Reproducible Research

Niklaus Zemp
22.01.2020

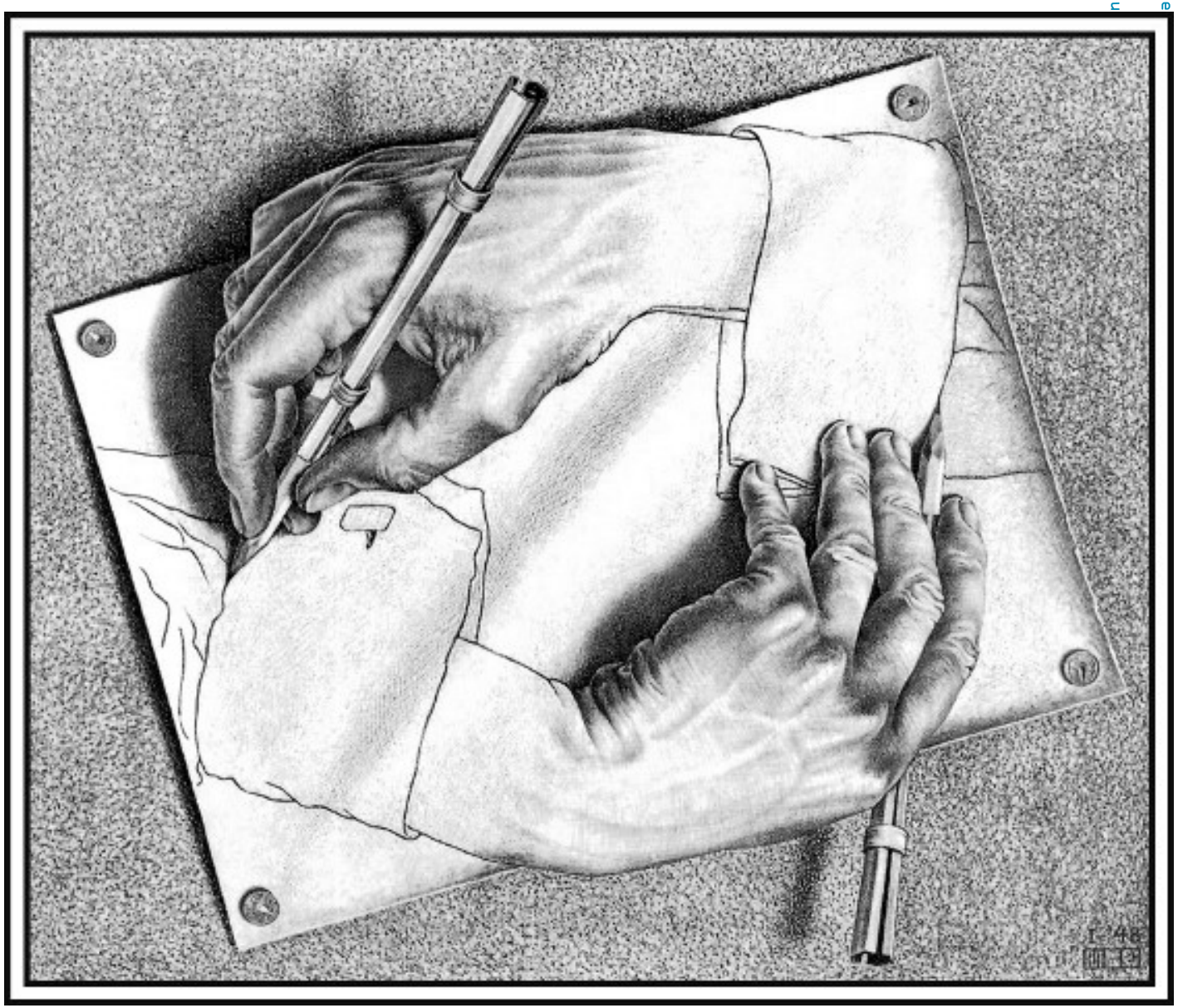
Genetic Diversity Centre (GDC)
Bioinformatics
ETH Zurich

Agenda

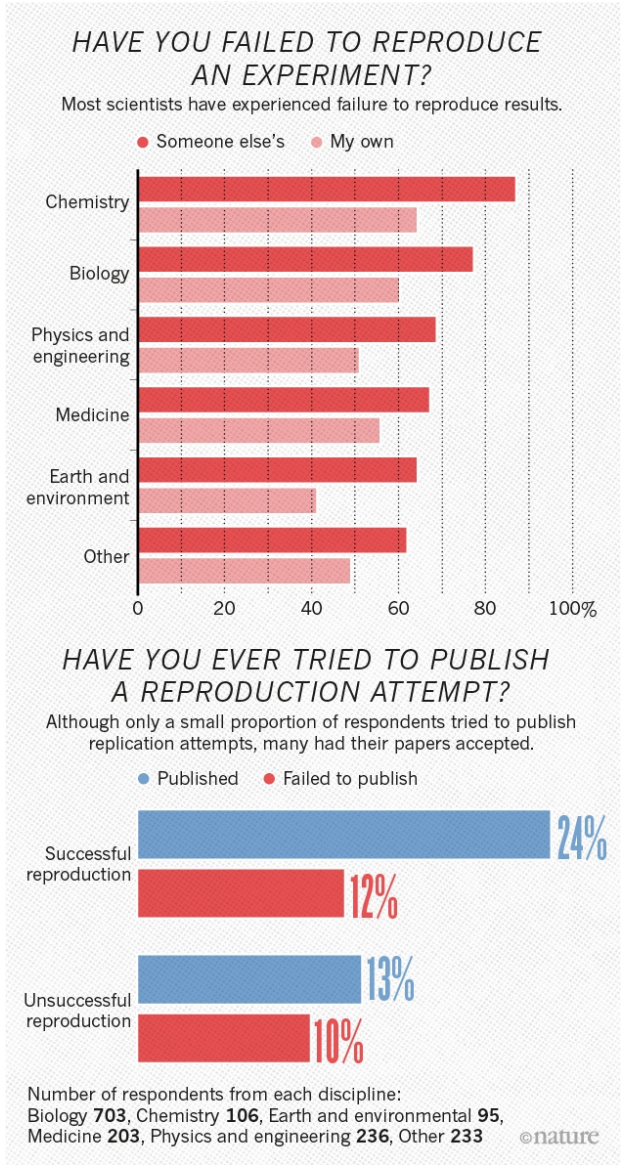
Input (13.30-14.00)

