Week 1: Introduction & Software

MATH-517 Statistical Computation and Visualization

Tomas Masak

Sep 21, 2022

Statistical **Computation** and Visualization

Offered the choice between mastery of a five-foot shelf of analytical statistics books and middling ability at performing statistical Monte Carlo simulations, we would surely choose to have the latter skill.

– Press et al., *Numerical Recipes*

Statistical **Computation** and Visualization

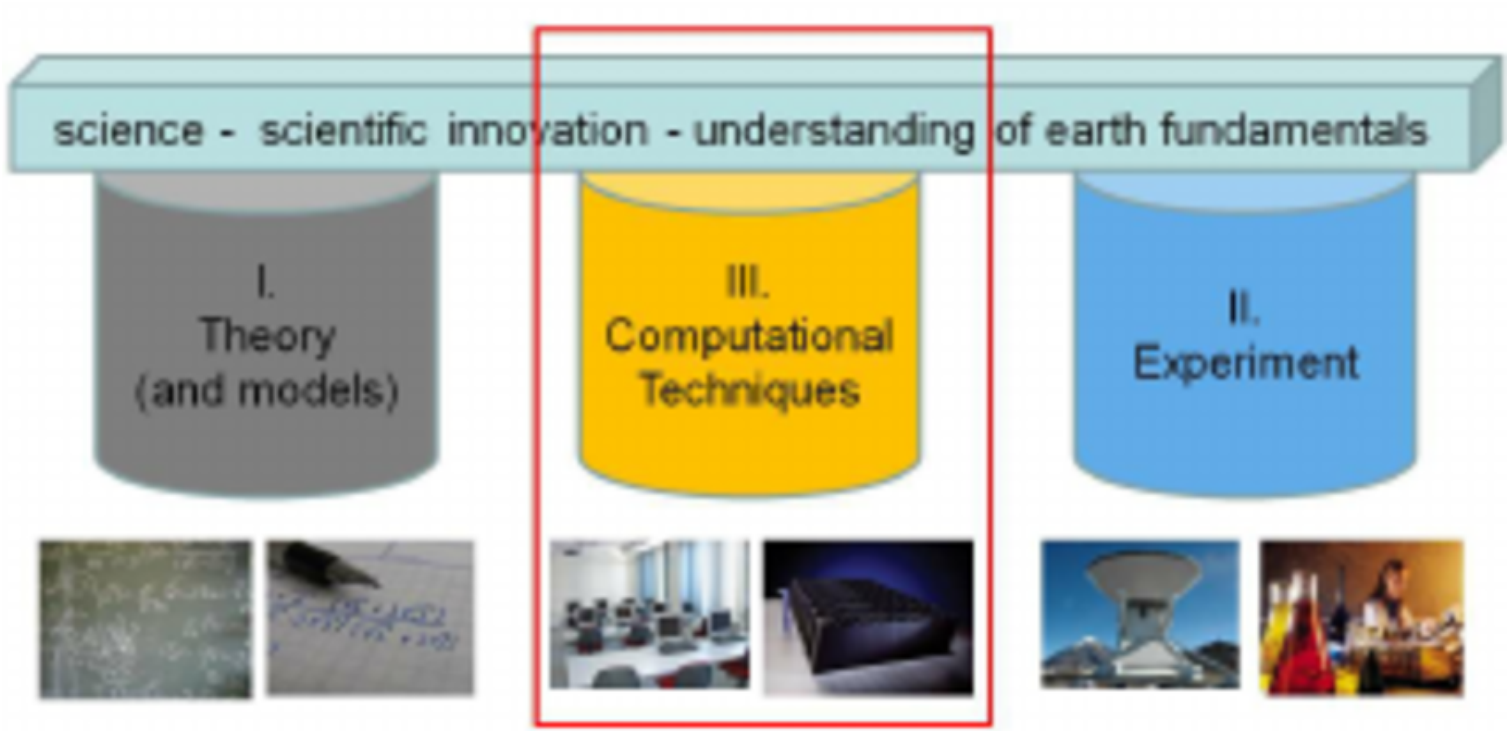
Offered the choice between mastery of a five-foot shelf of analytical statistics books and middling ability at performing statistical Monte Carlo simulations, we would surely choose to have the latter skill.

