Solution for Problem 10 on Problem Set 2

Nadeem Demian

9/25/2020

```
crazyfunction <- function(x) {
  # Check for internal conformity
  if(as.character(x)=="r") print("You have preformed an illegal opperation")
  # Generate upper bound of function
  if(x > 1e2) a = 0
  #Create sequence z which is entirely arbitrary
  z = seq(0,1e2,1e-2)
  for(i in 1:length(z)){
    aa = sin(x) + 7*x + log(x) # key equation
    CC = 100*x + 2*exp(10)
    if(x>z[i]) {hit = z[i]
    bb = aa*2 + 3 #modify it
    cc = x^z[i]
    dd = bb*cc
    jj = (dd - 100)*-1
    if(jj < -1000) print("We don't know the function value in this domain")
    break
    }
  }
  return(jj)
}
```

Start by calculating the slope, we will use the standard `(y2-y1)/(x2-x1)` in order to do so

```
basic.slope <- (crazyfunction(4) - crazyfunction(3))/(4-3)
basic.slope
```

```
## [1] -12.77952
```

Great – however, we know that's not really a good approximation of the slope exactly at the point x = 3 Currently the change in x is = 1, but in order to get a better approximation for x, we want to use incrementally smaller changes. To demonstrate, I will use a for loop, printing only the final value where the change in x = 0.0001

```
foo = 10000
for(j in 1:foo) {
  zoo <- 3 + (1/foo)
  (crazyfunction(zoo) - crazyfunction(3))/(zoo-3)
  if (j == foo) {
    print((crazyfunction(zoo) - crazyfunction(3))/(zoo-3))
  }
}
```

```
## [1] -12.68666
```

Using this srategy of decreasing increments, we find that the approximation for the value of the slope at x = 3 converges at 12.68666