (data$mpg - mpg_p)^2
```

```
MSE.SLR <- sum(sqerr)/length(data$mpg)
slope:", m,
length (dataSmpg),
```

OUTPUT:

```
> rm(list = ls ())
> data <- mtcars
> GRADIENT.DESCENT <- function (y, x, alpha, conv_threshold, n, max_iter) {
+   plot (x, y, col = "blue", pch = 20)
+   m <- runif(1, 0, 1)
+   c <- runif(1, 0, 1)
+   yhat <- m * x + c
+   MSE <- sum((y - yhat) ^ 2) / n
+   converged = F
+   iterations = 0
+   while(converged == F) {
+     m_new <- m - alpha * ((1 / n) * (sum((yhat - y) * x)))
+     c_new <- c - alpha * ((1 / n) * (sum (yhat - y)))
+     m <- m_new
+     c <- c_new
+     yhat <- m * x + c
+     MSE_new <- sum((y - yhat) ^ 2) / n
+     if(MSE - MSE_new <= conv_threshold){
+         abline(c, m)
+         converged = T
+         return(paste("Optimal intercept:", c, "Optimal slope:", m, "No of iterations:", iterations, "MSE:", MSE_new))
+     }
+     iterations = iterations + 1
+     if(iterations >= max_iter) {
+       abline(c , m)
+       converged = T
+       return(paste("Optimal intercept:", c, "Optimal slope:", m, "No of iterations:", iterations, "MSE:", MSE_new))
+     }
+   }
+ }
>
> GRADIENT.DESCENT(data$mpg, data$wt, 0.25, 0.001, length(data$mpg), 2500)
[1] "Optimal intercept: 4.69138829676791 Optimal slope: 13.1012890281482 No of iterations: 0 MSE: 1039.87236235623"
> slr <- lm(mpg ~ wt, data = mtcars)
> slr$coef
(Intercept)          wt
  37.285126   -5.344472
> mpg_p <- predict (slr)
> sqerr <- (data$mpg - mpg_p)^2
> MSE.SLR <- sum(sqerr)/length(data$mpg)
> slope:", m,
+ length (dataSmpg),
```


y
10
15
20
25
30
2
3
4
5
x