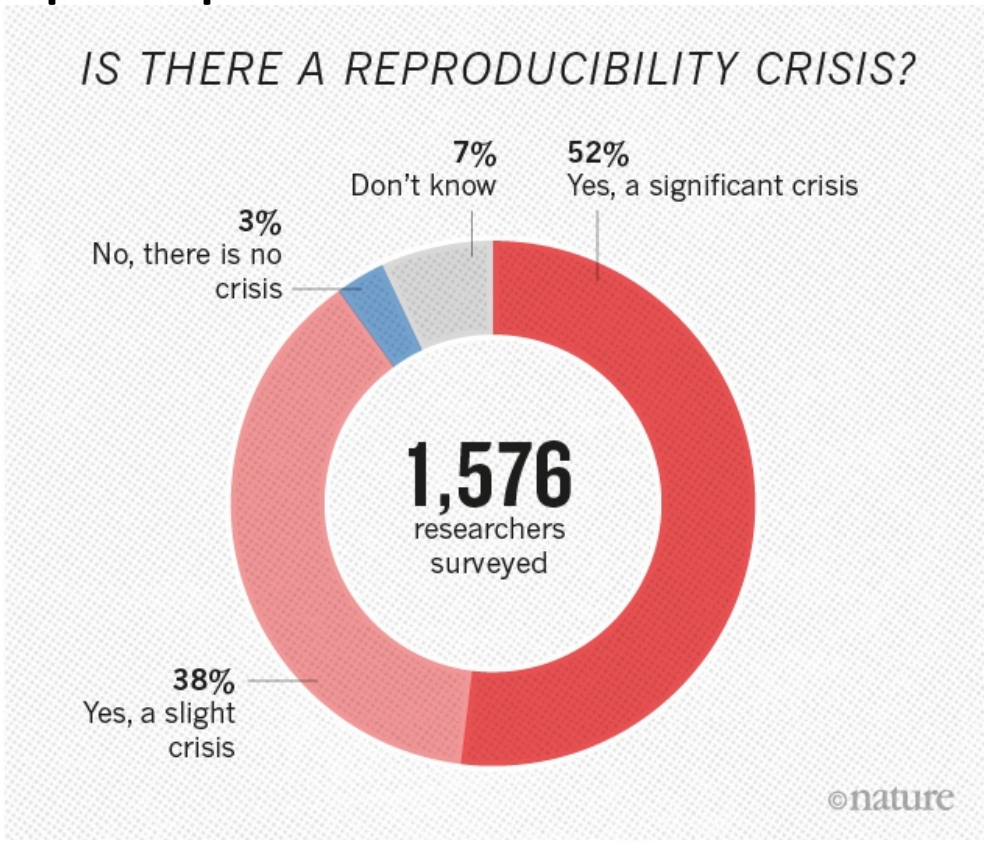
Exercise (14.00-14.45)

Discussion (14.45-15.00)



Scientific recipe-publications



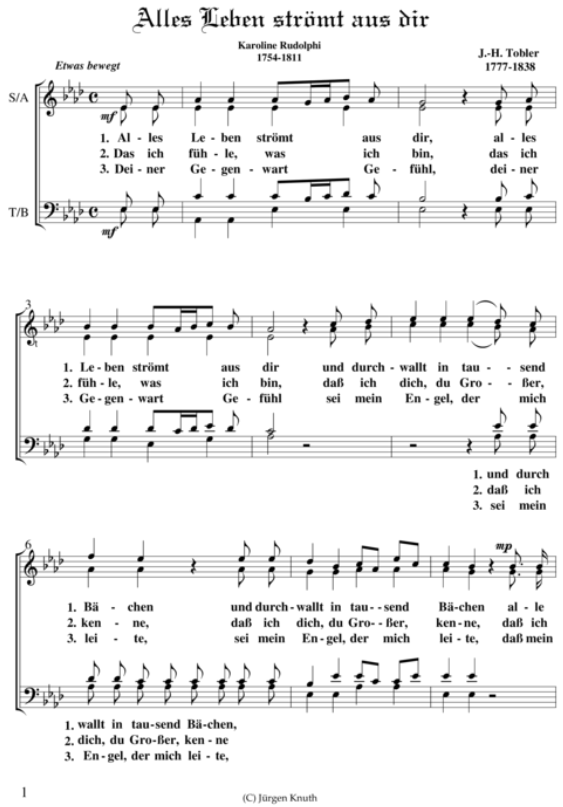


Reality check on reproducibility

A survey of *Nature* readers revealed a high level of concern about the problem of irreproducible results. Researchers, funders and journals need to work together to make research more reliable.

25 May 2016

Recipe



Ingredients

3 fresh red chillies

2 onions

4 cloves of garlic

4 large plum tomatoes

1 bunch of fresh coriander

4 large free-range chicken legs, skin on

olive oil

2 teaspoons garam masala

1 tablespoon crumbled dried curry leaves

1 tablespoon mustard seeds

2 tablespoons white wine vinegar

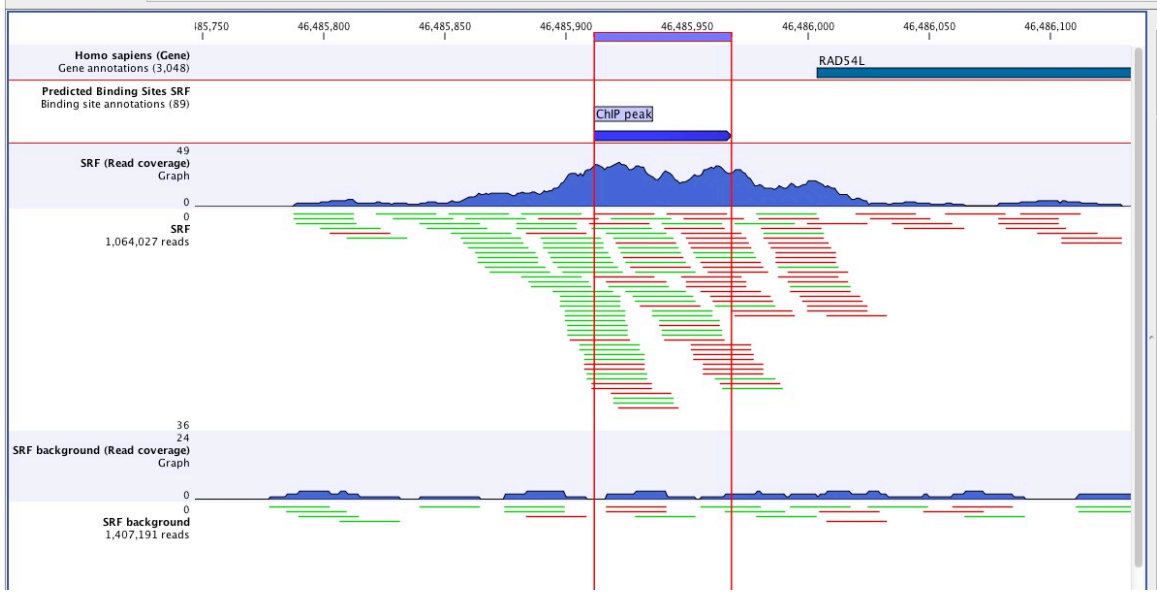
fat-free natural yoghurt

Method

1. Halve the chillies (deseed if you like), peel and finely slice the onions, then peel and crush the garlic. Quarter the plum tomatoes, and pick the coriander leaves.
2. Rub the chicken legs all over with a drizzle of oil and the garam masala, then transfer to a large non-stick ovenproof pan.
3. Add another drizzle of oil and fry the chicken over a medium heat for 10 to 15 minutes, or until lovely and dark golden all over. Be brave and let it get really nice and dark – it will make such a difference to the end result if you get it right at this stage. Drain off any excess fat.
4. At this point, preheat the oven to 180ºC/350ºF/gas 4.
5. Next, add the curry leaves, mustard seeds, chillies, onion and garlic to the chicken. Cook, stirring often, for 5 minutes, then add the tomatoes and white wine vinegar.

