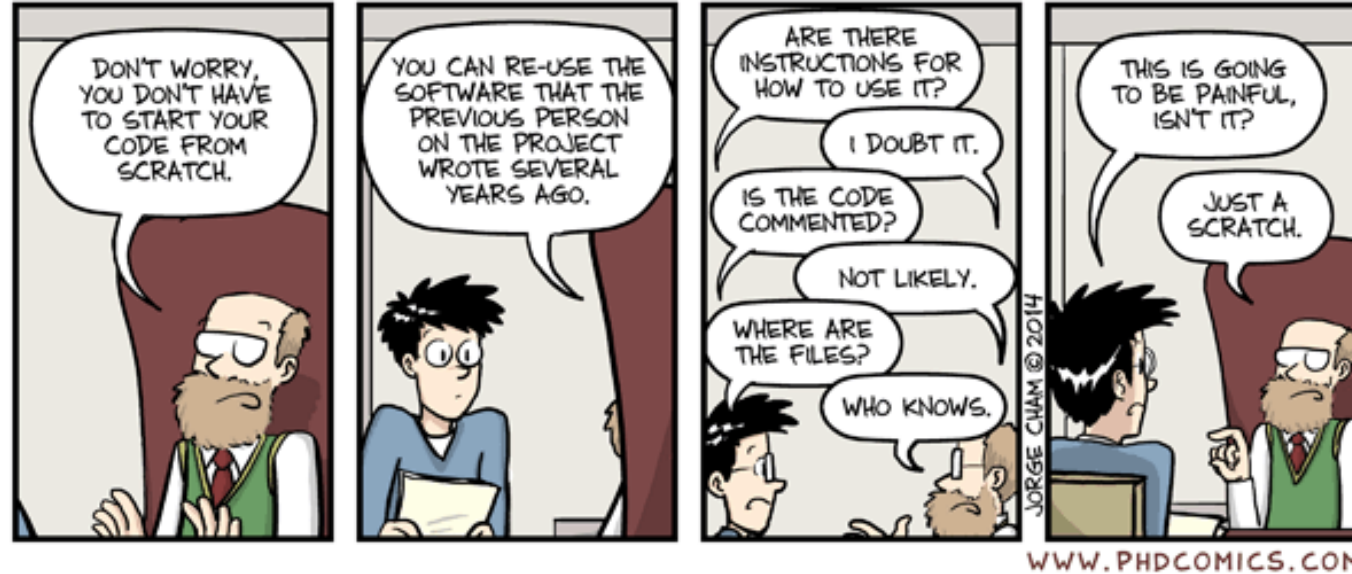


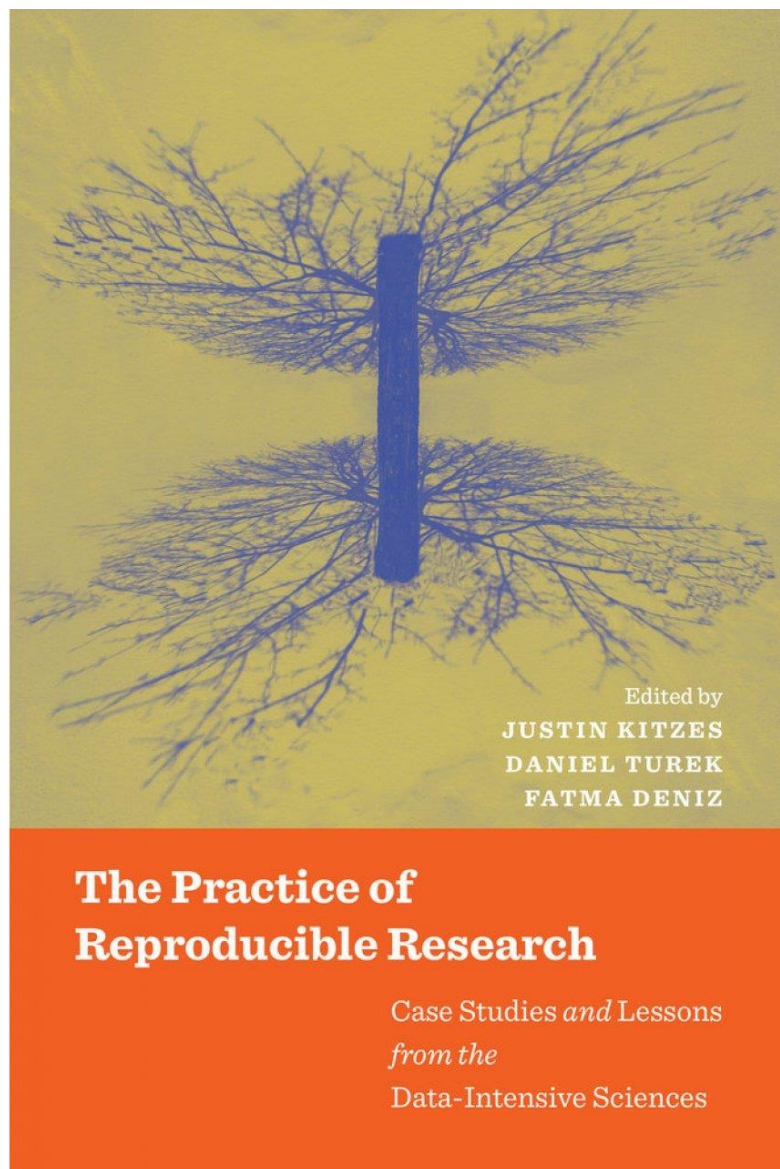
- > Computationally reproducible research with R
- >
- > R-user meeting, Wageningen UR
- > March 14th, 2018
- > Luc Steinbuch

Error: unexpected symbol in "Luc Steinbuch"