– Press et al., *Numerical Recipes*

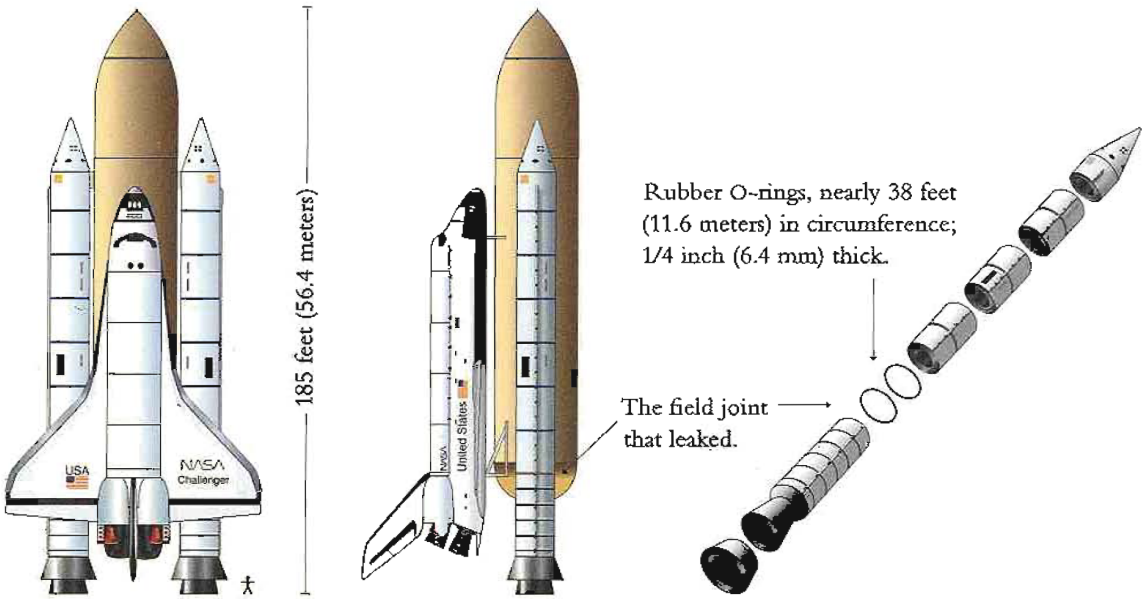
Apart from Monte Carlo (MC), we will cover (re-)sampling methods such as

