**Module 1 Homework**

Chengbo Gu

**Problem 1 (40 points)** Choose the answers in the following questions:

**(a)** What is the class of the object defined be vec <-c(5,TRUE) ?

 Numeric

 Integer

 Matrix

 Logical

> vec <- c(5,TRUE)

> class(vec)

[1] "numeric"

**(b)** Suppose I have vectors x <- 1:4 and y <- 1:2. What is the result of the expression x + y?

 A numeric vector with the values 1, 2, 5, 7

 A numeric vector with the values 2, 4, 2, 4

 An integer vector with the values 2, 4, 4, 6

 An error

> x <- 1:4

> y <- 1:2

> x+y

[1] 2 4 4 6

**(c)** Suppose I define the following function in R: fsin<-function(x) sin(pi\*x)

What will be returned by fsin(1) ?

 The number 0 is returned

 The number 1 is returned

 A warning is given with no value returned

 An error is returned because 'pi' is not specified in the call to 'fsin'

> fsin <- function(x) sin(pi\*x)

> fsin(1)

[1] 1.224606e-16

**(d)** What is returned by the R command c(1,2) %\*% t(c(1,2)) ?

 The number 5

 A one by two matrix

 A two by two matrix

 An error is returned because the dimensions mismatch

> c(1,2)%\*%t(c(1,2))

[,1] [,2]

[1,] 1 2

[2,] 2 4

**(e)** Suppose I define the following function in R:

Consider the following function:

f <- function(x) {

g <- function(y) {

y + z

}

z <- 4

x + g(x)

}

If I then run in R the following statements

z <- 15

f(3)

What value is returned?

 16

 7

 10

 4

> f <- function(x){

+ g <- function(y){

+ y+z

+ }

+ z <- 4

+ x + g(x)

+ }

> z <- 15

> f(3)

[1] 10

**Problem 2 (20 points)**

Use R to calculate       

Please hand in your R commands and the results you produce by running those commands.

> sum <- 0

> for (i in 1:1000) {

+ sum <- sum + i^2

+ }

> sum

[1] 333833500

The result of the summation is 333833500.

**Question 3 (40 points)**

Write an R script that does all of the following:

**a)** Create a vector X of length 20, with the kth element in X = 2k, for k=1…20. Print out the values of X.

[1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40

**b)** Create a vector Y of length 20, with all elements in Y equal to 0. Print out the values of Y.

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

**c)** Using a “for” loop, reassigns the value of the k-th element in Y, for k = 1…20. When k < 12, the kth element of Y is reassigned as the cosine of (3k). When the k ≥ 12, the kth element of Y is reassigned as the value of integral .

[1] -0.98999250 0.96017029 -0.91113026 0.84385396 -0.75968791 0.66031671

[7] -0.54772926 0.42417901 -0.29213881 0.15425145 -0.01327675 27.71281603

[13] 31.24811456 34.92213953 38.72983781 42.66667146 46.72853567 50.91169396

[19] 55.21272615 59.62848609

The script is showed below:

rm(list=ls())

X **<-** c()

**for** (i **in** 1:20){

X[i] **<-** 2\*i

}

print(X)

Y **<-** rep(0, 20)

print(Y)

integrand **<-** **function**(x) sqrt(x)

**for** (k **in** 1:20){

**if** (k < 12){

Y[k] = cos(3\*k)

}

**else** {

Y[k] = integrate(integrand, lower = 0, upper = k)$value

}

}

print(Y)