aissgnment2

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Exercise 2

1

```
#(a)
A = matrix(data = c(1,1,3,5,2,6,-2,-1,-3),nrow = 3,ncol = 3,byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    3
## [2,]    5    2    6
## [3,]   -2   -1   -3
```

```
A^3
```

```
##      [,1] [,2] [,3]
## [1,]    1    1   27
## [2,]  125    8  216
## [3,]   -8   -1  -27
```

```
#(b)
A[,3] <- A[,2]+A[,3]
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    4
## [2,]    5    2    8
## [3,]   -2   -1   -4
```

2

```
B = matrix(data = rep(c(10,-10,10),times = 15),nrow = 15,byrow = TRUE)
B
```

```
##       [,1] [,2] [,3]
##  [1,]   10  -10   10
##  [2,]   10  -10   10
##  [3,]   10  -10   10
##  [4,]   10  -10   10
##  [5,]   10  -10   10
##  [6,]   10  -10   10
##  [7,]   10  -10   10
##  [8,]   10  -10   10
##  [9,]   10  -10   10
## [10,]   10  -10   10
## [11,]   10  -10   10
## [12,]   10  -10   10
## [13,]   10  -10   10
## [14,]   10  -10   10
## [15,]   10  -10   10
```

```
crossprod(B,B)
```

```
##       [,1]  [,2]  [,3]
## [1,]  1500 -1500  1500
## [2,] -1500  1500 -1500
## [3,]  1500 -1500  1500
```

3

```
matE = matrix(rep(0,36),nrow = 6,byrow = TRUE)
matE
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    0    0    0    0    0
## [2,]    0    0    0    0    0    0
## [3,]    0    0    0    0    0    0
## [4,]    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0
```

```
row(matE)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    1    1    1    1    1
## [2,]    2    2    2    2    2    2
## [3,]    3    3    3    3    3    3
## [4,]    4    4    4    4    4    4
## [5,]    5    5    5    5    5    5
## [6,]    6    6    6    6    6    6
```

```
col(matE)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    2    3    4    5    6
## [2,]    1    2    3    4    5    6
## [3,]    1    2    3    4    5    6
## [4,]    1    2    3    4    5    6
## [5,]    1    2    3    4    5    6
## [6,]    1    2    3    4    5    6
```

```
matE[abs(row(matE)-col(matE))==1] <- 1
matE
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    1    0    0    0    0
## [2,]    1    0    1    0    0    0
## [3,]    0    1    0    1    0    0
## [4,]    0    0    1    0    1    0
## [5,]    0    0    0    1    0    1
## [6,]    0    0    0    0    1    0
```

4

```
a = 0:4
outer(a,a,"+")
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    2    3    4
## [2,]    1    2    3    4    5
## [3,]    2    3    4    5    6
## [4,]    3    4    5    6    7
## [5,]    4    5    6    7    8
```

5

```
#(a)
outer(0:4,0:4,"+")%%5
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    2    3    4
## [2,]    1    2    3    4    0
## [3,]    2    3    4    0    1
## [4,]    3    4    0    1    2
## [5,]    4    0    1    2    3
```

```
#(b)
outer(0:9,0:9,"+")%%10
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    1    2    3    4    5    6    7    8     9
##  [2,]    1    2    3    4    5    6    7    8    9     0
##  [3,]    2    3    4    5    6    7    8    9    0     1
##  [4,]    3    4    5    6    7    8    9    0    1     2
##  [5,]    4    5    6    7    8    9    0    1    2     3
##  [6,]    5    6    7    8    9    0    1    2    3     4
##  [7,]    6    7    8    9    0    1    2    3    4     5
##  [8,]    7    8    9    0    1    2    3    4    5     6
##  [9,]    8    9    0    1    2    3    4    5    6     7
## [10,]    9    0    1    2    3    4    5    6    7     8
```

```
#(c)
M <- outer(8:0,8:0,"+")%%9
M <- M[, c(9, (1:ncol(M))[-9])]
#t(outer(8:0,8:0,"+")%%9)
apply(M, 2, rev)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
##  [1,]    0    8    7    6    5    4    3    2    1
##  [2,]    1    0    8    7    6    5    4    3    2
##  [3,]    2    1    0    8    7    6    5    4    3
##  [4,]    3    2    1    0    8    7    6    5    4
##  [5,]    4    3    2    1    0    8    7    6    5
##  [6,]    5    4    3    2    1    0    8    7    6
##  [7,]    6    5    4    3    2    1    0    8    7
##  [8,]    7    6    5    4    3    2    1    0    8
##  [9,]    8    7    6    5    4    3    2    1    0
```

