Reproducible Research with Rmarkdown

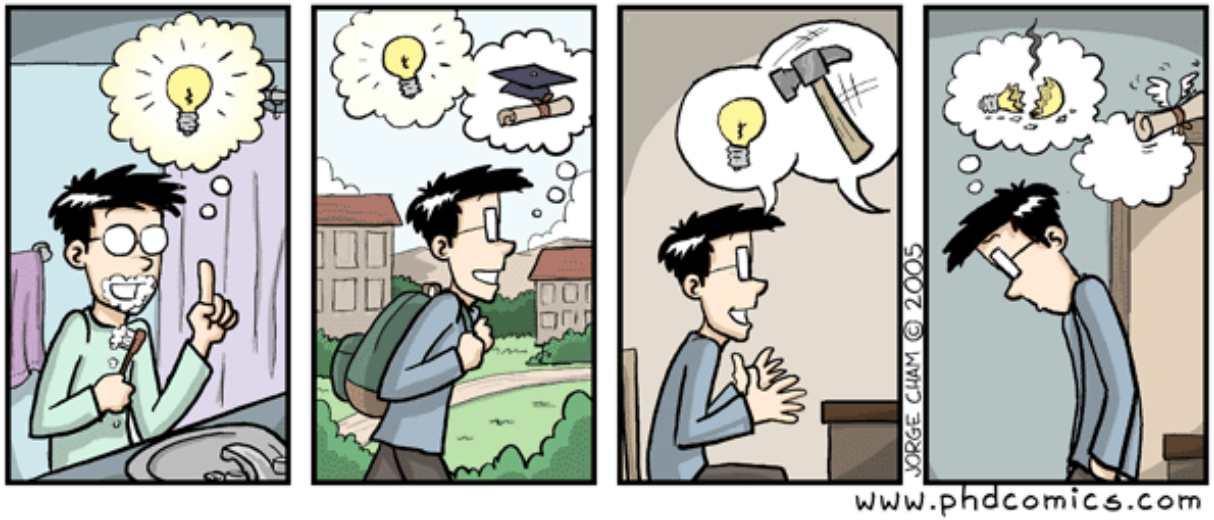
Data management, analysis, and paper writing all-in-one

Francisco Rodriguez-Sanchez [tinyurl.com/frod-san]
24/01/2014

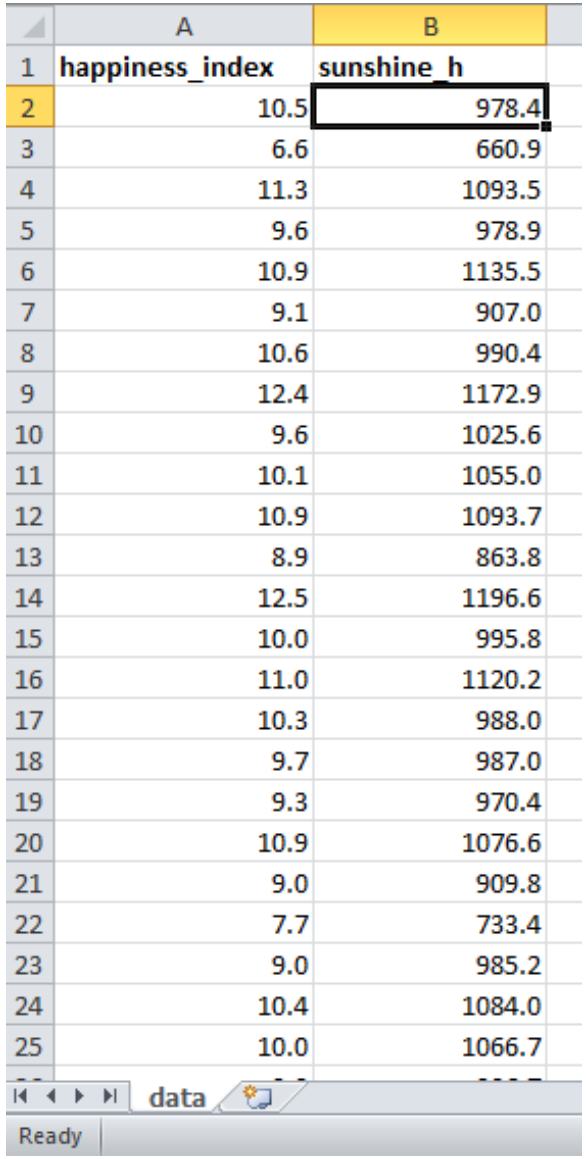
Typical workflow of many research projects

First have an idea

e.g. Does sunshine influence happiness?