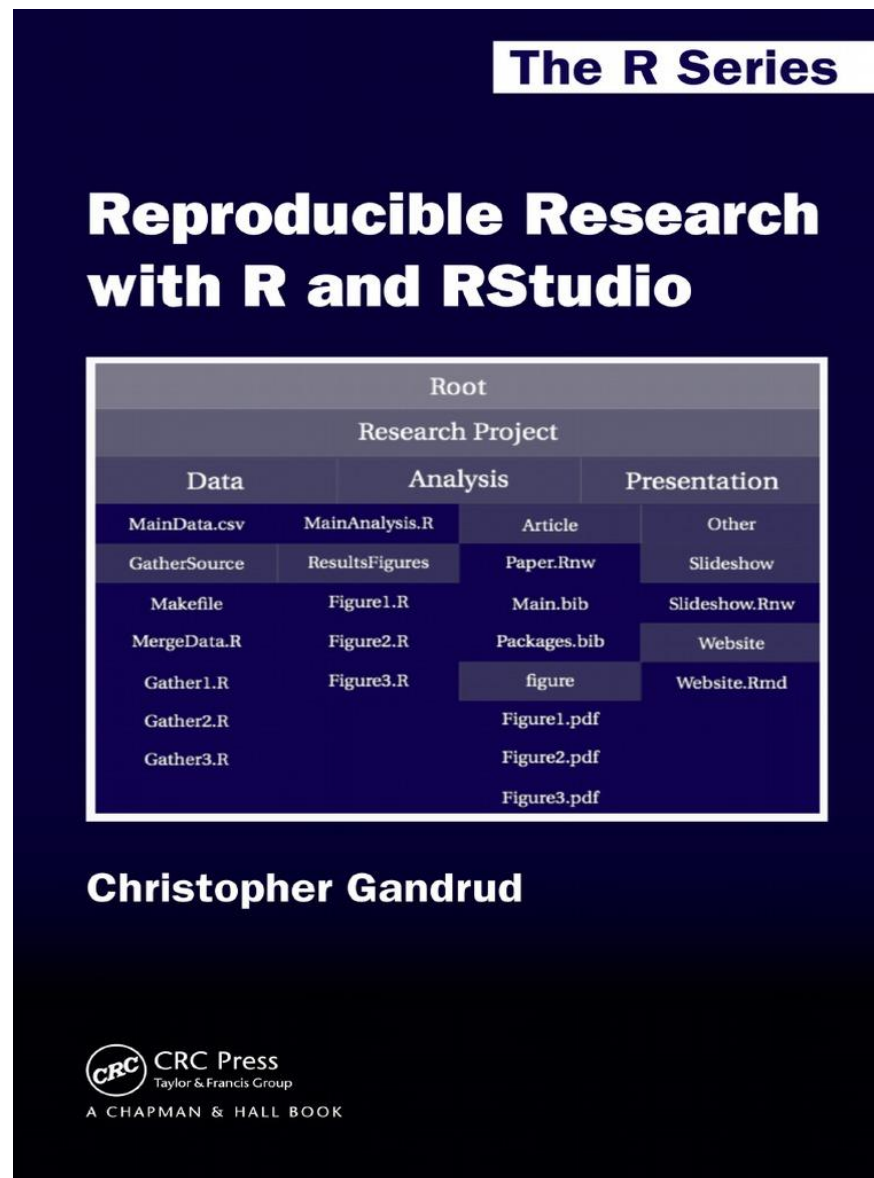


- > [1] Why, what and how reproducible research?
- > [2] Readable code
- > [3] Brief overview RMarkdown & application

Based on:

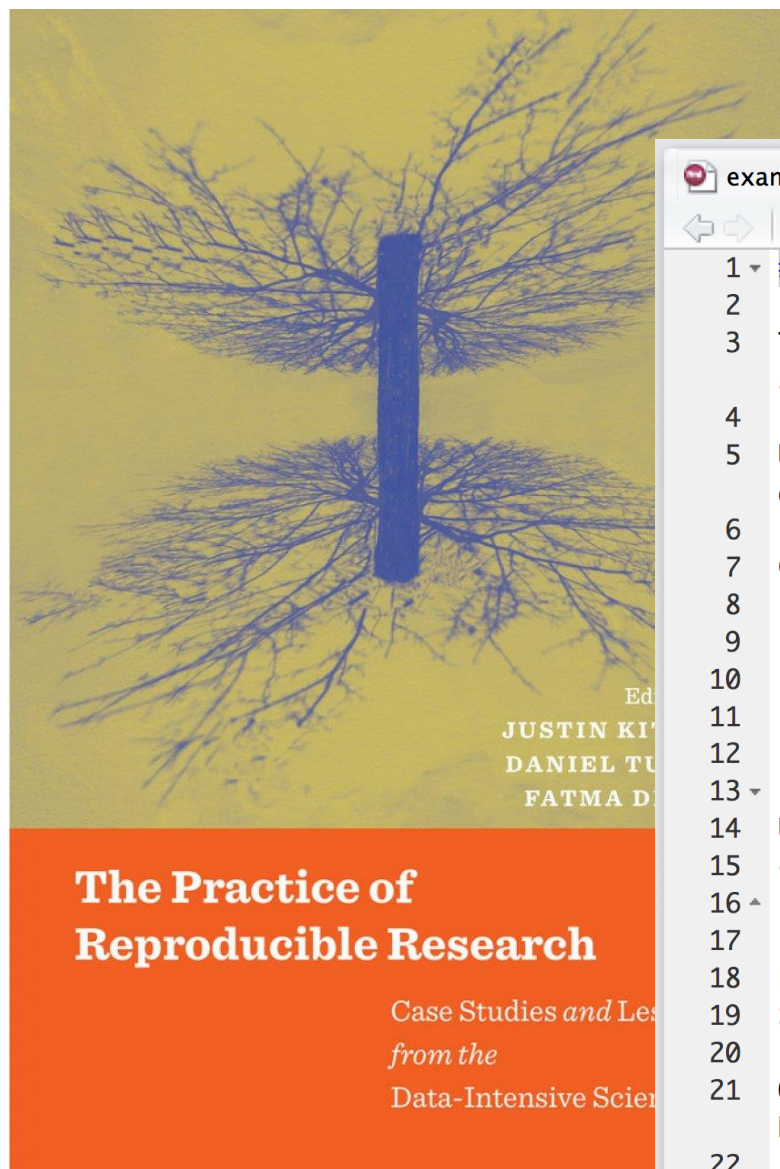


<https://www.practicereproduciblesearch.org/>



<https://englianhu.files.wordpress.com/2016/01/reproducible-research-with-r-and-studio-2nd-edition.pdf>

Based on:



<https://www.practicereproducibleresearch.org/>

The R Series

example.Rmd

1

Header 1

2

3 This is an R Markdown document. Markdown is a simple formatting syntax for authoring webpages.

4

5 Use an asterisk mark to provide emphasis, such as *italics* or **bold**.

6

7 Create lists with a dash:

8

9 - Item 1

10 - Item 2

11 - Item 3

12

13 ````

14 Use back ticks to create a block of code

15

16 ````

17

18 Embed LaTeX or MathML equations,

19
$$\frac{1}{n} \sum_{i=1}^n x_i$$

20

21 Or even footnotes, citations, and a bibliography. ^[1]

22

23 ^[1]: Markdown is great.

24

1:1

Header 1

R Markdown

example.html

Open in Browser

Find

Header 1

This is an R Markdown document. Markdown is a simple formatting syntax for authoring web pages.