6

```
xs = abs(outer(1:5,5:1,"+")-6)+1
ys = matrix(c(7,-1,-3,5,17),nrow = 5,byrow = TRUE)
solve(xs,ys)
```

```
##      [,1]
## [1,]   -2
## [2,]    3
## [3,]    5
## [4,]    2
## [5,]   -4
```

```
#apply(M2, 2, rev)
```

7

```
set.seed(75)
aMat <- matrix( sample(10, size=60, replace=T), nr=6)

#(a)
rowSums(aMat>4)
```

```
## [1] 4 7 6 2 6 7
```

```
#(b)
equal7 <- rowSums(aMat==7)
match(2,equal7)
```

```
## [1] 5
```

```
#(c)
myfunc <- function(v){
  #result <- list()
  for (i in c(1:9)) {
    column1 <- v[,i]
    sum1 <- sum(column1)
    #print(i+1)
    for (j in c(i:10)) {
      column2 <- v[,j]
      success <- list(i)
      #print(sum1+sum(column2))
      if(sum1+sum(column2)>75){
        #list.append(success,c(j))
        if(i!=j){
        print(c(i,j))}
        #list.append(result,success)
      }
    }
  }
  #return(result)
```

```r
}

myfunc(aMat)
```

```
## [1] 2 6
## [1] 2 8
## [1] 6 8
```

8

```r
#(a)
sum((1:20)^4) * sum(1/(3+(1:5)))
```

```
## [1] 639215.3
```

```r
#(b)
myfunc1 <- function(v){
  result <- 0
  for (i in c(1:20)) {
    for (j in c(1:5)) {
      result <- result + ((i^4)/(3+i*j))
    }
  }
  return(result)
}
myfunc1(1)
```

```
## [1] 89912.02
```

```r
#(c)
myfunc2 <- function(v){
  result <- 0
  for (i in c(1:20)) {
    for (j in c(1:i)) {
      result <- result + ((i^4)/(3+i*j))
    }
  }
  return(result)
}
myfunc2(1)
```

```
## [1] 137295.9
```

Exercises 3. Simple Functions

1

```r
#(a)
tmpFn1 <- function(v){
  lengthv <- length(v)
  return(v^c(1:lengthv))
}
```

```
tmpFn1(c(1:3))
```

```
## [1]  1  4 27
```

```
tmpFn2 <- function(v){
  lengthv <- length(v)
  return((v^c(1:lengthv))/c(1:lengthv))
}

tmpFn2(c(1:3))
```

```
## [1] 1 2 9
```

```
#(b)
tmpFn3 <- function(x,n){
 return(1+sum(x^c(1:n)/c(1:n)))
}
tmpFn3(2,10)
```

```
## [1] 238.3079
```

2

```
tmpFn <- function(xVec){
  if(is.vector(xVec)){
    lengthx <- length(xVec)
    return((xVec[c(1:(lengthx-2))]+xVec[c(2:(lengthx-1))]+xVec[c(3:(lengthx))])/3)
  }
}

#tmpFn(c(1,3,6,7,8,9,12,23))
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
```