   Transfer the pan to the oven. Cook, uncovered, for 50 minutes, or until the chicken is cooked through and falling off the bone.
6. Pop the pan on the hob and reduce the liquid until sticky. Scatter with the coriander leaves and serve with the cooling yoghurt. Delicious with rice or couscous, and a crisp, refreshing

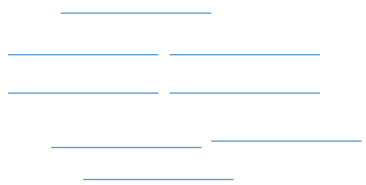
GUI tools

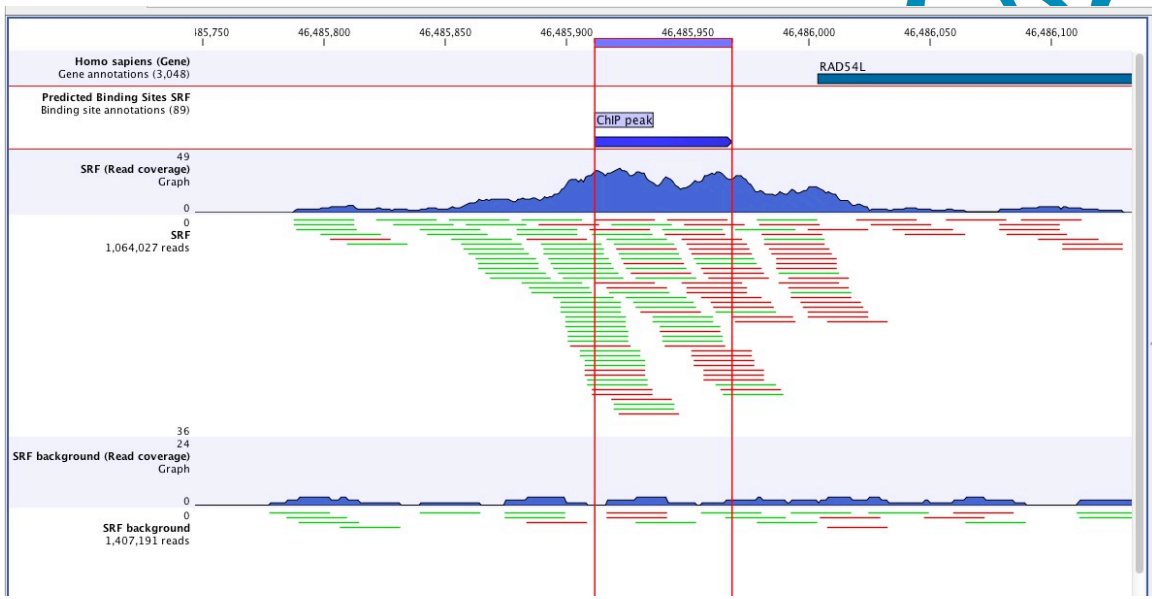

185,750
46,485,800
46,485,850
46,485,900
46,485,950
46,486,000
46,486,050
46,486,100
Homo sapiens (Gene)
Gene annotations (3,048)
RAD54L
Predicted Binding Sites SRF
Binding site annotations (89)
ChIP peak
49
SRF (Read coverage)
Graph
0
0
SRF
1,064,027 reads
36
24
SRF background (Read coverage)
Graph
0
0
SRF background
1,407,191 reads


CLC Genomics Workbench
GET INTO PC
CLC bio


geneious

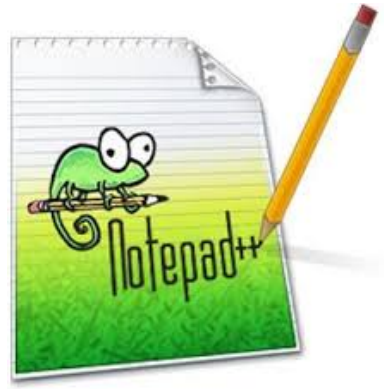
Command line Tools






```
bwa mem Ref Read_R1 Read_R2 > alignment.sam
```

Scientific recipe



Genetic Diversity Centre Zurich

Log-file

```
################################################
Nik Zemp, niklaus.zemp@env.ethz.ch, GDC, ETH Zurich
################################################
#####ddRAD log file, Ivo Widmer, p432
################################################

#######################################################
####Download data
#######################################################
module load eth_proxy
bsub -n1 -W 4:00 -R "rusage[mem=1000]" \
"wget -r --no-parent --reject="index.htm*" http://gc3fstorage.u
bsub -n1 -W 4:00 -R "rusage[mem=1000]" \
"wget -r --no-parent --reject="index.htm*" http://gc3fstorage.u
bsub -n1 -W 4:00 -R "rusage[mem=1000]" \
"wget -r --no-parent --reject="index.htm*" http://gc3fstorage.u
bsub -n1 -W 4:00 -R "rusage[mem=1000]" \
```

script

```
#!/bin/bash
#BSUB -J "processradtags"
#BSUB -R "rusage[mem=10000]"
#BSUB -n 1
#BSUB -W 24:00


module load gcc/4.9.2 gdc perl/5.18.4
export PATH=$PATH:/cluster/project/gdc/shared/tools/stacks-1.48
#module load gcc/4.8.2 gdc perl/5.18.4 stacks/1.40
source /cluster/apps/gdc/perl5/etc/bashrc



mkdir samples

process_radtags -i gzfastq -f /cluster/project/gdc/people/buckleyj/Hiseq_
```

