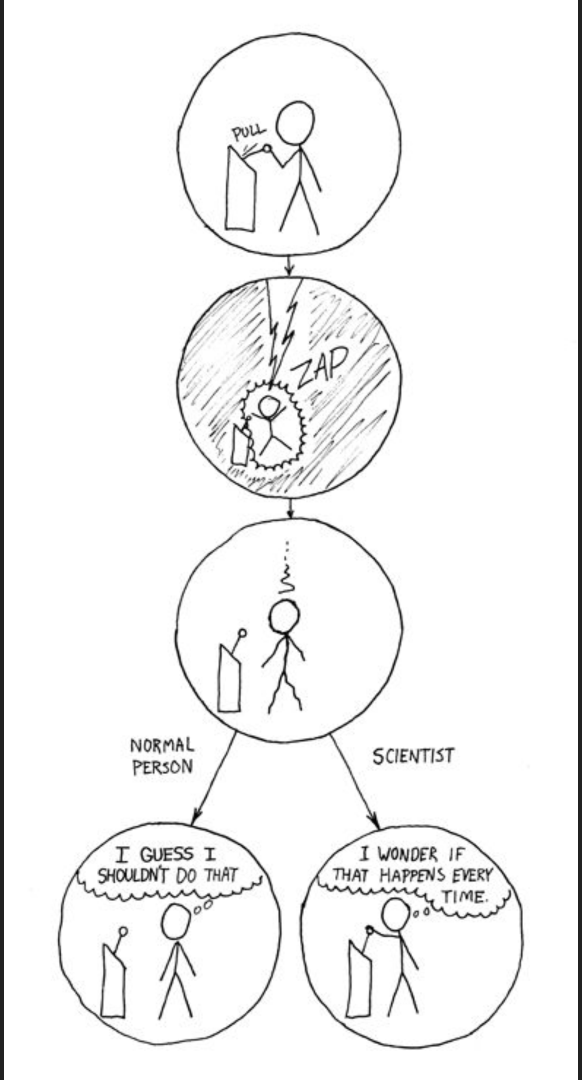
Reproducibility in Data Science

Ivan Marin
Daitan Group
imarin@daitangroup.com

Definitions

https://xkcd.com/242/


PULL
ZAP
NORMAL PERSON
SCIENTIST
I GUESS I SHOULDN'T DO THAT
I WONDER IF THAT HAPPENS EVERY TIME.

Definitions

Replicability

is the capacity of an experiment or study to be ***replicated***

by ***another group***

independently e ***completely,*** collecting new data

Definitions

Reproducibility

is the capacity of an experiment or study to be ***reproduced***

by the ***same*** researcher or ***another group***

using the same raw data sets and software

Definitions

Replicability is stronger than ***reproducibility***

Replicability introduces other variables like different researchers, equipment, etc and the associated variations

Reproducibility concerns only to redo the original experiment with the same data and code

Replicability crisis in Science

“The test of replicability, as it’s known, is the foundation of modern research. Replicability is how the community enforces itself. It’s a safeguard for the creep of subjectivity. Most of the time, scientists know what results they want, and that can influence the results they get. The premise of replicability is that the scientific community can correct for these flaws.”

What happens when this fails?

Replicability crisis in Science

Why Most Published Research Findings Are False, John P. A. Ioannidis, *PLOS Medicine 2(8): e124*, 30 Aug 2005

Estimating the reproducibility of psychological science, Open Science Collaboration, *Science* vol. 349, Issue 6251, 28 Aug 2015

1,500 scientists lift the lid on reproducibility, Monya Baker, *Nature* 25 May 2016

and so on...

Replicability crisis in Science

The most common problems:

- apophenia: seeing patterns in random data
- confirmation bias: focus on evidence that agrees with expectation
- hindsight bias: being predictable only after it already happened
- p-hacking: analysing the data to minimize p-values
- pressure to publish only positive results
- bad experiment design

Replicability crisis in Science

What can be done?

