Assignment 2

Maharaj Teertha Deb, 40227747

2024-02-01

Assignment 2

TextBook Problems:

Question 1:

```
Create the vector:
[1] 0 0 0 1 1 1 1 0 0 0 1 1 1 1
[34] 1 1 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1

and convert it to a factor. Identify the levels of the result, and then change the level
labels to obtain the factor:

[1] Male Male Male Female Female Female Female Male
[10] Male Female Female Female Female Male Male Male Male Male Male Female
      Female Female Female
[19] Female Female Female Male
[28] Female Male Male Male

Levels: Male Female
```

Answer:

```r
# Create the vector
vector <- c(0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
            1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1)

# Convert it to a factor
gender_factor <- factor(vector)
levels(gender_factor)[1] <- "Male"
levels(gender_factor)[2] <- "Female"

# Print the result
print(gender_factor)
```

```
##  [1] Male   Male   Male   Female Female Female Female Male   Male   Male
## [11] Female Female Female Female Female Female Male   Female Female Female
```

```
## [21] Female Male   Male   Male   Female Female Female Female Male   Male
## [31] Male   Female Female
## Levels: Male Female
```

Question 2:

```
  Use the `more.colors` vector, `rep()`, and `seq()` to create the vector

"red" "yellow" "blue" "yellow" "blue" "green"
"blue" "green" "magenta" "green" "magenta" "cyan"
```

Answer:

```
# R code to create the vector
more.colors <- c("red","yellow","blue","green","magenta","cyan")
more.colors[rep(seq(0 , 2) , times = 4) + rep( seq(0 : 3) , each = 3)]
```

```
##  [1] "red"     "yellow"  "blue"    "yellow"  "blue"    "green"   "blue"
##  [8] "green"   "magenta" "green"   "magenta" "cyan"
```

Question 3:

```
Convert the character vector from the preceding exercise to a factor.
What are the levels of the factor?
Change the "blue" label to "turquoise".
```

Answer:

```
# R code to convert the character vector to a factor
factor_vector <- factor(more.colors)
cat("Labels for the factor vector: " , labels(factor_vector))
```

```
## Labels for the factor vector:  1 2 3 4 5 6
```

```
levels(factor_vector)[levels(factor_vector) == "blue"] <- "turquoise"
factor_vector
```

```
## [1] red       yellow    turquoise green     magenta   cyan
## Levels: turquoise cyan green magenta red yellow
```

Text Book problem:

Question 4:

Use R to identify the elements of the sequence $\{2^1, 2^2, \ldots, 2^{15}\}$ that exceed the corresponding elements of the sequence $\{1^3, 2^3, \ldots, 15^3\}$.

Answer:

```
n <- 1 : 15
m1 <- 2^n # sequence 1
m2 <- n^3 # sequence 2

print("The elements in the first sequence that are greater than the last sequence:")
```

```
## [1] "The elements in the first sequence that are greater than the last sequence:"
```

```
m1[m1 > m2]
```

```
## [1]     2  1024  2048  4096  8192 16384 32768
```

Question 5:

Evaluation of a square root is achieved using the `sqrt()` function, but a warning will be issued when the argument is negative. Consider the following code which is designed to test whether a given value is positive before checking whether the square root of the value is less than 5.

```
testValue <- 7
(testValue > 0) & (sqrt(testValue) < 5)
## [1] TRUE

testValue <- -7
if (testValue > 0) {
  result <- (sqrt(testValue) < 5)
} else {
  result <- FALSE
}
result
## [1] FALSE

Modify the code so that it continues to give the correct answer but without the warning.
```

Answer:

```
testValue <- 7
(testValue > 0) && (sqrt(testValue) < 5)
```

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

```
testValue <- -7
(testValue > 0) && (sqrt(testValue) < 5) # show answer no warning.
```

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

Addictional Problems:

1:

Generate the following sequences using seq() and rep():

a:

2, 4, 6, 8, . . . , 22.

Answer:

```
seq(2 , 22 , by = 2)
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20 22
```

b:

1, 5, 25, 125, . . . , 9765625

Answer:

```
5^(0 : 10)
```

```
##  [1]       1       5      25     125     625    3125   15625   78125  390625
## [10] 1953125 9765625
```

c:

11, 12, 13, 11, 12, 13, . . . 13, 11, 12, 13 (each entry appearing 7 times).

Answer:

```
rep((11 : 13) , 7)
```

```
##  [1] 11 12 13 11 12 13 11 12 13 11 12 13 11 12 13 11 12 13 11 12 13
```

d:

1, 1, 1, 2, 2, 2, . . . , 6, 6, 6

Answer:

```r
rep((1 : 6) , each = 3)
```

```
##  [1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6
```

e:

60, 57, 54, . . . , 15, 12

Answer:

```r
seq(60, 12, by = -3)
```

```
##  [1] 60 57 54 51 48 45 42 39 36 33 30 27 24 21 18 15 12
```

f:

1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000,20000, 50000

Answer:

```r
initial <- c(1 , 2 , 5)

c(initial , initial ^ 10 , initial ^ 100 , initial ^ 1000 , initial ^ 10000)
```

```
##  [1]  1.000000e+00  2.000000e+00  5.000000e+00  1.000000e+00  1.024000e+03
##  [6]  9.765625e+06  1.000000e+00  1.267651e+30  7.888609e+69  1.000000e+00
## [11] 1.071509e+301           Inf  1.000000e+00           Inf           Inf
```

2:

Create a vector containing each of the following sequences:

a:

$\tan\left(\frac{\pi n}{7}\right)$, for $n = 0, 1, 2, \ldots, 10$.

Answer:

```r
tan(pi * 0:10 / 7)
```

```
##  [1]  0.000000e+00  4.815746e-01  1.253960e+00  4.381286e+00 -4.381286e+00
##  [6] -1.253960e+00 -4.815746e-01 -1.224647e-16  4.815746e-01  1.253960e+00
## [11]  4.381286e+00
```

b:

1, 4, 23, 122, 621, 3120, 15619, 78118, 390617.

Answer:

```r
n <- 0 : 8

c(5^n - n)
```

```
## [1]      1      4     23    122    621   3120  15619  78118 390617
```

c:

3n - 3n, for $n = 0, 1, 2, \ldots, 10$.

Answer:

```r
n <- 0 : 10

c(3^n - 3*n)
```

```
##  [1]     1     0     3    18    69   228   711  2166  6537 19656 59019
```

d:

Compute the sequence $5n \mod 3$ for $n = 0, 1, 2, \ldots, 10$ (i.e., the remainder of $5n/3$ for $n = 0, 1, 2, \ldots, 10$).

Answer:

```r
c(5 * seq(0, 10) %% 3)
```

```
##  [1]  0  5 10  0  5 10  0  5 10  0  5
```

e:

Let $S_n = \sum_{i=1}^{n} \frac{2i-1}{(-1)^{i+1}} = 1 - \frac{3}{1} + \frac{5}{1} - \frac{7}{1} + \ldots + \frac{2n-1}{(-1)^{n+1}}$. It can be shown that $\lim_{n\to\infty} S_n = \frac{\pi}{4}$. Evaluate $4S_{10}$, $4S_{100}$, and $4S_{1000}$. (Hint: use the `sum()` function).

Answer:

```r
# Function to calculate Sn
Sn <- function(n) {
  sum((2*(1:n)-1) / (-1)^(1:n))
}


S_10 <- 4 * Sn(10)
S_100 <- 4 * Sn(100)
S_1000 <- 4 * Sn(1000)


c(S_10, S_100, S_1000)
```

```
## [1]   40  400 4000
```