

R for trial & model-based cost-effectiveness analysis, 2019

Code challenge

Overview

This is a virtual code challenge hosted via GitHub for participants in the R for trial & model-based cost-effectiveness analysis workshop taking place 9 July 2019 at University College London. The intention is to provide something for the more advanced R user to get their teeth into in advance of the workshop itself. An award will be presented on the day for the best submission, judged accounting for

- Giving the right answer
- Speed
- Readability, transparency and clarity
- Simplicity, elegance and “cleanness”

There is no single right way to answer the challenges. We are not primarily interested in the values of the solutions or indeed the problem per-se. We are really interested in how the problems are tackled in code. That is, how the code is structured, what data structures are used and how they are handled. The decision of the judges is final :)

We will provide some ‘model solutions’ to the challenges following the workshop. These will not be the ‘perfect’ solutions and we well anticipate that participants’ submitted code will demonstrate superior ability!

Content

The challenges can be found [here](#). The challenge has an overarching problem which is then split into smaller challenges. Hopefully, it’ll be good fun if not a little work. You can also look through the alternative versions provided by others and learn something new, about different ways of approaching and solving problems.

Submissions

Participants should email their code solutions to nathan.green@imperial.ac.uk by **24th June 2019**. We are fairly flexible in how you would like to structure your submission. This could be as a package, project or a single .R file. In all cases include a `main.R` or `main.Rmd` file so we know where to start.

Measuring computational speed

We suggest measuring your code performance using microbenchmarking and the package (not surprisingly) `microbenchmark`. In this way we can compare running time even for things that only take a very small amount of time.