DS311 - Basic R Lab Exercise

R Lab Exercise

Brian Solis

4/27/2025

Basic R Exercise

Section 1 - Data Type

Key Functions - typeof() - as.numeric() - as.character()

Numeric

```
n1 <- 15
n1
```

```
## [1] 15
```

```
typeof(n1)
```

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

```
n2 <- 1.5
n2
```

```
## [1] 1.5
```

```
typeof(n2)
```

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

Character

```
c1 <- "c"
c1
```

```
## [1] "c"
```

```r
typeof(c1)
```

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

```r
c2 <- "a string of text"
c2
```

```
## [1] "a string of text"
```

```r
typeof(c2)
```

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

Logical

```r
l1 <- TRUE
l1
```

```
## [1] TRUE
```

```r
typeof(l1)
```

```
## [1] "logical"
```

```r
l2 <- F
l2
```

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

```r
typeof(l2)
```

```
## [1] "logical"
```

Transforming Numerics and Characters

```r
num <- 10
numToChar <- as.character(num)
paste("num Type: ", typeof(num), " | numToChar: ", typeof(numToChar))
```

```
## [1] "num Type:  double  | numToChar:  character"
```

```r
char <- "10"
charToNum <- as.numeric(char)
paste("char Type: ", typeof(char), " | charToNum: ", typeof(charToNum))
```

```
## [1] "char Type:  character  | charToNum:  double"
```

Challenge:

```
a <- as.integer(500)
b <- as.double(500)
c <- as.character(500)

# Checking data types
typeof(a)
```

```
## [1] "integer"
```

```
typeof(b)
```

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

```
typeof(c)
```

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

```
is.double(a)
```

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

```
# Check data type of variable 'd'
d <- a / b
typeof(d)
```

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

```
is.character(d)
```

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

Section 2 - Data Structure

- is.vector()
- is.matrix()
- cbind()
- as.data.frame()

Vector

```r
v1 <- c(1, 2, 3, 4, 5)
v1
```

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

```r
is.vector(v1)
```

```
## [1] TRUE
```

```r
v2 <- c("a", "b", "c")
v2
```

```
## [1] "a" "b" "c"
```

```r
is.vector(v2)
```

```
## [1] TRUE
```

```r
v3 <- c(TRUE, TRUE, FALSE, FALSE, TRUE)
v3
```

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

```r
is.vector(v3)
```

```
## [1] TRUE
```

Matrix

```r
m1 <- matrix(c(T, T, F, F, T, F), nrow = 2)
m1
```

```
##      [,1]  [,2]  [,3]
## [1,] TRUE FALSE  TRUE
## [2,] TRUE FALSE FALSE
```

```r
is.matrix(m1)
```

```
## [1] TRUE
```

```r
m2 <- matrix(c("a", "b",
               "c", "d"),
               nrow = 2,
               byrow = T)
m2
```

```
##      [,1] [,2]
## [1,] "a"  "b"
## [2,] "c"  "d"
```

```r
is.matrix(m2)
```

```
## [1] TRUE
```

Challenge:

```r
# Vector of lowercase alphabet
alphabet_lower <- letters
alphabet_lower
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```r
# 2x13 Matrix of uppercase alphabet
alphabet_upper_matrix <- matrix(LETTERS, nrow = 2, byrow = TRUE)
alphabet_upper_matrix
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,] "A"  "B"  "C"  "D"  "E"  "F"  "G"  "H"  "I"  "J"   "K"   "L"   "M"
## [2,] "N"  "O"  "P"  "Q"  "R"  "S"  "T"  "U"  "V"  "W"   "X"   "Y"   "Z"
```

DataFrame

```r
vNumeric   <- c(1, 2, 3)
vCharacter <- c("a", "b", "c")
vLogical   <- c(T, F, T)

df1 <- cbind(vNumeric, vCharacter, vLogical)
df1
```

```
##      vNumeric vCharacter vLogical
## [1,] "1"      "a"        "TRUE"
## [2,] "2"      "b"        "FALSE"
## [3,] "3"      "c"        "TRUE"
```

```r
df2 <- as.data.frame(cbind(vNumeric, vCharacter, vLogical))
df2
```

```
##   vNumeric vCharacter vLogical
## 1        1          a     TRUE
## 2        2          b    FALSE
## 3        3          c     TRUE
```

Section 3 - Setup Working Directory and Installing Packages

Key Functions: - getwd() - setwd() - install.packages() - library()

Setting up your working directory

```
# wd1 <- getwd()
# paste("Current Working Directory: ", wd1)

# setwd("c://.../project")
# wd2 <- getwd()
# paste("Current Working Directory: ", wd2)
```

Installing and Loading Packages

Section 4 - Problem Solving

```
# Part a
x <- 4

# Part b
y <- 12

# Part c
print(paste("x:", x, "| y:", y))
```

```
## [1] "x: 4 | y: 12"
```

```
# Part d
z <- y / x

# Part e
print(paste("y divided by x is equal to", z))
```

```
## [1] "y divided by x is equal to 3"
```

```
# Do not need to change the following code!
if (exists("x") == TRUE | exists("y") == TRUE | exists("z") == TRUE){
  if (x == 4 & y == 12 & z == 3) {
    print("Congratulation!!  You completed the first exercise in this section!!")
  } else {
    print("Sorry, you got it wrong!")
  }
} else {
  print("You did not complete the last problem!")
}
```

```
## [1] "Congratulation!!  You completed the first exercise in this section!!"
```