Scientific recipe

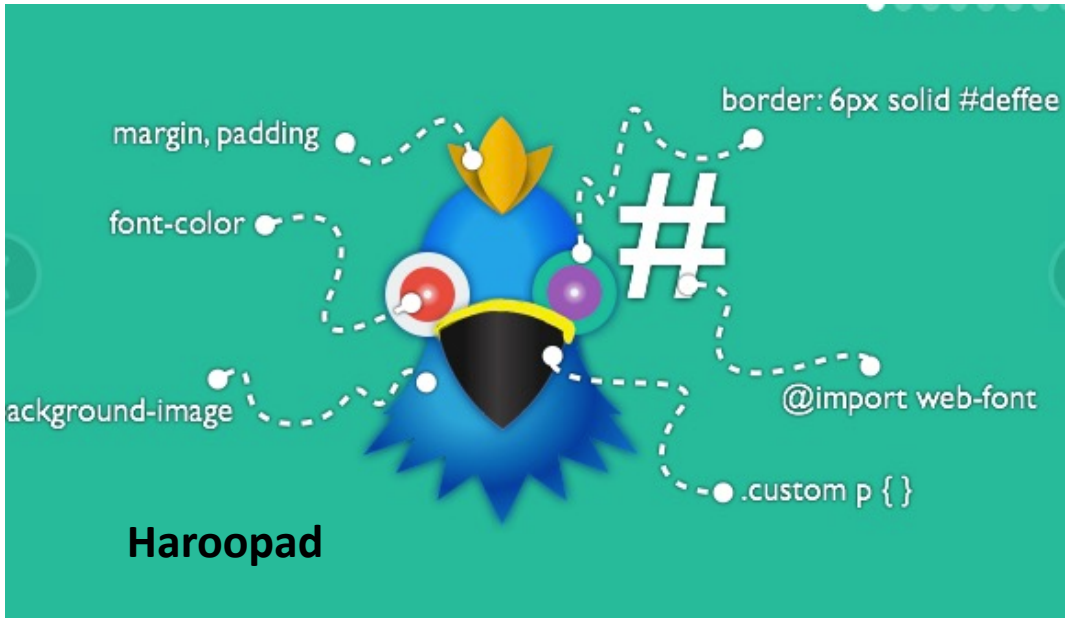
- (original) author or source
- your name
- date
- version of the tool
- version of the script
- reproducible code with comments (more comments than code)
- use style guides
- syntax coloring

```
#!/bin/bash
#BSUB -J "processradtags"
#BSUB -R "rusage[mem=10000]"
#BSUB -n 1
#BSUB -W 24:00


module load gcc/4.9.2 gdc perl/5.18.4
export PATH=$PATH:/cluster/project/gdc/s
#module load gcc/4.8.2 gdc perl/5.18.4 s
source /cluster/apps/gdc/perl5/etc/bashr



mkdir samples

process_radtags -i gzfastq -f /cluster/p
```

Markdown

Slightly modified based on dDocent Version 2.6.1; overlapping paired-end reads; September 2018; Nik Zemp based on the Tutorial

```Bash
bsub -n 2 -W 4:00 -R "rusage[mem=5000]" -Is bash

module load gcc/4.8.2 gdc python/2.7.11 java/1.8.0_73 perl/5.18.4 freebayes/0.9.20 trimmomatic/0.35 bwa/0.7.12 cd-hit/
export PATH="$PATH:/cluster/project/gdc/shared/tools/pear-0.9.6-bin-64"
export PATH="/cluster/project/gdc/shared/tools/seqtk:$PATH"
```



