Exercises 3

Doug Goon

April 27, 2018

1.

(a)

```r
tmpFn1 <- function(xVec){
  return(xVec^(1:length(xVec)))
}

tmpFn2 <- function(xVec2){
  n = length(xVec2)
  return(xVec2^(1:n)/(1:n))
}
```

```r
x <- c(2, 5, 3, 8, 2, 4)
tmpFn1(x)
```

```
## [1]    2   25   27 4096   32 4096
```

```r
tmpFn2(x)
```

```
## [1]    2.0000   12.5000    9.0000 1024.0000    6.4000  682.6667
```

(b)

```r
tmpFn3 <- function(x3,n){
  1 + sum((x3^(1:n))/(1:n))
}
tmpFn3(1,3)
```

```
## [1] 2.833333
```

2.

```r
tempFn <- function(xVec){
  n <- length(xVec)
  (xVec[1:(n-2)] + xVec[2:(n-1)] + xVec[3:n])/3
}
tempFn(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.

```r
tmpFn_Q3 <- function(xVec){
  ifelse(xVec < 0, xVec^2 + 2*xVec + 3, ifelse(xVec < 2, xVec + 3, xVec^2 + 4*xVec - 7))
}
```

4.

```r
oddmatrix <- function(aMat){
  aMat[aMat%%2 == 1] <- 2 * aMat[aMat%%2 ==1]
}
```

5.

```
tmpFn_Q5 <- function(n,k){
  emat <- diag(k, nr = n)
  emat[abs(row(tmp) - col(tmp)) == 1] <- 1
  emat
}
```

6.

```
quadrant <- function(alpha){
 floor(alpha/90)%%4 + 1
}
```

7.

(a)

```
weekday <- function(day,month,year){
  month <- month - 2
  if (month <= 0) {
      month <- month + 12
      year <-  year -1
  }

  century <- as.integer(substring(as.character(year*100), 1, 2) )
  year <- year %% 100
  tmp <- floor(2.6*month - 0.2) + day + year + year %/% 4 + century %/% 4 - 2 * century
  c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")[1+tmp%%7]
}

weekday(01,02,2018)
```

```
## [1] "Thursday"
```

(b)

```
weekday_vec <- function(day,month,year){
  adj <- month <= 2
  month <- month - 2 + 12*adj
  year <- year - adj
  century <- as.integer(substring(as.character(year*100), 1, 2) )
  year <- year %% 100
  tmp <- floor(2.6*month - 0.2) + day + year + year %/% 4 + century %/% 4 - 2 * century
  c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")[1+tmp%%7]
  adj
}
```

8.

(a)

```
testloop <- function(n){
  xVec <- rep(NA, n-1)
    xVec[1] <- 1
    xVec[2] <- 2
  for( j in 3:(n-1) )
    xVec[j] <- xVec[j-1] + 2/xVec[j-1]
  xVec
}
```

(b)

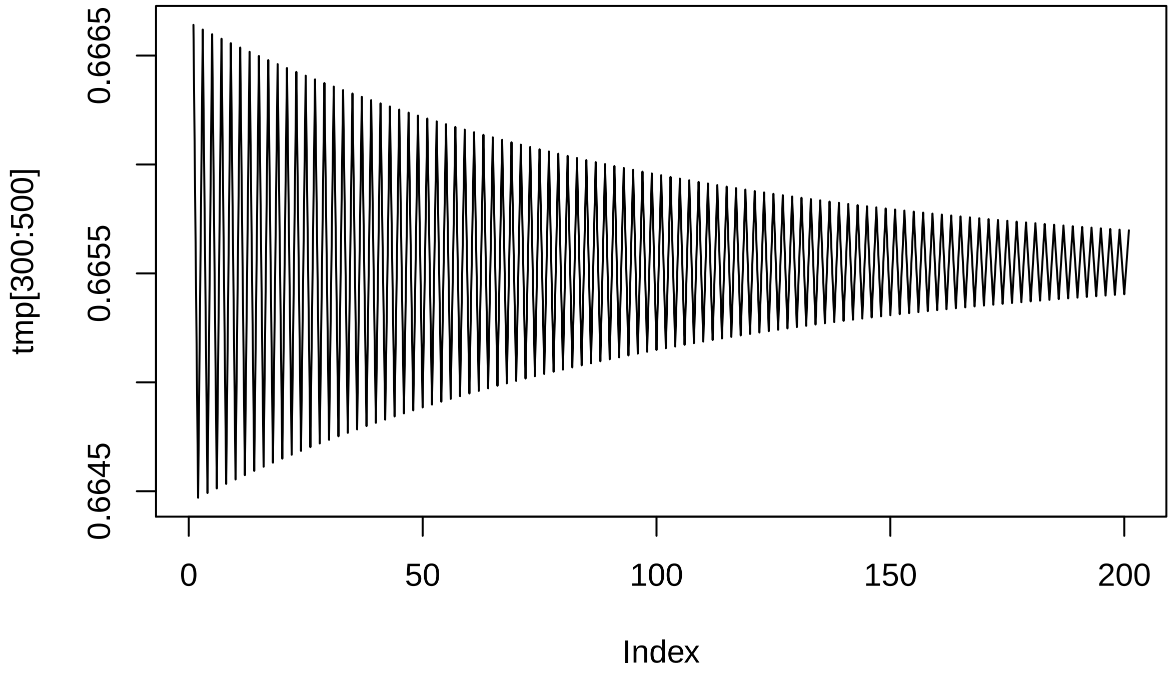
```
testloop2 <- function(yVec){
  n <- length(yVec)
  if(n <= 0)
    {
      0
    }
  else
    {
      sum(exp(1:n))
    }
}
```

9.

(a)

```
quadmap <- function(start,rho,nither){
  xVec <- rep(NA,nither)
    xVec[1] <- start
  for(i in 1:(nither-1))
    {
      xVec[i+1] <- rho* xVec[i] * (1 - xVec[i])
    }
  xVec
}

tmp <- quadmap(0.95,2.99,500)
plot(tmp[300:500], type="l")
```



(b)

```
quadmap2 <- function(start,rho){
  x1 <- start
  x2 <- rho*x1*(1-x1)
  nither <- 1
  while(abs(x1-x2)>=0.02){
    x1 <- x2
    x2 <- rho*x1*(1-x1)
    nither <- nither + 1
  }
  nither
}

quadmap2(0.95,2.99)
```

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

10.

(a)

```
tmpFn_10A <- function(xVec){
  xdiff <- xVec - mean(xVec)
  n <- length(xVec)
  r1 <- sum(xdiff[2:n]*xdiff[1:(n-1)])/sum(xdiff^2)
  r2 <- sum(xdiff[3:n]*xdiff[1:(n-2)])/sum(xdiff^2)
  list(r1=r1,r2=r2)
}
xv <- seq(2,56, by=3)

tmpFn_10A(xv)
```

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

(b)

```
tmpFn_10B <- function(xVec,k){
  xdiff <- xVec - mean(xVec)
  n <- length(xVec)
  div <- sum(xdiff^2)
  tmpFn <- function(j){ sum( xdiff[(j+1):n] * xdiff[1:(n-j)] )/div }
c(1, sapply(1:k, tmpFn))
}
xv <- seq(2,56, by=3)

tmpFn_10B(xv,3)
```

```
## [1] 1.0000000 0.8421053 0.6859649 0.5333333
```