

# Assignment1

2023-01-16

Problem 1 (a) This type of vector is numeric used type of function to predict the class of the object

```
vec <-c(5,TRUE)
typeof(vec)
```

```
## [1] "double"
```

Problem 1 (b) To calculate the expression of given vectors x+y

```
x <- 1:4
y <- 1:2
x+y
```

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

Problem 1 (c) the R command `c(1,2) %*% t(c(1,2))` The code returns a two by two matrix

```
c(1,2) %*% t(c(1,2))
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    2    4
```

Problem 1 (d) Running the provided R statement gives the output as 10

```
f <- function(x) {
  g <- function(y) {
    y+z
  }
  z<-4
  x+g(x)
}

z<-15
f(3)
```

```
## [1] 10
```

Problem 2 Use R to calculate the provided expression in question

```
sum <- 0
for(x in 1:1000){

  sum = sum +x^2

}
print(sum)
```

```
## [1] 333833500
```

Question 3 (a) Create a vector named “age” to represent reasonable age of a person

```
age_years <- c(10,15,20,25,30,35,40,45,50,55)
age_years
```

```
## [1] 10 15 20 25 30 35 40 45 50 55
```

(b) Multiply each person’s age by 12 (to convert into months).

```
person_age_months <- age_years*12
person_age_months
```

```
## [1] 120 180 240 300 360 420 480 540 600 660
```

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

```
sum(person_age_months)
```

```
## [1] 3900
```

d) Find the age of the youngest person

```
min(person_age_months)
```

```
## [1] 120
```

e) Find the age of the oldest person.

```
max(person_age_months)
```

```
## [1] 660
```

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)

```
sqrt(person_age_months)
```

```
## [1] 10.95445 13.41641 15.49193 17.32051 18.97367 20.49390 21.90890 23.23790
## [9] 24.49490 25.69047
```

Question 4 (g) Create a vector X of length 30, with the kth element in  $X = 3k$ , for  $k=1 \dots 30$ . Print out the values of X.

```
X <- seq(1,30,by=3)
x1 <- 3*X
x1
```

```
## [1] 3 12 21 30 39 48 57 66 75 84
```

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

```
Y <- rep(0,30)
Y
```

```
## [1] 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  $k = 1 \dots 30$ . When k less than 20, the kth element of Y is reassigned as the sine of (2k). When the k greater than 20, the kth element of Y is reassigned as the value of integral . (You may want to use \$value at the end of the line to get the integration with R clean out unwanted values)

```
Y <- rep(0,30)
for (k in 1:30){
  if (k<20){
    Y[k] = sin(2*k)
  } else if (k>= 20){
    Y[k] = integrate(sqrt, lower = 0, upper = k)$value
  }
}
print(Y)
```

```
## [1] 0.909297427 -0.756802495 -0.279415498 0.989358247 -0.544021111
## [6] -0.536572918 0.990607356 -0.287903317 -0.750987247 0.912945251
## [11] -0.008851309 -0.905578362 0.762558450 0.270905788 -0.988031624
## [16] 0.551426681 0.529082686 -0.991778853 0.296368579 59.628486093
## [21] 64.156066931 68.792772200 73.536091612 78.383680568 83.333342688
## [26] 88.383014823 93.530754108 98.774726701 104.113197958 109.544523798
```