Replicability crisis in Science

Recommendations:

- More transparency and openness on methodology, source code and data
- More incentives to share data and methodology
- Better rewards to other research groups to reproduce experiments
- Open source tools and workflow integration
- Improve research methodology
- Statistical training
- Study pre-registration

A manifesto for reproducible science, Munafo et al, *Nature Human Behavior* 2017

Replicability crisis

Other approaches that try to help:

- Open Science Framework (https://osf.io/)
- Reproducible Research Net (http://reproducibleresearch.net/)
- Protocols.io (https://www.protocols.io/)

Replicability crisis

But what about outside academia?

What about outside academia?

Replicability can also benefit teams in a business setting

- Substantiate conclusions and decisions
- Allow for project enhancement, duplication or continuity
- Re-analyse the process in case of questions or doubts
- Prevent losing knowledge generated by the team
- Compare the process between different projects to highlight success parameters
- Maintain accountability of project development

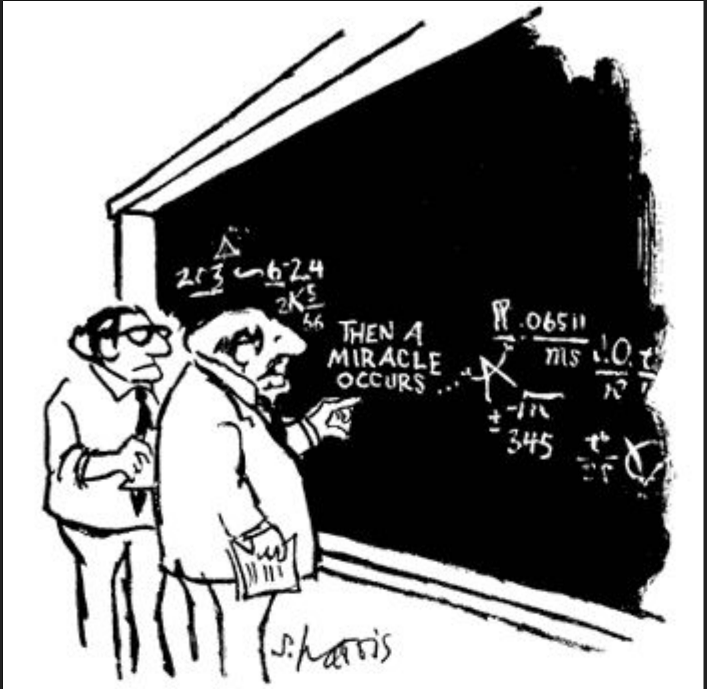
Who can benefit from replicability?

- Researchers
- Data Scientists
- Software Engineers
- Project Managers
- Developers
- Support teams
- Sales

and entire organizations

Reproducible experiments

Sidney Harris



"I think you should be more explicit here in step two."

Reproducible experiments

We will focus on reproducibility of a computational experiment:

- very common outside academia
- most Data Science projects
- business value
- relative small changes to existing processes
- open source tooling

Anatomy of a reproducible experiment

Computational experiment *c*:

- at time *t*
- on hardware/operating system *s*
- on data *d*

Must generate consistent results if executed at time *t´*, on operating system *s'* and with similar or equal data *d'*

Anatomy of a reproducible experiment

For that, we need:

- **metadata**: description of what the input data means, how it was generated, its format, dependencies and relationships
- **environment description**: from the hardware to all the tools used
- **executable specification of the experiment**: the code itself or the steps followed to derive the results, from the data gathering to the interpretation

Anatomy of a reproducible experiment

Levels of reproducibility:

- **depth**: how much of the original experiment is available
- **portability**: the dependency on the original environment
- **coverage**: how much of the original experiment can be reproduced

Reproducible experiment concepts

- Literate Programming
- Data provenance
- Data pipeline
- Execution workflow
- Version control for data and source code
- Experiment automation
- External and internal documentation

