STAT 579 Homework 6

Yifan Zhu

October 25, 2016

Problem 1

(a)

```
h1 <- function(x, n) {
    s <- 0
    for (i in 0:n) s <- s + x^i
    return(s)
}
```

(b)

```
h2 <- function(x, n) {
    s <- 0
    i <- 0
    while (i < n + 1) {
        s <- s + x^i
        i <- i + 1
    }
    return(s)
}
```

(c)

```
system.time(h1(0.3, 500))
```

```
##    user  system elapsed
##       0       0       0
```

```
system.time(h2(0.3, 500))
```

```
##    user  system elapsed
##       0       0       0
```

```
system.time(h1(1.01, 500))
```

```
##    user  system elapsed
##       0       0       0
```

```
system.time(h2(1.01, 500))
```

```
##    user  system elapsed
##       0       0       0
```

```
system.time(h1(0.3, 5000))
```

```
##    user  system elapsed
##    0.01    0.00    0.01
```

```
system.time(h2(0.3, 5000))
```

```
##    user  system elapsed
##    0.02    0.00    0.02
```

```
system.time(h1(1.01, 5000))
```

```
##    user  system elapsed
##       0       0       0
```

```
system.time(h2(1.01, 5000))
```

```
##    user  system elapsed
##    0.00    0.00    0.02
```

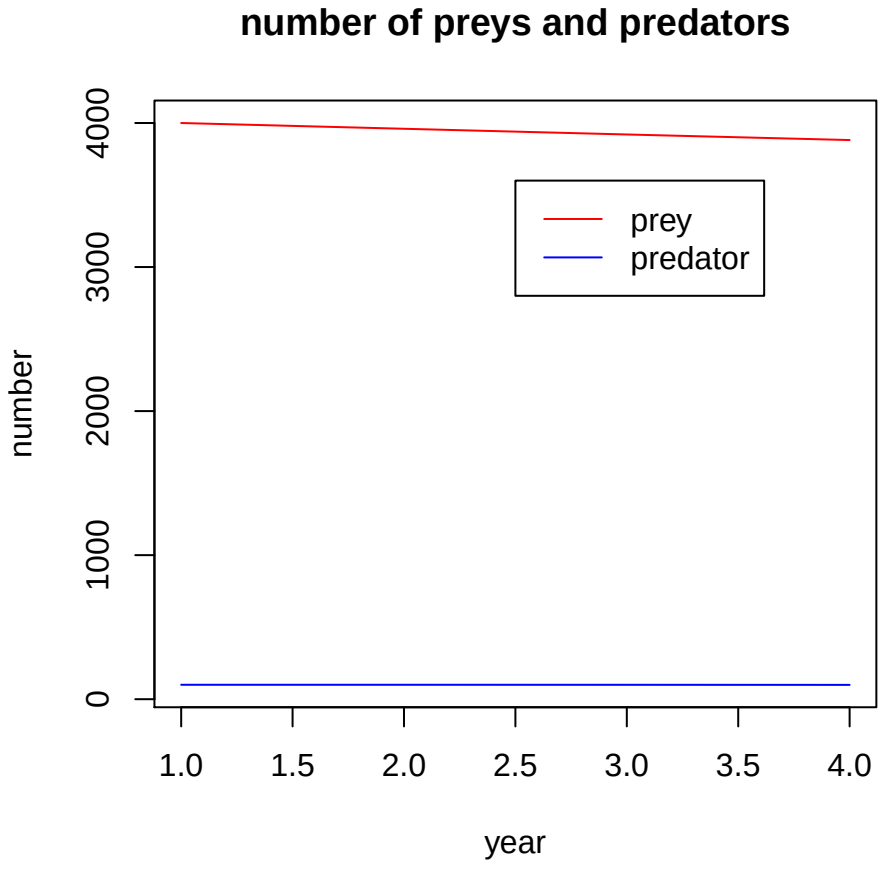
(d)

```
h <- function(x, n) {
    return(sum(x^(0:n)))
}

h(0.3, 500)
```

```
## [1] 1.428571
```