Use an asterisk mark to provide emphasis, such as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

Use back ticks to create a block of code

Embed LaTeX or MathML equations, $\frac{1}{n} \sum_{i=1}^n x_i$

Or even footnotes, citations, and a bibliography. ¹

1. Markdown is great.↩

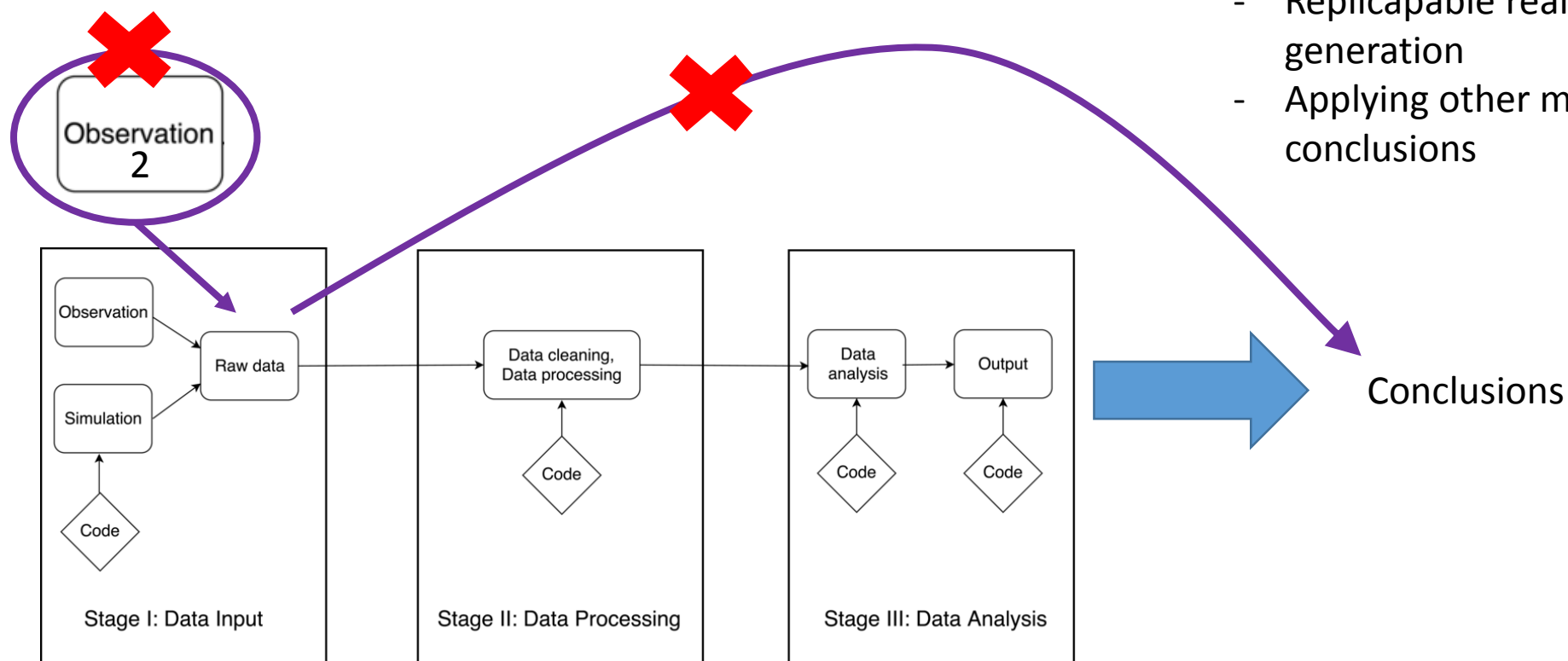
What?

“A research project is **computationally reproducible** if a second investigator (including you in the future) can recreate the final reported results of the project, including key quantitative findings, tables, and figures, given only a set of files and written instructions.”

(Kitzes et al., 2017)

Note that we do not consider here:

- Replicable real-world raw data generation
- Applying other methods to validate conclusions



Why? Scientific principles



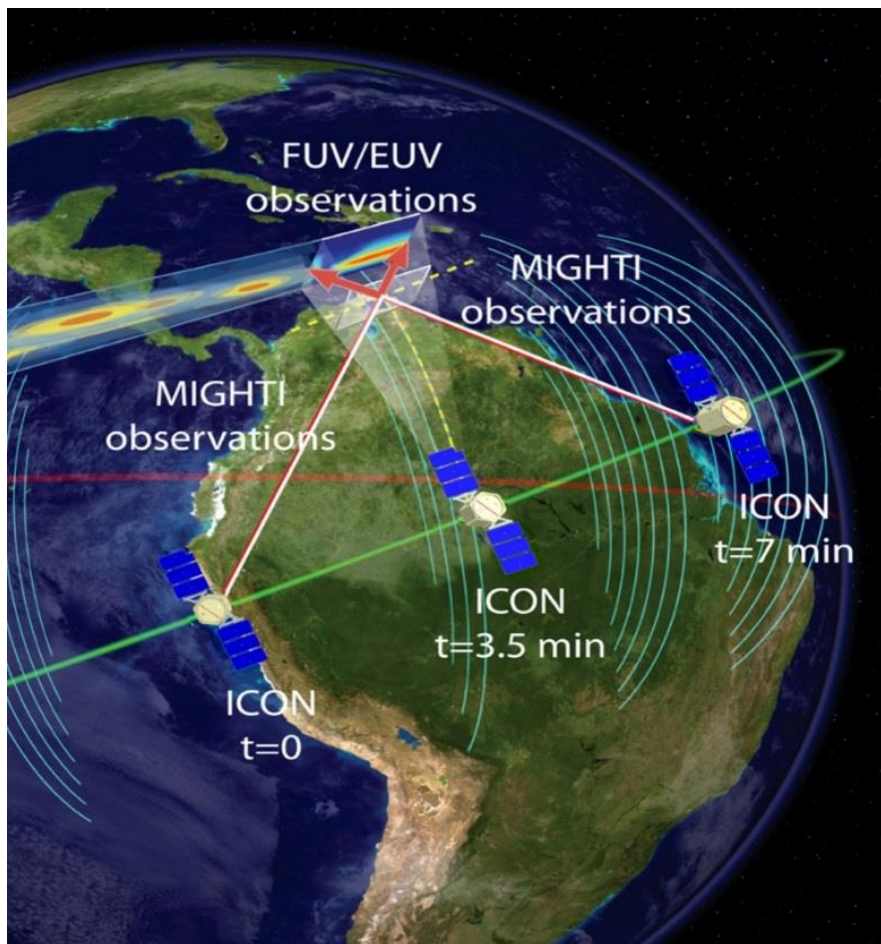
“Take nobody's word for it”

