Lab2.R

Feipeng Huang

2022-09-23

```
#Run the following code to create a large vector containing randomly
generated integers between 1 and 12:
n = 12345
vec_1 = sample(12, n, replace = TRUE)
head(vec_1)
```

```
## [1]  1 10 11  6  8  8
```

```
#Use a logical test operator to create a Boolean vector (called vec_2) whose
entries are TRUE if the corresponding entry in vec_1 is 3 and FALSE
otherwise.
###Q1
vec_2 = c(vec_1 == 3)
#Self test: you can use vec_2 to retrieve all of the 3 elements of vec_1
using the following:
#vec_1[vec_2]
###Q2
# the vector is very large, easy to lose track

n = 12345
vec_1 = sample(12, n, replace = TRUE)
head(vec_1)
```

```
## [1]  1  7  6 11 11  5
```

```
length(vec_1)
```

```
## [1] 12345
```

```
sum(vec_1 == 3)
```

```
## [1] 1062
```

```
n = 10
vec_1 = sample(12, n, replace = TRUE)
paste0("Sum of elements with value 3: ", sum(vec_1 == 3))
```

```
## [1] "Sum of elements with value 3: 0"
```

```
###Q3
#The numbers are randomly generated each time.
###Q4
#Number and order of 3 change each time. A logical test always makes the
correct selection in a second.
```

```r
###Q5
#By-hand subsetting is very time-consuming when data sets are large.
#By-hand subsetting has more room for error.
#You do not have any code to reuse or share.

###Q6
for (i in 1:10)
{
  print(paste0("This is loop iteration: ", i))
}
```

```
## [1] "This is loop iteration: 1"
## [1] "This is loop iteration: 2"
## [1] "This is loop iteration: 3"
## [1] "This is loop iteration: 4"
## [1] "This is loop iteration: 5"
## [1] "This is loop iteration: 6"
## [1] "This is loop iteration: 7"
## [1] "This is loop iteration: 8"
## [1] "This is loop iteration: 9"
## [1] "This is loop iteration: 10"
```

```r
###Q7
n <- 20
for (i in 1:n)
{
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```

```r
###Q8
#Create an integer variable, n, that holds the value 17.
n = 17
#Write code to create a vector called vec_1 of length n. vec_1 should contain
[pseudo]randomly generated integers between 1 and 10.
vec_1 = sample(10, n, replace = TRUE) ##randomly generated integers between 1
and 10, length n (17)
vec_1
```

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

```r
for (i in 1:n) #Iterates n times (once for each element of vec_1)
{
  print(paste0("The element of vec_1 at index ", i, " is ", vec_1[i]))
#Prints a message that includes the iteration number as well as the
corresponding element of vec_1
}
```

```
## [1] "The element of vec_1 at index 1 is 9"
## [1] "The element of vec_1 at index 2 is 1"
## [1] "The element of vec_1 at index 3 is 6"
## [1] "The element of vec_1 at index 4 is 9"
## [1] "The element of vec_1 at index 5 is 10"
## [1] "The element of vec_1 at index 6 is 3"
## [1] "The element of vec_1 at index 7 is 7"
## [1] "The element of vec_1 at index 8 is 2"
## [1] "The element of vec_1 at index 9 is 9"
## [1] "The element of vec_1 at index 10 is 5"
## [1] "The element of vec_1 at index 11 is 6"
## [1] "The element of vec_1 at index 12 is 3"
## [1] "The element of vec_1 at index 13 is 1"
## [1] "The element of vec_1 at index 14 is 4"
## [1] "The element of vec_1 at index 15 is 8"
## [1] "The element of vec_1 at index 16 is 8"
## [1] "The element of vec_1 at index 17 is 5"
```

```r
###Q9
create_and_print_vec = function(n, min = 1, max = 10)
{
  vec = sample(min:max, n, replace = TRUE)
  for (i in 1:n)
  {print(paste0("The element at index ", i, " is ", vec[i]))
  }
}
create_and_print_vec(20, min = 1, max = 10)
```

```
## [1] "The element at index 1 is 9"
## [1] "The element at index 2 is 9"
## [1] "The element at index 3 is 9"
## [1] "The element at index 4 is 6"
## [1] "The element at index 5 is 9"
```

```
## [1] "The element at index 6 is 5"
## [1] "The element at index 7 is 1"
## [1] "The element at index 8 is 6"
## [1] "The element at index 9 is 5"
## [1] "The element at index 10 is 2"
## [1] "The element at index 11 is 1"
## [1] "The element at index 12 is 8"
## [1] "The element at index 13 is 6"
## [1] "The element at index 14 is 6"
## [1] "The element at index 15 is 7"
## [1] "The element at index 16 is 8"
## [1] "The element at index 17 is 5"
## [1] "The element at index 18 is 5"
## [1] "The element at index 19 is 9"
## [1] "The element at index 20 is 2"
```

```
create_and_print_vec(10, min = 100, max = 2000)
```

```
## [1] "The element at index 1 is 672"
## [1] "The element at index 2 is 1155"
## [1] "The element at index 3 is 1682"
## [1] "The element at index 4 is 583"
## [1] "The element at index 5 is 1069"
## [1] "The element at index 6 is 1764"
## [1] "The element at index 7 is 610"
## [1] "The element at index 8 is 589"
## [1] "The element at index 9 is 1592"
## [1] "The element at index 10 is 1783"
```