Assignment2_2.R

Francesco Ignazio Re

Fri Feb 02 12:42:39 2018

```
#1.a

tmpFn1 <- function(xVec){

  return(xVec ^ (1 : length(xVec)))

}


tmpFn2 <- function(xVec){

  return((xVec ^ (1 : length(xVec))) / (1 : length(xVec)))

  }


#1.b

tmpFn3 <- function(x, n){

  return (1 + sum(x ^ (1 : n) / (1 : n)))

}

#2

tmpFn <- function(xVec){

 x <- xVec

 y <- c(1 : (length(x) - 2))

 for(i in 1 : (length(x) - 2)) {

   y[i] <- (x[i] + x[i + 1] + x[i + 2]) / 3 }

  return(y)
}

tmpFn(c(1 : 5, 6 : 1))
```

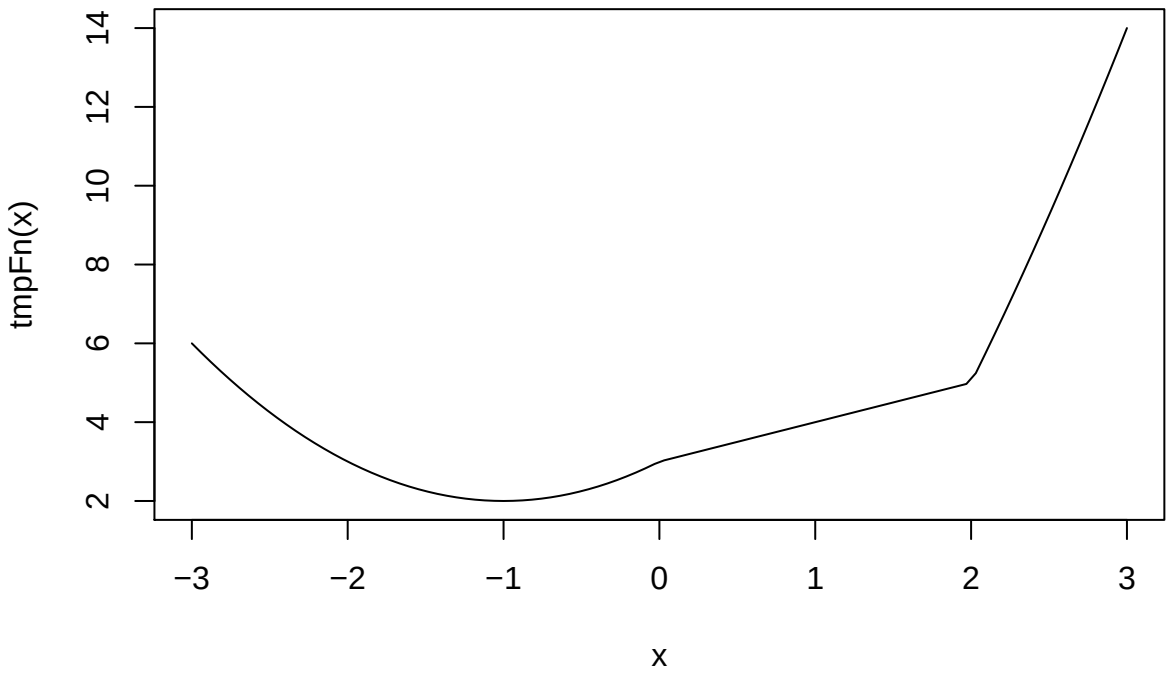
```
## [1] 2.000000 3.000000 4.000000 5.000000 5.333333 5.000000 4.000000 3.000000
## [9] 2.000000
```

```
#or

tmpFn <- function(xVec)
{
```

```
   n  <- length(xVec)

   (xVec[1 : (n - 2)] + xVec[2 : (n - 1)] + xVec[3 : n]) / 3


}

tmpFn(c(1 : 5, 6 : 1))
```

```
## [1] 2.000000 3.000000 4.000000 5.000000 5.333333 5.000000 4.000000 3.000000
## [9] 2.000000
```

```
#3

tmpFn <- function(xVec){

    ifelse(xVec < 0, xVec ^ 2 + 2 * xVec + 3, ifelse(xVec > 2, xVec ^ 2 + 4 * xVec - 7, xVec + 3))

  }

x <- seq(-3, 3, len=100)

plot(x, tmpFn(x), type="l")
```




```
#4

mat_1 <- function(A){
```

