**Aim**: Understand the following operations/functions on ‘Mtcars’ data.

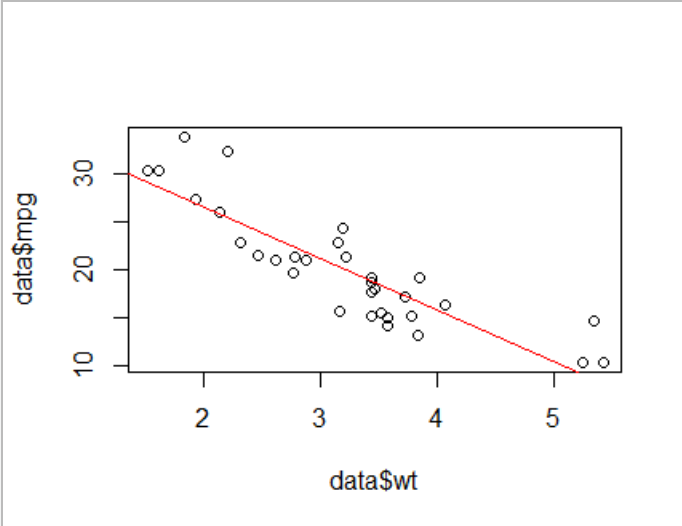
**Algorithm:**

* removing all the values from the global environment.
* Create a function named gd with the parameters of x, y, m, c, alpha, com\_thr, iter.
* Create and initialize the variable iterations with 0.
* Create and initialize the variable Lf with 0.
* Start the while loop with condition that if the iterations value should be less than iter.
* Create a variable name y\_p and assignee value as m\*x-c.
* Create a new variable named Lf\_new and assignee value of 0.5\*(sum(y\_p-y)^2).
* Create a new variable called m and assignee the value with m-alpha\*sum((y\_p-y)\*x). it can be n number of m’s because if the system has one independent variable then we will have one m value or else we have m’s values according to the no of independent variable we use.
* Create a new variable called c and assignee the value with c-alpha\*sum(y\_p-y).
* There will be use of an condition now if the absolute difference of Lf and Lf\_new is less or equal to conv\_thr then we break the while loop and return optimal intercept and optimal slop values.
* Store the dataset of mtcars in data variable.
* Call the function gd and give the input variables in this way (data$wt, data$mpg, -0.2,32, 0.001, 0.0001, 1000).
* Plot the data$wt and data$mpg and later find the lm values using lm function or these 2 variables.

**Result:**

**Dataset: mtcars:**

**One independent and one dependent variable:**



**Inference**:

**Dataset: mtcars:**

**One independent and one dependent variable:**

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

**Two independent and one dependent variable:**

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

**Statistics**:

**Dataset: mtcars:**

**One independent and one dependent variable:**

**Gradient decent:**

Optimal Intercept 36.7393256646, Optimal slop -5.187718039126

**Linear Regression:**

(Intercept) data$wt

37.285 -5.344

**Two independent and one dependent variable:**

**Gradient decent:**

Optimal Intercept 31.7905329118 Optimal slop one -4.92233682447 Optimal slop two 1.15226031532

**Linear Regression:**

(Intercept) data$wt data$drat

30.290 -4.783 1.442

**Code:**

**One independent and one dependent variable:**

rm(list=ls())

gd<- function(x,y,m,c,alpha,conv\_thr,iter){

iterations<-0

Lf<-0

while(iterations<=iter){

y\_p=m\*x+c

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

m=m-alpha\*sum((y\_p-y)\*x)

c=c-alpha\*sum(y\_p-y)

if(abs(Lf-Lf\_new)<=conv\_thr){

break

}

Lf=Lf\_new

iterations=iterations+1

}

return(paste("Optimal Intercept ",c,"Optimal slop ",m))

}

data<-mtcars

gd(data$wt,data$mpg,-0.2,32,0.001,0.0001,1000)

plot(data$wt,data$mpg)

reg<-lm(data$mpg~data$wt)

reg

abline(reg,col='red')

**Two independent and one dependent variable:**

rm(list=ls())

gd1<- function(x1,x2,y,m1,m2,c,alpha,conv\_thr,iter){

iterations<-0

Lf<-0

while(iterations<=iter){

y\_p=m1\*x1+m2\*x2+c

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

m1=m1-alpha\*sum((y\_p-y)\*x1)

m2=m2-alpha\*sum((y\_p-y)\*x2)

c=c-alpha\*sum(y\_p-y)

if(abs(Lf-Lf\_new)<=conv\_thr){

break

}

Lf=Lf\_new

iterations=iterations+1

}

return(paste("Optimal Intercept ",c,"Optimal slop one ",m1,"Optimal slop two ",m2))

}

data<-mtcars

view(data)

gd1(data$wt,data$drat,data$mpg,-0.2,-0.4,32,0.001,0.00001,10000)

reg1<-lm(data$mpg~(data$wt+data$drat))

reg1