**Name:** Vikram Surya R

**Reg No:** 19BCE1490

**Essentials of Data Analytics**

**Tasks for Week-10: Moment-based Gradient Descent**

## AIM:

To understand the moment-based gradient descent following operations/functions on 'mtcars' dataset based on given instructions.

## Algorithm:

1. Clear the environmental variables using rm function
2. Create a function named mgd for gradient descent with attributes 'x1','x2','y','m1','m2','c','aplha','gama','iter'.
3. Initialize iterations and Lf values with 0.
4. While iterations less than iter.
5. Calculate y_predicted as m1*x1+m2*x2+c
6. New loss function equals 0.5*sum of (difference of y and y_predicted)^2.
7. Calculate gradient descent.
8. Update the values of c and slope using gradient descent function.
9. Check if the value of loss function is less than threshold (Loss function – new loss function) if not break the loop else repeat.
10. Return the optimal, m1 & m2, c, loss.
11. Close the mgd function.
12. Retrieve the dataset mtcars.
13. Call the mgd function with attributes passed.
14. Apply a linear model from mpg to sum of hp and wt and print the result.

## RESULTS:

```
> mgd(data$wt,data$hp,data$mpg,-0.2,-0.2,32,0.0000045,0.98,15000)
[1] "Optimal Intercept  37.2259097539394 Optimal slope one  -3.8774106273879 Optimal slope two  -0.0317735876402436 loss fun
ction: 5.4087824042049e-06 iterations: 15001"
> lr<-lm(data$mpg~(data$wt+data$hp))
> lr

Call:
lm(formula = data$mpg ~ (data$wt + data$hp))

Coefficients:
(Intercept)      data$wt      data$hp
   37.22727     -3.87783     -0.03177
```

## INFERENCE:

The values of the gradient decent are almost similar to Linear Regression model so the moment-based gradient decent can be acceptable.

## CODE:

```
rm(list=ls())
mgd<- function(x1,x2,y,m1,m2,c,alpha,gamma,iter){
  iterations=0
  Lf<-0
  u_m1=0
  u_m2=0
  u_c=0
  while(iterations<=iter){
    y_p=m1*x1+m2*x2+c
    Lf_new<-0.5*sum(y_p-y)^2
    nu_m1=gamma*u_m1+alpha*sum((y_p-y)*x1)
    nu_m2=gamma*u_m2+alpha*sum((y_p-y)*x2)
    nu_c=gamma*u_c+alpha*sum(y_p-y)
    m1<-m1-nu_m1
    m2<-m2-nu_m2
    c<-c-nu_c
    u_m1<-nu_m1
```

```
    u_m2<-nu_m2
    u_c<-nu_c
    Lf<-Lf_new
    iterations=iterations+1
  }
  return(paste("Optimal Intercept ",c,"Optimal slope one ",m1,"Optimal slope two ",m2,"loss function:",Lf,"iterations:",iterations))
}
data<-mtcars
plot(data$mpg,data$wt,col="red",pch=20)
mgd(data$wt,data$hp,data$mpg,-0.2,-0.2,32,0.0000045,0.98,15000)
lr<-lm(data$mpg~(data$wt+data$hp))
lr
```