

All About Vectors

Introduction to R

Preliminaries

After each question, you will see the following:

```
# FILL ME IN
```

This, in R Markdown parlance, is a “code chunk.” To answer the question, replace this line with your answer. Note that anything following a “#” symbol is a comment (or is code that is “commented out”). Also note that you do not need to remove the question or make other edits. Just fill in the code chunks.

To run the chunk to see if it works, simply put your cursor *inside* the chunk and, e.g., select “Run Current Chunk” from the “Run” pulldown tab. Alternately, you can click on the green arrow at the upper right-most part of the chunk, or use “<cntl>-<return>” as a keyboard shortcut.

Here is an example (that only makes sense if you are looking at the .Rmd file):

Question 0

Print “Hello, world.”

```
print("Hello, world.")
```

```
## [1] "Hello, world."
```

When you have finished answering the questions, you can click on the “Knit” button to see a “compiled” version of your answers. Note that R Markdown may prompt you to install packages to get the knitting to work; do install these.

Question 1

Initialize a vector `y` with one logical value, one numeric value, and one character value, and determine the type of `y`.

```
y <- c(TRUE, 3.14, "a")
typeof(y)
```

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

Question 2

Sort the vector `y` into ascending order. Note how the data are ordered.

```
sort(y)
```

```
## [1] "3.14" "a"      "TRUE"
```

y is sorted in lexicographic order: numbers first, then letters in alphabetical order.

Question 3

Initialize a vector y of integers, with first value 4 and last value -4, stepping down by 1. Do this *two* different ways. After each initialization, print the vector.

```
# For instance:  
y <- 4:-4  
y
```

```
## [1]  4  3  2  1  0 -1 -2 -3 -4
```

```
y <- seq(4,-4,by=-1)  
y
```

```
## [1]  4  3  2  1  0 -1 -2 -3 -4
```

```
y <- c(4,3,2,1,0,-1,-2,-3,-4)  
y
```

```
## [1]  4  3  2  1  0 -1 -2 -3 -4
```

Question 4

What variable type is 1? Divide 1 by 2. Note to yourself whether you get zero or 0.5.

```
typeof(1)
```

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

```
1/2
```

```
## [1] 0.5
```

Question 5

Take the vector x defined below and display the elements that are less than -1 or greater than 1 . Do this using the logical or symbol, and again via the use of the `abs()` function (for absolute value).

```
set.seed(199)
x <- rnorm(20)
x[x< -1|x>1]
```

```
## [1] -1.909143 -2.216337 -1.132455 -1.763385  1.291574
```

```
x[abs(x)>1]
```

```
## [1] -1.909143 -2.216337 -1.132455 -1.763385  1.291574
```

Question 6

What proportion of values in the vector x are less than 0.5 ? Use `sum()` and `length()` in your answer.

```
sum(x<0.5)/length(x)
```

```
## [1] 0.5
```

Question 7

Sort all the values of x in *decreasing* order. Do this two different ways.

```
sort(x,decreasing=TRUE)
```

```
## [1]  1.29157391  0.98448947  0.95177053  0.80563530  0.75756594  0.73688424
## [7]  0.58058560  0.56505908  0.55516667  0.54160719  0.49414548  0.27542518
## [13] -0.04973667 -0.06946347 -0.29888471 -0.58057099 -1.13245520 -1.76338525
## [19] -1.90914272 -2.21633653
```

```
rev(sort(x))
```

```
## [1]  1.29157391  0.98448947  0.95177053  0.80563530  0.75756594  0.73688424
## [7]  0.58058560  0.56505908  0.55516667  0.54160719  0.49414548  0.27542518
## [13] -0.04973667 -0.06946347 -0.29888471 -0.58057099 -1.13245520 -1.76338525
## [19] -1.90914272 -2.21633653
```

Question 8

Replace all positive values in the vector x with zero, using `which()`.

```
w <- which(x>0)
x[w] <- 0
```