ECON 370: Assignment 1

Drew Van Kuiken

Due: Before class on September 11th, 2024

Directions

As a reminder, LLMs and AIs are strictly forbidden from use on this assigment.

Create a R script and write code to answer the following questions. Please break the code up into four sections as the assignment does: Reading Documentation, Creating Objects, Creating Data Sets, and Indexing. For the first section, please write your answers using *comments* in an organized way that makes it clear which question you are answering. As well, make it clear where one question starts and ends. Please comment your code. While I will not be explicitly grading your style and comments, the easier your code is to read, the more likely I will be able to figure out what you're doing.

Each question is worth three points. Note that in the "Reading Documentation" section, a "question" is a bullet point. So each numbered question will be worth one point.

You may work in groups of up to three people for this homework. Please submit one script per group and clearly indicate who was in your group for the assigment.

Reading Documentation

In this section, I will be asking you to read the documentation for functions that we have yet to cover or use. Remember, if the name of a function is "fun1", you can pull up its documentation by running either help(fun1) or ?fun1 in the R Console. Also, when answering what each function does, *please do not just copy and paste the description of the function in the documentation.* Please use your own words.

- Pull up the documentation for the "log" function and answer the following questions:
  1. Describe what this function does.
  2. What are its arguments and what type should each argument be?
  3. What is the default base for the logarithm? Hint: Remember that the natural log is "log base e." Does there appear to be a difference between the log and logb function? What does the log10 function do? Please answer each part as they are all related.

- Pull up the documentation for the "rnorm" function and please answer the following questions. Note that this is a little more challanging than the above question as it pulls up the documentation for a *family* of functions and not just the "rnorm" function, so please keep this in mind.

  1. What does the *rnorm* function specifically do? Read the description in the documentation very carefully and match it with the "usage" section.
  2. What are the arguments for the "rnorm" function along with their types? Are there any default arguments? What are they?
  3. Write the code to draw 100 values from a normal distribution with mean 5 and *variance* of 4. Note: remember that the variance is the standard deviation squared.

- Pull up the documentation for the "optim" function. This will be the most challenging question. Please answer the following questions:

  1. What does the optim function do? Note that "optimization" means to maximize or minimize some function.
  2. Which arguments *must* be supplied to the optim function?
  3. The "control" argument is a list of options that control the algorithm used for the specific optimization method used. What is the default maximum number of iterations? Note that there will be three different values depending upon the "method." What is the typical value of "reltol"? Hint: Follow the instructions in the "control" argument.

Creating Objects

1. Create a vector of you and your group members' names called "group_names"
   - Note: if you are submitting alone, you can still create a vector of only one element.
2. Create a vector of the integers from 1 to 5 called "my_vec".
3. Update "my_vec" by adding 10 to each element of "my_vec".
   - Hint: make use of vectorized operations.
   - Note that you should still call this new vector "my_vec".
4. Create a vector of integers from 1995 to 2024 called "years_alive."
5. Create a vector of logicals testing if an element in the "years_alive" vector is a leap year.
6. Draw 1000 simulations from a $N(0, 1)$ distribution and call it "norm_draws".
   - Hint: use rnorm.
7. Create a vector of that is the log of the draws from item 6 which is called "log_draws".
   - Note: this will produce NaNs and that's okay.
8. Create a logical vector called "draws_NaN" that tests which elements of "log_draws" are NaNs.
9. Create a vector of integers from 1 to N, where N is the number of NaN elements in "log_draws".
10. Use *modular arithmetic operations* to create a vector called "class_length" whose first element is the number of whole hours in one of our class meetings and the second is the number of leftover minutes. E.g. 50 would be "c(0,50)".
    - Note: Our class meetings are 50 minutes long.

Creating Data Sets

1. Create a variable called "mtcars_list" that is each variable in mtcars as a different in the list.
    - Hint: use as.list()
2. Create a 2 × 13 matrix of the letters of the alphabet where the letters appear from *from left to right*. Call this matrix "letter_mat".
3. Create a matrix that is 5 × 5 of new $N(0, 1)$ draws called "norm_draws_mat".
    - Hint: How many draws will we need for a 5 × 5 matrix?
4. Using the "iris" data set, store the Species variable as a factor variable named "iris_species" where "versicolor" is the first level, "virginica" is the second, and "setosa" is the third.
5. Create a sequence of numbers from 1.2 to 5.3 by 0.05 called "first_seq".
6. Create a sequence of numbers from 1.2 to 5.3 that is 100 elements long called "second_seq".
7. Load the AER library and attach the GSOEP9402 data set (data("GSOEP9402")). What are the dimensions of the data set? What are the names of the variables? Rename the variables to the same name, but all in uppercase rather than lowercase.
8. Create the following vectors:
    - "person_id" which is a numeric vector from 1 to 50 that increases by 1.
    - "time" which is a numeric vector from 2001 to 2020 that increases by 1.
9. Finally, create a data.frame called "panel_data" that have the following variables:
    - "id" where each element of "person_id" is repeated 20 times.
    - "time" where the time vector is repeated 50 times.
    - "draw" that is the vector "norm_draws".
    - Note: this data set is pointless; this is just to get you familiar with creating data.frames. Do not copy and paste an object 20/50 times.

Indexing

1. Store a vector of the id's of the elements in"draws_NaN" that are NaN called "NaN_ids".
    - Note: use the "which()" function with "draws_NaN".
2. Store a vector of the id's of the elements in "norm_draws" that are positive called "pos_ids".
    - Note that positive means *strictly* greater than 0 (i.e. does not include 0).
3. Store the elements with *even indices* from "norm_draws" called "even_id_draws".
    - Hint: use the "seq()" function.
    - Style Tip: *Do not* hard code in 1000. Use the "length()" function.
4. Extract the value from the 20th observation of the "mpg" variable in the "mtcars" data set using *three* different ways we discussed in class.
5. Extract the value from the 17th observation of the "cyl" and "hp" variables in the "mtcars" data set.

6. (Harder) Extract all observations from “mtcars” that have 8 cylinders (cyl) using logical expressions/indexing.

7. Load the “Titanic” data set. This data set documents the survivors and victims of The Titanic. Note that it is the Titanic data set and not the titanic data set.

   Use the “str()” function to determine the dimensions and structure of the data set. Print the outcomes for the following subgroups of passengers:

   (a) The outcomes of the female, adult, crew.

   (b) The outcomes for the male, first-class, adults.

   (c) The outcomes of the female, first-class, children.

   (d) The outcomes of the male, second-class, adults.

8. Using “mtcars_list”, extract the “wt” *as an atomic vector* using two different ways of indexing a list discussed in class.

9. Using “mtcars_list”, extract the “wt” variable *as a list*.