Lab experiment – 9

Name: Prithviraj Guntha

Reg. No.: 20BRS1188

Subject: Essentials of data analytics

Subject code: CSE3506

Professor: A Sheik Abdullah

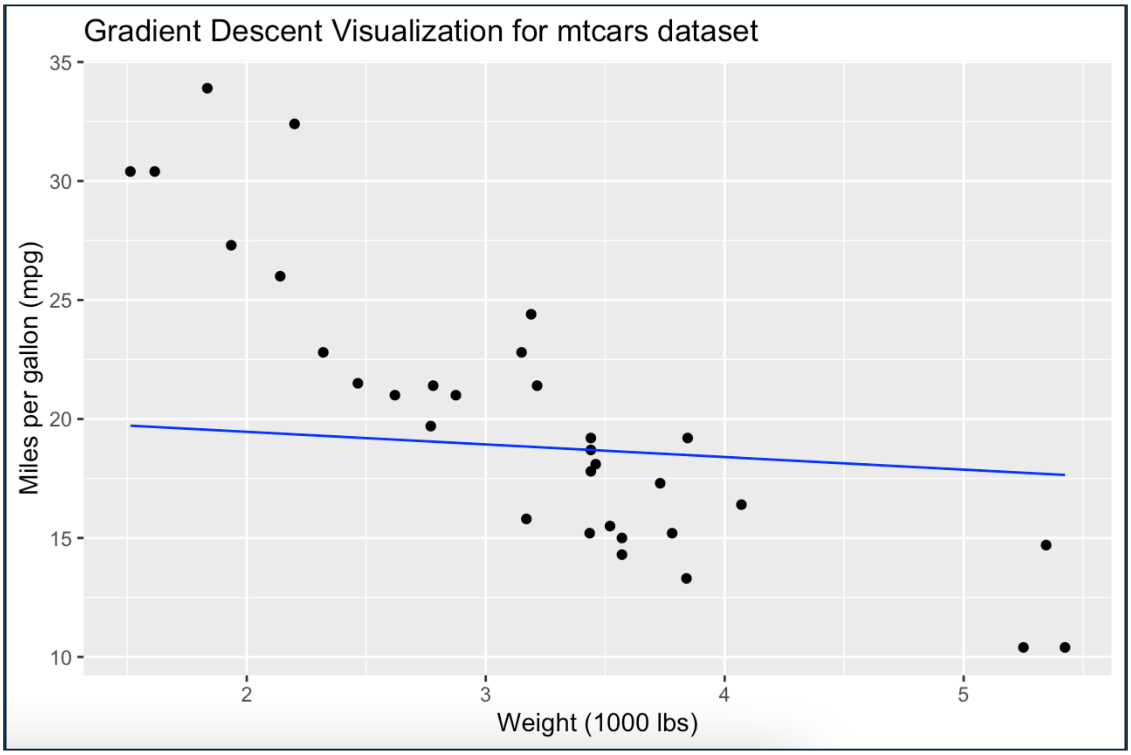
Slot: L55+L56

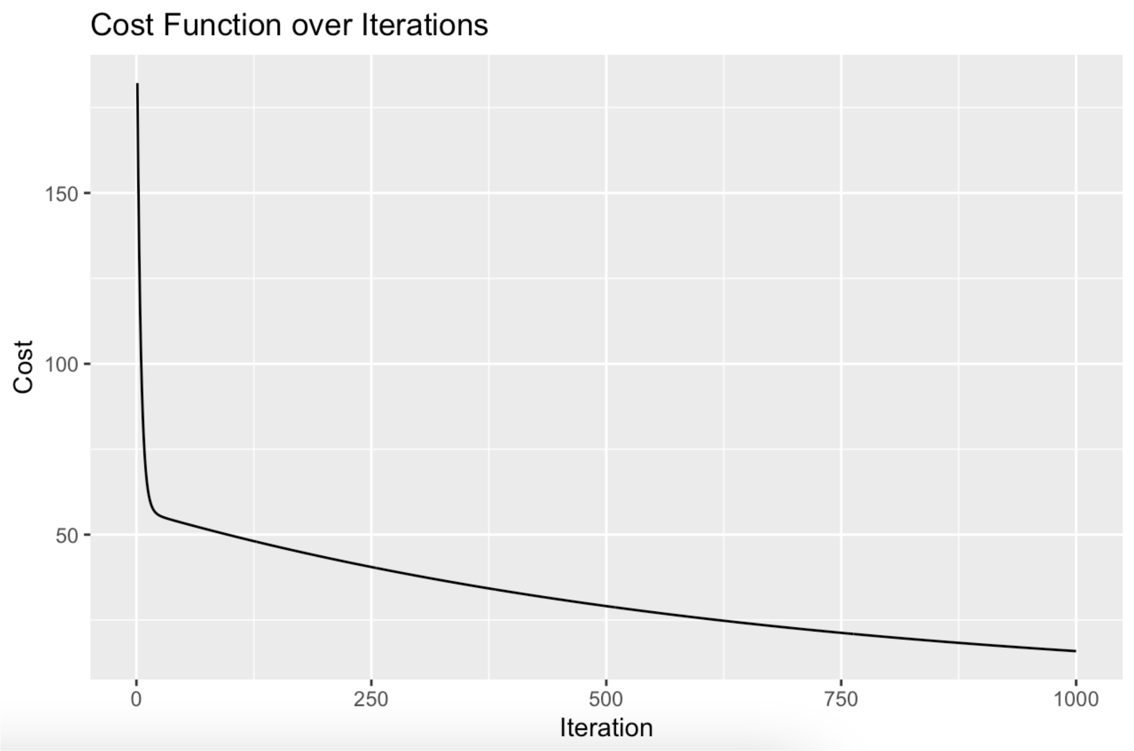
1. **Write R code to visualise gradient descent.**

Code:

```
# Load required libraries
library(ggplot2)

# Load mtcars dataset
data(mtcars)

# Define the cost function
cost_function <- function(theta, x, y) {
  predictions <- theta[1] + theta[2] * x
  errors <- predictions - y
  cost <- (1 / (2 * length(y))) * sum(errors^2)
  return(cost)
}

# Define the gradient descent function
gradient_descent <- function(x, y, alpha = 0.01, iterations =
1000) {
  theta <- c(0, 0)
  costs <- rep(0, iterations)
  for (i in 1:iterations) {
    predictions <- theta[1] + theta[2] * x
    errors <- predictions - y
    gradient <- c(sum(errors) / length(y), sum(errors * x) /
length(y))
    theta <- theta - alpha * gradient
    costs[i] <- cost_function(theta, x, y)
  }
  results <- list(theta = theta, costs = costs)
  return(results)
}

# Run gradient descent on the mtcars dataset
results <- gradient_descent(mtcars$wt, mtcars$mpg)

# Visualize the results
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_line(aes(y = results$theta[1] + results$theta[2] * wt),
color = "blue") +
  ggtitle("Gradient Descent Visualization for mtcars dataset")
+
  xlab("Weight (1000 lbs)") +
  ylab("Miles per gallon (mpg)")


# Create a dataframe for storing costs over iterations
costs_df <- data.frame(iteration = 1:length(results$costs),
cost = results$costs)

# Visualize the cost function over iterations
ggplot(data = costs_df, aes(x = iteration, y = cost)) +
  geom_line() +
ggtitle("Cost Function over Iterations") +
xlab("Iteration") +
ylab("Cost")
```

Output:


Gradient Descent Visualization for mtcars dataset
Miles per gallon (mpg)
Weight (1000 lbs)


Cost Function over Iterations
Cost
Iteration