Matthew Jusino

Lab 2 Response Questions

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

vec_2 = vec_1 == 3

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 is because vec_1 has 12345 values in the vector, so visual inspection for 3 or TRUE will take forever. No one has that kind of time. The next reason is human error. The longer you stare at the screen counting 3's or TRUEs, the more likely you are to accidentally add or omit from the count, skewing your results.

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

Because each time you run the lines of code it's taking a different random sample of the integers between 1 and 12, so the first time you run it there might be no threes, the next there might be 2, then the next there might be none again.

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`.

Using a logical test is a safe way to select entries with the value of 3 because each new vector is being checked for values of 3 by R. While the number of 3's in the vector will change each time you run the code as a random sample, R will use the logical test each time to pull the correct sum of 3's.

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.

Performing logical 'by hand' subsetting is a bad practice because 'by hand' is not only time-consuming depending on the size of the data set, it is also just a waste of time because when you use random sampling methods, it's going to give a different sum each time, and thus it doesn't make sense to do it by hand. Additionally, if sharing code with collaborators, they will have to do the subsetting themselves because if your code doesn't include a logical test, since it is randomly sampled, they will have to use the

'by hand' method themselves or edit the code to include a logical test, which wastes their time and effort as well.

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: ",
      print(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.

```
for (i in 1:n)
{
  n = 100
  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 = 100, max = 2000)

{

 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)
```