

Assignment 5 Solutions

BIOL 1007A

Winter 2023

Working with matrices, lists, and data frames

1. Assign to the variable `n_dims` a single random integer between 3 and 10.

```
set.seed(752)
n_dims<-sample(3:10, 1)
```

- Create a vector of consecutive integers from 1 to `n_dims`.

```
vec <- seq(1:n_dims)
```

- Use the `sample` function to randomly reshuffle these values.

```
randVec <- sample(vec)
```

- create a square matrix with these elements.

```
m<-matrix(data=randVec, nrow=n_dims, ncol=n_dims)
```

- print out the matrix.

```
print(m)
```

- find a function in `R` to transpose the matrix.
- print it out again and note how it has changed.

```
t(m) ##rows become columns
```

- calculate the sum and the mean of the elements in the first row and the last row.

```
sum(m[1,]); mean(m[1,])
sum(m[n_dims,]); mean(m[n_dims,])
```

- if have set your code up properly, you should be able to re-run it and create a matrix of different size because `n_dims` will change.

This means you should not have used any integer values when setting up the matrix. Use the `n_dims` variable instead so the code can run repeatedly with different values.

2. Create a list with the following named elements:

- `my_matrix`, which is a 4 x 4 matrix filled with random uniform values
- `my_logical` which is a 100-element vector of TRUE or FALSE values. Do this efficiently by setting up a vector of random values and then applying an inequality to it.
- `my_letters`, which is a 26-element vector of all the lower-case letters in random order.

```
my_matrix <- matrix(data=runif(16), nrow=4, ncol=4)
my_logical <- as.logical(sample(0:1, 100, replace=T))
my_letters <- sample(tolower(LETTERS))
```

- create a new list, which has the element[2,2] from the matrix, the second element of the logical vector, and the second element of the letters vector.

```
myList <- list(my_matrix[2,2], my_logical[2], my_letters[2])
#myList
```

- use the `typeof()` function to confirm the underlying data types of each component in this list

```
typeof(myList[[3]])
```

```
## [1] "character"
```

```
str(myList)
```

```
## List of 3
## $ : num 0.45
## $ : logi FALSE
## $ : chr "x"
```

- combine the underlying elements from the new list into a single atomic vector with the `c()` function.
- what is the data type of this vector?

```
myVec<- c(myList[[1]],myList[[2]],myList[[3]])
typeof(myVec[[3]]) # transformed to character
```

```
## [1] "character"
```

3. Create a data frame with following two variables (= columns) and 26 observations (= rows) below:

- call the first variable `my_unis` and fill it with a random uniform value from 0 to 10
- call the second variable `my_letters` and fill it with 26 capital letters in random order.

```
df<-data.frame(my_unis=runif(26, min=0, max=10),my_letters=sample(LETTERS))
```

- for the first variable, use a single line of code in R to select 4 random rows and replace the numerical values in those rows with NA.

```
df[sample(1:nrow(df), 4),1]<-NA
```

- for the first variable, write a single line of R code to identify which rows have the missing values.

```
which(!complete.cases(df))
```

- for the second variable, sort it in alphabetical order

```
df[order(df$my_letters),]
```

- calculate the column mean for the first variable.

```
mean(df$my_unis, na.rm =T)
```

```
## [1] 4.005096
```

- rename the second column 'my_LETTERS'

```
names(df)
```

```
## [1] "my_unis" "my_letters"
```

```
names(df)[2]<- "my_LETTERS"  
names(df)
```

```
## [1] "my_unis" "my_LETTERS"
```