<https://royalsociety.org/>

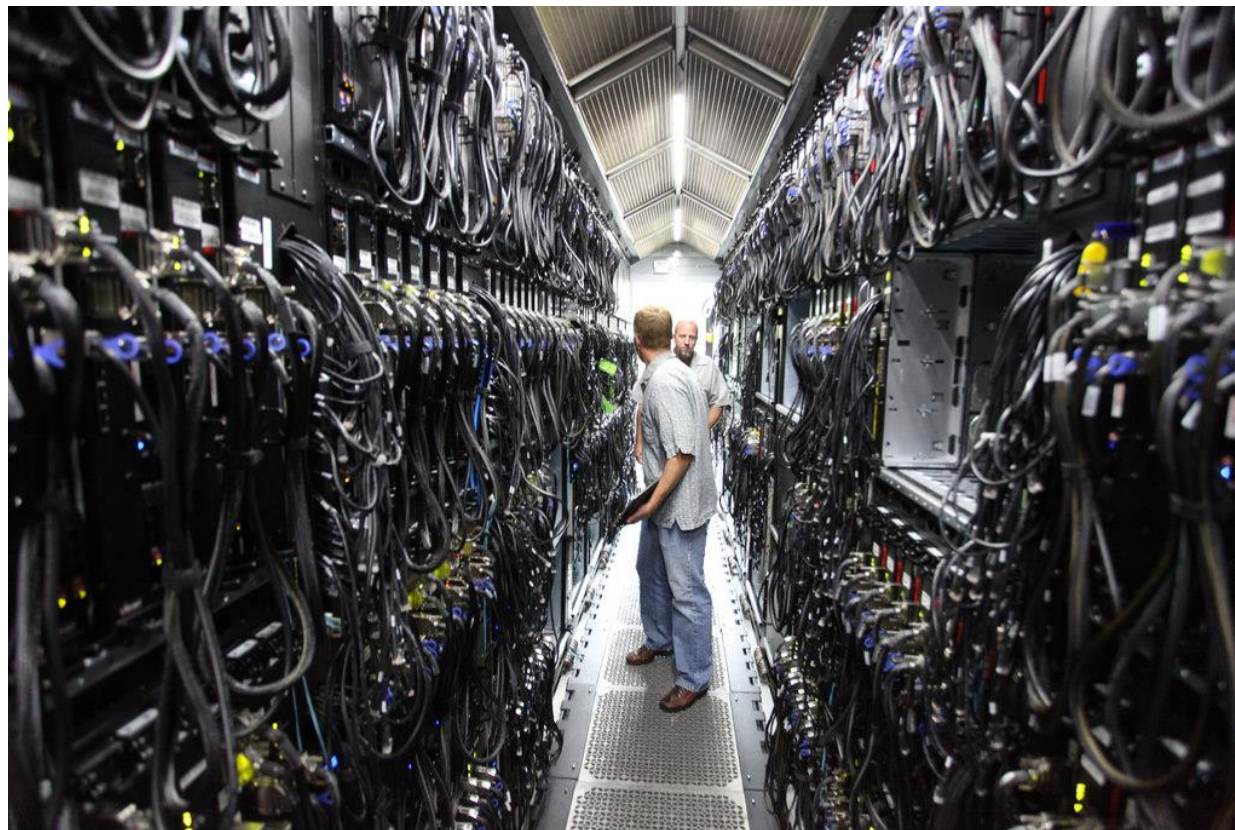


<https://www.nature.com/scitable/ebooks/english-communication-for-scientists-14053993/writing-scientific-papers-14239285>

Why? Data available and shareable



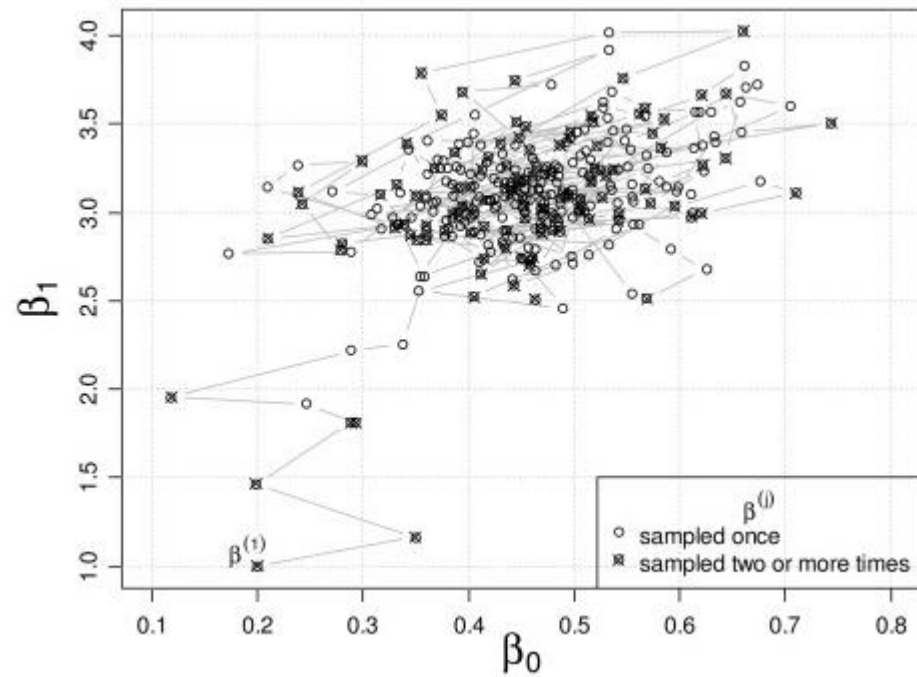
Source: <https://commons.wikimedia.org>



<https://www.flickr.com/photos/scobleizer/4870003098>

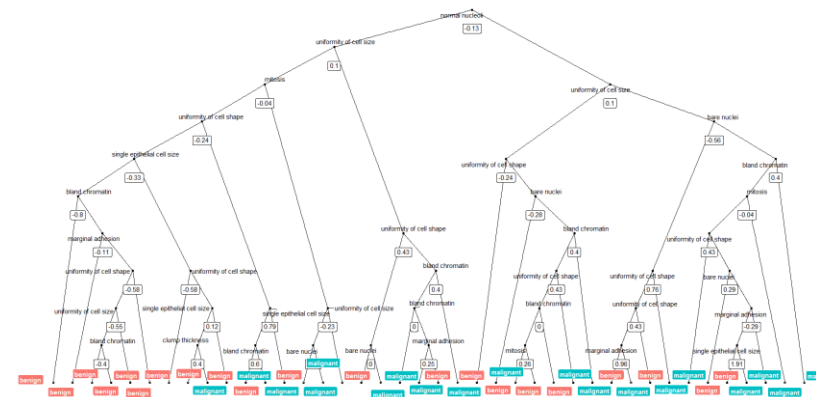
Why? Data processing more and more important

MCMC



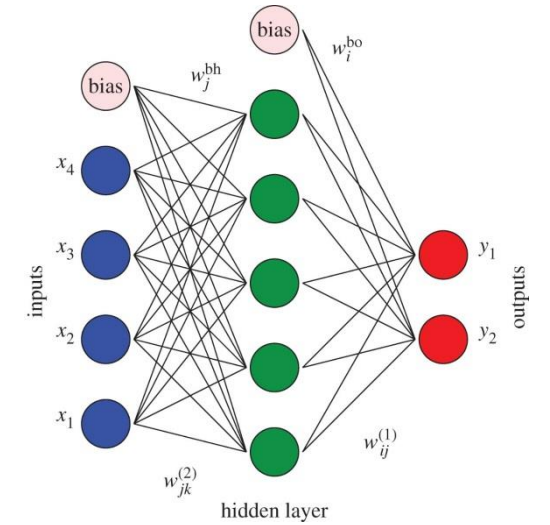
Doi 10.1016/j.geoderma.2017.12.010

Random Forests



https://shiring.github.io/machine_learning/2017/03/16/rf_plot_gggraph

Neural networks



Doi 10.1098/rsos.170175

Where to store data and/or code

Depository	4TU.Centre for Research Data.	DANS Easy (KNAW)
Focus	Technical + β sciences	All sciences, including humanities, social sciences, archeology etc
Size	10 GB; contact for more	100 MB zipped; contact for more
Duration	At least 15 years	At least 10 years

<https://www.wur.nl/en/Expertise-Services/Data-Management-Support-Hub/Browse-by-Subject/Publishing-your-dataset-in-a-repository.htm>

Other options:

- On **github** or **Git@WUR**
- Something **domain-specific**
- With an **R-package** on CRAN (advice: < 5 MB for code, documentation and data combined)
- At website scientific **journal** (€€€?);
- **Mendeley**: Max 10 GB per dataset, for free (yet) <https://data.mendeley.com/faq>
- With paper and R code at **Researchgate**; max 512MB for your whole profile (contact for more); citation and download overview for free – <https://explore.researchgate.net/display/support/Data>

Focuspoints when sharing data

Requirements

Data files: format

Meta-data (often fixed form with title, access restrictions etc.)

