Empirical research in economics, finance, and management using R: Essentials, real examples, and troubleshooting

Dr. Andreï V. Kostyrka
University of Luxembourg

1. Brief description

The aim of this doctoral course is to help researchers learn the basics of the R programming language, to teach them how to apply it to answer typical and atypical research questions, how to produce useful diagnostics and visualise the results, and how to proceed in case of troubles or errors. The material covered contains advice that is difficult to find in the literature or online resources. The practical part of this course is based on the questions asked by Ph.D. students in DSEFM over the course of several years. The applied topics covered in the course (see 'Course details' below, items 8–10) will depend on the interest of the participants. The theoretical knowledge of numerical methods and algorithms received in this course is applicable to other statistical packages used for research in the field.

Upon successful completion of this course, students will be able to:

- Understand the logic of the R language and use its strongest features to manage and transform data;
- Use numerical optimisation based on deterministic and stochastic algorithms to answer questions ubiquitous in applied research;
- Create their own functions, routines, and simulations for cutting-edge research to increase productivity, even when there is no existing implementation or solution;
- Produce neat publication-ready plots in 2D and 3D and create vibrant animations to better illustrate the research problem;
- Troubleshoot errors, debug functions, look for the possible source of error or performance bottlenecks;
- Understand how computers store and process data and where accuracy is typically lost.

2. Intended audience

This course is intended for Ph.D. students in economics, management, and finance. However, all postdoctoral researchers, Ph.D. or master's students with a natural interest in quantitative methods are encouraged to participate.

3. Text and materials

Required study material for this course:[1]

- Davies, T. M. (2016). *The book of R: A first course in programming and statistics*. No Starch Press
- Wickham, H. (2019). *Advanced R*. Chapman & Hall/CRC
- Burns, P. (2011). *The R inferno*
- Sauer, T. (2017). *Numerical analysis* (3rd ed.). Pearson Education
- Nash, J. C. (2014). *Nonlinear parameter optimization using R tools*. John Wiley & Sons
- Boyd, S., & Vandenberghe, L. (2004, March). *Convex optimization*. Cambridge University Press
- Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). Springer
- Muenchen, R. A. (2011). *R for SAS and SPSS users* (2nd ed.). Springer
- Kleiber, C., & Zeileis, A. (2008). *Applied econometrics with R*. Springer
- Cotton, R. (2013). *Learning R: A step-by-step function guide to data analysis*. O'Reilly Media, Inc.

[1] The vastly popular book 'R for Data Science' by Wickham et al. (2017) is not recommended *in the context of the present course* because it heavily relies on an ecosystem of packages with syntax different from that of base R and rather showcases particular applications of specific R packages. This course focuses not on a dialect of R, but rather on general problem solving in R. It teaches how to build methods from scratch for research in economics, finance and management because more often than not there is no pre-packaged method to help solving a research problem.

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning with applications in R* (2nd ed.). Springer
- Kabacoff, R. I. (2015). *R in action: Data analysis and graphics with R* (2nd ed.). Manning Publications Co.[2]
- Matloff, N. (2011). *The art of R programming: A tour of statistical software design*. No Starch Press
- Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th ed.). Springer; Venables, W. N., & Ripley, B. D. (2000). *S programming*. Springer Science+Business Media
- Hubbard, J. H., & Hubbard, B. B. (2015). *Vector calculus, linear algebra, and differential forms: A unified approach* (5th ed.). Matrix Editions
- UCLA: Statistical Consulting Group. (2023). *R learning resources*.

The following open-source software are used in this course:

- The R project for statistical computing (`https://www.r-project.org`);
- RStudio (`https://posit.co`).

Other course material (lecture slides, additional data sets, code written during the sessions, computer exercises) will be made available on the Moodle site of this course.

4. Lecture schedule

The course will be given in a hybrid format, which means that the participants who cannot be present physically due to the limited room capacity / health issues can join the session via this Webex link.