Problem 2

```
x <- 4000
y <- 100
while (x[length(x)] > 3900) {
    xt <- x[length(x)]
    yt <- y[length(y)]
    xt1 <- xt + 0.04 * xt - 5e-04 * xt * yt
    yt1 <- yt + 0.1 * 5e-04 * xt * yt - 0.2 * yt
    x <- c(x, xt1)
    y <- c(y, yt1)
}

numbers <- list(prey = x, predator = y)

t <- 1:length(x)

plot(0, 0, xlim = c(1, 4), ylim = c(100, 4000), "n", xlab = "year", ylab = "number",
    main = "number of preys and predators")
lines(x = t, y = numbers$predator, col = "blue")
lines(x = t, y = numbers$prey, col = "red")
legend(2.5, 3600, c("prey", "predator"), lty = 1, col = c("red", "blue"))
```



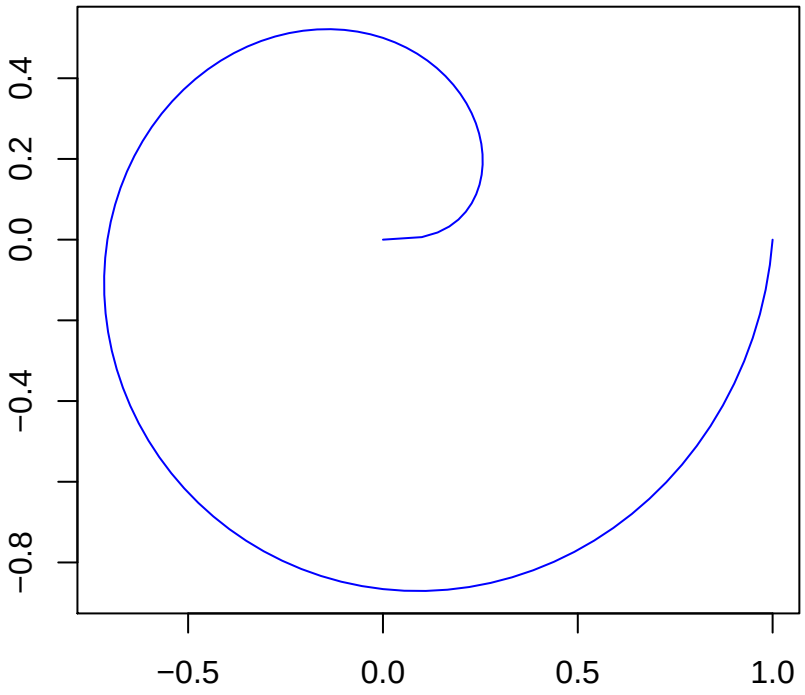
Problem 3

```
i <- 1
dice1 <- sample(x = 1:6, size = 1)
dice2 <- sample(x = 1:6, size = 1)
dicesum0 <- dice1 + dice2
if (dicesum0 == 7 || dicesum0 == 11) {
    cat("trial 1 : \n number on dice 1 :", dice1, "\n number on dice 2 :", dice2,
        "\n sum of the two dices :", dicesum0, "\n Player 1 wins")
} else {
    cat("trial 1 : \n number on dice 1 :", dice1, "\n number on dice 2 :", dice2,
        "\n sum of the two dices :", dicesum0, "\n")
    repeat {
        i <- i + 1
        dice1 <- sample(x = 1:6, size = 1)
        dice2 <- sample(x = 1:6, size = 1)
        dicesum <- dice1 + dice2
        if (dicesum == dicesum0) {
            cat("trial", i, ": \n number on dice 1 :", dice1, "\n number on dice 2 :",
                dice2, "\n sum of the two dices :", dicesum, "\n Player 1 wins")
            break
        } else {
            if (dicesum == 7 || dicesum == 11) {
                cat("trial", i, ": \n number on dice 1 :", dice1, "\n number on dice 2 :",
                  dice2, "\n sum of the two dices :", dicesum, "\n Player 2 wins")
                break
            } else {
                cat("trial", i, ": \n number on dice 1 :", dice1, "\n number on dice 2 :",
                  dice2, "\n sum of the two dices :", dicesum, "\n")
            }
        }
    }
}
```

```
## trial 1 :
##  number on dice 1 : 6
##  number on dice 2 : 2
##  sum of the two dices : 8
## trial 2 :
##  number on dice 1 : 6
##  number on dice 2 : 5
##  sum of the two dices : 11
##  Player 2 wins
```