- cross-validation
- bootstrap
- jackknife
- Bayesian MC extensions

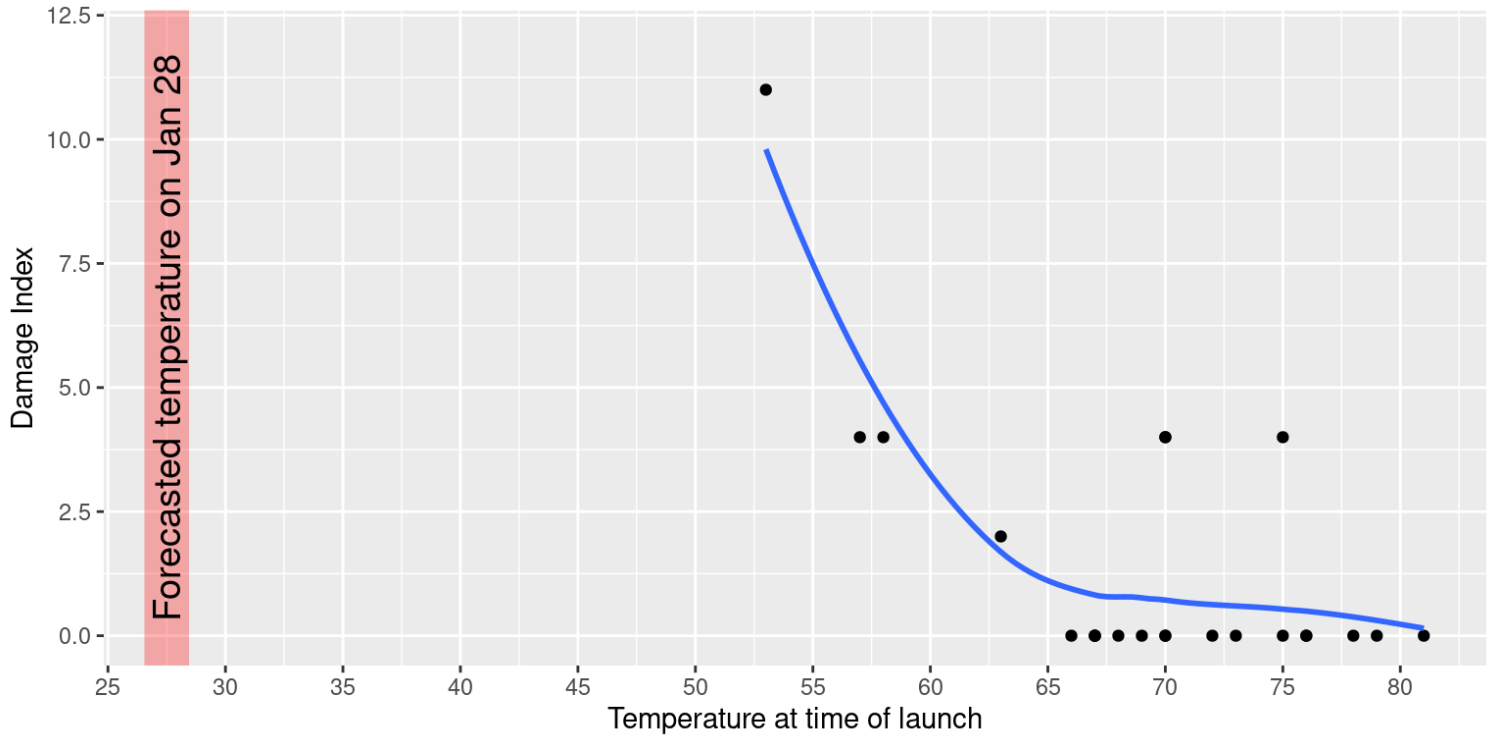
The Three Pillars of Science



Statistical Computation and **Visualization**



Statistical Computation and **Visualization**



Course Content

- Soft Start
  - R and other software
  - reproducibility and ethics
  - data wrangling and visualization
- Course Core
  - non-parametric regression
  - cross-validation
  - simulations
  - Monte Carlo (MC)
  - bootstrap
  - EM algorithm
- Bayesian Dessert
  - basic thinking
  - Markov Chain Monte Carlo (MCMC)

