MIS 5330 Module 4 User-Defined Functions Exercises

Here are three exercises for module 4. These exercises are due in this dropbox not later than September 26th at 11:59 PM. These exercises require that you understand and integrate several important, fundamental topics and issues about data structures and functions in R. We may build on these in subsequent classes.

Rule #1: Keep it simple, simple, simple !

Rule #2: Keep it simple, simple, simple !

Rule #3: Keep it simple, simple, simple !

Get something (one line of script that you know to be "in the right direction") to 'work', then build on it. Approach this exercise with as much 'granularity' as you can muster. Begin by tackling the most basic, fundamental, and yet essential 'activity' (i.e. 'piece' of a process). Then, once that 'works,' build on it. This is a good strategy to successfully accomplish these exercises, and, is a good approach for programming in R, in general.

See the R script 'Vector_makers_exercise_stub.R' and modify this script to turn

Functions that can potentially help you: **`sample()`**.

These are not hard to do ! The hard part is getting started ! They can be fun !

Do not use the function* `vector()` *inside of any of your user-defined functions !

Vector-Makers Exercises

1. Create a user-defined function "**`vector.maker.num()`**" that generates a vector that returns **exactly** 5 randomly-generated numbers in the range 1 to 100. The function should have no formal arguments. Explore the use of the sample() function inside your vector.maker.num() function. The function should have no formal arguments.
2. Then create a user-defined function "**`vector.maker.alpha()`**" that generates a vector that returns **exactly** 5 randomly-generated lower case letters from the alphabet. (**Note:** the R object '**`letters`**' will generate all 26 lower case letters.) The function should have no formal arguments. The sample() function can help you here, too.
3. Then create a user-defined function "**`vector.maker.bool()`**" that generates a vector that returns **exactly** 5 randomly-generated values of TRUE and/or FALSE. The function should have no formal arguments. The sample() function can help you here, too.