Problem 4

```r
t <- seq(0, 1, 0.01)
r <- sqrt(t)
theta <- 2 * pi * t

# convert to the cartesian coordinate
x <- r * cos(theta)
y <- r * sin(theta)

plot(x, y, "l", col = "blue", xlab = "", ylab = "")
```



Problem 5

```
x <- matrix(rnorm(n = 500), ncol = 5)
varx <- var(x)

# find the standard deviation
sdx <- sqrt(diag(varx))

# divide each row with the standard deviation
R1 <- sweep(x = varx, MARGIN = 1, STATS = sdx, FUN = "/")

# divide each column with the standard deviation
R <- sweep(x = R1, MARGIN = 2, STATS = sdx, FUN = "/")

R
```

```
##              [,1]        [,2]        [,3]         [,4]         [,5]
## [1,]  1.000000000 -0.19049964 -0.14304716  0.003637058  0.005327144
## [2,] -0.190499635  1.00000000  0.08594290 -0.027312473 -0.045047270
## [3,] -0.143047161  0.08594290  1.00000000 -0.079907873  0.065013284
## [4,]  0.003637058 -0.02731247 -0.07990787  1.000000000 -0.136115795
## [5,]  0.005327144 -0.04504727  0.06501328 -0.136115795  1.000000000
```

Problem 6

Part a

i.

```
temp <- array(scan(file = "C:/Users/fanne/Desktop/STAT579/STAT579hw6/etcpod_05-400_102307_1_trig01.dat"
    dim = c(72, 72, 150))
```

ii. A. Estimation of background noise.

```
# average the first five frame
background <- apply(X = temp[, , 1:5], MARGIN = c(1, 2), FUN = mean)
```

B. Elimination of background noise.

```
temp <- sweep(x = temp, MARGIN = c(1, 2), STAT = background)
```

C. Identifying frame with hottest signal.

```
tempmax <- max(temp)

# find the index of frame with max temperature
maxid <- (1:150)[apply(X = temp == tempmax, MARGIN = 3, FUN = any)]

maxid
```

```
## [1] 93
```

