Homework 3

Christopher Brunswick

10/26/2021

# 1. Use the function c to create a vector with the average high   
# temperatures in January for Beijing, Lagos, Paris, Rio de Janeiro,   
# San Juan, and Toronto, which are 35, 88, 42, 84, 81, and 30 degrees   
# Fahrenheit. Call the object temp.  
  
temp <- c(35,88,42,84,81,30)  
temp

## [1] 35 88 42 84 81 30

# 2. Now create a vector with the city names and call the object city.  
  
city <- c("Beijing", "Lagos", "Paris", "Rio de Janeiro", "San Juan", "Toronto")  
city

## [1] "Beijing" "Lagos" "Paris" "Rio de Janeiro"  
## [5] "San Juan" "Toronto"

# 3. Use the names function and the objects defined in the previous   
# exercises to associate the temperature data with its corresponding   
# city.  
  
names(city) <- temp  
city

## 35 88 42 84   
## "Beijing" "Lagos" "Paris" "Rio de Janeiro"   
## 81 30   
## "San Juan" "Toronto"

# 4. Use the [ and : operators to access the temperature of the   
# first three cities on the list.  
  
temp[1:3]

## [1] 35 88 42

# 5. Use the [ operator to access the temperature of Paris and   
# San Juan.  
  
temp[5:6]

## [1] 81 30

# 6. Use the : operator to create a sequence of numbers 12,13,14,…,73.  
  
12:73

## [1] 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36  
## [26] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61  
## [51] 62 63 64 65 66 67 68 69 70 71 72 73

# 7. Create a vector containing all the positive odd   
# numbers smaller than 100.  
  
seq(1, 100, 2)

## [1] 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49  
## [26] 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

# 8. Create a vector of numbers that starts at 6, does not pass 55,   
# and adds numbers in increments of 4/7: 6, 6 + 4/7, 6 + 8/7, and so   
# on. How many numbers does the list have? Hint: use seq and length.  
  
seq(6, 55, 4/7)

## [1] 6.000000 6.571429 7.142857 7.714286 8.285714 8.857143 9.428571  
## [8] 10.000000 10.571429 11.142857 11.714286 12.285714 12.857143 13.428571  
## [15] 14.000000 14.571429 15.142857 15.714286 16.285714 16.857143 17.428571  
## [22] 18.000000 18.571429 19.142857 19.714286 20.285714 20.857143 21.428571  
## [29] 22.000000 22.571429 23.142857 23.714286 24.285714 24.857143 25.428571  
## [36] 26.000000 26.571429 27.142857 27.714286 28.285714 28.857143 29.428571  
## [43] 30.000000 30.571429 31.142857 31.714286 32.285714 32.857143 33.428571  
## [50] 34.000000 34.571429 35.142857 35.714286 36.285714 36.857143 37.428571  
## [57] 38.000000 38.571429 39.142857 39.714286 40.285714 40.857143 41.428571  
## [64] 42.000000 42.571429 43.142857 43.714286 44.285714 44.857143 45.428571  
## [71] 46.000000 46.571429 47.142857 47.714286 48.285714 48.857143 49.428571  
## [78] 50.000000 50.571429 51.142857 51.714286 52.285714 52.857143 53.428571  
## [85] 54.000000 54.571429

# 9. What is the class of the following object a <- seq(1, 10, 0.5)?  
  
a <- seq(1, 10, .5)  
a

## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0  
## [16] 8.5 9.0 9.5 10.0

class(a)

## [1] "numeric"

# 10. What is the class of the following object a <- seq(1, 10)?  
  
a <- seq(1, 10)  
a

## [1] 1 2 3 4 5 6 7 8 9 10

class(a)

## [1] "integer"

# 11. The class of class(a<-1) is numeric, not integer.   
# R defaults to numeric and to force an integer, you need to add   
# the letter L. Confirm that the class of 1L is integer.  
  
class(a <- 1L)

## [1] "integer"

# 12. Define the following vector:  
# x <- c("1", "3", "5")  
# and coerce it to get integers.  
  
x <- c("1", "3", "5")  
class(x)

## [1] "character"

y <- as.numeric(x)  
y

## [1] 1 3 5

class(y)

## [1] "numeric"

```