Reproducible experiment workflow

Diagram

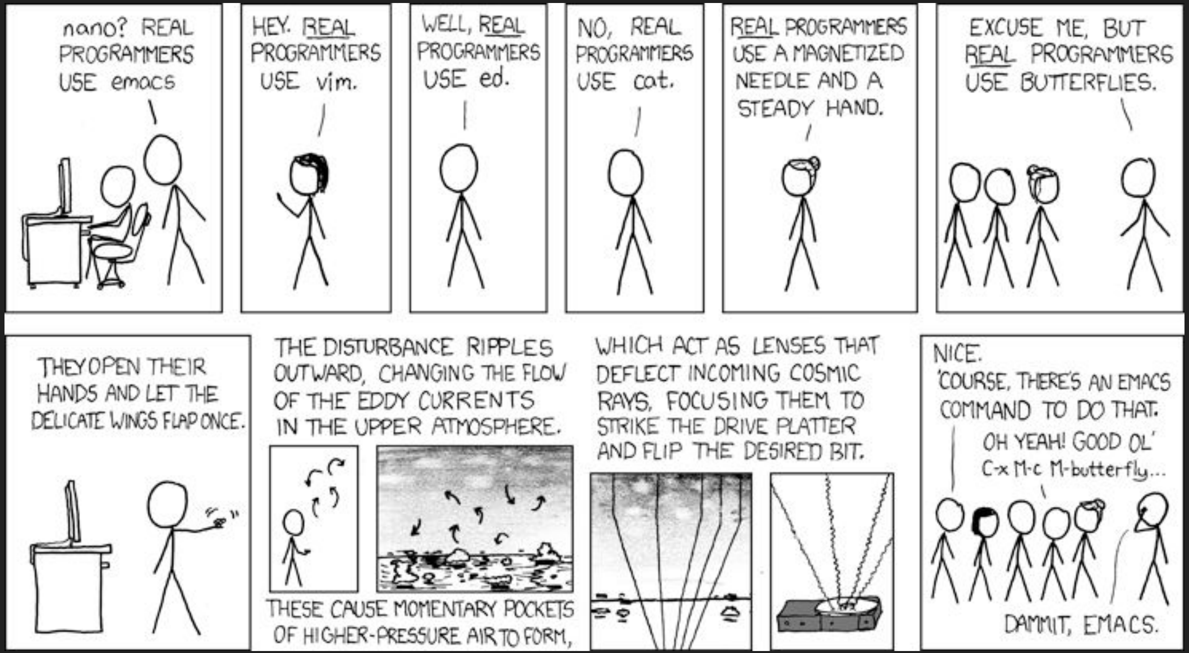
data pipeline

run code

generate results

Reproducible Pipeline Example

https://xkcd.com/378/


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

Workflow entities

For the Reproducible Pipeline, we will provide a few recommendations for each of the pipeline entities:

- Data artifacts
- Source code
- Platform/Environment
- Documentation
- Results and reports

Data Artifacts

Recommendations for Data Artifacts

- Raw data is NEVER modified
- Transformed data is versioned
- All data transformations are recorded - including code used
- Data description and metadata changes automatically with data transformation
- Intermediate steps should be stored, considering volume/storage space

Source Code

Recommendations for Source Code

- All source code is versioned, from the beginning of development
- Create internal documentation during development
- Generate tests for trivial cases
- Document example use cases
- Collaboration is done through version control

Platform/Environment

Recommendations for Platform/Environment

- Dependency from Operating System should be minimal
- Execution environment should be isolated
- Installation/Deploy should be automatic
- Libraries and package versions are static and part of the experiment
- External tools needed (compilers, etc) are also part of the environment
- All code, library and packages should be portable to other Operating Systems
- All code, library and packages should be portable to other architectures