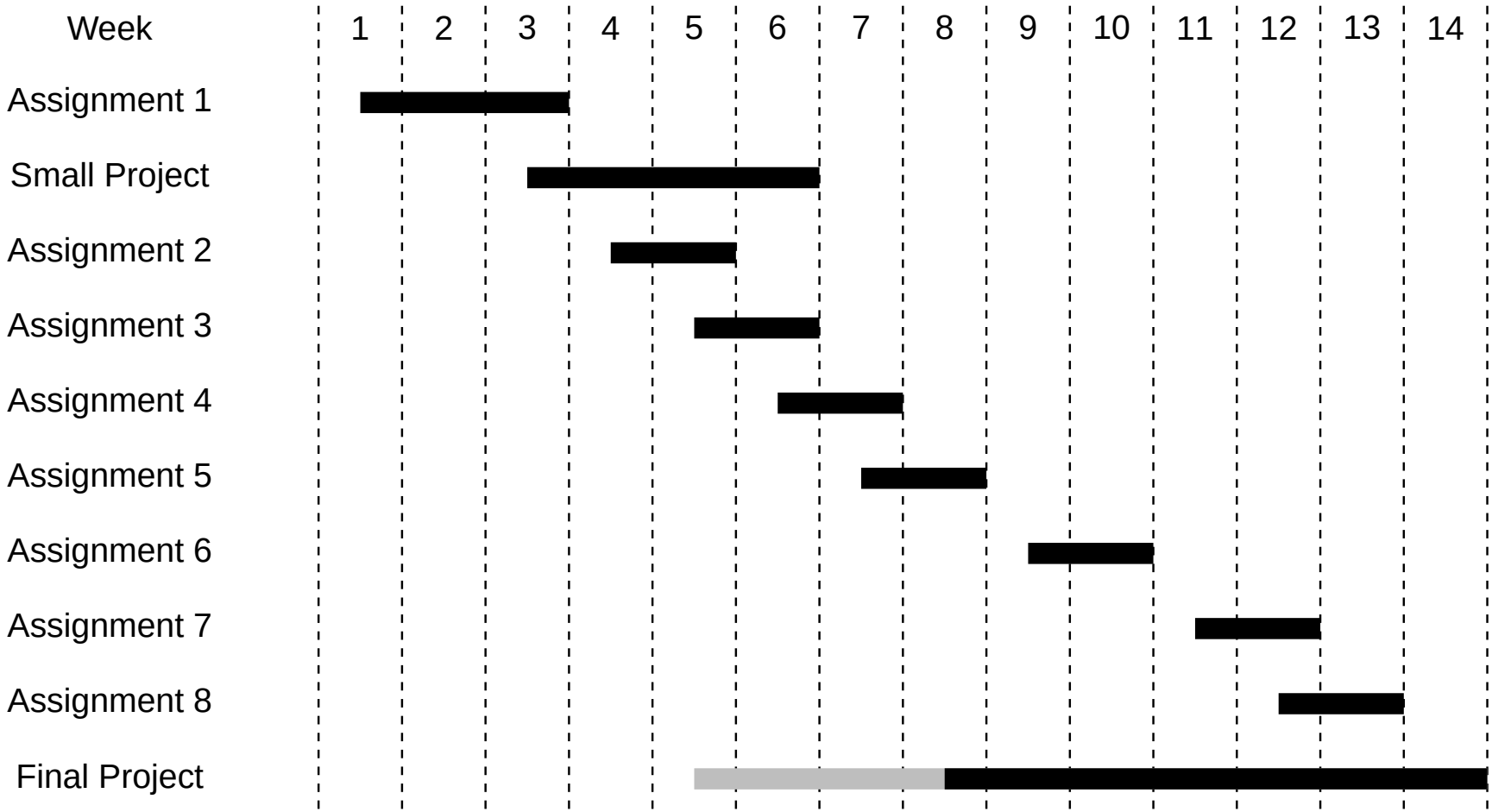
Polls

- Have you ever written a for-loop and if-else statement?
- Have you ever worked with R?
- Have you ever worked with Python, Matlab, etc.?
- Have you taken a class dedicated to linear models?
  - prerequisite
- Can you define the p-value?

Course Requirements

- assignments
  - 40 % of the grade (say 8 assignments of 5 %)
  - to be solved during the exercise classes
  - graded on the binary scale
  - collaboration (and questions) encouraged, but individual submissions required (avoid perfect copies!)
- data exploration – small project
  - 20 % of the grade
  - if the chosen data set too simple, can be composed of multiple data sets
  - in groups of 2-3 students
- project: data exploration+analysis *or* simulation study
  - 30+10 % of the grade
  - the 10 % for added value
  - in groups of 2-3 students

Expected Progress



Course Requirements

- 1 assignment = 5 % of the grade = 0.25 on the 1-6 grade scale
  - missing all assignments ⇒ final grade 4.0 at best!
- R, Markdown and GitHub for the assignments and projects will be needed
  - submission of GitHub links to the Moodle needed for grading purposes
  - this is not a programming course, learn by doing!
- 2 hours of lecture per week
  - going through the course content
- 2 hours of exercises per week
  - working on assignments and projects
  - keeping up with the lecture (e.g. with R)

active participation = success in this course

Questions and feedback are always appreciated.

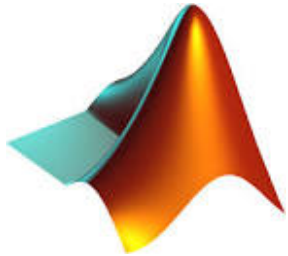
Evaluation starts right away!

Business Software

- all commercial
  - it has pros and cons
  - all (claim to) offer free academic versions
- popular in different fields
  - SAS: biomedicine, clinical research, etc.
  - SPSS: psychology, social sciences, etc.
  - STATA: econometrics, finance, etc.