Documentation (extended textual description)

You can get a DOI, Digital object identifier

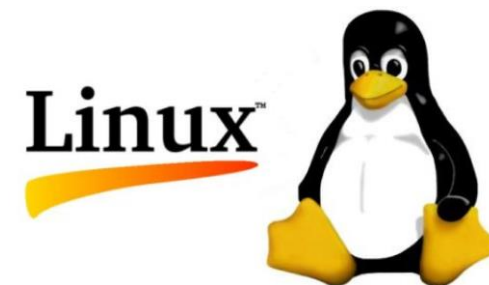
.txt, .csv etc.

But also (from researchgate):

- Nucleotide Sequences (.gb, .fas. .fasta)
- Alignments (.gb, .fas. .fasta)
- Protein Sequences (.gb, .fas. .fasta)
- Excel (.xls, .xlsx)
- Images (.png or similar)
- Animations (.gif)

Focuspoints when sharing code

- Access restrictions etc.
- Platform
- R, packages version
- Checked on errors
- Understandable, readable
- Explicit random seed



```
> sessionInfo()
R version 3.4.0 (2017-04-21)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

Matrix products: default

locale:
[1] LC_COLLATE=Dutch_Netherlands.1252  LC_CTYPE=Dutch_Netherlands.1252
[3] LC_MONETARY=Dutch_Netherlands.1252 LC_NUMERIC=C
[5] LC_TIME=Dutch_Netherlands.1252

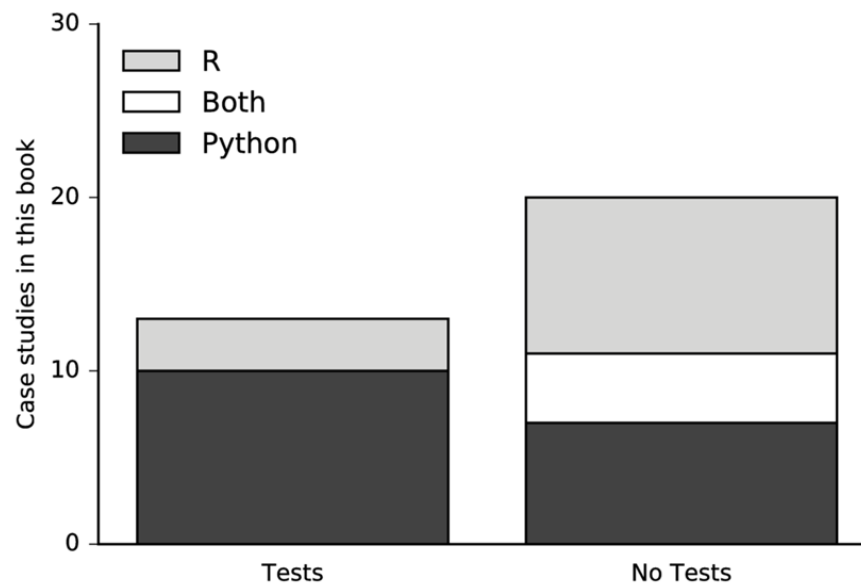
attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] knitr_1.20      tictoc_1.0      DEoptim_2.2-3   raster_2.5-8    rgdal_1.1-10
[6] magrittr_1.5    reshape2_1.4.2 ggplot2_2.1.0   MCMCpack_1.4-0  coda_0.19-1
[11] sp_1.2-5        MHadaptive_1.1-8 MASS_7.3-47

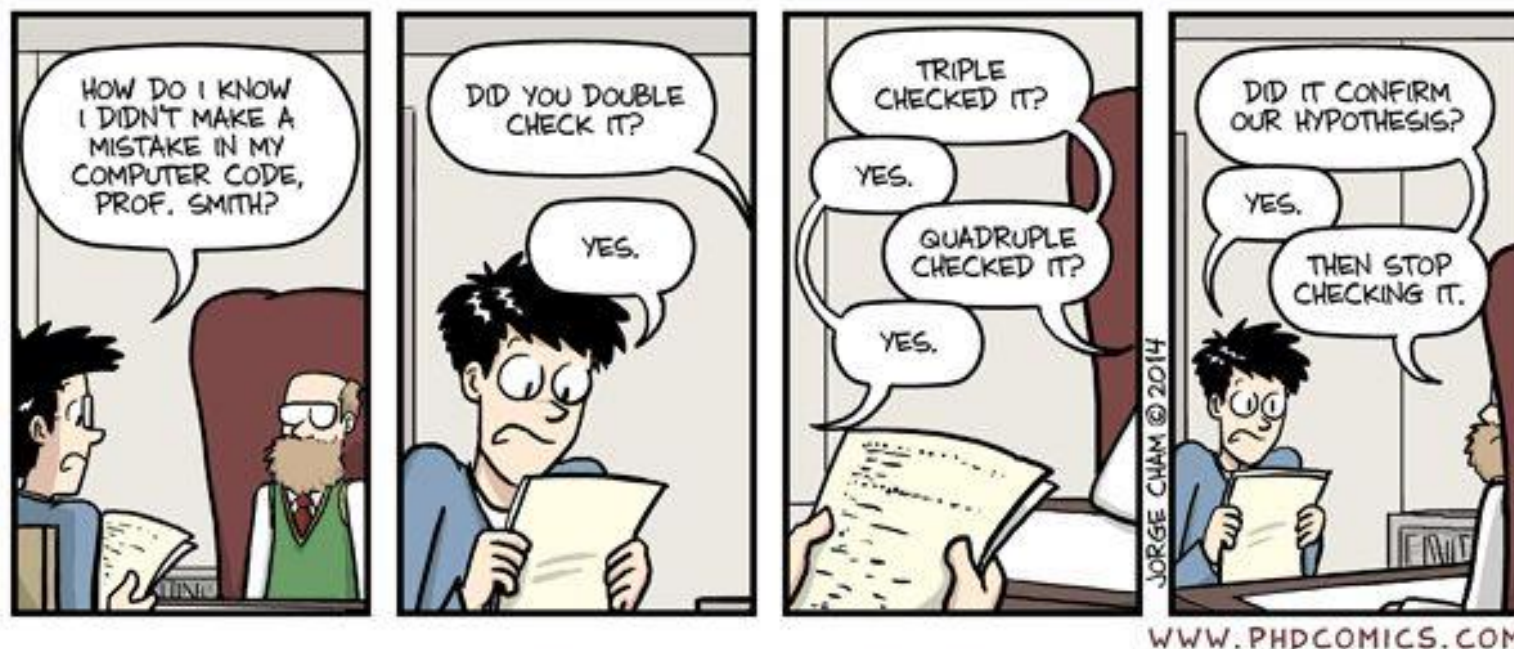
loaded via a namespace (and not attached):
[1] Rcpp_0.12.15    munsell_0.4.3    colorspace_1.3-2 lattice_0.20-35
```