Documentation

Recommendations for Documentation

- Documentation should be auto generated
- Documentation process should be integral part of development work
- All notes and documents should be versioned in sync with code/data
- External documentation are part of the project documentation and should be included
- Binary documentation formats are to be avoided - prefer plain text and autogeneration

Results and reports

Recommendations for results and reports

- All results must be generated automatically, from data extraction to final result, without any kind of intervention
- The results should be versioned in sync with data, code and documentation
- Reports should be auto generated in case of modifications in any part of the workflow
-

Project Example: Sonar Data

https://xkcd.com/378/
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4565690/

Sonar Data

Let's create an example project using a data set from UCI Machine Learning Repository:

Connectionist Bench (Sonar, Mines vs. Rocks)

http://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks)

We will use Python for this example, but the ideas should work for any language or environment.

Project Example

Project folder structure:

- doc
- data
- src
- notebooks
- reports

Project Example

Project folder structure:

- doc: project documentation and references
    - literature
    - Readme
    - index

Project Example

Project folder structure:

- data: input and processed data
  - raw
  - processed
  - external

Project Example

Project folder structure:

- src: source files and executables for entire pipeline
    - data
    - features
    - models
    - visualization

Tooling

Some tools that we will use:

- git
- Pyenv
- Virtualenv
- cookiecutter
- Jupyter Lab
- Makefiles

Tooling

Some tools that we will use:

- git:
  - Distributed Version Control system
  - https://git-scm.com/

Tooling

Some tools that we will use:

- Pyenv:
  - Compile our own Python locally
  - https://github.com/pyenv/pyenv

Tooling

Some tools that we will use:

- Virtualenv:
  - Create a isolated Python environment for libraries
  - https://virtualenv.pypa.io/en/stable/

Tooling

Some tools that we will use:

- cookiecutter:
    - Python library to deploy folder structure and files for project based on a template
    - https://github.com/audreyr/cookiecutter

Tooling

Some tools that we will use:

- Jupyter Lab:
  - Extensive environment for literate and reproducible computing
  - https://github.com/jupyterlab/jupyterlab

Tooling

Some tools that we will use:

- Makefile:
  - File to execute set of executions, created to compile programs
  - https://www.gnu.org/software/make/manual/make.html

Project Example: Steps

Let's create the base environment. We are assuming Debian Stretch AMD64.

1. Install required packages:

```
apt-get install build-essential libssl-dev libsqlite3-dev python-devgit virtualenv
virtualenvwrapper zlib1g-dev libbz2-dev libreadline-dev
```

Project Example: Steps

2. Install Pyenv and compile the Python interpreter:

```
git clone https://github.com/pyenv/pyenv.git ~/.pyenv

echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile

echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile

echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n  eval "$(pyenv init -)"\nfi' >> ~/.bash

source .bashrc

pyenv install 3.6.4

pyenv local 3.6.4
```

Project Example: Steps

1. Criar o ambiente de desenvolvimento com diretórios, virtualenv e pip
2. Popular definições do problema em doc e em results
3. Estruturar o código para ingestão de dados
4. Fazer a ingestão dos dados brutos em data
5. E que comecem as análises
6. Parte iterativa: a cada etapa, fazer um ciclo entre
   a. análise
   b. visualização

Here be dragons

Os resultados são mais importantes que a estrutura

O código em um projeto de Ciência de Dados ***não*** é o objetivo final

A estrutura deve ser adaptada ao projeto a ser executado (e não vice-versa)

"*A foolish consistency is the hobgoblin of little minds*"

(Ralph Waldo Emerson)

Perguntas?

Obrigado!