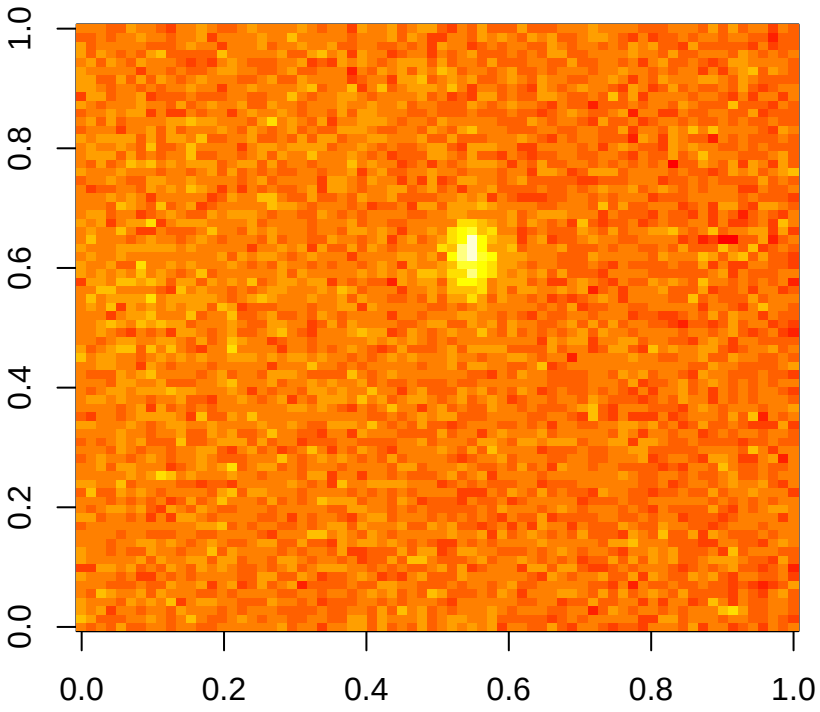
D.

```r
tempright <- array(rep(NA, 74 * 74 * 150), dim = c(74, 74, 150))
templeft <- array(rep(NA, 74 * 74 * 150), dim = c(74, 74, 150))
tempup <- array(rep(NA, 74 * 74 * 150), dim = c(74, 74, 150))
tempdown <- array(rep(NA, 74 * 74 * 150), dim = c(74, 74, 150))
tempcenter <- array(rep(NA, 74 * 74 * 150), dim = c(74, 74, 150))
tempright[2:73, 1:72, ] <- temp
templeft[2:73, 3:74, ] <- temp
tempdown[1:72, 2:73, ] <- temp
tempup[3:74, 2:73, ] <- temp
tempcenter[2:73, 2:73, ] <- temp

# combine these 5 shifted temp frames to one array
temparound <- array(c(tempcenter, tempup, tempdown, templeft, tempright), dim = c(74,
    74, 150, 5))

# compute the mean using the around temperature
tempmean <- apply(X = temparound, MARGIN = c(1, 2, 3), FUN = mean, na.rm = TRUE)

tempmean <- tempmean[2:73, 2:73, ]

tempmeanmax <- max(tempmean)

# find the index of frame with max temperature
maxmeanid <- (1:150)[apply(X = tempmean == tempmeanmax, MARGIN = 3, FUN = any)]

maxmeanid
```

```
## [1] 93
```

F.

```r
image(temp[, , maxmeanid])
```



Part b

i.

```
library(readxl)
testdata <- read_excel(path = "C:/Users/fanne/Desktop/STAT579/STAT579hw6/PreliminaryData.xlsx")

testdata <- data.frame(x = testdata$x, y = testdata$y, old = testdata$"DD (old method)",
    L1 = testdata$"DD (L1 method)")
```

ii.

```
xrange <- testdata$x[testdata$y == 0]
yrange <- testdata$y[testdata$x == 0]

xgrid <- rep(xrange, times = length(yrange))
ygrid <- rep(yrange, each = length(xrange))

any(testdata$x == xgrid)
```