1. Prepare data (Excel)

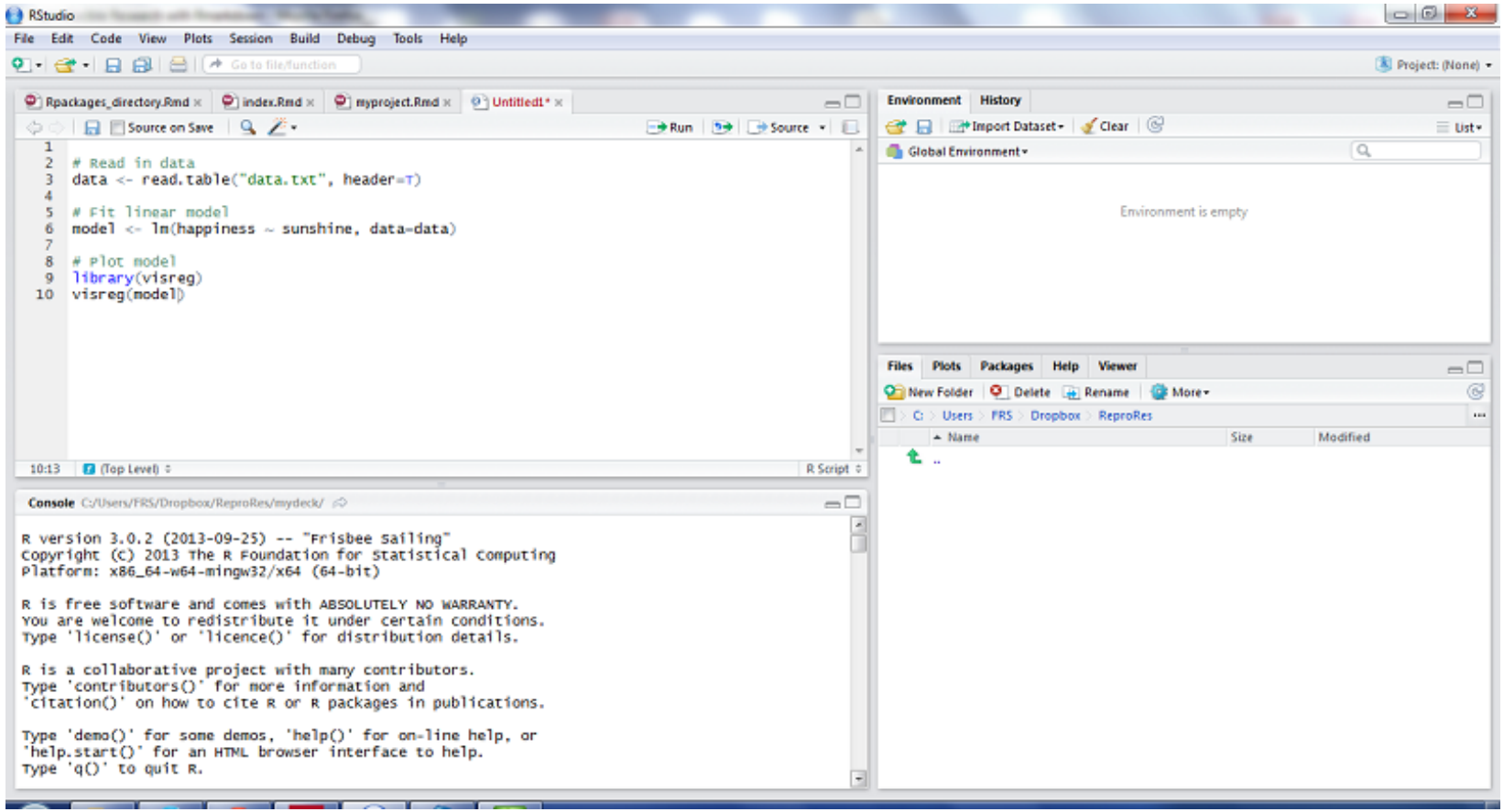


	A	B
1	happiness_index	sunshine_h
2	10.5	978.4
3	6.6	660.9
4	11.3	1093.5
5	9.6	978.9
6	10.9	1135.5
7	9.1	907.0
8	10.6	990.4
9	12.4	1172.9
10	9.6	1025.6
11	10.1	1055.0
12	10.9	1093.7
13	8.9	863.8
14	12.5	1196.6
15	10.0	995.8
16	11.0	1120.2
17	10.3	988.0
18	9.7	987.0
19	9.3	970.4
20	10.9	1076.6
21	9.0	909.8
22	7.7	733.4
23	9.0	985.2
24	10.4	1084.0
25	10.0	1066.7