#	Date	Room	Time
1	18/09/2023	CK BLG 001	14:00 – 17:00
2	20/09/2023	CK BLG 001	14:00 – 17:00
3	25/09/2023	CK BLG 001	14:00 – 17:00
4	27/09/2023	CK BLG 001	14:00 – 17:00
5	02/10/2023	CK BLG 001	14:00 – 17:00
6	04/10/2023	CK BLG 001	14:00 – 17:00
7	09/10/2023	CK BLG 001	14:00 – 17:00
8	11/10/2023	CK BLG 001	14:00 – 17:00
9	16/10/2023	CK BLG 001	14:00 – 17:00
10	18/10/2023	CK BLG 001	14:00 – 17:00

1. How computers compute, how data are stored. Data formats. Programming concepts. Comparison of statistical packages. Showcasing R as a language.
2. Best practices for development. How to get help. R packages. RStudio features. Data types in R, object types, variables, vectors, matrices, functions.
3. Special data types. Classes and methods. Logical operations, conditions, loops. Subsetting, data manipulation. Lists, vectorised operations. Data transformation best practices. Text manipulation.
4. Functions in R. How to efficiently create a user function. Environments. Debugging. Parallel computing. Speeding up, benchmarking, profiling.
5. Graphics in R: scatter plots, line plots, heat maps, histograms, bar plots, box-and-whisker plots. Plot clean-up. Formulae, summarisation, aggregation, describing data. Producing animations.
6. Numerical optimisation (convex and non-convex). Derivative-based, derivative-free and stochastic optimisers. Speeding up slow optimisation problems.
7. Applied economic analysis in R: OLS, 2SLS, GMM, panel models, non-linear and non-parametric methods, time-series methods, robust estimation. Hypothesis testing and inference.
8. Practical session #1: fetching data from the web; merging 'dirty' data sets; diagnosing panel models; variable selection via LASSO/Elastic Net.

[2] The 3rd edition (2022) introduces high-level abstractions earlier, whilst the 2nd edition makes a stronger focus on the base R throughout the text and is therefore recommended for the scope of this course.

9. Practical session #2: detecting non-linearities in relationships, parametric and semi-parametric specification testing, improving estimation efficiency, principal component analysis and dimensionality reduction.
10. Practical session #3: custom conditional-density models; avoiding numerical instabilities; reproducing non-standard plots; demographic calculations; recreational statistics.

All sessions **will be recorded**; the recordings will be made available only to the registered participants via a link on the secure Moodle platform.

5. Grading

Regular class participation: 10% (every class meeting yields 1%). Students need to come to the classroom; active participation and questions from the audience are encouraged.

Assignment 1: 15%. *Due: 02/10/2023 23:59.* The first assignment tests the participants' ability to manipulate basic R objects and structures (vectors, matrices, data frames). The main goal is to teach the R-specific features not found or only partially implemented in other programming languages.

Assignment 2: 15%. *Due: 16/10/2023 23:59.* The second assignment focuses on the core features of this course: functions and custom plots. Each student is expected to write their own functions, benchmark and debug them, and prepare publication-ready visualisations similar to those found in top economic journals.

Assignment 3: 15%. *Due: 06/11/2023 23:59.* The third homework contains several short research problems in economics from which the student is required to pick one. The goal is to simulate the data-generating process, write a solver for the model, compare the behaviour of several estimators, and find the weak spot of said model.

Final project: 45%. *Due: 31/12/2023 23:59.* The participants may choose one of the following tasks:

- *Published code translation.* Take any working Matlab / Stata / SPSS / SAS / EViews code base from any article published in a scientific journal that conducts simulations and/or estimation, and implement the same methods in R. Identify the parts that work faster or slower. Fully document the process, produce clear and concise visualisations; showcase the model, the assumptions and key dependencies in it. The results need not match the exact figures from the article – they need to be consistent or similar. The size of the source code chosen for re-writing must exceed 5 kB. Examples:
  https://economics.mit.edu/people/faculty/josh-angrist/angrist-data-archive
  https://sites.google.com/view/korteweg/home/data-code
- *Own code translation.* Take any working Matlab / Stata / SPSS / SAS / EViews code from your research and rewrite it in R. Identify the parts that work faster or slower, produce clear visualisations; showcase the model, the assumptions and key dependencies. It need not be one script; you may outsource different chunks of multiple projects into R – the choice is flexible. The parts chosen for re-writing, stripped of comments and initial white space, must exceed 1600 bytes in size when *compressed into a ZIP archive* with standard settings.[3]
- *Tidying bad 'tidy' code.* Take any working R code base from any article available online that conducts simulations and/or estimation and relies heavily on `tidyverse` (possibly abusing it), and implement the same methods in base R. Implement the best programming practices, make the code readable, eliminate the redundancies. Identify the parts that work faster or slower. Fully document the process, produce clear and concise visualisations. The results need not match the exact figures from the article – they need to be consistent or similar. The total number of characters in the lines containing changes, stripped of white-space indentation, must exceed 7 kB. Example:
  https://github.com/arpitrage/Dividend_Strip/blob/master/Code/PEmodel/