Focuspoints when sharing code

- Access restrictions etc.
- Platform
- R, packages version
- **Checked on errors**
- Understandable, readable
- Explicit random seed



(Kitzes et al., 2017) ; n = 31



Focuspoints when sharing code

- Access restrictions etc.
- Platform
- R, packages version
- **Checked on errors**
- Explicit random seed
- Understandable, readable



```
> n      <- 5
> mean   <- 4
> var    <- 3.5
> rnorm(n, mean, var)
[1]  1.7841321 -0.9912195  6.2078209  2.0045731 -0.7611575
```

Focuspoints when sharing code

- Access restrictions etc.
- Platform
- R, packages version
- **Checked on errors**
- Explicit random seed
- Understandable, readable

Help function:

```
rnorm(n, mean = 0, sd = 1)
```

```
> number_repetitions <- 5
> mean_population <- 4
> variance_population <- 3.5
>
> rnorm(n = number_repetitions,
+       mean = mean_population,
+       sd = sqrt(variance_population)
+       )
[1] 5.895903 3.875858 3.022181 4.511389 3.301425
```

```
> n <- 5
> mean <- 4
> var <- 3.5
> rnorm(n, mean, var)
[1] 1.7841321 -0.9912195 6.2078209 2.0045731 -0.7611575
```

Focuspoints when storing code

- Access restrictions etc.
- Platform
- R, packages version
- Checked on errors
- **Explicit random seed**
- Understandable, readable

```
> rnorm(n      = number_repetitions,
+       mean    = mean_population,
+       sd      = sqrt(variance_population)
+       )
[1] 5.895903 3.875858 3.022181 4.511389 3.301425
> rnorm(n      = number_repetitions,
+       mean    = mean_population,
+       sd      = sqrt(variance_population)
+       )
[1] 2.518553 6.506036 5.512475 7.199004 2.932298
> set.seed(671)
> rnorm(n      = number_repetitions,
+       mean    = mean_population,
+       sd      = sqrt(variance_population)
+       )
[1] 3.0133336 -0.2559971 5.8221417 3.0651584 3.5877066
> set.seed(671)
> rnorm(n      = number_repetitions,
+       mean    = mean_population,
+       sd      = sqrt(variance_population)
+       )
[1] 3.0133336 -0.2559971 5.8221417 3.0651584 3.5877066
```


Why aren't people sharing their data and code?

1. It takes **effort to clean up data and code** to put them in a format where you can share them.
2. **Privacy** issues (for example social sciences)
3. You do **not want to give away your data** too soon. Science is competitive!
4. Because **others might go through it and find mistakes in your analysis**
5. You're John Lott and **you lost all traces** of your data – so nothing to share.
6. You're Diederik Stapel and you never did the study in the first place. You can't share your data because the data **never existed**.

And: plain ownership of the data

Inspired on: <http://andrewgelman.com/2015/09/14/its-not-so-easy-to-share-data-and-code-and-there-are-lots-of-bureaucrats-who-spend-their-time-making-it-even-more-difficult>

From Wikipedia:

“However, in 2000 **Lott** was unable to produce the data or any records showing that the survey had been undertaken. He said the 1997 hard drive crash ...”

“In September 2011, Tilburg University suspended **Stapel** due to his fabrication of data used in research publications.”

More about readable code

Let's take an example, from https://www.researchgate.net/publication/322741243_S1_R-script

Data File available

S1 R-script

January 2018

License · [CC BY 4.0](#)

 Katarzyna Wojczulanis-Jakubas ·  Marcelo Araya-Salas ·  Dariusz Jakubas

Overview

Comments

Citations

References

Related research (10+)

Description

R-codes (Monte Carlo routines) to analyze coordinated provisioning in the Dovekie. (R)

Linked Research (1)

Seabird parents provision their chick in a coordinated manner

Article

Full-text available

Jan 2018 · PLoS ONE

Pair collaborative behavior may play an important role in avian reproduction. However, evidence for this mainly comes from certain ecological groups (e.g. passerines). We studied the coordination of parents in foraging and its effect on ...

[View](#) [Download](#)

134 Reads

File



pone.0189969.s002.R

9.79 KB

[Download](#)

```
#We suggest removing all objects in your enviroment fist
rm(list = ls())

#read supplementary data
prov.data <- read.csv("Supplementary materials_Data.csv")
```

S1 Data

Data

File available

Jan 2018

Row data in csv format. (CSV)

[View](#)

[Recommend](#)

2 Reads

More about readable code

```

1  #We suggest removing all objects in your enviroment fist
2  rm(list = ls())
3
4  #read supplementary data
5  prov.data <- read.csv("Supplementary materials_Data.csv")
6
7
8  #install metap (for combining probabilities) if you haven't before
9  if(!"metap" %in% installed.packages()[,"Package"]) install.packages("metap")
10
11
12 ▾ ##### TEST 1 #####
13 ▾ ##### Probability that only one bird is at the colony at any time by shuffling all 10 min segment:
14  #set number of iterations
15  reps = 10000
16
17  #t1 == test1
18  t1.pair.period <- parallel::mclapply(unique(prov.data$pair.obs.period), mc.cores = 3, function(i)
19 ▾ {
20    #subset for each nest.observation (nest+observation period)
21    a <- prov.data[prov.data$pair.obs.period == i,]
22
23    #put male and female data as independent columns (sex is a single column in p)
24    d <- cbind(as.character(a$strip.type[a$sex == "m"]), as.character(a$strip.type[a$sex == "f"]))
25
26    #convert NAs to at the colony (CO)
27    # d <- apply(b, 2, function(x) {levels(x)<-c("ST","LT","CO")
28    # x[is.na(x)]<-"CO"
29    # return(x)})
30

```

In total 264 lines of code, in one script

More about readable code

```

1  #We suggest removing all objects
2  rm(list = ls())
3
4  #read supplementary data
5  prov.data <- read.csv("Suppl
6
7
8  #install metap (for combining
9  if(!"metap" %in% installed.p
10
11
12  #### TEST 1 #####
13  #### Probability that only
14  #set number of iterations
15  reps = 10000
16
17  #t1 == test1
18  t1.pair.period <- parallel::mclapply(unique(prov.data$pair.obs.period), mc.cores = 3, function(i)
19  {
20    #subset for each nest.observation (nest+observation period)
21    a <- prov.data[prov.data$pair.obs.period == i,]
22
23    #put male and female data as independent columns (sex is a single column in p)
24    d <- cbind(as.character(a$strip.type[a$sex == "m"]), as.character(a$strip.type[a$sex == "f"]))
25
26    #convert NAs to at the colony (CO)
27    # d <- apply(b, 2, function(x) {levels(x)<-c("ST","LT","CO")
28    # x[is.na(x)]<-"CO"
29    # return(x)})
30