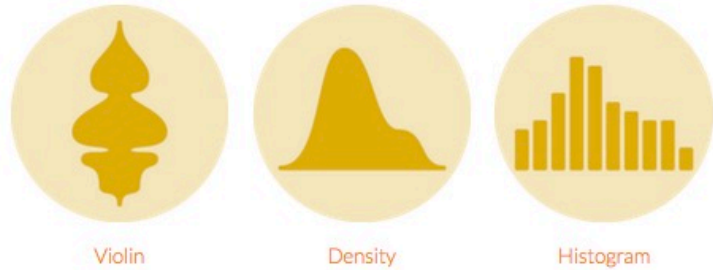
Cheat sheet

Reproducible Research

scripts
logs

scripts
logs

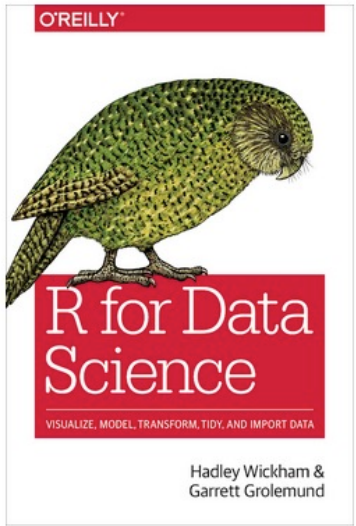
scripts
logs



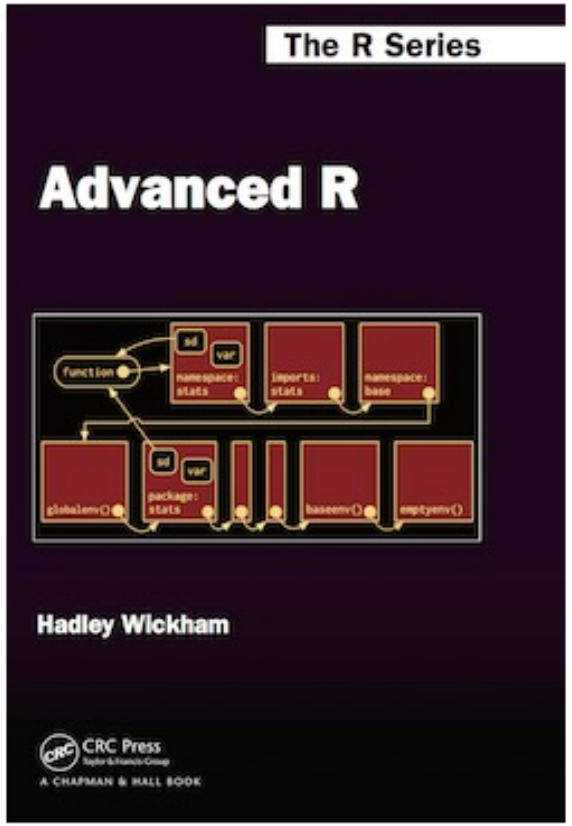
Reproducible Research

Resources

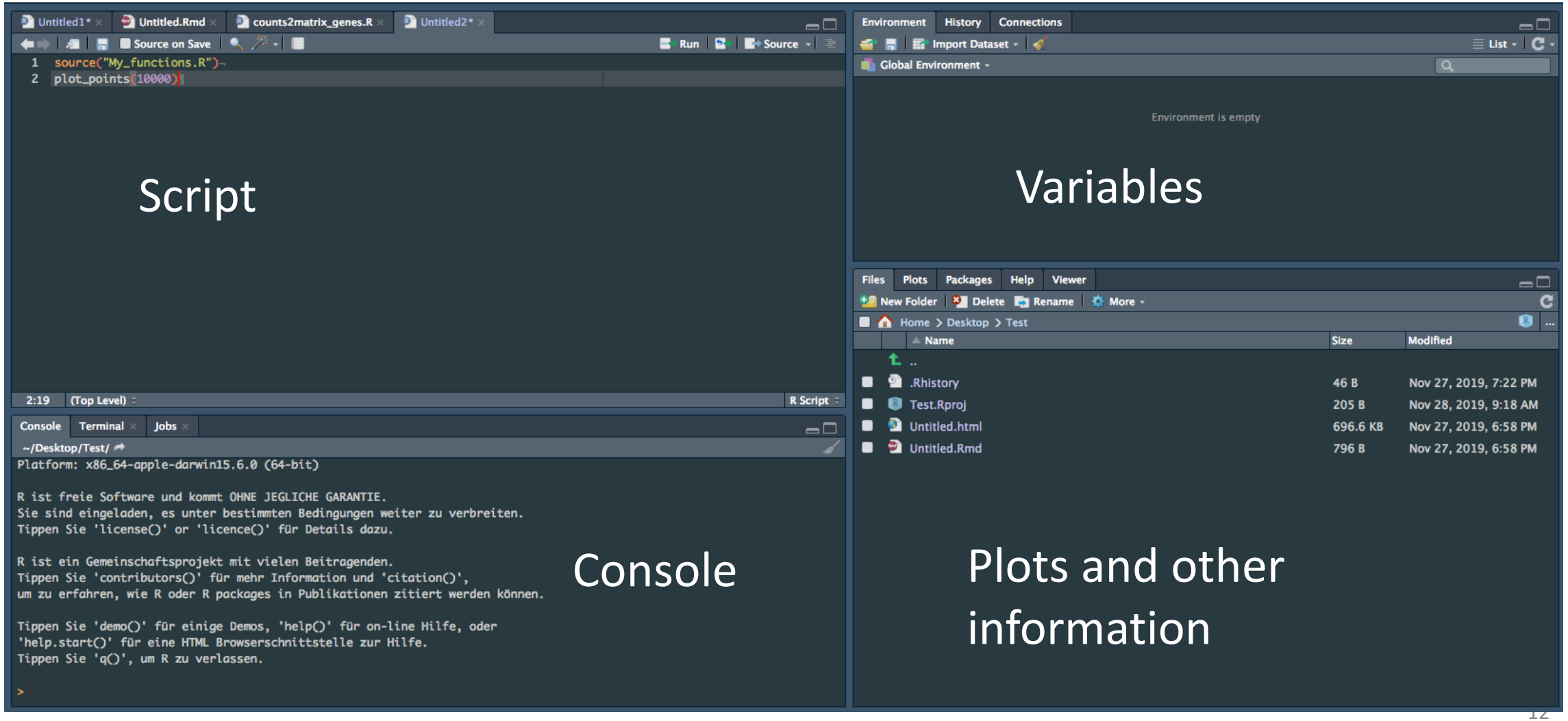
Many many tutorials, forum, YouTube videos posts and books available



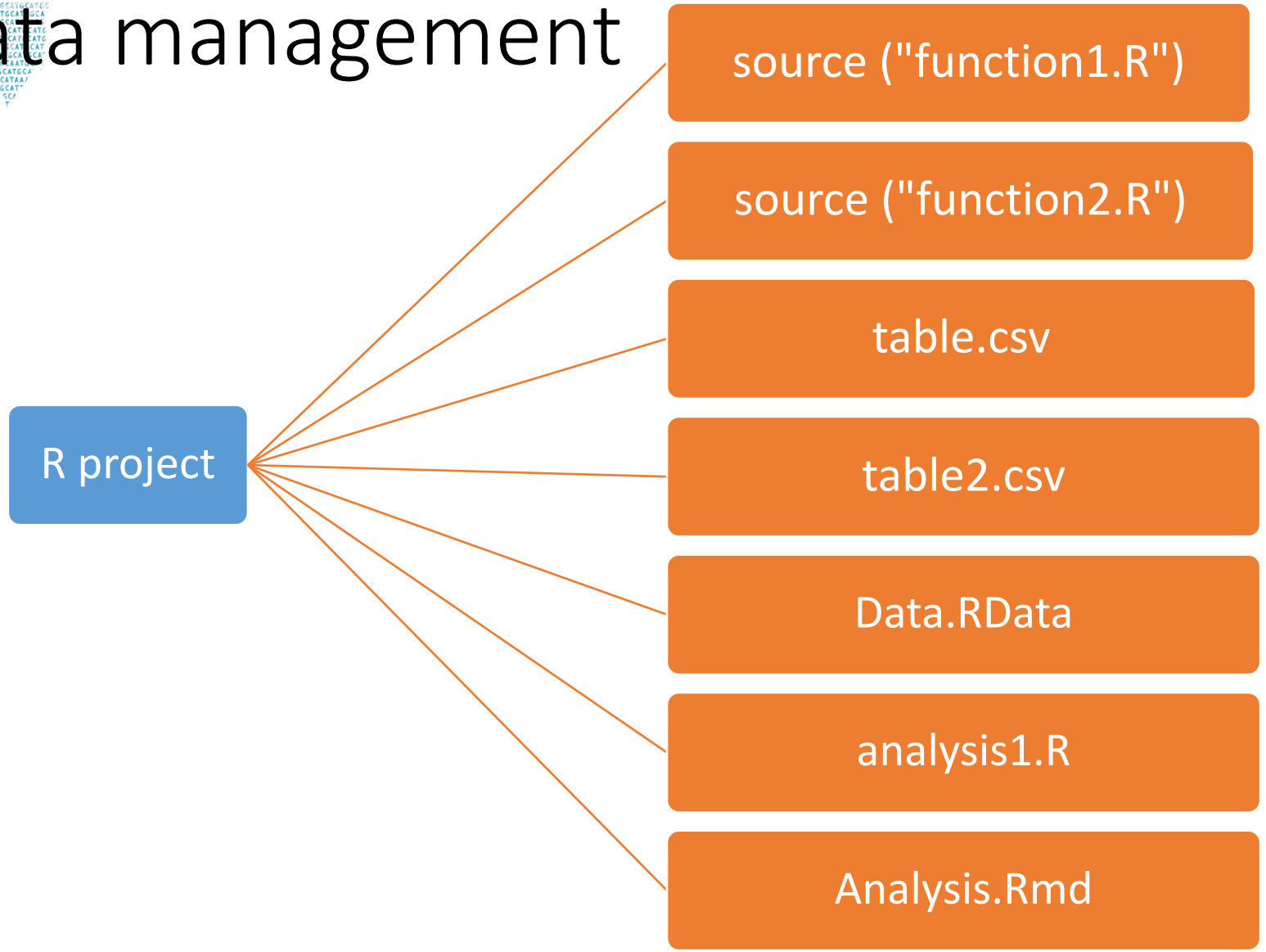
R-bloggers
R news and tutorials contributed by (750) R bloggers



Rstudio


Untitled1*
Untitled.Rmd
counts2matrix_genes.R
Untitled2*
Source on Save
Run
Source
1 source("My_functions.R")
2 plot_points(10000)
Script
2:19
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Variables
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home > Desktop > Test
Name
Size
Modified
.Rhistory 46 B Nov 27, 2019, 7:22 PM
Test.Rproj 205 B Nov 28, 2019, 9:18 AM
Untitled.html 696.6 KB Nov 27, 2019, 6:58 PM
Untitled.Rmd 796 B Nov 27, 2019, 6:58 PM
Plots and other information
Console
Terminal
Jobs
~/Desktop/Test/
Platform: x86_64-apple-darwin15.6.0 (64-bit)
R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.
R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.
>
Console

Data management



Packages

Available Packages

Currently, the CRAN package repository features 13884 available packages.

