Statistics: Changes since I was an undergrad

Part 4: Reproducible research with R and friends

Pedro J. Aphalo

Department of Biosciences, University of Helsinki

SenPEP

20 November 2014

Lectures in this series

Changes in statistics: 1975–2014

This lecture is the last in a series of four related lectures.

1. Computing capacity has dramatically increased and programming languages have evolved.
2. New statistical methods have been developed.
3. Statistical software is dramatically more powerful and varied.
4. Accountability and reproducibility have become requirements in scientific research.

Outline

1 Introduction

2 Reproducible research

3 Tools

4 Examples

5 Conclusions

Today’s roadmap

- Why reproducibility?
- Achieving reproducibility
- Available tools
- Combining the tools

The focus on R

- R and knitr is what I use myself
- They are well accepted by the research community
- They are already stable tools
- BUT: reproducible research is not limited to these tools
- It is just an approach to doing research
- Similar tools are available for other languages
- Phyton: Sphinx, Doxygen
- Ruby: Glyph

What is reproducible research

- Reproducibility is clearly a basic feature of any research. . .
- and one of the philosophical bases behind the scientific method.
- So, what is this new fuzz about?
- Well, some people have tried to systematically reproduce published research. . .
- and found that only a small fraction of it can in practice be reproduced.

What has been the reaction?

- alarm bells ringing. . .
- in funding agencies. . .
- and publishers.
- e.g. Announcement in *Nature*
- Requirements of openness and for full disclosure tightened.
- e.g. for EU funding.

The origin of the movement

- It seems to have started in relation to computational methods and data analysis.
- Many people use this narrow meaning, and it is also the main focus this talk.
- However, reproducible research in a broader sense includes full disclosure of methods and reporting limitations and assumptions involved.
- An of course the use of valid protocols at every step.
- and the exact correspondence of what has been done at all stages of the research with what is reported.

Why it was not a problem earlier?

- To some extent it was, and even 40 years ago, there was ‘lab folk’ knowledge that some researchers would systematically omit a crucial step in every lab protocol they described, so as to delay the work of competing research teams.
- However, smaller data sets and simpler methods made reproducing results frequently possible. . .
- In addition few systematic attempts to reproduce earlier research were done. . .
- Large data and complex analyses have made the problem more prevalent.

The objectives

For data analysis

- Keep a trace of all steps of data analysis.
- Ensure that the methods description is consistent with what has been actually done.
- Ensure that the description does not omit any steps.
- Include a human-readable and understandable explanation in addition to the instructions for the computer.

The approach

- Link the human-readable description, the computer instructions and the results or output into a single file (or piece of code) to guarantee that results and code are always consistent.
- Make the computer code as human readable as possible and keep the explanations next to the code, so that they can be updated at the same time when code is edited.
- We already saw some examples of this in earlier lectures.
- Today I will discuss one particular set of tools which I have been using myself, and which my whole research group is adopting at the moment. . .

Tools

- Two philosophies to data processing
  1. Small tools that are combined to get a job done originated with Unix, and is based on connecting tools using ‘pipes’ and ‘tees’, to assemble an add hoc solution.
  2. Large monolithic applications that attempt to do it all, the philosophy behind many Windows applications and many old-time COBOL programs.
- An intermediate solution, very common for software development, is to have a windowed interface that behind the scenes calls the small simple tools that originated with Unix and its command shell.
- These front-ends are usually called **IDE**s, for *integrated development environment*.

IDE

Integrated Development System

- The currently most popular IDE for R is *RStudio*, which is free.
- In addition to a version that runs locally, there is a server based version accessed through web browsers.
- RStudio makes writing the reports used for reproducible research relatively easy.
- RStudio makes writing R packages painless, and has debugging integrated into the interface.
- Other tools are available, several popular editors have addon's for R: Winedt and Emacs are the most common.
- I use both RStudio and Winedt, depending on the project I am working on.