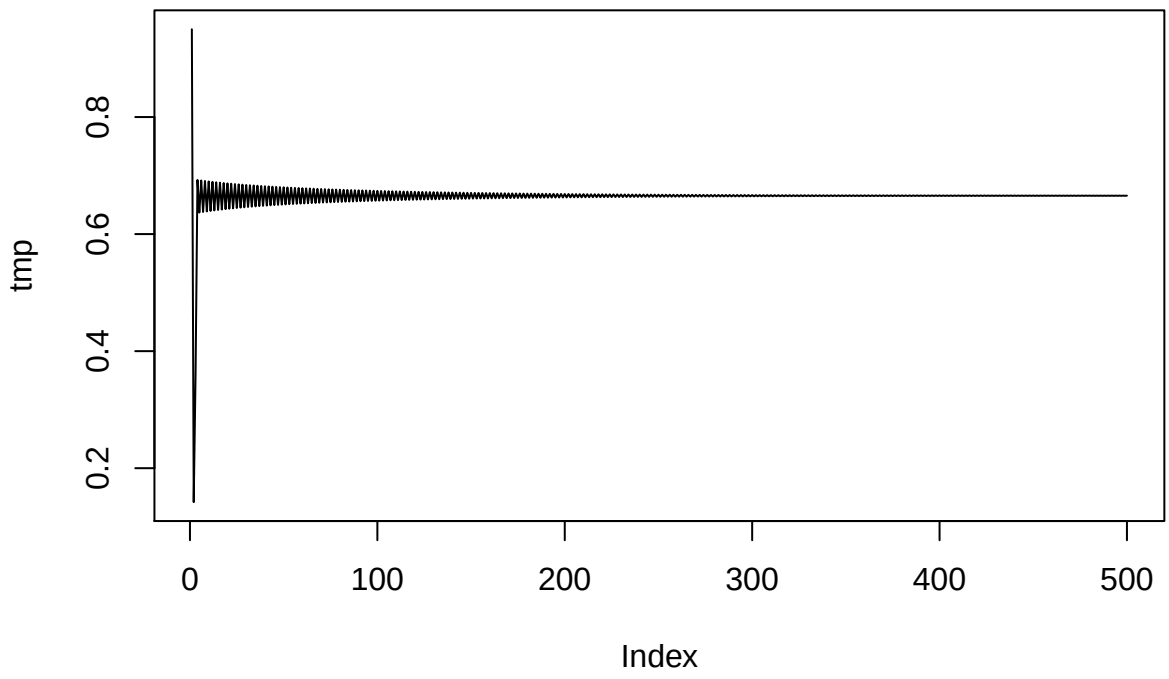
```r
  Ao <- as.vector(A)

  A1 <- ifelse(Ao %% 2 == 1, A * 2, A)

  return(matrix(A1, dim(A)))

}

A <- matrix(c(1, 1, 3, 5, 2, 5, -2, -1, -3), byrow= TRUE, nrow= 3)

mat_1(A)
```

```
##      [,1] [,2] [,3]
## [1,]    2    2    6
## [2,]   10    2   10
## [3,]   -2   -2   -6
```

