Newton Raphson Assignment

```r
NewtonRaphson <- function(func, StartingValue,Tolerance,MaxNumberOfIterations){
  Deviation <- func(StartingValue)[1]
  i <- 0
  newX <- StartingValue
  val <- data.frame(i = NULL, x = NULL, y= NULL)
  while ((i < MaxNumberOfIterations) && (Deviation > Tolerance)) {
    tmp <- func(newX)
    val[i + 1, 1:3] <- c(i + 1, newX, tmp[1])
    if ((tmp[1]=="NaN") || (tmp[2]=="NaN")) {
      cat("Function or derivative not defined error.")
      cat("\n", newX, tmp)
      break
    }
    newX <- newX - tmp[1]/tmp[2]
    newVal <- func(newX)
    Deviation <- abs(newVal[1])
    i <- i + 1
    cat(paste("\nIteration ", i, ":   X=", newX,"  Y=", newVal))
  }
  if (Deviation < Tolerance) {
    cat(paste("\nFound the root point: ", newX, " after ", i, "iterations"))
  } else {
    cat(paste("\nConvergence failure. Deviation: ", Deviation, "after ", i, "iterations"))}
  return(val)
}
```

Function 1

$f(x) = 10x^2 + 3x - 4$

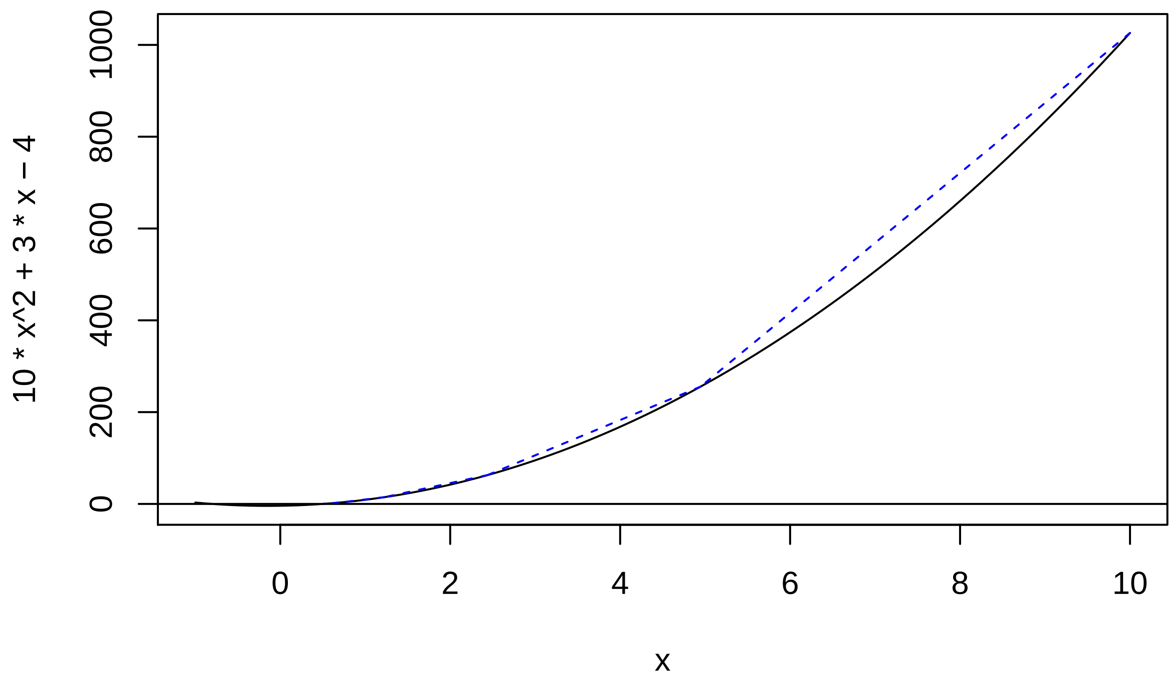
```r
f1 <- function(x) {
  return(c(10*x**2 + 3*x - 4, 20*x +3))
}

a <- NewtonRaphson(f1, 10, 1e-3, 40)
```

```
##
## Iteration  1 :   X= 4.94581280788177   Y= 255.448081729719
## Iteration  1 :   X= 4.94581280788177   Y= 101.916256157635
## Iteration  2 :   X= 2.43936200836836   Y= 62.8229561038143
## Iteration  2 :   X= 2.43936200836836   Y= 51.7872401673672
## Iteration  3 :   X= 1.22626480718943   Y= 14.7160481950816
## Iteration  3 :   X= 1.22626480718943   Y= 27.5252961437886
## Iteration  4 :   X= 0.691627573199036   Y= 2.85836971968898
## Iteration  4 :   X= 0.691627573199036   Y= 16.8325514639807
## Iteration  5 :   X= 0.521815544059812   Y= 0.288361252403806
## Iteration  5 :   X= 0.521815544059812   Y= 13.4363108811962
## Iteration  6 :   X= 0.500354202851388   Y= 0.0046058916646432
## Iteration  6 :   X= 0.500354202851388   Y= 13.0070840570278
## Iteration  7 :   X= 0.50000009645487   Y= 1.25391339977909e-06
```

```
## Iteration  7 :   X= 0.50000009645487   Y= 13.0000019290974
## Found the root point:  0.50000009645487  after  7 iterations
```

```r
curve(10*x**2 + 3*x -4, -1, 10)
for(i in 1:(nrow(a) - 1)) {
  segments(x0 = a[i, 2], y0 = a[i, 3], x1 = a[i + 1, 2], y1 = a[i + 1, 3], col = "blue", lty = 2)
}
abline(h=0)
```




```r
dev.off()
```

```
## null device
##           1
```