[3]The ZIP compression ratio of Stata-like codes, given the typical syntactic redundancies, is usually ≈3–6. The input should be diverse enough – not a single chunk copied and pasted many times and replaced with one loop; ZIP compression mitigates this.

- *Speeding up slow code.* Similar to the task above: take any working R code base for a published article fitting the description above and speed it up. Identify the bottlenecks, profile the code. If possible, introduce parallelisation and replace sequential loops with `lapply`-like functions. Fully document the process, produce clear and concise visualisations. The total number of characters in the lines containing changes, stripped of white-space indentation, must exceed 7 kB. The speed gains achieved must be 100% or higher.
- *Numerical optimisation.* Make a comprehensive test suite of optimisation algorithms implemented in R: choose 10 objective functions found in real-world economic applications and benchmark the speed and reliability of 20 algorithms (derivative-based, derivative-free, stochastic, or even custom rules). Study which tweaking parameters are the most important for high-dimensional applications. Write a report with animations.
- *Simulation & visualisation.* Contribute to popular science. Produce 3 small simulations and visualisations of applications of mathematics / statistics / economics / algorithms in real life that would be of interest to the general public: simulating the work of a casino and testing different strategies, modelling the spread of a pandemic in the world, generating fractal art, devising maths-based magic tricks, designing and testing board games, estimating the cost of cryptocurrency mining / experience farming in video games in terms of electricity consumption etc. Animate one of these simulations.

Hint: **choose the task that is the most relevant for your research** or the one that can be later reused in other projects. You may recycle your existing material in the assignment.

6. Technical requirements for the final project

1. The submission must consist of a script file (named `yoursurname.R`) and, if necessary, data files in `.RData` format, image files in PDF or PNG format, or videos in MP4 format (H264, H265, or AV1 codec). Compress multiple files into a single ZIP archive.
   Example: `smith.zip` containing `smith.R`, `smith.RData`, `smith-1.png`, `smith-2.pdf`, `smith-3.mp4`.
2. The scripts must run in a new session without an error.
   - If external data sets are used, they must be loadable from the current working directory without user interaction; no absolute paths are allowed.
   - If functions from external packages are used, these packages must be loaded earlier.

   If a script stops due to an error, the assignment will not be graded.
   Example: The script `smith.R` uses the `ivreg` function from the `AER` package to estimate a model on a custom data set `mydata`. The script contains `load("smith.RData")`, which loads `mydata`, and `library(AER)` before `ivreg(..., data = mydata)` is called.
3. The implementation should rely on base R + 'CRAN recommended packages', although other packages can be loaded, too, on condition that **their version be stated in a comment** to make your research reproducible even after decades.
   Example: `library(cubature) # Tested with ver. 1.4.`
4. The following packages **should be avoided:** `ggplot2`, `dplyr`, `tidyr`, `purrr`, `tibble`, `stringr`, `forcats`, `magrittr`, `tidymodels`. **Exception:** if your model is hard to solve, or cumbersome to simulate, and if there exists a dedicated package for working with such models that internally relies on any of the `tidyverse` packages, you may use it. Likewise, if a package provides visualisations for your class of models implemented in `ggplot2` syntax, use whatever has been developed, but **stating the versions of all of its dependencies in a comment is mandatory**: many published R scripts from 2019 relying on `tidyverse` will not run today due to the breaking changes.
5. The functions from the `data.table` package (and the `data.table` class) may be used only if an accompanying benchmark shows that the code runs at least 50% faster or if execution of the similar built-in function halts.

Students are allowed to ask for assistance with their project if they are stuck at a point where the code halts for more than 5 minutes or produces an out-of-memory error.

7. Contact information

- Office: Campus Kirchberg, room G214
- Email: andrei.kostyrka@uni.lu
- Course homepage: https://moodle.uni.lu/course/view.php?id=7454
- Office hours: By appointment