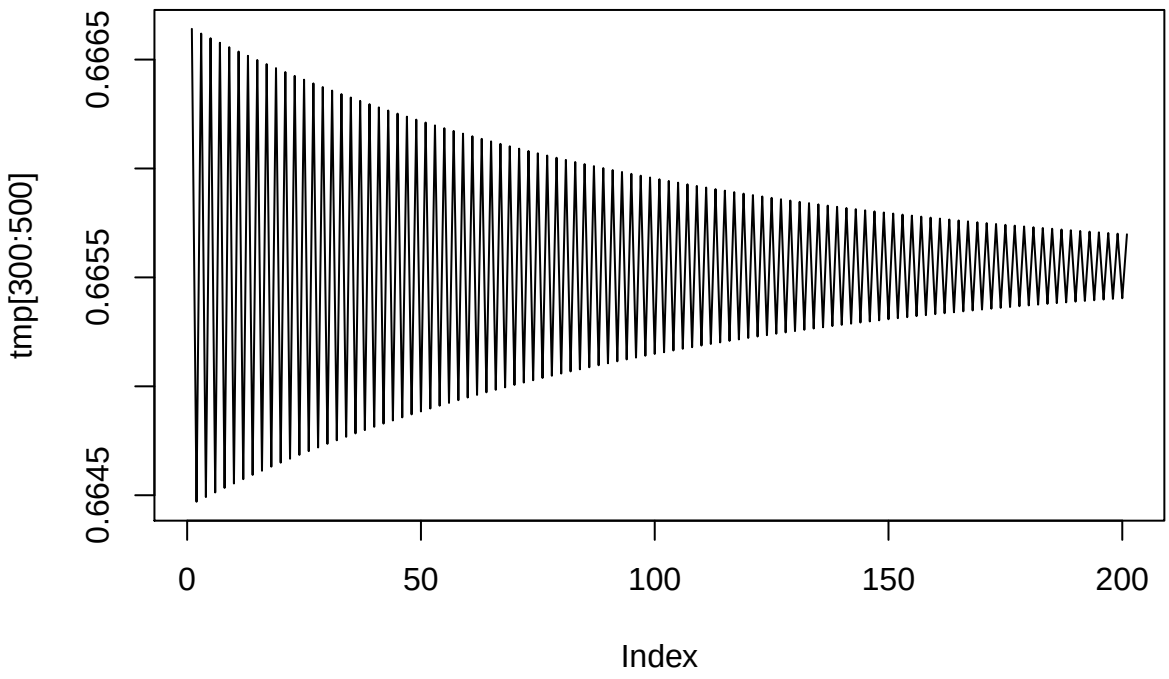
```r
#5
fun1 <- function(n, k){

  A1 <- diag(x = k, nrow = n)

  A1[abs(col(A1) - row(A1)) == 1] <- 1

  return(A1)

}

#6
quadrat <- function(alpha){

  return(1 + (alpha %% 360) %/% 90)

}

#7

weekday <- function(day, month, year){

  month <- month -2

  if(month <= 0){

      month <- month + 12

      year  <- year - 1

  }

  c <- year %/% 100

  y <- year - c * 100

  f <- (as.integer(2.6 * month - 0.2) + day + y + as.integer(y / 4) + as.integer(c / 4) - 2 * c)
```

```r
  return(c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")[f %% 7 + 1])
}

#in this case vectors don't work because of the if statement

#The problem gets solved introducing a logical variable

weekday <- function(day, month, year){

  tmp   <- month <= 2              #Boolean value that tells us whether the condition is true or not

  month <- month - 2 + 12 * tmp # 2 = Feb <- 2 - 2 + 12 = 12

  year  <- year - tmp

  c <- year %/% 100

  y <- year %% 100

  f <- (floor(2.6*month - 0.2) + day + y + (y %/% 4) + as.integer(c %/% 4) - 2 * c)

  return(c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")[f %% 7 + 1])}

#8.a
testLoop <- function(n){

if(n < 4) {return(NA)}

  nl <- c(1 : (n - 1))

  nl[1] <- 1
  nl[2] <- 2

  for(i in 3 : (n - 1))  {

    nl[i] <- (nl[i - 1] + 2 / nl[i - 1])
  }

  return(nl)
}

#8.b

testLoop2 <- function(yVec){


 nl <- c(1 : length(yVec))

 e1 <- rep(exp(1), length(yVec))

 return(sum((e1 ^ nl)))
```

```
}

#9.a

quadmap <- function(start, rho, niter){

  xVec <- rep(start, niter)

  for(i in 2 : niter)

  { xVec[i] = rho * start * (1 - start)

    start = xVec[i]

  }

  xVec

}

tmp <- quadmap(start=0.95, rho=2.99, niter=500)

plot(tmp, type="l")
```



```r
plot(tmp[300 : 500], type = "l")
```




```r
#9.b

quadmap1 <- function(start, rho){


  diff <- 1

  i = 0
  while (diff > 0.02)

  {
    i = i + 1

    xplus <- rho * start * (1 - start)

     diff <- abs(start - xplus)

    start <- xplus

  }

  i

}
```

```r
quadmap1(0.95, 2.99)
```

```
## [1] 84
```

```r
#10.a

tmpFn <- function(xVec){


   n <- length(xVec)

  x1 <- xVec - mean(xVec)

  xVec1 <- x1[2 : n]
  xVec2 <- x1[1 : (n - 1)]

  r1 <- sum((xVec1 * xVec2)) / sum(x1 ^ 2)

  xVec1 <- x1[3 : n]
  xVec2 <- x1[1 : (n - 2)]

  r2 <- sum((xVec1 * xVec2)) / sum(x1 ^ 2)

  list(r1 = r1, r2 = r2)

}

xp <- seq(2, 56, by = 3)

tmpFn(xp)
```

```
## $r1
## [1] 0.8421053
##
## $r2
## [1] 0.6859649
```

```r
#10.b

tmpFnG <- function(xVec, k)
{


  n  <- length(xVec)

  x1 <- xVec - mean(xVec)

  rk <- function(k){

  sum(x1[(k + 1) : n] * x1[1 : (n - k)]) / (sum(x1 ^ 2))   }

  c(1, sapply(seq(1 : k), rk))
```

```
}
```