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

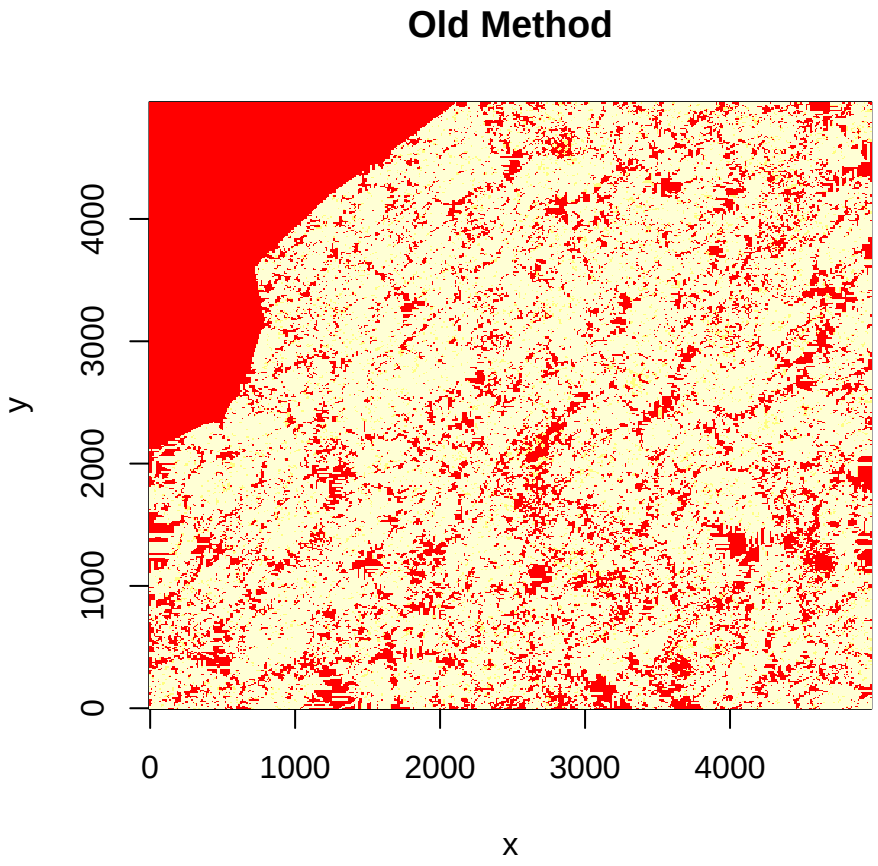
```
any(testdata$y == ygrid)
```

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

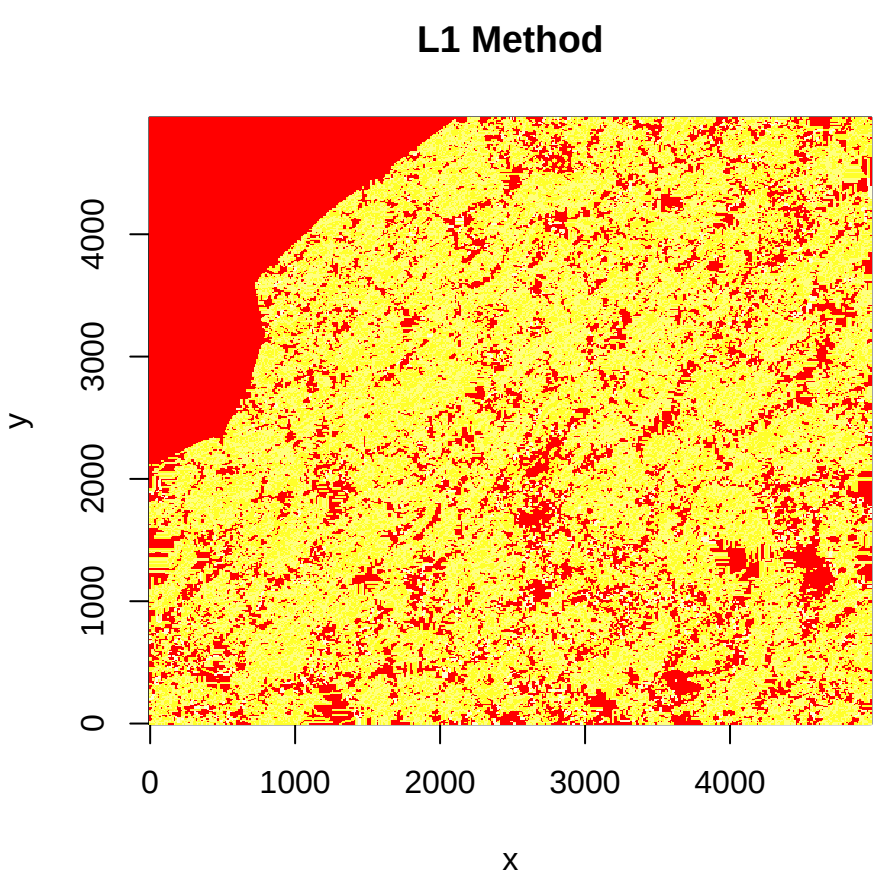
data are complete.

iii.

```
oldmat <- matrix(testdata$old, nrow = length(xrange))/10
L1mat <- matrix(testdata$L1, nrow = length(xrange))/10

image(xrange, yrange, oldmat, xlab = "x", ylab = "y", main = "Old Method")
```




```
image(xrange, yrange, L1mat, xlab = "x", ylab = "y", main = "L1 Method")
```


L1 Method
y
0
1000
2000
3000
4000
x
0
1000
2000
3000
4000