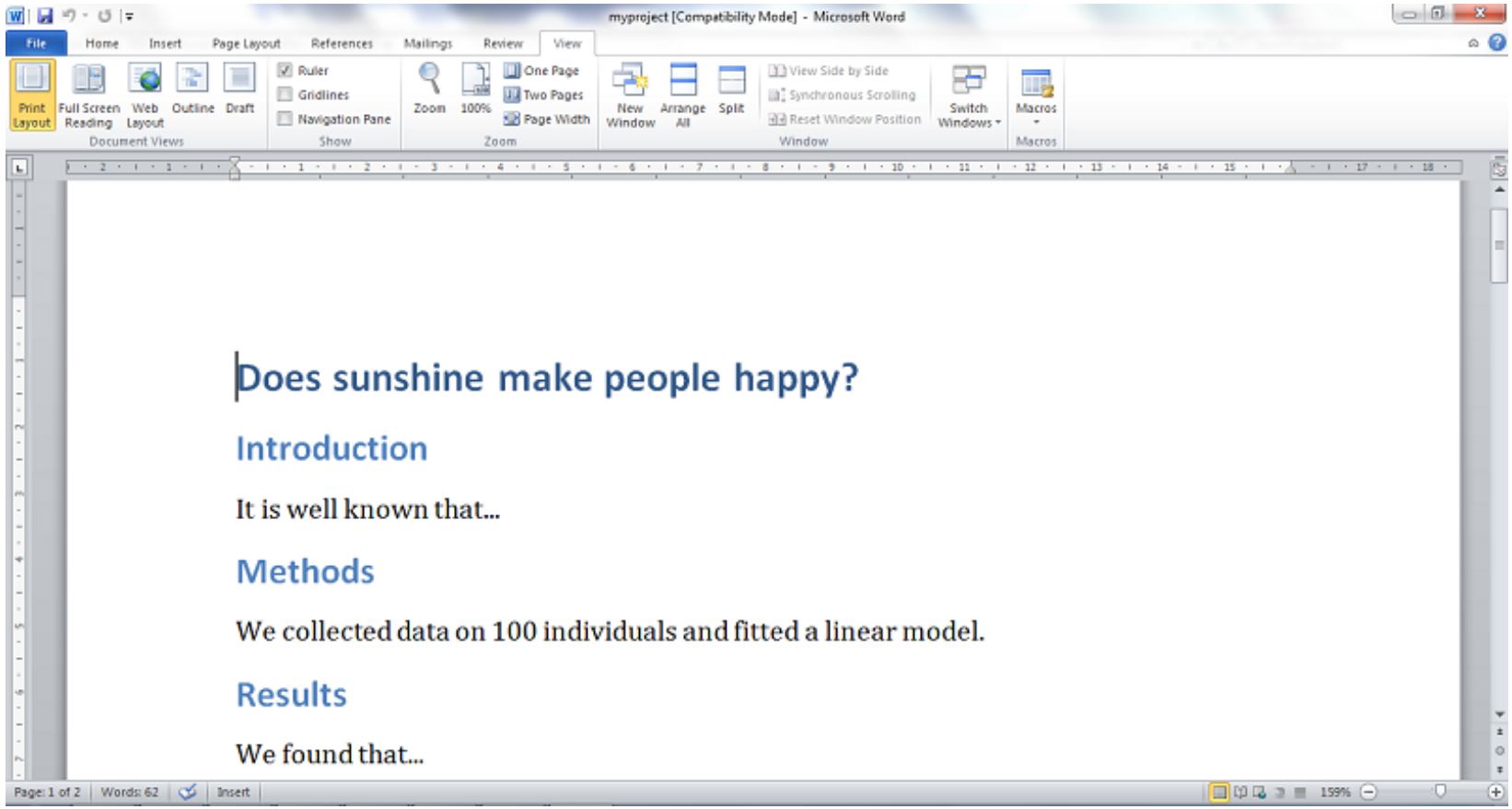
data

Ready

2. Do some analysis (R)



3. Write a report/paper (Word)

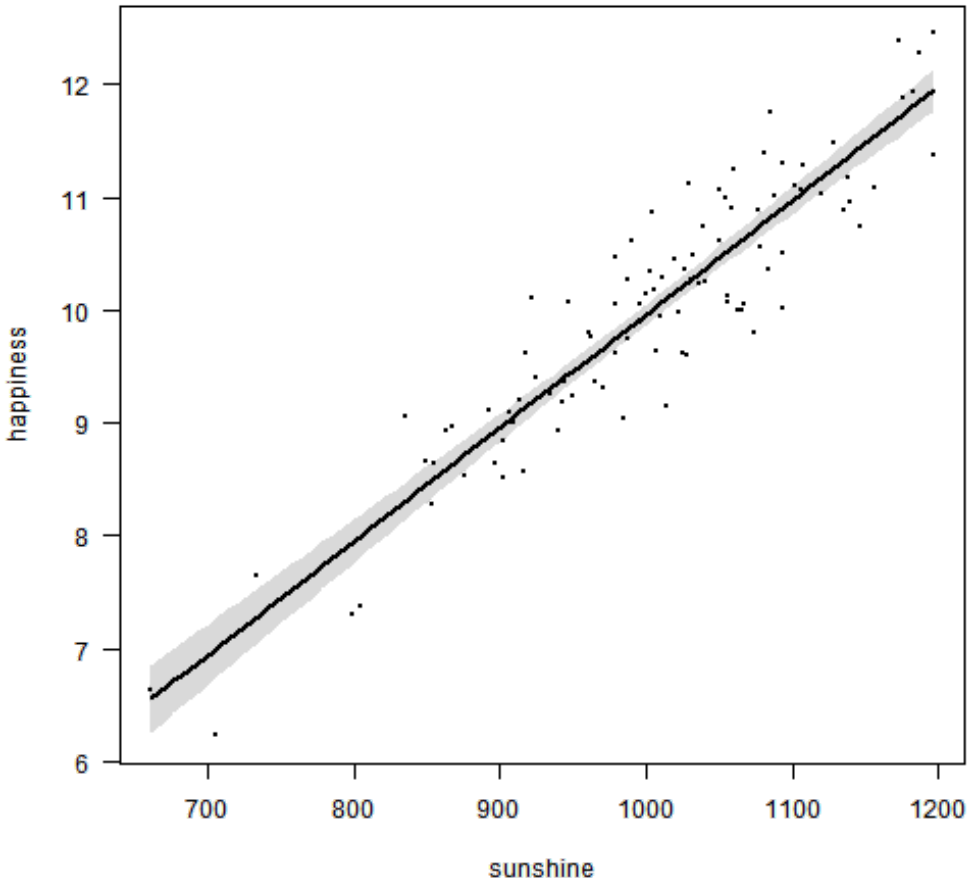


All results (figures, tables) **manually** imported to Word

This workflow is broken

1. Collect and manage data (**EXCEL**)
2. Analysis (**R**)
3. Write up (**WORD**)

Problems brought by the broken workflow



- What analysis is behind this figure? Did you account for [...] in the analysis?
- What dataset was used (e.g. final vs preliminary dataset)?
- Oops, there is an error in the data. Can you repeat the analysis? And update figures/tables in Word!
- As a coauthor/reader, I'd like to see the whole research process (how you arrived to that conclusion), rather than cooked manuscript with inserted tables/figures.

Rmarkdown allows us to fix the disconnect

Integrating

1. Data management
2. Data analysis
3. Writing up results

in a single dynamic document

REPRODUCIBLE RESEARCH!