Table of available packages, sorted by date of publication

Table of available packages, sorted by name

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Install »

- Discover 1649 software packages available in *Bioconductor* release 3.8.

Get started with *Bioconductor*

- Install *Bioconductor*
- Get support
- Latest newsletter
- Follow us on twitter
- Install R

Tidyverse


```
install.packages("tidyverse")
```

Set up an R script

MDA 2020, GDC Nik Zemp, January 2020, Version 0.1

Set working directory

```
setwd("~/Desktop/")
```

Remove all variables

```
rm(list = ls())
```

set seed

```
set.seed(1000)
```

load packages

```
library(tidyverse)
```

Source functions

```
source("My_Functions.R")
```

UseR packages

```
library(tidyverse)

dplyr::filter()
stats::filter()
```

> *@ijlyttle a package is a like a book, a library is like a library; you use library() to check a package out of the library #rsats*
>
> *— Hadley Wickham (@hadleywickham) December 8, 2014*

datawox=jittermap(datawox, amount=1e-6);datauto=jittermap(datauto, amount=1e-6) autosome;data=jittermap(data,amount=1e-6);data2=data;datauto$pheno$gender=as.numeric(datauto$pheno$gender)
datauto$pheno$ffd[68] <- NA;datauto$pheno$ffd=(datauto$pheno$ffd-min(datauto$pheno$ffd,na.rm=T))/(max(datauto$pheno$ffd,na.rm=T)-min(datauto$pheno$ffd,na.rm=T));datauto$pheno$ffd=asin(sqrt(datauto$pheno$ffd));datauto$pheno$ffd_height=log(datauto$pheno$ffd_height)
datauto$pheno$open_flowers_ffd5=sqrt(datauto$pheno$open_flowersffd5);datauto$pheno$open_flowers_rate=log(datauto$pheno$open_flowers_rate)

```
datauto=jittermap(datauto, amount=1e-6) # w/ X chromosome, but treated as
data=jittermap(data,amount=1e-6)
data2=data
datauto$pheno$gender=as.numeric(datauto$phen
datauto$pheno$ffd[68] <- NA
datauto$pheno$ffd=(datauto$pheno$ffd-min(dat
datauto$pheno$ffd=asin(sqrt(datauto$pheno$ff
datauto$pheno$ffd_height=log(datauto$pheno$f
datauto$pheno$open_flowers_ffd5=sqrt(datauto
datauto$pheno$open_flowers_rate=log(datauto$
```

```
# w/o X chromosome
datawox <- jittermap(datawox, amount = 1e-6)


# w/ X chromosome, but treated as an autosome
datauto <- jittermap(datauto, amount = 1e-6)
data <- jittermap(data, amount = 1e-6)
data2 <- data


# convert f to 0, m to 1
datauto$pheno$gender <- as.numeric(datauto$pheno$gender) - 1 # convert f to 0, m to 1
datauto$pheno$ffd[68] <- NA
datauto$pheno$ffd <- (datauto$pheno$ffd - min(datauto$pheno$ffd, na.rm = T)) / (max(datauto$pheno$ffd, na.
datauto$pheno$ffd <- asin(sqrt(datauto$pheno$ffd))
datauto$pheno$ffd_height <- log(datauto$pheno$ffd_height)

datauto$pheno$open_flowers_ffd5 <- sqrt(datauto$pheno$open_flowers_ffd5)
datauto$pheno$open_flowers_rate <- log(datauto$pheno$open_flowers_rate)
```

R with style

The tidyverse style guide

Hadley Wickham

https://style.tidyverse.org/index.html

Google R-style

https://google.github.io/styleguide/Rguide.xml

Styler

styler

The goal of styler is to provide non-invasive pretty-printing of R source code while adhering to the tidyverse formatting rules. styler can be customized to format code according to other style guides too.

Installation

You can install the package from CRAN:

```
install.packages("styler")
```

Some examples

```
# Good
day_one
day_1

# Bad
DayOne
dayone
```

```
# Good
x <- 5

# Bad
x = 5
```

```
# Good
if (y < 0) {
  stop("Y is negative")
}
```

```
# Bad
if (y < 0) stop("Y is negative")
```

```
# Good
"Text"
'Text with "quotes"'
'<a href="http://style.tidyverse.org">A link</a>'

# Bad
'Text'
'Text with "double" and \'single\' quotes'
```

