Gradient Descent Algorithm

2024-03-18

library(grid)  
library(dplyr)

##   
## Attaching package: 'dplyr'

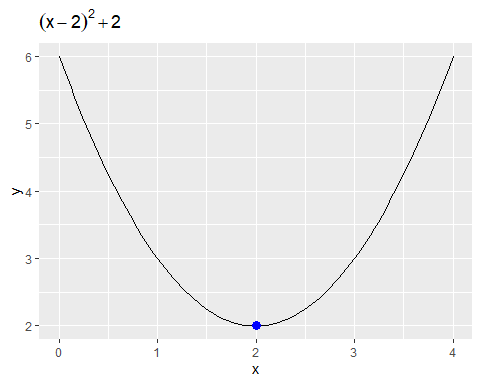
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(Deriv)

## Warning: package 'Deriv' was built under R version 4.3.3

library(ggplot2)  
  
#original formula  
formula = function(x){  
 (x-2)^2+2  
}  
  
ggplot(data.frame(x = c(0,4)), aes(x))+  
 stat\_function(fun = formula)+  
 geom\_point(data = data.frame(x = 2,y=formula(2)),aes(x,y),   
 color = "blue", size = 3)+  
 ggtitle(expression((x-2)^2+2))



#first derivative of the formula above  
derivative = Deriv(formula,"x");derivative

## function (x)   
## 2 \* (x - 2)

#defining learning rate  
l\_rate = 0.2  
#defining initial value  
xold = 2.2  
(iteration = data.frame(x = xold, y = formula(xold)))

## x y  
## 1 2.2 2.04

## Iteration:1  
#apply the formula of gradient descent  
xnew = xold - l\_rate\*derivative(xold)  
#output  
rbind(iteration, c(xnew,formula(xnew)))

## x y  
## 1 2.20 2.0400  
## 2 2.12 2.0144

#define the epsilon value, maximum iteration allowed  
epsilon = 0.05  
step = 2  
iteration = 10  
#records the x and y values: add initial guess  
xtrace = list();ytrace = list()  
xtrace[[1]] = xold; ytrace[[1]] = formula(xold)  
xtrace[[2]] = xnew; ytrace[[2]] = formula(xnew)  
cbind(xtrace,ytrace)

## xtrace ytrace  
## [1,] 2.2 2.04   
## [2,] 2.12 2.0144

while(abs(xnew-xold)>epsilon & step <= iteration)  
{  
 #update iteration count  
 step = step +1  
 #gradient descent  
 xold = xnew  
 xnew = xold - l\_rate\*derivative(xold)  
   
 # record keeping  
 xtrace[[step]] = xnew  
 ytrace[[step]] = formula(xnew)  
}  
#create the data points in dataframe  
record = data.frame(x = do.call(rbind,xtrace), y = do.call(rbind,ytrace))  
record

## x y  
## 1 2.200 2.040000  
## 2 2.120 2.014400  
## 3 2.072 2.005184