Revision control system

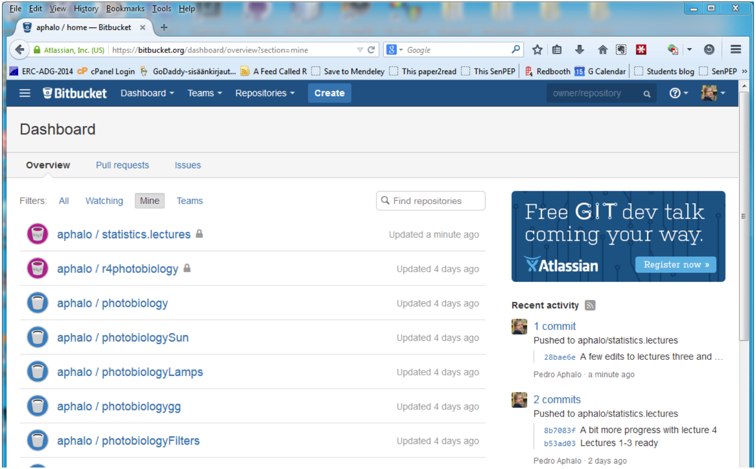
- There are several to choose from
- Currently Git and Subversion (SVN) are popular, and integrated into RStudio.
- A revision control system keeps track of the history of a file, not automatically, but for each commit that the author explicitly makes.
- Changes are stored as 'deltas' in other words only the changes are stored, but as they are indexed, any past commit can be restored.
- These systems do not track individual files, but rather projects, in other word each restore, at least by default, retrieves a consistent state of the whole project.
- Furthermore, modern systems, are geared to collaboration, allowing automatic and manual merging of changes done in parallel, and branching (keeping more than one parallel version of a project).

On-line repositories

- From the idea of collaboration follows the problem of exchanging files
- On line repositories come to the rescue. . .
- For Git two well known ones are Github and Bitbucket.
- As software used is open-source one can run a private repository on one's own server, or even locally.
- This means that several people, anywhere in the world with internet access, can work in parallel, on the same data analysis problem.

Revision control system

Bitbucket: my home page



File Edit View History Bookmarks Tools Help
aphalo / home — Bitbucket
Atlassian, Inc. (US) https://bitbucket.org/dashboard/overview?section=mine
Google
ERC-ADG-2014 cPanel Login GoDaddy-sisäänkirjaut... A Feed Called R Save to Mendeley This paper2read This SenPEP Redbooth G Calendar Students blog SenPEP
Bitbucket Dashboard Teams Repositories Create
owner/repository

Dashboard

Overview Pull requests Issues

Filters: All Watching Mine Teams

Find repositories

aphalo / statistics.lectures	Updated a minute ago
aphalo / r4photobiology	Updated 4 days ago
aphalo / photobiology	Updated 4 days ago
aphalo / photobiologySun	Updated 4 days ago
aphalo / photobiologyLamps	Updated 4 days ago
aphalo / photobiologygg	Updated 4 days ago
aphalo / photobiologyFilters	Updated 4 days ago

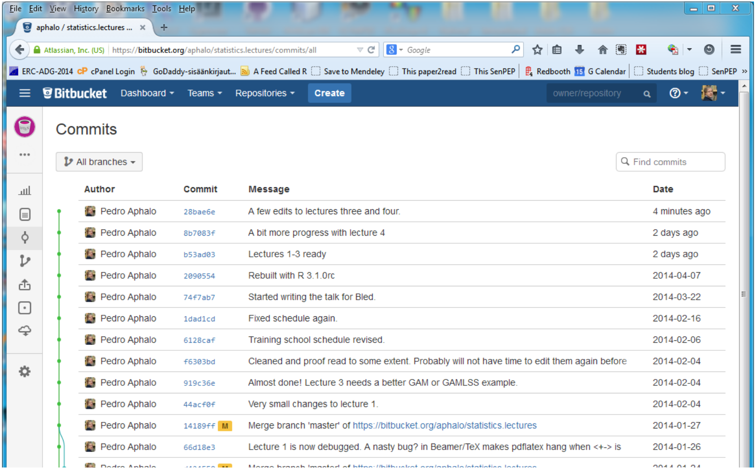
Recent activity

1 commit
Pushed to aphalo/statistics.lectures
28bae6e A few edits to lectures three and ...
Pedro Aphalo · a minute ago

