Homework 1

Brady Lamson

1/28/2022

Problems 1-5:

Problem 1:

a) Try the commands pi, round(pi), round(pi, digits = 4), and trunc(pi), ceiling(pi), floor(pi). What are the results?

```
## The output of pi is 3.14159265358979
```

```
## The output of round(pi) is 3
```

```
## The output of round(pi, digits = 4) is 3.1416
```

```
## The output of trunc(pi) is 3
```

```
## The output of ceiling(pi) is 4
```

```
## The output of floor(pi) is 3
```

b) Try the commands sqrt(16), 16ˆ0.5. Are the results the same?

```
## The output of sqrt(16) is 4
```

```
## The output of 16^0.5 is 4
```

```
## Are the two commands the same? TRUE
```

c) Write a command that computes 4^3

```
4^3
```

```
## [1] 64
```

d) Try the commands log10(1000), log(1000). Try the command log2(64). What are the results?

```
## The output of log10(1000) is 3
## The output of log(1000) is 6.90775527898214
```

```
## The output of log2(64) is 6
```

e) Does the text of the help file for log() match your observations?

- Yes it does! The number next to the log is the base, so log 10 uses a base of 10. The one thing to keep in mind is that log() uses a base of e by default (exp(1) in R).

Problem 2.

Manipulate the following character vector using square brackets [ ] to accomplish the following goals.

```
1) Barry arrives (and gets in the last position of the line)
2) Steve is served (and so he leaves)
3) Pam arrives and talks her way to the front of the line (with just one item)
4) Barry gets impatient and leaves
```

```r
queue <- c("Steve", "Russell", "Alison", "Liam")
queue[length(queue) + 1] <- "Barry"
queue <- queue[-1]
queue <- c("Pam", queue)
queue <- queue[-length(queue)]
queue
```

```
## [1] "Pam"     "Russell" "Alison"  "Liam"
```

Problem 3.

```
a) Write a command that lists the objects in your Workspace.

b) Write a command that removes x from the Workspace.

c) Write a command that removes *all* the objects from your Workspace.
```

```r
w <- 6
x <- 7
y <- 8
z <- 9

ls()
```

```
## [1] "queue" "w"     "x"     "y"     "z"
```

```r
rm(x)
rm(list = ls())
```

Problem 4

Consider the below vector.

a) What is the output of x == 0

```
x <- c(3, 2, 0, 1, 4, 5, 9, 0, 6, 7, 2, 8)
x == 0
```

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

b) Write a command involving sum() and the “logical” vector x == 0 that counts the number of elements of x that are equal to 0.

```
logical_vector <- x == 0
```

```
## The output of sum(logical_vector) is 2
```

c) Write a command that determines the *proportion* of elements of x that are equal to 0, assuming you *don’t know* the number of elements in x.

```
proportion <- (sum(logical_vector) / length(logical_vector)) |>
    round(digits = 3)
```

```
## The proportion of elements of x that are equal to 0 is 0.167
```

Problem 5:

Using the following data frame:

```r
numVec <- c(2, 4, 6, 5, 9, 8, 2, 4, 7, 8)
charVec <- c("a", "b", "c", "c", "b", "c", "a", "b", "b", "c")
myData <- data.frame(x1 = numVec, x2 = charVec, stringsAsFactors = FALSE)
```

a) The following commands do the same thing:

- myData$x1
- myData[["x1"]]
- myData[[1]]

What do they do?

```
- These return the first column of the data set, which in this case is all of numVec.
```

b) What kind of object is returned by the commands in part a?

```r
is.vector(myData$x1)
```

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

If they return a *vector*, what type of vector is it?

```r
glue::glue("
           Is myData$x1 a numeric vector or character vector?
           Numeric? {is.numeric(myData$x1)}
           Character? {is.character(myData$x1)}
           ")
```

```
## Is myData$x1 a numeric vector or character vector?
## Numeric? TRUE
## Character? FALSE
```

c) What do the following commands do?

```r
myData[2, ]
```

```
##   x1 x2
## 2  4  b
```

```r
myData[, 2]
```

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

myData[2,] returns the second row of the data frame. So (4, b)

myData[, 2] returns the second column of the data frame. This will be the full vector of characters.

d) What class of object is myData?

```
glue::glue("myData is of class {class(myData)}.")
```

```
## myData is of class data.frame.
```

e) What happens when you pass myData into the summary() command?

```
summary(myData)
```

```
##        x1             x2           
##  Min.   :2.00   Length:10         
##  1st Qu.:4.00   Class :character  
##  Median :5.50   Mode  :character  
##  Mean   :5.50                     
##  3rd Qu.:7.75                     
##  Max.   :9.00                     
```

This command provides the summary statistics, length, class and mode.