DS311 - Basic R Lab Exercise

R Lab Exercise

Your Name

8/23/2022

Basic R Exercise

Section 1 - Data Type

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

Numeric

```r
# Numeric - Double precision by default

n1 <- 15
n1
```

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

```r
typeof(n1)
```

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

```r
n2 <- 1.5
n2
```

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

```r
typeof(n2)
```

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

Character

```r
# 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
# Logical

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
# Transforming numeric into characters
num <- 10
numToChar <- as.character(num)
paste("num Type: ", typeof(num), " | numToChar: ", typeof(numToChar))
```

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

```r
# Transforming characters into numeric
char <- "10"
charToNum <- as.numeric(char)
paste("char Type: ", typeof(num), " | charToNum: ", typeof(numToChar))
```

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

Challenge:

Complete the following tasks:

```r
# Check the data type of the following variables
a <- as.integer(500)
b <- as.double(500)
c <- as.character(500)

# Enter your code here!
typeof(a)
```

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

```r
typeof(b)
```

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

```r
typeof(c)
```

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

```r
# Check the data type of the following variable
d <- a / b

# Enter your code here!
typeof(d)
```

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

Section 2 - Data Structure

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

Vector

```
# Vector

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

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

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

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

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

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

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

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

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

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

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

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

Matrix

```
# Matrix

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

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

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

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

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

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

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

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

Challenge:

1. Create a vector of the 26 alphabet lower case letters in sequence.
2. Create a 2 by 13 matrix for the 26 English upper case letter in sequence.

Hint: Check out the "letters" and "LETTERS" key words in R.

```
# Enter your code here.
x <- c(letters[1:26])
x
```

```
##  [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"
```

```
print(LETTERS[1:13])
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M"
```

```
print(LETTERS[14:26])
```

```
##  [1] "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
```

DataFrame

```
# Data Frame

# Can combine vectors of the same length
vNumeric   <- c(1, 2, 3)
vCharacter <- c("a", "b", "c")
vLogical   <- c(T, F, T)

df1 <- cbind(vNumeric, vCharacter, vLogical)
df1  # Coerces all values to most basic data type
```

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

```
df2 <- as.data.frame(cbind(vNumeric, vCharacter, vLogical))
df2  # Makes a data frame with three different data types
```

```
##   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

```
# Check your current working directory
wd1 <- getwd()
paste("Current Working Directory: ", wd1)
```

```
## [1] "Current Working Directory:  /Users/jeffreydanford"
```

```
# Setting the working directory for a project
# setwd("c://.../project")
wd2 <- getwd()
paste("Current Working Directory: ", wd2)
```

```
## [1] "Current Working Directory:  /Users/jeffreydanford"
```

Installing packages

```
# Install a new package, note the quotation marks
install.packages("dplyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/j1/6dkt51nd5fx4whtgj_fwghkh0000gn/T//RtmpwAtFZp/downloaded_packages
```

```
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/j1/6dkt51nd5fx4whtgj_fwghkh0000gn/T//RtmpwAtFZp/downloaded_packages
```

```
# Loading the packageh, note no quotation marks
library(dplyr)
```

```
##
## Attaching package: ’dplyr’
```

```
## The following objects are masked from ’package:stats’:
##
##     filter, lag
```

```
## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
# Checking the package version
packageVersion("dplyr")
```

```
## [1] ’1.0.10’
```

```
# Checking the session info
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_1.0.10
##
## loaded via a namespace (and not attached):
##  [1] rstudioapi_0.14  knitr_1.41       magrittr_2.0.3   tidyselect_1.2.0
##  [5] R6_2.5.1         rlang_1.0.6      fastmap_1.1.0    fansi_1.0.3
##  [9] stringr_1.5.0    tools_4.2.1      xfun_0.35        utf8_1.2.2
## [13] DBI_1.1.3        cli_3.4.1        htmltools_0.5.3  assertthat_0.2.1
## [17] yaml_2.3.6       digest_0.6.30    tibble_3.1.8     lifecycle_1.0.3
## [21] vctrs_0.5.1      glue_1.6.2       evaluate_0.18    rmarkdown_2.18
## [25] stringi_1.7.8    compiler_4.2.1   pillar_1.8.1     generics_0.1.3
## [29] pkgconfig_2.0.3
```

Section 4 - Problem Solving

Write the code that accomplish the following tasks:

Part a: Assign 4 to variable x

Part b: Assign 12 to variable y

Part c: Print both x and y to check their values

Part d: Divide y by x and assign it to variable z

part e: Print a statement to report your answer in Part d.

Once you finished and knit the RMarkdown file into html file, you should be able to see the message "Congratulation!! You completed the first exercise in this section!!" in the html document.

```r
# Write your code here!
# Part a
x = 4

# Part b
y = 12

# Part c
x
```

```
## [1] 4
```

```r
y
```

```
## [1] 12
```

```r
# 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"
```

```r
# 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 activity in this class!!")
  } else {
    print("Sorry, you got it wrong!")
  }
} else {
  print("You did not complete the last problem!")
}
```

```
## [1] "Congratulation!!  You completed the first activity in this class!!"
```