Function 2

$f(x) = (x - 2)^3 - 6x$

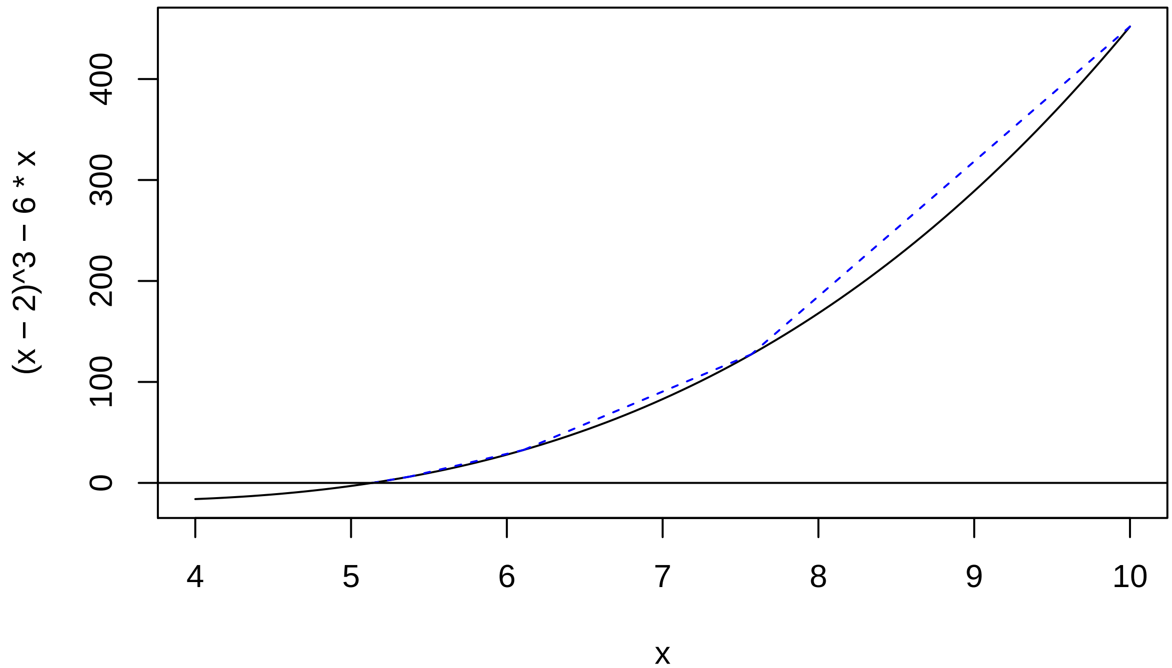
```r
f2 <- function(x) {
  return(c((x - 2)**3 - 6*x, 3*(x - 2)**2 - 6))
}

a <- NewtonRaphson(f2, 10, 1e-3, 40)
```

```
##
## Iteration  1 :   X= 7.56989247311828   Y= 127.379330322233
## Iteration  1 :   X= 7.56989247311828   Y= 87.071106486299
## Iteration  2 :   X= 6.10695791925136   Y= 32.6307361646652
## Iteration  2 :   X= 6.10695791925136   Y= 44.6013100515045
## Iteration  3 :   X= 5.3753485533609   Y= 6.20318000683711
## Iteration  3 :   X= 5.3753485533609   Y= 28.1789335700266
## Iteration  4 :   X= 5.15521318197001   Y= 0.48003626931828
## Iteration  4 :   X= 5.15521318197001   Y= 23.8661106710319
## Iteration  5 :   X= 5.13509946189492   Y= 0.00382129831852751
```

```
## Iteration  5 :   X= 5.13509946189492   Y= 23.4865459079214
## Iteration  6 :   X= 5.1349367603068   Y= 2.48970923877323e-07
## Iteration  6 :   X= 5.1349367603068   Y= 23.4834854733687
## Found the root point:  5.1349367603068  after  6 iterations
```

```
curve((x - 2)**3 - 6*x, 4, 10)
for(i in 1:(nrow(a) - 1)) {
  segments(x0 = a[i, 2], y0 = a[i, 3], x1 = a[i + 1, 2], y1 = a[i + 1, 3], col = "blue", lty = 2)
}
abline(h=0)
```




```
dev.off()
```

```
## null device
##           1
```