rmarkdown

https://github.com/rstudio/cheatsheets/raw/master/rmarkdown-2.0.pdf

Keep data

save specific objects to a file

```
save(iris,file="iris.RData")
```

load object

```
load("iris.RData")
```

save workspace

```
save.image(file='image.RData')
```

load workspace

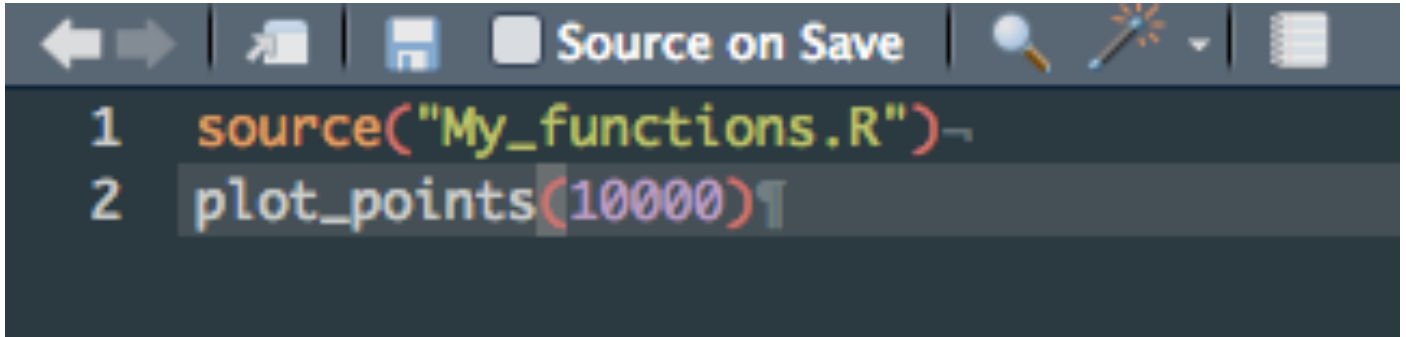
```
load("image.RData")
```

Customized functions

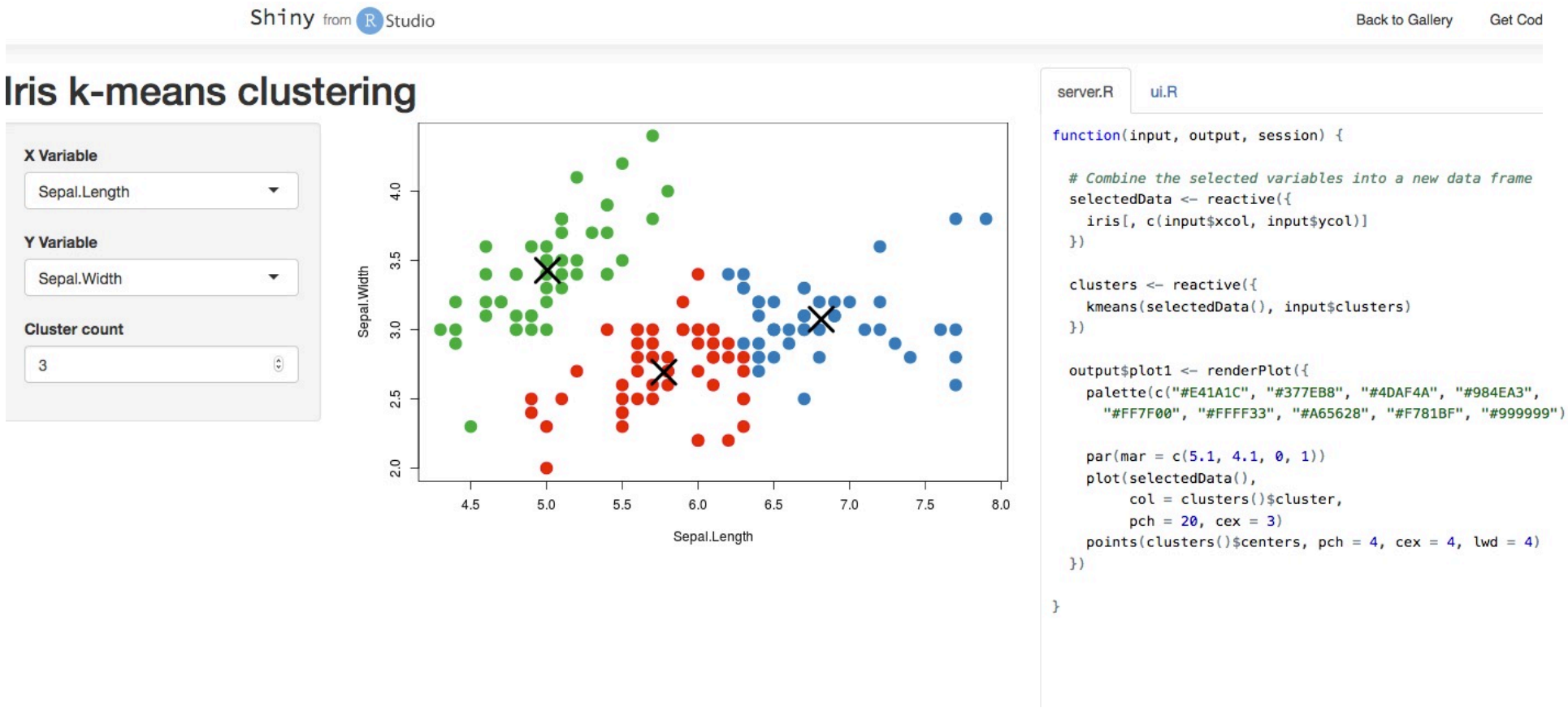
```
Myfunction <- function(variables) {
Functions
}
```

```
plot_points <- function(n_points) {
dat <- rnorm(n_points, 100, 5)
plot(dat)
}
```

My_functions.R



Shiny Apps



https://shiny.rstudio.com/gallery/kmeans-example.html

Workflows on a HPC cluster

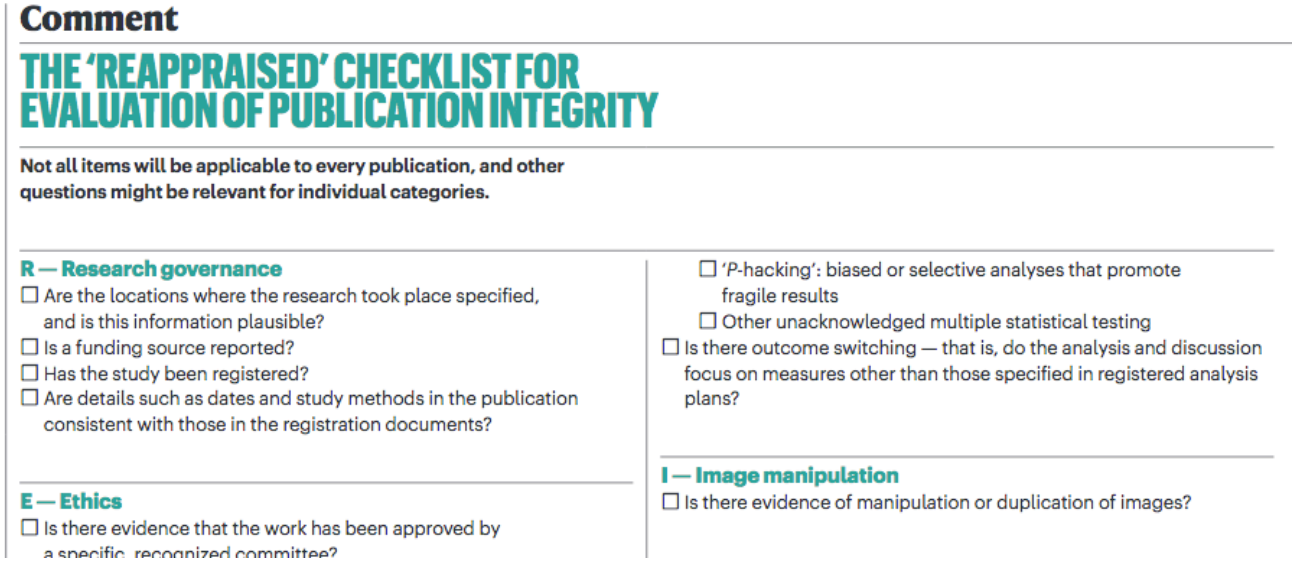
Read.R

```
#!/usr/bin/env Rscript
args <-   commandArgs(trailingOnly=TRUE)

##read table
samples <- read.table(args[1], header = F)
```

```
nik$ Rscript --vanilla Read.R table.csv
```

Publication

- keep all scripts
- deposit raw data to public database
- use default settings or mention if not
- provide version information
- provide commands in supplemer
- deposit scripts on github/gitlab
- Check-list



Comment

THE 'REAPPRAISED' CHECKLIST FOR EVALUATION OF PUBLICATION INTEGRITY

Not all items will be applicable to every publication, and other questions might be relevant for individual categories.

