

Elyse McCormick
9/14/2022
ECO 602

Lab 2: R Fundamentals

Q1 (2 pts.): Show the R code you used to create vec_2.

```
vec_2 <- vec_1 == 3  
vec_1[vec_2]
```

Q2 (2 pts.): Give two reasons why determining which elements in vec_1 have value 3 by visual inspection is a bad idea.

One reason it is a bad idea to inspect for elements that have the value of 3 in vec_1 is that this vector is huge, there's no way you would be able to find everything by yourself in a vector this big, and if you did, you'd probably miss some. Second, if it's generating random integers and you're looking for 3, you could make a mistake and find something that's 33, or 333, this tells you exactly which are three, and where they're located.

Q3 (1 pt.): Why didn't you always get the same count of 3 entries each time?

Because the vector is generating random numbers each time, and generating different vectors with each time, it can only find instances of 3 when they're randomly generated.

***Q4 (3 pts.): Considering the different vectors generated each time, explain why using a logical test is a safe way to select entries with a value of 3.**

A logical test is a good way to determine entries with a value of three because they MUST equal three to be selected within the vector. This would be incredibly complicated to try to do visually, because it would be so difficult to try to keep track of how many 3s had been generated within each vector, especially if you needed to run this multiple times.

***Q5 (5 pts.): Explain why performing logical 'by hand' subsetting is very very bad practice. You may want consider re-usability of code, working with different sized data sets, and sharing code with collaborators.**

→ **Your answer should cite at least two reasons why 'by hand' subsetting is bad.**

If you as an individual can't guarantee your ability to keep track of the exact number of 3s randomly generated, that already makes 'by-hand' subsetting a bad idea. But, if you wanted to share your code with a collaborator, there's 1) no guarantee they would get the same results and 2) no guarantee they would interpret your subsetting the same way you did. Additionally, if you go to publish and your code is not re-interpretable/re-usable, then your data can be called into question.

***Q6 (3 pts.): Provide the code for your modified loop. It must run as a self-contained example on a fresh R session on my computer.**

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

Q7 (2 pts.): Provide the code for the modified loop that executes n times. It needs to be a self contained example. I should be able to set the value of n and then run your loop on my computer.

```
n <- 1:17
for (i in n)
{
  print(i)
}
```

****Q8 (4 pts.): Provide the code you used to create the n, vec_1, and the loop. As always, it should run as a stand-alone example in a fresh R session on my computer.**

```
n <- 17
vec_1 = sample(10, n, replace = TRUE)
for (i in 1:n)
{
  print(paste0("The element of vec_1 at index ", i , " is ", vec_1[i]))
}
```

Q9 (10 pts.): Provide the code you used to build your function.

➔ **To receive full credit your code must run without error on a new R session and produce output similar to the examples given in the instructions.**

```
create_and_print_vec = function(n, min = 1, max = 10)
{
  vec_1 = sample(x = min:max, size = n, replace = TRUE)

  for(i in 1:n)
  {
    print(paste0("The element at index ", i , " is ", vec_1[i]))
  }
}
create_and_print_vec(10, min = 1, max = 10)
```

**Questions discussed with Kato Csanadi-Schwartz*

***Question discussed with Matt Jusino and Kato Csanadi-Schwartz*