```


[Publish](#)
[About](#)

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Seabird parents provision their chick in a coordinated

Katarzyna Wojczulanis-Jakubas , Marcelo Araya-Salas, Dariusz Jakubas

Published: January 10, 2018 • <https://doi.org/10.1371/journal.pone.0189969>

In total 264 lines of code, in one script

More about readable code

Another example https://www.researchgate.net/publication/320272803_R_script_and_data

```

1  # Methods S2: R script used to perform the statistical analyses
2
3  rm(list=ls(all=TRUE))
4
5  library(vegan)
6
7  # bioenv function modified to take a distance matrix as input
8  # to see the original script: print(bioenv.default)
9
10 bioenv_DIST <- function (comm, env, method, index, upto, trace, partial)
11 {
12   if (is.null(partial)) {
13     corfun <- function(dx, dy, dz, method) {
14       cor(dx, dy, method = method)
15     }
16   }
17   else {
18     corfun <- function(dx, dy, dz, method) {
19       rxy <- cor(dx, dy, method = method)
20       rxz <- cor(dx, dz, method = method)
21       ryz <- cor(dy, dz, method = method)
22       (rxy - rxz * ryz)/sqrt(1 - rxz * rxz)/sqrt(1 - ryz * ryz)
23     }
24   }
25   if (!is.null(partial))
26     partpart <- deparse(substitute(partial))
27   else partpart <- NULL
28   if (!is.null(partial) && !inherits(partial, "dist"))
29     partial <- dist(partial)
30   if (!is.null(partial) && !pmatch(method, c("pearson", "spearman"),
31                                     nomatch = FALSE))
32     stop("method ", method, " invalid in partial bioenv")

```

```

> print(bioenv.default)
Error in print(bioenv.default) : object not found

```

In total 213 lines of code, in one script

More about readable code

```
154
155 vpd      <- compo_BIO[,11:14] # weather variables
156 t_mean   <- compo_BIO[,15:18]
157 t_sd     <- compo_BIO[,19:22]
158 dew      <- compo_BIO[,23:26]
159 aur_fr    <- compo_BIO[,27:30] # climate variables
160 aur_pp    <- compo_BIO[,31:34]
161 aur_tm    <- compo_BIO[,35:38]
162
163 # Geographical distance between elevation sites
164
165 geo      <- vegdist(compo_BIO[,c("x","y","z")], "euc", diag=F, upper=F, binary=F)
166
167 # # BIOENV selection of the best season for climate variables, and the best time span for weather variable
168
169 result <- array(NA, c(7,2))
170 dimnames(result)[[1]] <- c("vpd", "t_mean", "t_sd", "dew", "aur_fr", "aur_pp", "aur_tm")
171 dimnames(result)[[2]] <- c("selected_period", "mantel_R")
```



```

154
155 vpd      <- compo_BIO[,11:14]
156 t_mean   <- compo_BIO[,15:18]
157 t_sd     <- compo_BIO[,19:22]
158 dew      <- compo_BIO[,23:26]
159 aur_fr    <- compo_BIO[,27:30]
160 aur_pp    <- compo_BIO[,31:34]
161 aur_tm    <- compo_BIO[,35:38]
162
163 # Geographical distance between
164
165 geo      <- vegdist(compo_BIO
166
167 # # BIOENV selection of the be
168
169 result <- array(NA, c(7,2))
170 dimnames(result)[[1]] <- c("v
171 dimnames(result)[[2]] <- c("s

```

Table 2 List of climatic and weather variables calculated for each elevation site

Abbreviation	Description
Climatic variables	
<i>t_m_S</i>	Mean temperature in season <i>S</i>
<i>pp_S</i>	Mean precipitation in season <i>S</i>
<i>fr_S</i>	Mean no. of days of frost in season <i>S</i>
Weather variables	
<i>t_m_D</i>	Mean temperature over the <i>D</i> days before sampling
<i>t_sd_D</i>	Temperature standard deviation over the <i>D</i> days before sampling
<i>dew_point_D</i>	Number of hours above dew point over the <i>D</i> days before sampling
<i>VDP_10max_D</i>	Mean of the ten highest vapor pressure deficit values over the <i>D</i> days before sampling



More about readable code

Final example <http://data.4tu.nl/repository/uuid:8dd8a61b-ca67-447a-ac20-e2d23472142e>

Dataset | **Code for alpha irradiated FNTD processing and microdosimetry**

▶▶▶▶▶ Link/cite as <https://doi.org/10.4>

▼ go to DATA section ▼

title	?	Code for alpha irradiated FNTD processing and microdosimetry
creator	?	 Kouwenberg, J.J.M. (Jasper)
contributor	?	TU Delft, Faculty of Applied Sciences, Department of Radiation Science & Technol
date accepted	?	2017-10-04
date created	?	2016 through 2017
date published	?	2017
description	?	Collection of R and Java scripts for the processing of alpha irradiated Fluorescent I images and spheroid simulation. To be used together with the uuid:684161d8-fbd4- a modified version of a 2016 fork of the public https://github.com/FNTD/R-package
language	?	en
publisher	?	Delft University of Technology
subject	?	Alpha microdosimetry ◇ Fluorescent Nuclear Track Detectors (FNTD) ◇ FNTD proc
▲ in collection	?	General collection of datasets
is software for	?	Am241 irradiated Fluorescent Nuclear Track Detectors
licence	?	 General terms of use

DATA

[readme.txt - dataset documentation](#) (text/plain)

 **Dataset files (91.6 MiB) >> [download complete dataset \(zip\)](#) | [download separate files](#)**

 bag-info

 contents of this dataset, 317 files

FNTD processing, Fluorescent Nuclear Track Detectors, Alpha
microdosimetry, Spheroid simulation dataset

Code files

/R-package
 (...)

/R

Collection of scripts for processing and analyzing FNTDs.
Relies heavily on the accompanying R-package library. Also contains
code for calculation and simulation of microdosimetric spectra in
cells.

/R/Add.Scan.To.DB.R

Script for adding FNTD scans to the database. Requires surface
plane equation and an ID.

/R/CellHit/Cell.LET.distribution.R

Calculates the LET distribution given a cell image and a
collection of FNTD images. Only works for alpha particles.