2 commits
Pushed to aphalo/statistics.lectures
8b7083f A bit more progress with lecture 4
b53ad03 Lectures 1-3 ready
Pedro Aphalo · 2 days ago

Revision control system

Bitbucket: my commits page

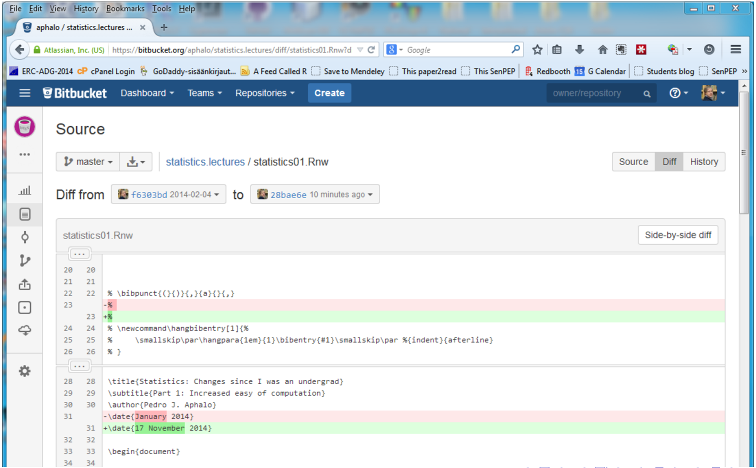


Commits

Author	Commit	Message	Date
Pedro Aphalo	28bae6e	A few edits to lectures three and four.	4 minutes ago
Pedro Aphalo	8b7083f	A bit more progress with lecture 4	2 days ago
Pedro Aphalo	b53ad03	Lectures 1-3 ready	2 days ago
Pedro Aphalo	2090554	Rebuilt with R 3.1.0rc	2014-04-07
Pedro Aphalo	74f7ab7	Started writing the talk for Bled.	2014-03-22
Pedro Aphalo	1dad1cd	Fixed schedule again.	2014-02-16
Pedro Aphalo	6128caf	Training school schedule revised.	2014-02-06
Pedro Aphalo	f6303bd	Cleaned and proof read to some extent. Probably will not have time to edit them again before	2014-02-04
Pedro Aphalo	919c36e	Almost done! Lecture 3 needs a better GAM or GAMLSS example.	2014-02-04
Pedro Aphalo	44acf0f	Very small changes to lecture 1.	2014-02-04
Pedro Aphalo	14189ff M	Merge branch 'master' of https://bitbucket.org/aphalo/statistics.lectures	2014-01-27
Pedro Aphalo	66d18e3	Lecture 1 is now debugged. A nasty bug? in Beamer/TeX makes pdflatex hang when <+-> is	2014-01-26
Pedro Aphalo	[illegible] M	Merge branch 'master' of https://bitbucket.org/aphalo/statistics.lectures	2014-01-24

Revision control system

Bitbucket: some diffs from my morning lecture



https://bitbucket.org/aphalo/statistics.lectures/diff/statistics01.Rnw?d

Source

master | statistics.lectures / statistics01.Rnw

Source | Diff | History

Diff from f6303bd 2014-02-04 to 28bae6e 10 minutes ago

statistics01.Rnw

```


% \bibpunct{(}{)}{,}{a}{}{,}
-%
+%
% \newcommand\hangbibentry[1]{%
%     \smallskip\par\hangpara{1em}{1}\bibentry{#1}\smallskip\par %{indent}{afterline}
% }
```

```
\title{Statistics: Changes since I was an undergrad}
\subtitle{Part 1: Increased easy of computation}
\author{Pedro J. Aphalo}
-\date{January 2014}
+\date{17 November 2014}

\begin{document}