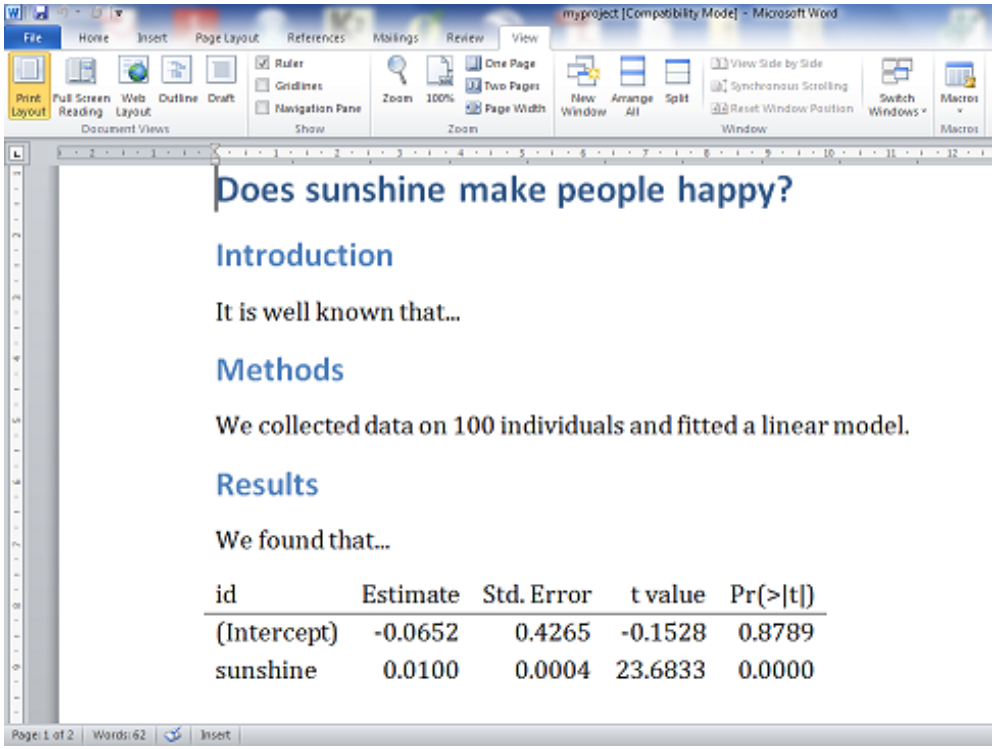
Let's make our sunshine-happiness research reproducible

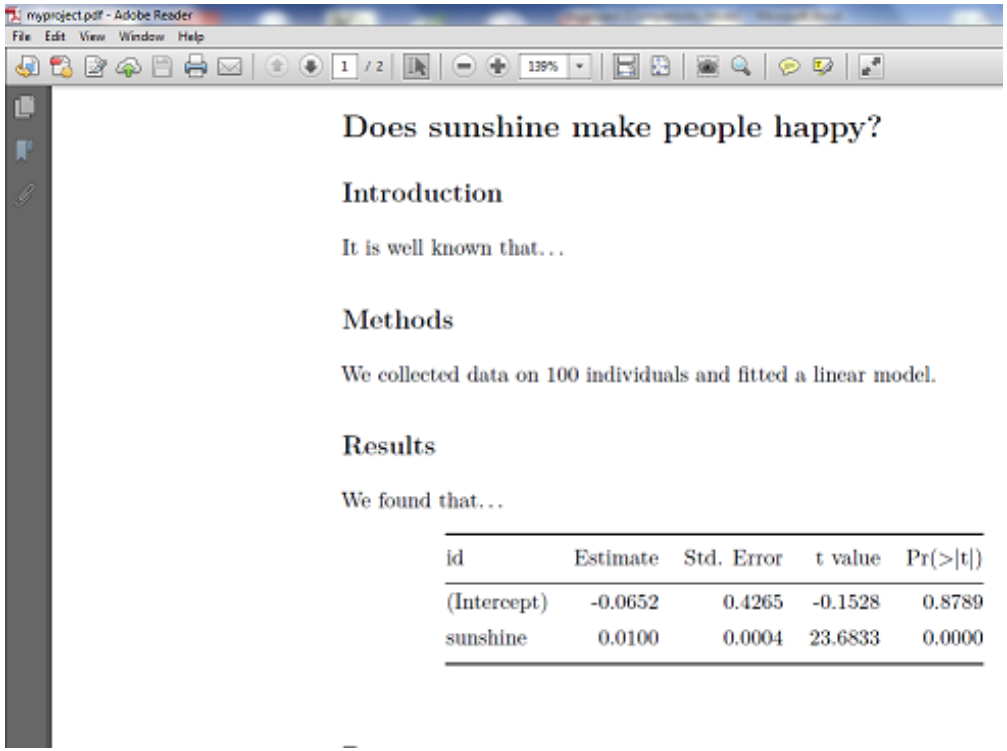
- Before starting any project read this: Designing projects at Nice R Code

```
proj/
├── R/
├── data/
├── doc/
├── figs/
├── output/
└── analysis.R
```

- Now see myproject.Rmd

With PANDOC, conversion to PDF or Word is straightforward





If spotting error in the data, or using different dataset...

make changes in Rmarkdown and report will update automatically

So... main advantages

Does sunshine make people happy?

