BIOS 545 Spring 2017

Pittard

February 6, 2017

1 Tabulation

Write a function called **mytable** that, given an input vector, produces the same output as would the **table** function that is built into R. This implies of course that you cannot use the actual **table** function or other existing counting or aggregation functions (e.g. **xtabs**) to solve this problem. Of course you can use the **table** function to verify that the output from your **mytable** function matches the output of **table**.

```
mytable <- function(somevec) {
# A function to sort a numeric vector in ascending or descending order
# INPUT: "somevec" an unsorted numeric vector
#
# OUTPUT: a named vector that presents the total number of times that each
#         unique value in the vector appears

   # Your code goes here
}
```

```
> # Let's try it on an example vector
>
> set.seed(123)
> somevec <- as.integer(runif(30,1,10))
> mytable(somevec)

1 2 3 4 5 6 7 8 9
3 1 4 1 5 4 3 2 7

> table(somevec)

somevec
1 2 3 4 5 6 7 8 9
3 1 4 1 5 4 3 2 7

> # Let's try another example
>
```

```
> set.seed(321)
> somevec <- as.integer(runif(30,1,10))
> #
> mytable(somevec)

1 2 3 4 5 6 7 8 9
1 2 4 5 3 7 2 1 5

> table(somevec)

somevec
1 2 3 4 5 6 7 8 9
1 2 4 5 3 7 2 1 5
```

Here is some general psuedocode:

```
Determine the unique values in the input vector somevec
Sort the resulting vector

  For each unique value determine how many times it appears in
      the input vector somevec - use functions you've seen previously

  After processing each unique value assign names to the return
    vector - the names and arrangement of your counts should match those
       from the output of the table function
 Return a vector with your computed counts
```