

HW 4: Vectors and Matrices

Due Date: Mar 13, 2019

1 Assessments guide

1. Turn in homeworks for grading on or before the “ **Due Date** ” on D2L.

2. **Assignments Etiquette:**

- (a) Create a **script file for every homework**. Every homework should have the file name starting with your last name, your first initial and then assignment number i.e. HW#. For example:

sahiR-HW1.R

- (b) Every script file in the beginning should have the following:

```
# HW (The correct number)
# Raman Sahi (Your name here)
```

- (c) Make sure you have numbered the questions correctly using the # symbol.

- (d) There should be some space between any two questions.

- (e) *Label/name your homework appropriately and give detailed headings, comments descriptions etc to your solutions.*

2 Questions

1. Create the following vectors using `c()`, `:`, `seq()`, `rep()` and show the output. You can use the help feature of R.

- (a) `v1 = (2 3 4 5 6 7 -1 -2 -3 -4)`
- (b) `v2 = (5 6 7 8 9 10 11 12)`
- (c) `v3 = ("Hard work" "Practice" "Perseverance" "Success")`
- (d) `v4 = (TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE)`
- (e) `v5 = (4 4 4 4 4 4 4)` **Note:** Use the function `rep()`

2. Apply the following to vectors given in Q.1 and show the output.

- (a) `length(v4)` # Length of the vector v4
- (b) `length(v5)` # Length of the vector v5
- (c) `sum(v1)` # sum all the elements of vector v1
- (d) `range(v1)` # range of the elements of vector v1
- (e) create a new vector: `v6 <- c(v2, v5, -1)`
- (f) `length(v6)`
- (g) `v1[8]*v2[4]`
- (h) `v1[8]/v2[4]`
- (i) create a new vector: `v7 <- v2[2:3]` # creating a new vector by using a range of values with the help of a colon (`:`) operator
- (j) create a new vector: `v8 <- v2[4:8]`
- (k) create a new vector: `v9 <- v6[6:10]`
- (l) `v8 * v9`

3. In here we will learn to do sorting of numeric and character vectors. Create vectors given below and then do the following.

```
vnum <- c(-11, 0, 45, 22:25, -6:-2, -1, 11)
```

```
vchar <- c("dollars", "rupees", "pound", "naira", "dinar", "sol")
```

- (a) Using the help feature of R, write 3-4 lines about the function: `sort()`
- (b) Using the `sort()`, write the vectors: `vnum` and `vchar` in increasing order.
- (c) Using the `sort()`, write the vectors: `vnum` and `vchar` in decreasing order.

4. **Matrix Computations.** In here, you will learn how to

- Create matrices of different sizes: Square Matrix (number of rows = number of columns) and Rectangular Matrix (number of rows \neq number of columns)
- Add row and column names to the matrices
- Access elements of the matrices
- Add, Subtract, Multiply and Divide element wise the matrices

(a) Create a Square matrix by writing the command:

```
matSq <- matrix(1:9, nrow=3, ncol=3)
```

(b) Print the above square matrix, i.e., show the output.

(c) Type the following commands that shows the various elements of **matrix: matSq**

- i. `matSq[2, 3]`
- ii. `matSq[1,]`
- iii. `matSq[,3]`

(d) Apply row and column names to the **matrix: matSq** by typing the command:

```
rownames = c("row1", "row2", "row3")
```

```
colnames = c("col1", "col2", "col3")
```

Now type the following and see the results.

- i. `matSqnam <- matrix(matSq, nrow = 3, ncol=3, dimnames = list(rownames, colnames))`
`print(matSqnam)`
- ii. `matSqnam[1,]`
- iii. `matSqnam[,3]`
- iv. What do you notice when you see the output for Q.4. c (ii, iii) and Q. 4. d (ii, iii). Write your observations.

(e) Create a rectangular matrix called: `matRec` by writing the commands:

```
rowRecnames = c("row1", "row2", "row3", "row4")
```

```
colRecnames = c("col1", "col2", "col3")
```

```
matRec <- matrix(1:12, nrow=4, ncol=3, byrow = TRUE, dimnames = list(rowRecnames, colRecnames))
```

```
matRec
```

(f) Add, subtract, multiply and divide the matrices. **Note:** These arithmetic operations are done on matrices element wise. Type the following commands.

- i. `matSq + matSq`
- ii. `matSq + matRec`

- iii. `matSq - matSq`
- iv. `matSq - matRec`
- v. `matSq * matSq`
- vi. `matSq * matRec`
- vii. `matSq / matSq`
- viii. `matSq / matRec`
- ix. What do you notice when you see the output for Q.4. f (ii, iv, vi, viii). Write your observations.

5. **Matrix Constructions and Deconstruction.** In here, you will learn how to

- Transpose a matrix. It means that rows and columns of a matrix are flipped. For example, if a matrix has row =4 and column =3, then the transpose matrix will have row = 3 and column = 4. Matrix transpose is created by the function: `t()`
 - Combine matrices columns and rowwise by using the function: `cbind()` and `rbind()` respectively.
 - Deconstruct a matrix, i.e., create a vector from a matrix
- (a) Create a transpose of matrix by writing the command:
`matRecT <- t(matRec)`
- (b) Print the above square matrix, i.e., show the output.
- (c) Type the following commands that shows the various elements of **matrix: matSq**
- i. `matRec[2, 3]`
 - ii. `matRecT[2, 3]`
 - iii. `matRecT[3, 2]`
 - iv. What do you notice when you see the output for Q.5. c (i, ii, iii). Write your observations.
 - v. `matRec[4,]`
 - vi. `matRecT[4,]`
- (d) Combine Matrices by typing the command:
- i. `cbind(matSq, matSq)`
 - ii. `cbind(matSq, matRec)`
 - iii. `cbind(matSq, matRecT)`
 - iv. What do you notice when you see the output for Q.4. d (iii). Write your observations