

Data Camp

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Section 1: Variables

```
a <- "Devon"
b1 <- 45.6
b2 <- "45.6"
c1 <- 0:3
```

Questions 1-6

1. Character
2. Numeric
3. Character
4. It says error in b1 + b2 : non-numeric argument to binary operator. This is because the variables have not both been assigned numeric values.
5. b1 and c1 are different variables, as b1 is numeric and c1 is an integer.
6. When adding b1 and c1, it produces 4 different number outcomes. This is because both variables have numeric values and are able to work together, and since 0:3 is an integer for 4 different values, R added 45.6 to each integer.

Section 2: Vectors

```
v1_vector <- c(-2, -1, 0, 1, 2)
v2_vector <- c(-6, -3, 0, 3, 6)
sum(v2_vector)
```

```
## [1] 0
```

Questions 7-9

7. v1_vector <- c(-2, -1, 0, 1, 2)
8. v2_vector <- c(-6, -3, 0, 3, 6)
9. sum(v2_vector)

Section 3: Matrices

```
vec_4 <- c(1:12)

mat_1 <- matrix(vec_4, nrow = 3, ncol = 4, byrow = TRUE)
mat_1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
```

```
mat_2 <- matrix(vec_4, nrow = 3, ncol = 4, byrow = FALSE )
mat_2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```

Questions 10-11

10. mat_1 <- matrix(vec_4, nrow = 3, ncol = 4, byrow = TRUE)
11. mat_2 <- matrix(vec_4, nrow = 3, ncol = 4, byrow = FALSE)

Section 3: Lists

```
my_list_1 <- list(5.2, "five_point_two", 0:5)
names(my_list_1) <- c("two", "one", "three")
my_list_1$three
```

```
## [1] 0 1 2 3 4 5
```

```
my_list_1[1]
```

```
## $two
## [1] 5.2
```

Questions 12-14

12. my_list_1 <- list(5.2, "five_point_two", 0:5)
13. my_list_1\$three
14. my_list_1\$[1]

Section 4: Logical Test

subset to a logical vector, only pulls out the elements that correspond to values of TRUE

```
my_vec = rep(1:3, 5)
my_vec
```

```
## [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
my_vec == 3
```

```
## [1] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE
## [13] FALSE FALSE TRUE
```

```
my_bool_vec <- my_vec == 3
my_vec[my_bool_vec]
```

```
## [1] 3 3 3 3 3
```

```
my_bool_vec
```

```
## [1] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE
## [13] FALSE FALSE TRUE
```

```
my_bool_vec==my_vec
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE
```

```
data.frame(my_vec, my_bool_vec)
```

```
##      my_vec my_bool_vec
## 1         1      FALSE
## 2         2      FALSE
## 3         3       TRUE
## 4         1      FALSE
## 5         2      FALSE
## 6         3       TRUE
## 7         1      FALSE
## 8         2      FALSE
## 9         3       TRUE
## 10        1      FALSE
## 11        2      FALSE
## 12        3       TRUE
## 13        1      FALSE
## 14        2      FALSE
## 15        3       TRUE
```

Questions 15-16

15. my_bool_vec <- my_vec == 3
16. my_vec[my_bool_vec]