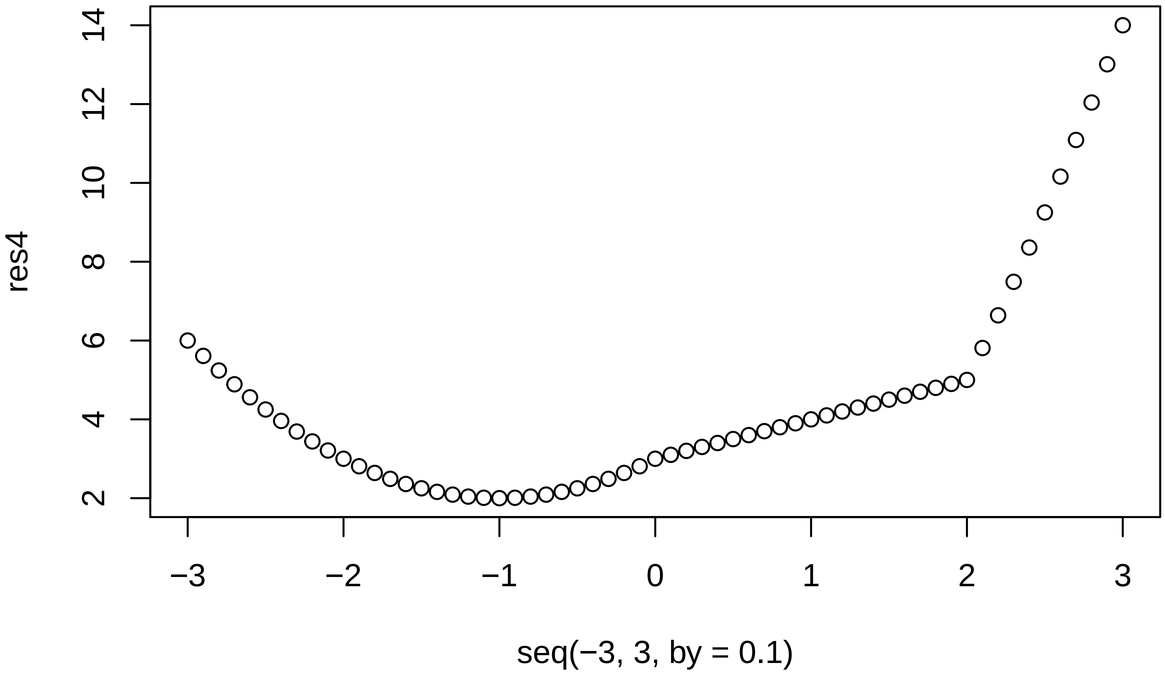
3

```
tmpFn4 <- function(xVec){
  for (i in c(1:length(xVec))) {
    varx <- xVec[i]
    #print(varx)
    if(varx<0){
      xVec[i] <- (varx^2)+(2*varx)+3
    }
    else if(varx>=0 && varx<2){
      xVec[i] <- varx+3
    }
    else{
      xVec[i] <- (varx^2)+(4*varx)-7
    }
  }
  return(xVec)
}

res4 <- tmpFn4(seq(-3,3,by=0.1))
res4
```

```
##  [1]  6.00  5.61  5.24  4.89  4.56  4.25  3.96  3.69  3.44  3.21  3.00
## [12]  2.81  2.64  2.49  2.36  2.25  2.16  2.09  2.04  2.01  2.00  2.01
## [23]  2.04  2.09  2.16  2.25  2.36  2.49  2.64  2.81  3.00  3.10  3.20
## [34]  3.30  3.40  3.50  3.60  3.70  3.80  3.90  4.00  4.10  4.20  4.30
## [45]  4.40  4.50  4.60  4.70  4.80  4.90  5.00  5.81  6.64  7.49  8.36
## [56]  9.25 10.16 11.09 12.04 13.01 14.00
```

```
plot(seq(-3,3,by=0.1),res4)
```



4

```
tmpFn5 <- function(xMatrix){
  result <- apply(xMatrix, 1, function(x){ifelse(x%%2==0, x, x*2)})
  return(t(result))
}

tmpFn5(matrix(c(1,1,3,5,2,6,-2,-1,-3),nrow = 3,byrow = TRUE))
```

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

5

```r
tmpFn6 <- function(k,n){
  matE = diag(k,nrow = n,ncol = n)

  matE[abs(row(matE)-col(matE))==1] <- 1
  return(matE)
}
tmpFn6(2,5)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2    1    0    0    0
## [2,]    1    2    1    0    0
## [3,]    0    1    2    1    0
## [4,]    0    0    1    2    1
## [5,]    0    0    0    1    2
```

6

```r
quadrant <- function(alpha){
  module <- alpha%%360
  if(module>=0 && module<90){
    return(1)
  }
  else if(module>=90 && module<180){
    return(2)
  }
  else if(module>=180 && module<270){
    return(3)
  }
  else{
    return(4)
  }
}
quadrant(92)
```

```
## [1] 2
```

```r
quadrant(430)
```

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

7

```r
weekday <- function(day,month,year){
  k <-  day
  m <- (month+10)%%13
  y <- year%%100-1+(month+10)%/%12
  c <- year%/%100
  list_weekdays <- c("Saturday","Sunday","Monday","Tuesday",
                     "Wednesday","Thursday","Friday")
  names(list_weekdays) = c(0:6)
  f <- (floor(2.6*m-0.2)+k+y+floor(y/4)+floor(c/4)-2*c)%%7
  print(f)
  return(list_weekdays[as.character(f)])
}
weekday(2,2,2018)
```

```
## [1] 6
```

```
##        6
## "Friday"
```

```
#(b)
#yes! it works
weekday(c(2,4),c(2,2),c(2018,2018))
```

```
## [1] 6 1
```

```
##        6        1
## "Friday" "Sunday"
```