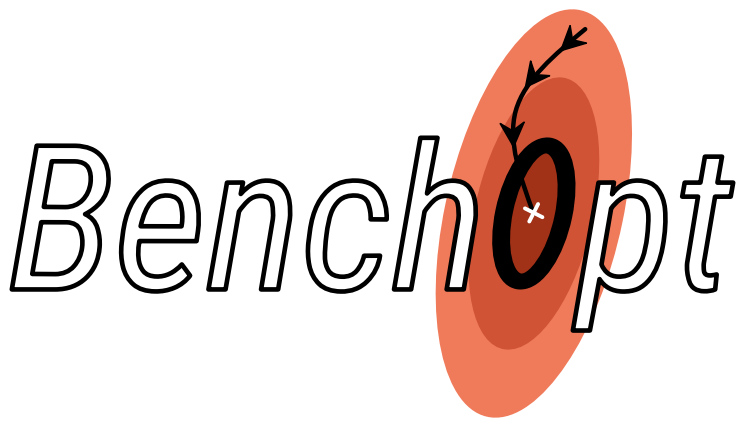
Code for Reproducible Research

Thomas Moreau



Reproducible research

Different goals:

- Reproduce the exact same results?
- Run with new parameters with robust results?
- Run with a new dataset?
- Extend the results with a new method?
- Provide tools for other to use?

Does not require the same set of tools!

Reproducible research

Different goals:

- Reproduce the exact same results?
- Run with new parameters with robust results?
- Run with a new dataset?
- Extend the results with a new method?
- Provide tools for other to use?

Does not require the same set of tools!

Here is my take.

Reproducing the same results

Minimal requirement for a research project:

- Write clean code:
  - Consistent style: *Use `black` or `flake8`.*
  - Use readable names.
  - No notebooks!
- Document your code: *docstring and `README.md`.*
- Determinist output: *Set a random seed!*

Optional but advised

- Document dependencies.
- Proper package organization
- Add some tests: *Use `pytest`.*

Extending the results?

These advice makes it easier to reproduce the same results, but we want to extend them!

Extending the results?

These advice makes it easier to reproduce the same results, but we want to extend them!

Pain points of a benchmark:

- competitors' methods do not work out of the box.
- re-code methods and tools to integrate a new method.
- hard to extend with new settings.

all of this started from scratch by every submission!

`Benchopt` produces **open**, **reproducible**, **extendable** benchmarks

Going further: creating a package

If you really want to make your research *reproductible* by other in different contexts, you need to properly package it.

- Documentation: *Sphinx*.
- Test on multiple platforms: *Continuous Integration*.
- Release on pypi/conda-forge
- Talk about it ! :)

Example of package:
https://github.com/tomMoral/test_package

Conclusion

Reproducible research needs more than just releasing code:

- Clean and Documented.
- Reusable.
- Extendable.

Use proper tools to make it possible!

Research is also collaborative: don't hesitate to report your issues and give feedback :)