(this list continues for about two pages)

```
1 rm(list = ls())
2 cat("\014")
3
4 library(rgl)
5
6 load(file = 'Data/FNTD_Refl_Regression.rda')
7
8 filename = ""
9 #reflectance plane
10 vec = 0:98.41
11 x = c(0, 45, 90)
12 y = c(0, 0, 0)
13 z = c(1, 1, 1)*.49
14 img_offset = (0)*.49
15
16 fit <- lm(z ~ x + y)
17
18 x1<-sapply(vec, function (x) rep(x,length(vec)))
19 x1<-as.vector(x1)
20 y1 = rep(vec, length(vec))
21 z1 = fit$coefficients[1] + fit$coefficients[2]*x1 + fit$coefficients[3]*y1
22 # plot3d(x1,y1,z1)
23
24 print(summary(fit))
```



```
1  ## Main script for "An analytical approach to Bayesian area-to-point kriging"
2  ## Luc Steinbuch, Wageningen UR
3  ## Spring 2018
4
5  ##### Load packages and supporting functions #####
6  source("packages_and_postParam_functions.R")
7  #file.edit("packages_and_postParam_functions.R")
8
9  ##### Prepare data #####
10 source("prepare_simulation_data.R")
11 #file.edit("prepare_simulation_data.R")
12
13 ##### Inspect & present data #####
14 source("inspect_simulation_data.R")
15 #file.edit("inspect_simulation_data.R")
16
17 ##### Estimating and adding linear models #####
18 source("add_lm_simulation_data.R")
19 #file.edit("add_lm_simulation_data.R")
20
21 ##### Analytical analysis and prediction #####
22 source("analytical_analysis_simulation_data.R")
23 #file.edit("analytical_analysis_simulation_data.R")
```

RMarkdown

- Run RMarkdown_example_1.Rmd (don't forget `install.packages("rmarkdown")`)
- Note environment:

```
> d
```



```
Error: object 'd' not found
```
- Note notebook in Rstudio (since 2016): run by chunk

```
12 ## Markdown with R code
```

```
13
```

```
14 ```{r}
```

```
15 d <- data.frame(participants=1:10,height=rnorm(10,sd=30,mean=170))
```

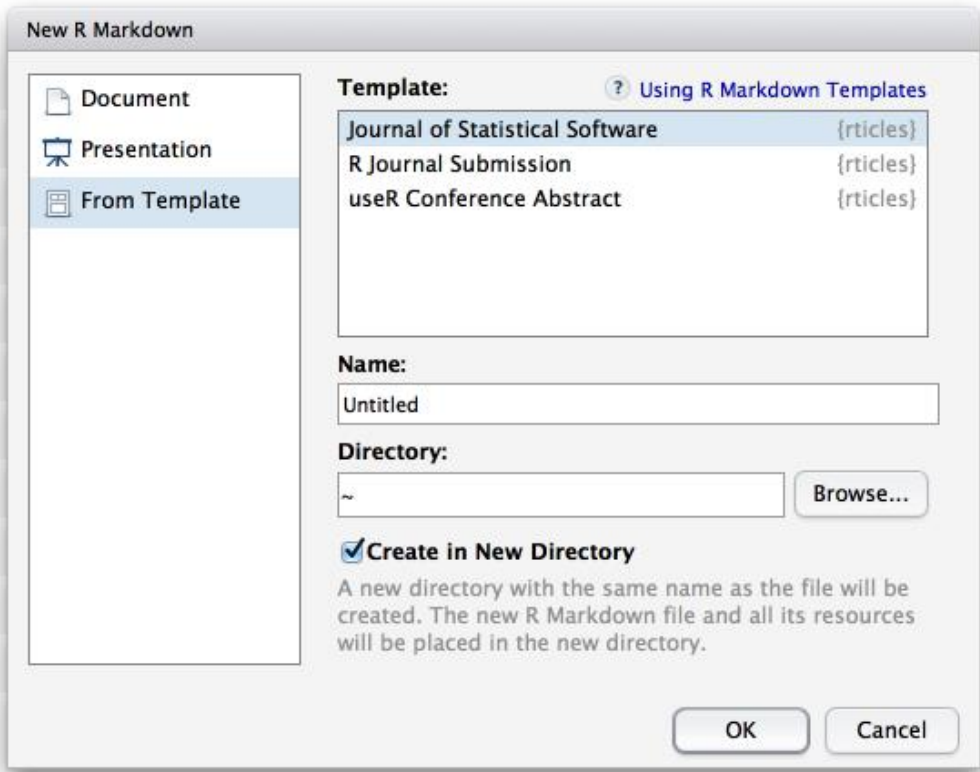
```
16 summary(d)
```

```
17 ```
```

participants	height
Min. : 1.00	Min. :127.7
1st Qu.: 3.25	1st Qu.:146.8
Median : 5.50	Median :191.3
Mean : 5.50	Mean :181.5
3rd Qu.: 7.75	3rd Qu.:209.0
Max. :10.00	Max. :234.0

RMarkdown

Powerful, versatile



<https://rmarkdown.rstudio.com/gallery.html>

Gallery

Check out the range of outputs and formats you can create using R Markdown.

Documents

With R Markdown, you write a single .Rmd file and then use it to render finished output in a variety of formats.



Interactive Documents

Combine R Markdown with htmlwidgets or the shiny package to make interactive documents.



<https://rmarkdown.rstudio.com/gallery.html>

RMarkdown

Powerful, versatile

```

1 ---
2 title: "Untitled"
3 output: pdf_document
4 bibliography: library.bib
5 ---
6
7 #Header 1
8
9 Lorem ipsum dolor sit amet, consectetur adipiscing elit
  [@author2000], sed do eiusmod tempor incididunt ut labore et
  dolore magna aliqua.
10
11 #References

```

Untitled

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit (Author 2000), sed do eiusmod tempor incididunt labore et dolore magna aliqua.

References

Author, A. 2000. "Title of article." *Journal of Tudududu* 3 (2): 112–901.

<https://rosannavanhespenresearch.wordpress.com/2016/02/17/writing-your-thesis-with-rmarkdown-2-making-a-chapter>

+ Table of contents, external images, etc etc.

But note:

- More complicated options depend on export format
- For good integration with \LaTeX , better use *sweave*; for example to use table captions and automated table numbering

RMarkdown

Why use RMarkdown or Sweave?	But...
Updating reports much easier and faster; no need for endless copying-and-pasting	... works best for stand-alone scripts.
Because no copying-and-pasting, lower risk on mistakes	
Convenient if you want to show code and results integrated	... only needed for very specialized journals (such as <i>Journal of Statistical Software</i>)
Consistent output	
Easy to learn	
	... sometimes need for plain .R file, beside the .Rmd: Ending up with two files doing the same thing 😞
Might help making research more reproducible	... is not a guarantee 😊

RMarkdown

“... sometimes need for plain .R file, beside the .Rmd”

Solution 1) Extract .R file from .Rmd

```
1 library(knitr)
2 purl('RMarkdown_example_1.Rmd')
3
```

2) Run .Rmd in global environment

```
1 library(knitr)
2 knit('RMarkdown_example_1.Rmd')
3
```

3) Create R file with the RMarkdown code behind special ('roxygen') comments, #' and #+

<pre>9 10 #' This was an **RMarkdown** document 11 #' 12 #' Now it is a **.R** document with 13 14 #' ## Markdown with R code 15 #+ chunk_name1 16 17 d <- data.frame(participants=1:10, 18 summary(d)</pre>	<pre>14 ## Create report from .R document with RMarkdown code 15 ## behind 'roxygen' comments 16 library(rmarkdown) 17 render(input = "RMarkdown_example_2.R", 18 output_format = "html_document", # other options: http://rmarkdown.rstudio.com/ 19 clean = TRUE # removes temporarily files 20) 21 22 browseURL('RMarkdown_example_2.html') 23</pre>
---	---

Anything additional to discuss?



Le Penseur, Auguste Rodin, picture by Tammy Lo, licensed under the terms of the cc-by-2.0