Academic Software

Python

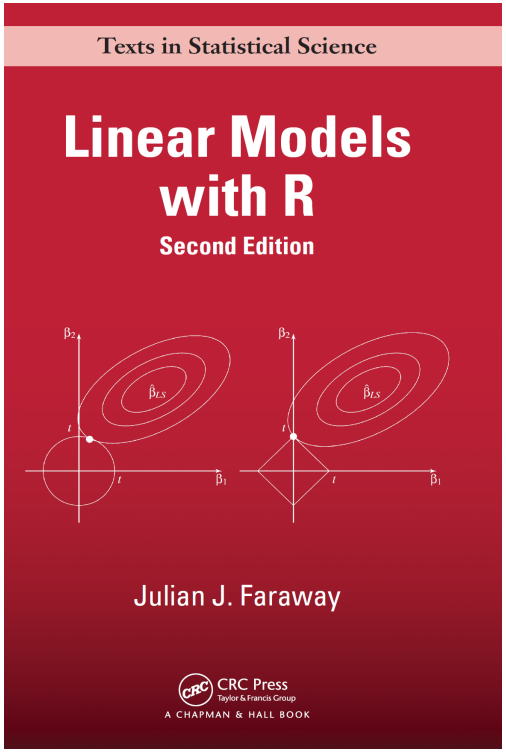


Matlab

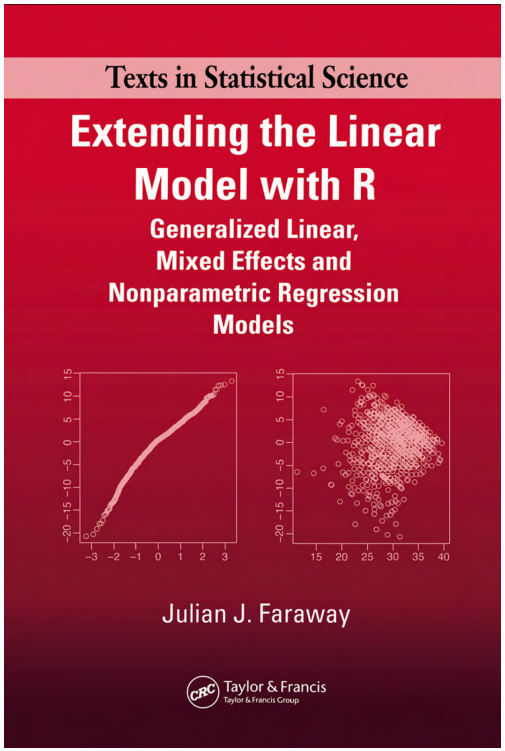
- all well documented, easy to use, with lots of examples and extensive community support
- each has its strengths and weaknesses, none is perfect
- we will use R!
- software packages are our **tools**, not skills!

free open source	free (mostly) open source (mostly)	paid (accessible) closed source
statistics	machine learning	numerical math
Data Science		Optimization

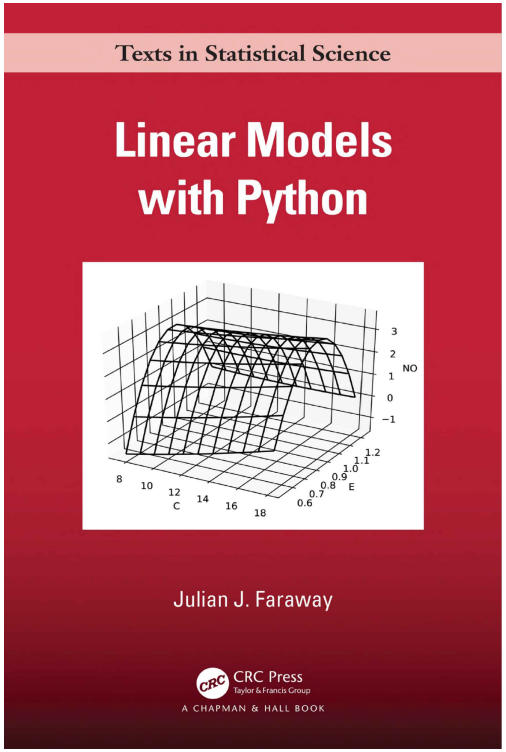
Statistics is done in R!