Introduction

It is well known that...

Methods

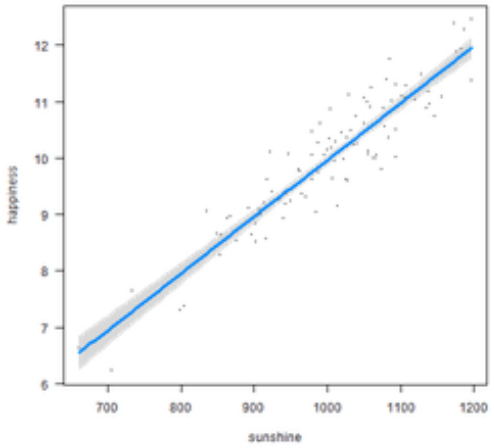
```
## Read data
data <- read.table("data.txt", header = T)
data[10, 1] <- 11  # correct error

# Fit linear model
model <- lm(happiness ~ sunshine, data = data)
```

We collected data on 100 individuals and fitted a linear model.

Results

We found that...

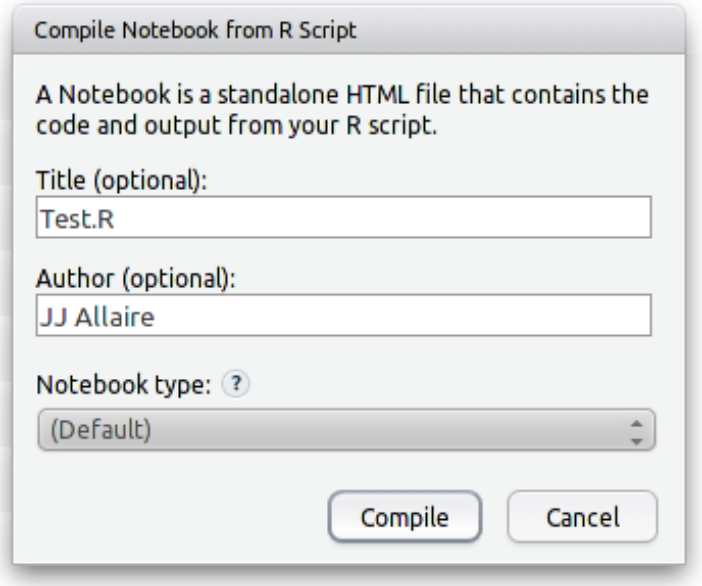
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.0986	0.4271	-0.2307	0.818
sunshine	0.0101	0.0004	23.7473	0.000



- Data management fully documented (no more manual changes in Excel!)
- Analysis fully documented
- Automated reports
- Lots of customisation options!

Convert your R scripts to Rmd, html, word, pdf...

- with one click in RStudio



- using knitr::spin or knitr::stitch [e.g. stitch("Rscript2convert.R")]

Some useful links:

- Rstudio docs
- Course on Reproducible Research by K. Broman
- Reproducible Research in Coursera
- Nice R code
- Reproducible Research with R and RStudio
- Example of full paper written in Rmd

More links (software/R packages)

- CRAN Task View on Reproducible Research
- knitr
- pandoc
- pander
- rapport
- reports

Next step: version control of Rmarkdown documents

Collaborative writing and version control without learning Git

- Draftin
- SciGit
- Authorea

To read more

OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Editorial

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve[1,2*], Anton Nekrutenko[3], James Taylor[4], Eivind Hovig[1,5,6]

Ten simple rules for Reproducible Research

1. For every result, keep track of how it was produced
2. Avoid manual data manipulation steps
3. Archive the exact versions of all external programs used
4. Version control all custom scripts
5. Record all intermediate results, when possible in standardized formats

Ten simple rules for Reproducible Research

1. For analyses that include randomness, note underlying random seeds
2. Always store raw data behind plots
3. Generate hierarchical analysis output, allowing layers of increasing detail to be inspected
4. Connect textual statements to underlying results
5. Provide public access to scripts, runs, and results

Find these slides at

- GitHub
- Figshare
- Code

END