Homework 2: Tweaking Resource-Allocation-by-Tweaking

36-350, Fall 2012

Due 11:59 pm on Thursday, 13 September 2012

INSTRUCTIONS: Submit a plain text file, whose name includes both your Andrew ID and the assignment number. Answer the problems in order, and indicate which answers go with which problems. Raw R output is not acceptable, and will be marked down accordingly; you are communicating with a human being, and need to write in a human language. Word files, PDFs, etc., will not be graded.

Direct objective: Practice with flow control in R.

Indirect objectives: Re-writing code; successive approximation.

1. (15) The file `hw-02.R` on the class website contains the code for the resource allocation example from lecture 3. Download it.

   (a) (5) Set the initial output levels to 30 cars and 20 trucks (as in the slides) and run the code. Repeat this five times (resetting the initial output levels each time). Why are the results different each time? What are the means and the standard deviations of the final output levels? (You should report two means and two standard deviations.)

   (b) (5) Repeat another five times. What are the means and standard deviations of the resources needed? Why are the standard deviations less than the `slack` levels?

   (c) (5) Re-run the code twice more, starting from an initial output level of 5 cars and 5 trucks, and from 12 cars and 19 trucks. Do you get the same results as before? Should you?

2. *Negative trucks? No trucks?* (20)

   (a) (4) Suppose that by accident we set the initial output levels to `c(-20,-30)`. What would happen with the code as given? *Hint:* think this through, rather just running it.

   (b) (3) Suppose that by accident we set the initial output levels to `c(0,0)`. What would happen with the code as given? *Hint:* See previous hint.

   (c) (10) Fix the code to check whether any of the output levels are ≤ 0, and replace those output levels with a small positive number.

(d) (1) Re-run the code starting from an output level of 30 cars and 20 trucks, and verify that it still works.

(e) (1) Does your code work starting from 0 cars and 0 trucks?

(f) (1) Does your code work starting from -20 cars and -30 trucks?

3. *Limiting run time* (25) The `slack` variables stop the iteration when the plan comes close to fully using the available resources. It can also be a good idea to stop when we've made a certain number of passes, to keep from spending forever tinkering.

(a) (5) Add a variable called `max.passes`, and a test inside the `repeat` body which halts the loop when the number of passes exceeds this number.

(b) (2) Check that the code works when the initial output is 30 cars and 20 trucks, and `max.passes` is set high enough. How high is high enough?

(c) (18) Replace the `repeat` loop with a `for` loop. Ensure that the code doesn't go through the loop more than `max.passes` times, but that once the plan is within acceptable limits, no time is wasted on extra loops. Check that the code still works on the 30 car/20 trucks test case.

4. *De-randomizing* (25) The random part of the change to the plan makes it hard to reproduce results. Let's get rid of that.

(a) (5) Which final good needs *relatively* more labor, cars or trucks? Which final good needs *relatively* more steel, cars or trucks?

(b) (3) If the plan demands too much labor but has leaves excess steel, should we produce more or fewer cars?

(c) (2) If the plan demands more steel than is available, but leaves excess labor, should we produce more or fewer trucks?

(d) (15) Modify the code to replace the random change to the plan when only one variable is in excess with one or more deterministic changes. Verify that it still works on the 30 cars/20 trucks test case.

5. *Generalizing* (15) Suppose we want to consider producing cars, trucks, motorcycles, and vacuum cleaners, using labor, steel, rubber, plastic, electric wire and glass.

(a) (5) What you would have to change in the original code? You do not have to implement the changes in this or subsequent parts, just describe it in words.

(b) (5) What you would have to change for the code as it stands at the end of question 2?

(c) (5) What you would have to change for the code as it stands at the end of question 3?