MATH7340

Module 1 Homework

Problem 1 (32 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

The class of the object defined by vec <- c(5, TRUE) is "Numeric." In R, if you generate a vector with members of various data types, R will convert those parts into a single data type.

(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

In R, the phrase x + y will result in: A numeric vector with the values 2, 4, 4, 6.

(c) 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

A two by two matrix.

The command c(1, 2) generates a numeric vector with the values 1 and 2.This vector is transposed by t(c(1, 2)), which results in a 2x1 matrix. A matrix multiplication of c(1, 2)%*% t(c(1, 2)) is then carried out.

(d) Suppose I define the following function in R:

```
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

10

The answer to the above question is 10.

Problem 2 (10 points)
Use R to calculate =

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

Code:
```
n <- 1000
result <- n*((n + 1) * (2 * n + 1)) / 6
print(paste("The result of ∑(x=1)^1000 x^2 =", result))
```

Output:
```
[1] "The result of ∑(x=1)^1000 x^2 = 333833500"
```

Question 3 (18 points)

This exercise is to make sure all of you understand how to create a vector in R and do simple operations. **All parts should be done using "R"** obviously.

Consider a group of 10 randomly selected people of **different** ages.

a) Create a vector named "age" to represent this. You can pick any **reasonable** age (whole numbers only please) for each person.

```
> age <- c(21, 25, 31, 35, 41, 45, 51, 55, 61, 65)
```

b) Multiply each person's age by 12 (to convert into months). (the answer should be a vector, hope you know this)

```
> age_converts_to_months <- age * 12
```

c) Find the sum of ages of all these people.

```
> total_sum_of_age_of_all_months <- sum(age_converts_to_months)
```

d) Find the age of the youngest person.

```
> youngest_among_of_all <- min(age)
```

e) Find the age of the oldest person.

```
> oldest_among_of_all <- max(age)
```

f) Find the square root of the age of each person. (Not sure what this means, but who cares?) (this also should be a vector)

```
> squareroot_of_age <- sqrt(age)
```

Question 4 (40 points)

Write an R script that does all of the following:

g) Create a vector X of length 30, with the k^{th} element in X = 3k, for k=1…30. Print out the values of X.

Code:
```
X <- 3 * (1:30)
```

print(X)

Output:
[1] 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75 78 81 84 87 90

h) Create a vector Y of length 30, with all elements in Y equal to 0. Print out the values of Y.

Code:
```
Y <- rep(0, 30)
cat(" Y:", Y, "\n")
```

Output:
Y: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

i) Using a "for" loop, reassigns the value of the k-th element in Y, for k = 1…30. When k < 20, the k^{th} element of Y is reassigned as the sine of (2k). When the k ≥ 20, the k^{th} element of Y is reassigned as the value of integral $\sqrt{\quad}$. (You may want to use $value at the end of the line to get the integration with R clean out unwanted values)

Code:

```
the_y_vector <- rep(0, 30)
for (k in 1:30) {
  the_y_vector[k] <- if (k < 20) sin(2 * k) else integrate(function(x) x, lower = 0, upper = k)$value
}
cat("Vector Y:", the_y_vector, "\n")
```

Output:
Vector Y: 0.9092974 -0.7568025 -0.2794155 0.9893582 -0.5440211 -0.5365729 0.9906074 -0.2879033 -0.7509872 0.9129453 -0.008851309 -0.9055784 0.7625585 0.2709058 -0.9880316 0.5514267 0.5290827 -0.9917789 0.2963686 200 220.5 242 264.5 288 312.5 338 364.5 392 420.5 450