(Algumas) referências

http://mfactor.sdf.org/data-science-workflow-with-reproducible-research.html

http://drivendata.github.io/cookiecutter-data-science/

https://en.wikipedia.org/wiki/Reproducibility

https://en.wikipedia.org/wiki/Reproducibility_Project

https://en.wikipedia.org/wiki/Replication_crisis

http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1182327/

(Algumas) referências

http://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020124

https://osf.io/ezcuj/

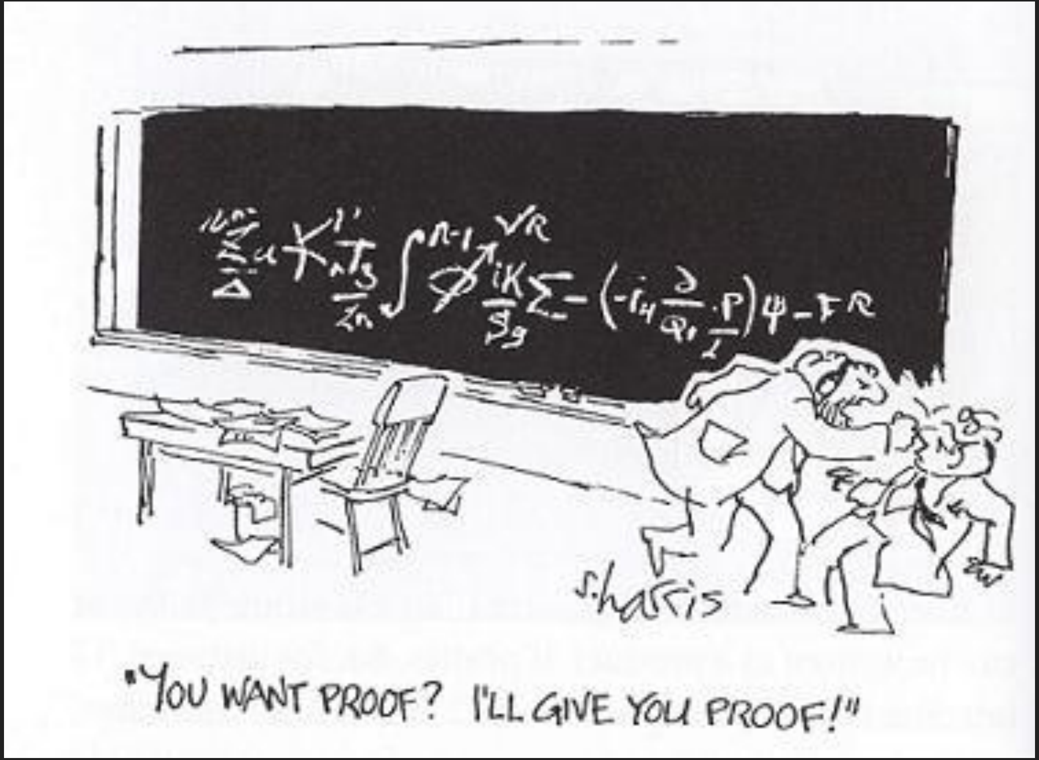
Útil?

“In any moment of decision, the best thing you can do is the right thing, the next best thing is the wrong thing, and the worst thing you can do is nothing.“

Theodore Roosevelt

What is ***not***

Sidney Harris



What is ***not***

A *tool* (or a *set* of tools)

There are tools that enable or facilitate reproducibility, but it can be implemented without dependency on specific ones

What is ***not***

A coding or documentation standard

Reproducible processes can be benefited from coding and documentation standards, but there is no “right” one

What is ***not***

Only useful in Academia

Well... Reproducibility is not doing very well in Academia, but it should be one of the cornerstones of Science

What is ***not***

Project planning tool

It can definitely help, but it's not restricted to the planning phase only

Replicability crisis

The Sokal Affair: *Transgressing the Boundaries: Towards a Transformative Hermeneutics of Quantum Gravity*, Social Text 46/47, 1996
(http://www.physics.nyu.edu/faculty/sokal/transgress_v2/transgress_v2_singlefile.html)

A hoax published on a humanities journal without peer review

Was it ethical?