**Aim**: Implementing the Momentum Gradient Descent Algorithm

**Algorithm:**

1. Clear the space by clearing the list using rm() function.
2. Then declare function MGD using function() with x1,x2,y,m1,m2,c,learning rate(alpha), gamma, number of iterations as the parameters of the function.
3. Declare a variable iterations and assign a value 0.
4. Then declare variables LF, u\_m1, u\_c and assign value 0.
5. Run a while loop and check the condition if iterations are less than or equal to given number of iterations.
6. Then declare the variable y\_pred and assign m1 \* x1 + m2 \* x2 +c value to y\_pred.
7. Then declare variable Lf\_new and assign 0.5 \* sum of difference of y\_pred and y power of 2.
8. Declare and assign variable nu\_m1 as gamma \* u\_m1 + alpha \*sum of difference of y\_pred and y \* x1.
9. Declare and assign variable nu\_m2 as gamma \* u\_m2 + alpha \*sum of difference of y\_pred and y \* x2.
10. Declare and assign variable nu\_c as gamma \* u\_c + alpha \*sum of difference of y\_pred and y.
11. Assign m1 as difference of m1 and nu\_m1.
12. Assign m2 as difference of m2 and nu\_m2.
13. Assign c as difference of c and nu\_m1.
14. Assign u\_m1 as nu\_m1.
15. Assign u\_m2 as nu\_m2.
16. Assign u\_c as nu\_c.
17. Assign u\_c as nu\_c.
18. Assign Lf as Lf\_new.
19. Then increase iterations to 1.
20. End the loop.
21. Then load the dataset.
22. Plot the two variables that chosen for applying momentum gradient descent using plot() function.
23. Then call the function Mgd by giving the parameters.
24. Apply linear model to selected variables.
25. Then analyse the summary.

**Inference**:

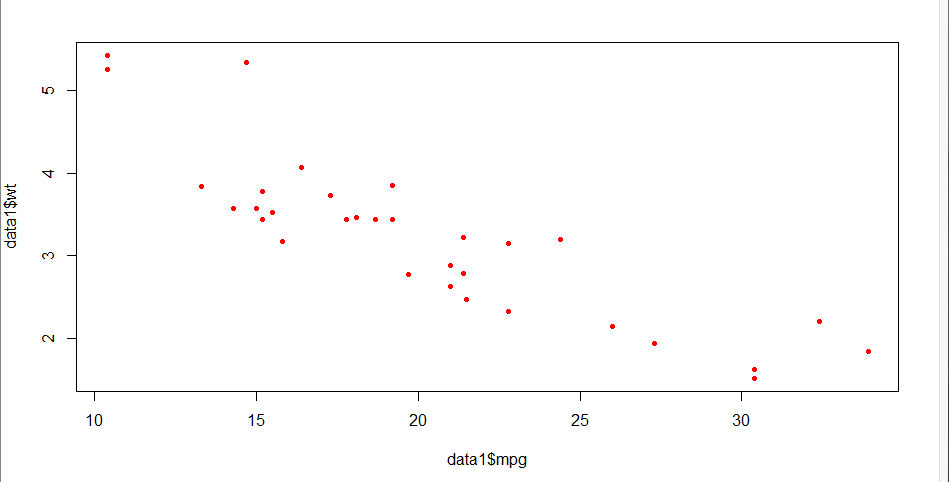
Dataset : mtcars X= hp and wt Y= mpg

**Mgd(data1$wt,data1$hp,data1$mpg,-0.2,-0.2,32,0.000002,0,1000000)**

According to the observation we saw that intercept and slope of linear regression and momentum gradient descent algorithm doesn’t have much difference but momentum gradient descent model performs a bit better than gradient descent model.

**Result:**

**Wt vs mpg:**



**Linear Model summary:**



**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\_pred<-m1\*x1+m2\*x2+c

Lf\_new<-0.5\*sum((y\_pred-y)^2)

nu\_m1<-gamma\*u\_m1+alpha\*sum((y\_pred-y)\*x1)

nu\_m2<-gamma\*u\_m2+alpha\*sum((y\_pred-y)\*x2)

nu\_c<-gamma\*u\_c+alpha\*sum(y\_pred-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,"optimatl slope:",m1,m2,"Loss funciton:",Lf,"iterations:",iterations))

}

data1<-mtcars

plot(data1$mpg,data1$wt,col="red",pch=20)

Mgd(data1$wt,data1$hp,data1$mpg,-0.2,-0.2,32,0.000002,0,1000000)

lr<-lm(data1$mpg~data1$hp+data1$wt)

lr