```

Typesetting engines

- ‘The’ typesetting engine for technical and mathematical typesetting is TEX, and LATEXmakes it more easy to use.
- LATEX is also very popular in some branches of humanities research, because it can typeset almost any contemporary language to even Egyptian hieroglyphs or Celtic runes, and combine them in the same document.
- However, TEX is not a WYSIWYG system, you edit a text file, and then obtained the final output in a separate step.
- Simpler engines based in markdown (and easy to use text markup) or HTML (what web pages are mostly written in) can be used.
- I use myself mostly LATEX but markdown should be easier to learn.

Typesetting engines

- ‘The’ typesetting engine for technical and mathematical typesetting is TEX, and LATEXmakes it more easy to use.
- LATEX is also very popular in some branches of humanities research, because it can typeset almost any contemporary language to even Egyptian hieroglyphs or Celtic runes, and combine them in the same document.
- However, TEX is not a WYSIWYG system, you edit a text file, and then obtained the final output in a separate step.
- Simpler engines based in markdown (and easy to use text markup) or HTML (what web pages are mostly written in) can be used.
- I use myself mostly LATEX but markdown should be easier to learn.

Typesetting engines

LaTeX showcase

http://tug.org/texshowcase/cheat.pdf
http://tug.org/texshowcase/chinese.pdf
http://tug.org/texshowcase/tengwar.pdf
http://tug.org/texshowcase/leaflet.pdf
http://tug.org/texshowcase/textopo-eg.pdf

Typesetting engines

LaTeX source example

```
One can use \textbf{bolface}, \textit{italics},
% insert comments
enter in-line equations such as $y = a + b^2$ or
displayed equations like
\begin{equation}
y = \int_{t = 0}^{20} a \cdot \frac{\alpha\cdot \beta}{n + 1} \delta t
\end{equation}
where $a_1$ is\ldots\\
or chemical reactions using the \textbf{chemmacros} package:
\ch{S + E <>[ $k_{\mathrm{SI}}$ ][ $k_{\mathrm{IS}}$ ]
                E.I <>[ $k_{\mathrm{PI}}$ ][ $k_{\mathrm{IP}}$ ] P + E}
```

Typesetting engines

LaTeX output example

One can use **bolface**, *italics*, enter in-line equations such as $y = a + b^2$ or displayed equations like

$$y = \int_{t=0}^{20} a \cdot \frac{\alpha \cdot \beta}{n+1} \delta t \tag{1}$$

where a_1 is...
or chemical reactions using the **chemmacros** package:

$$\mathrm{S} + \mathrm{E} \underset{k_{\mathrm{IS}}}{\overset{k_{\mathrm{SI}}}{\rightleftarrows}} \mathrm{E} \cdot \mathrm{I} \underset{k_{\mathrm{IP}}}{\overset{k_{\mathrm{PI}}}{\rightleftarrows}} \mathrm{P} + \mathrm{E}$$

Scripting/programing languages

- For statistics, R is the most common language used.
- For bioinformatics and some other types of data analysis Phyton is very popular.
- Most systems that accept instructions as text (as opposed to menu input) can be controlled with scripts.

Caveats

- One problem is that software evolves
- Even if our script and data do not change we cannot be sure that we will get exactly the same output in the future.
- To attain reproducibility, we need to be able to restore the same versions of all software used in a analysis (e.g. R and all the packages or extensions that we used).
- A very recent solution for this is an R package called packrat which packs together with the user script, all the R packages used in the analysis.
- Packrat is already integrated into RStudio.

RStudio

Here I present two short annotated videos of RStudio in action.