2004 – 1st Edition
2015 – 2nd Edition



2006



2020

Linear Models Recap

Data: $(Y_1, X_1)^\top, \ldots, (Y_N, X_N)^\top$, $Y_n \in \mathbb{R}$ is the response

Gaussian model: $Y = (Y_1, \ldots, Y_N)^\top \sim \mathcal{N}(X\beta, \sigma^2 I_{N\times N})$
- $X \in \mathbb{R}^{N\times p}$ is the model matrix containing (transformations of) $X_1, \ldots, X_N$

Model: $Y \sim (X\beta, \sigma^2 I_{N\times N})$ meaning that
- $\mathbb{E}\, Y_n = x_n^\top \beta$
- $\mathrm{var}(Y) = \sigma^2 I$

Least Squares: $\min_\beta \|Y - X\beta\|_2^2$ or $\min_\beta \sum_n (Y_n - x_n^\top \beta)^2$

Fit: $\widehat{\beta} = (X^\top X)^{-1} X^\top Y$ assuming X is full-rank
$\widehat{Y} = X(X^\top X)^{-1} X^\top Y$ are called fitted values

lm Example Takeaways

- doing statistics is more than just "massaging" data and taking whatever comes out
- linear models are the bread and butter of statistics, and as all statistical methods they are most easy to work with in R
- effective visualization is important beyond reporting results (code debugging, model verification, etc.)

lm Example Takeaways

- Jupyter Notebook (Python) or Live Script (Matlab) seem more interactive than R Markdown
  - not necessarily a good thing, e.g. reproducibility issues
- one can run R in Jupyter Notebook or conversely Python in R Markdown
  - I generally do *not* recommend either, except if you, say, work in R Markdown and want to use some Python packages (e.g. for deep learning)
- instead of R Markdown, one can use R Notebook (also in RStudio) to get some of the Jupyter Notebook interactivity

RStudio

Our Workflow

- open RStudio
- load some packages
- work in a (R Markdown) script, potentially inside a project
- save your progress
- push your changes to GitHub

Bookdown – a lot of interesting books written in R Markdown, including one of the references for this course, the R for Data Science book co-authored by Hadley Wickham.

Good Coding Practices

- consistency (following a certain style)
- indentation
- naming conventions – short but informative variable/function names
  - `camelCase`
  - `snake_case` – my personal preference
  - `dot.case` – not bad, but some languages will not allow it
  - others such as `PascalCase` or `kebab-case` are inferior
- simplicity
- comments (more the merrier!)
- load packages at the top
- loading data comes next (possibly in a separate script)
- functions come next (possibly in a separate script)
- use `set.seed()` if RNG present
- re-run with empty environment

Assignment

Go to Supplement 02 and set up R, RStudio and GitHub for yourself as described there.

Make your first submission (submit whatever you want) as described in the course requirements. [5 %]

Exercise

We have done most of the analysis of the `chredlin` data in R, but we have done some pieces in Python and Matlab. Do the full analysis in R, i.e. complete the Markdown script corresponding to R translating the pieces of code from Python and Matlab.

Freedman (2009) Statistical Models, Example 4, p. 75:
Suppose Y consists of 100 independent $\mathcal{N}(0,1)$ random variables. This is pure noise. Let $X \in \mathbb{R}^{100\times 50}$ be the design matrix with independent $\mathcal{N}(0,1)$ variables (just more noise). We regress Y on X (i.e. no intercept). The coefficient of determination R^2 will be about 50/100=0.5. Suppose we test each of the 50 coefficients at the 10 % level and keep only the "significant" variables. There will be about $50 \times 0.1 = 5$ keepers (just by chance). Running the regression on the keepers only (again, without intercept), we are likely to get a decent R^2 (like 0.2 – decent by the social-science standards) and dazzling t-statistics. Run a couple of simulations and see for yourself.

References

JJ Faraway (2015) Linear Models with R (2nd Edition)

JJ Faraway (2020) Linear Models with Python

Poldrack (2019) Statistical Thinking for the 21st Century (online)

Tufte (1997) Visual Explanations