R — Research governance

- ☐ Are the locations where the research took place specified, and is this information plausible?
- ☐ Is a funding source reported?
- ☐ Has the study been registered?
- ☐ Are details such as dates and study methods in the publication consistent with those in the registration documents?

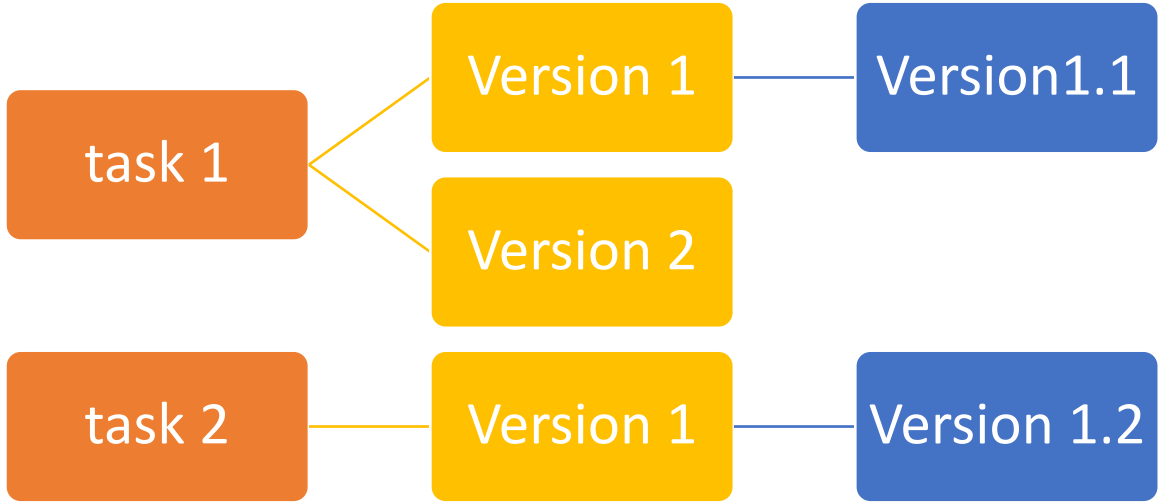
E — Ethics

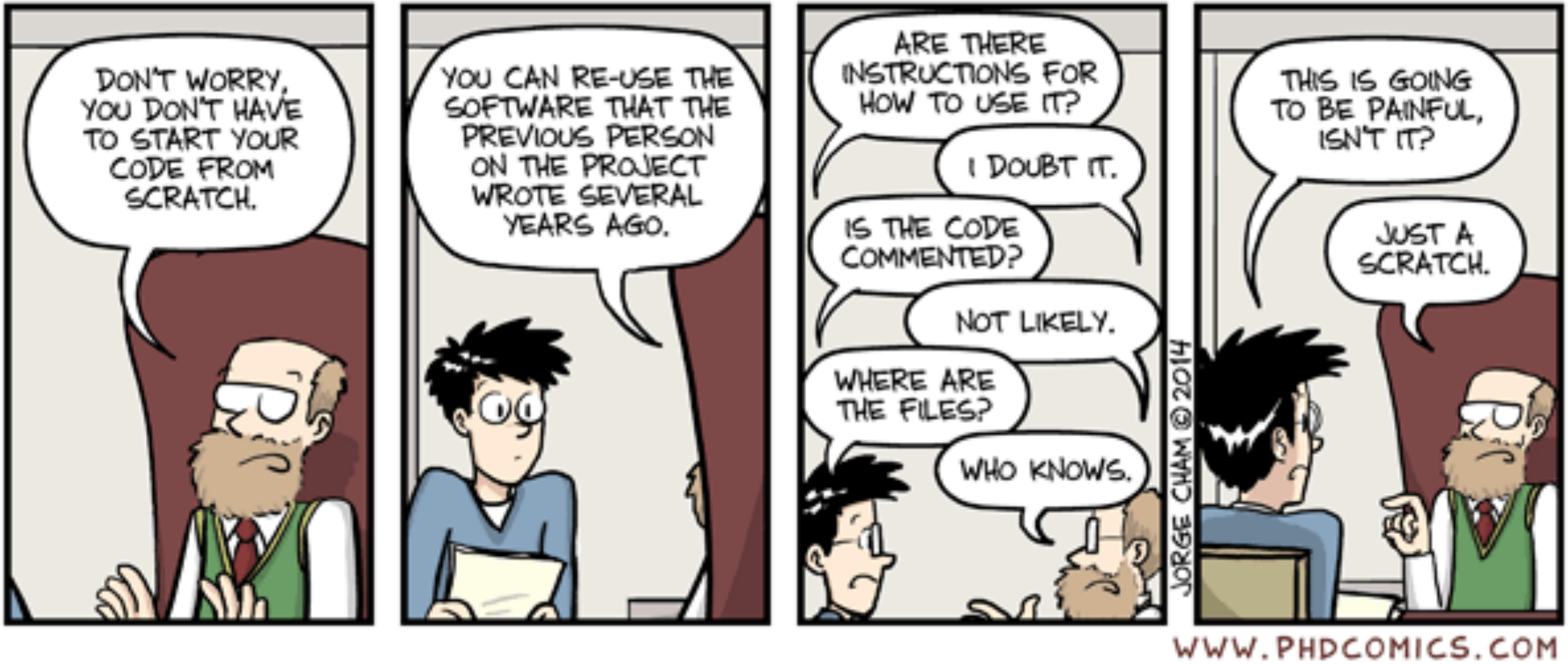
- ☐ Is there evidence that the work has been approved by a specific, recognized committee?
- ☐ 'P-hacking': biased or selective analyses that promote fragile results
- ☐ Other unacknowledged multiple statistical testing
- ☐ Is there outcome switching — that is, do the analysis and discussion focus on measures other than those specified in registered analysis plans?

I — Image manipulation

- ☐ Is there evidence of manipulation or duplication of images?

Exercise


task 1
Version 1
Version1.1
Version 2
task 2
Version 1
Version 1.2


DON'T WORRY, YOU DON'T HAVE TO START YOUR CODE FROM SCRATCH.
YOU CAN RE-USE THE SOFTWARE THAT THE PREVIOUS PERSON ON THE PROJECT WROTE SEVERAL YEARS AGO.
ARE THERE INSTRUCTIONS FOR HOW TO USE IT?
I DOUBT IT.
IS THE CODE COMMENTED?
NOT LIKELY.
WHERE ARE THE FILES?
WHO KNOWS.
THIS IS GOING TO BE PAINFUL, ISN'T IT?
JUST A SCRATCH.
JORGE CHAM © 2014
WWW.PHDCOMICS.COM

Take home message

- Do reproducible research
- Rstudio has a powerful functions implemented