Laboratory Exercise Week 1

Brian Tipton

8/23/23

Directions:

- Write your R code inside the code chunks after each question.
- Write your answer comments after the # sign.
- To generate the word document output, click the button `Knit` and wait for the word document to appear.
- RStudio will prompt you (only once) to install the `knitr` package.
- Submit your completed laboratory exercise using Blackboard's Turnitin feature. Your Turnitin upload link is found on your Blackboard Course shell under the Laboratory folder.

---

1. Create a vector of three elements `(2,4,6)` and name that vector `vec.a`. Create a second vector, `vec.b`, that contains `(8,10,12)`.
   a. Add these two vectors together and name the result `vec.c`.
   b. Create a vector, named `vec.d`, that contains only two elements `(14,20)`. Add this vector to `vec.a`. What is the result and what do you think `R` did (look up the recycling rule using `Google`)? What is the warning message that `R` gives you?
   c. Next add 5 to the vector `vec.a`. What is the result and what did `R` do? Why doesn't it give you a warning message similar to what you saw in the previous problem?

My Custom functions used from my local lab projects .Rprofile

```
source("../.Rprofile", chdir = TRUE)
```

```
catXWithString
```

```
## function (string, x, nl = TRUE, sep = " ")
## {
##     if (nl) {
##         cat(paste(string, toString(x), "\n", sep = sep))
##     }
##     else {
##         cat(paste(string, toString(x), sep = sep))
##     }
## }
```

Code chunk

```
# Insert your code for this question after this line
trial.vec <- 1:20
# trial.vec
vec.a <- c(2, 4, 6)
vec.b <- c(8, 10, 12)
vec.c <- vec.a + vec.b
```

```r
vec.d <- c(14, 20)

(vec.a + vec.d) |>
  catXWithString(string = "(VEC.a + VEC.d) -> ")
```

```
## Warning in vec.a + vec.d: longer object length is not a multiple of shorter
## object length
```

```
## (VEC.a + VEC.d) ->  16, 24, 20
```

```r
catXWithString("VEC.c: ", vec.c)
```

```
## VEC.c:  10, 14, 18
```

```r
(vec.a + 5) |>
  catXWithString(string = "(VEC.a + 5) -> ")
```

```
## (VEC.a + 5) ->  7, 9, 11
```

2. Generate the vector of even numbers `{2, 4, 6, . . . , 20}`
    a. Using the `seq()` function and
    b. Using the `a:b` shortcut and some subsequent algebra. *Hint: Generate the vector 1-10 and then multiple it by 2.*

```r
seq(from = 2, to = 20, by = 2) |>
  catXWithString(string = "ONE TO TWENTY EVENS USING SEQ:\n\t")
```

```
## ONE TO TWENTY EVENS USING SEQ:
##   2, 4, 6, 8, 10, 12, 14, 16, 18, 20
```

```r
2:20 |>
  (\(x) x[x %% 2 == 0])() |>
  catXWithString(string = "ONE TO TWENTY EVENS USING MODULO:\n\t")
```

```
## ONE TO TWENTY EVENS USING MODULO:
##   2, 4, 6, 8, 10, 12, 14, 16, 18, 20
```

3. Create a vector `y` containing `(2, 2, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8)` using the `rep()` function. You might need to check the help file for `rep()` by typing `?rep` in the console to see all of the options that `rep()` will accept. In particular, look at the optional argument `each=`.
    a. Find the mean of vector `y` using the function `mean()`.
    b. Use google search to find the function in `R` that computes the variance of a vector and find the variance of `y`.

```r
y <- rep(2^(1:3), each = 4)

catXWithString("Y: ", y)
```

```
## Y:  2, 2, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8
```

```r
mean(y) |>
  catXWithString(string = "MEAN OF Y: ")
```

```
## MEAN OF Y:  4.66666666666667
```

```r
var(y) |>
  catXWithString(string = "VARIANCE OF Y: ")
```

```
## VARIANCE OF Y:  6.78787878787879
```

4. The vector `letters` is a built-in vector to `R` and contains the lower case English alphabet.

a. Extract the 9th element of the `letters` vector.
b. Extract the sub-vector that contains the 9th, 11th, and 19th elements.
c. Extract the sub-vector that contains everything except the last two elements.

```
letters[9] |>
  catXWithString(string = "NINETH LETTER ALPHA: ")
```

```
## NINETH LETTER ALPHA:  i
```

```
letters[c(9, 11, 19)] |>
  catXWithString(
    string =
      "Alpha Subvector 9th, 11th and 19th indice: "
  )
```

```
## Alpha Subvector 9th, 11th and 19th indice:  i, k, s
```

```
head(letters, -2) |>
  catXWithString(string = "EVERYTHING BUT LAST 2: ")
```

```
## EVERYTHING BUT LAST 2:  a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x
```