RStudio report compilation video

RStudio panes video

RStudio is available from http://www.rstudio.com/

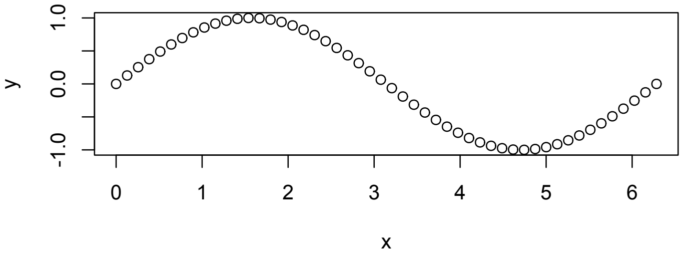
A simple example: the script

On the current page I show the unadorned script that produces the next slide.

```
<<>>=
x <- seq(0, pi/2, length=50)
y <- sin(x)
plot(y ~ x)
@
```

A simple example: the output

```
x <- seq(0, pi * 2, length=50)
y <- sin(x)
plot(y ~ x)
```



An inline example: the script

On the current page I show the unadorned script that produces the next slide.

```
Given $X = 2 \pi = \Sexpr{{}}2 * pi}$,
$y = \sin x = \Sexpr{{}}sin(2 * pi)}$,
which nicely shows how floating point arithmetics in
computers differs from mathematics' Real number
arithmetic.
```

A simple example: the output

Given $x = 2\pi = 6.2831853$, $y = \sin x = -2.4492127 \times 10^{-16}$, which nicely shows how floating point arithmetics in computers differs from mathematics' Real numbers arithmetics.

In this example I used the constant π, but I could have inserted in the text, inside or outside equations, any value returned by an R statement calculated based on any R object available from previous 'code chunks'.

Conclusions

- It is technically possible and not even difficult to achieve reproducible data analyses.
- Unless we are dealing with really ‘big data’ a desktop or laptop computer is all what is needed.
- There are R packages available that add support for parallel computing.
- National IT infrastructure such as CSC or the data centre of FMI at Södänkylä provide high computing capacity.
- Commercial cloud computing such as AWS (Amazon Web Services) is scalable and relatively cheap.
- R can run on all these systems, with scripts unchanged.
- So we can achieve portability, scalability and reproducibility.
- After a relatively slow start R has become almost the de facto standard for data analysis in academia and more recently it is being adopted in commercial enterprises including financial trading.

Resources

What statisticians blog about

http://matloff.wordpress.com/
http://robjhyndman.com/hyndsight/r/
http://robjhyndman.com/hyndsight/latex/
http://www.burns-stat.com/review-thinking-fast-slow-daniel-kahneman/

Resources

Reproducible research

http://www.economist.com/news/leaders/21588069-scientific-research-has-changed-world-now-it-need

http://www.nature.com/news/announcement-reducing-our-irreproducibility-1.12852

http://www.nature.com/news/science-joins-push-to-screen-statistics-in-papers-1.15509?WT.mc_id=FBK_NatureNews

doi:10.1016/j.csda.2008.03.004

Resources

Old and new tools

http://en.wikipedia.org/wiki/Literate_programming/
http://www.rstudio.com/
http://cran.r-project.org/
http://tug.org/
http://rmarkdown.rstudio.com/
http://www.rcpp.org/
https://www.slim.eos.ubc.ca/content/
repro-python-package-automating-reproducible-research-scie
http://h3rald.com/articles/glyph-050-released/
http://www.stack.nl/~dimitri/doxygen/
http://sphinx-doc.org/
http://www.burns-stat.com/r-navigation-tools/

Resources

My own stuff

http://blogs.helsinki.fi/kasbi-opiskelija/
http://r4photobiology.wordpress.com/
http://openinstruments.wordpress.com/
http://uv4growth.wordpress.com/
http://blogs.helsinki.fi/senpep-blog/
http://blogs.helsinki.fi/aphalo/

The End

- I prepared this slide presentation with LaTeX and R, on the RStudio IDE. I used Beamer and knitr, ggplot2 and other packages. This is all free open-source software, available for MS-Windows, OS-X, Linux, and Unix.
- This whole presentation (including examples) is coded in a single text file (except for the logos).
- To be continued. . . Part 4: Reproducible research.
- Thanks for listening!