Reproducible research

current challenges and future prospects

Rich FitzJohn

@phylorich

R can be irreproducible

R can be irreproducible

```
setwd(”myproject/final2/works”)
```

R can be irreproducible

Graphs that need manual tweaking

R can be irreproducible

Manually edit your input

R can be irreproducible

Undocumented dependencies

R can be reproducible

Don't do those things

R can be reproducible

Reproducibility depends on tools & workflows **around** R

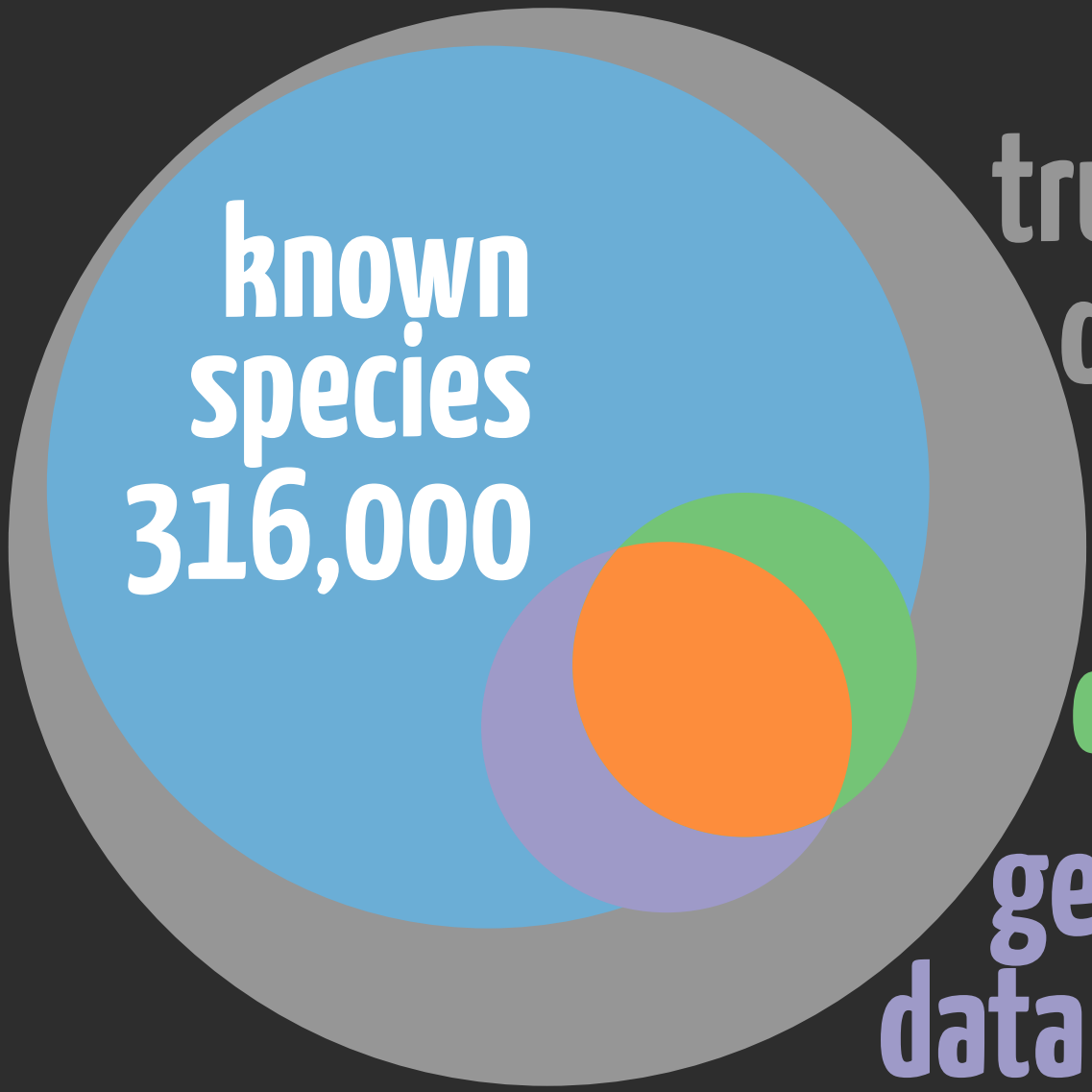
A simple case of reproducible research

- Open data
- No experiments
- No confidentiality
- Straightforward analysis

QLD grassland by Willem van Aken

QLD rainforest by Willem van Aken

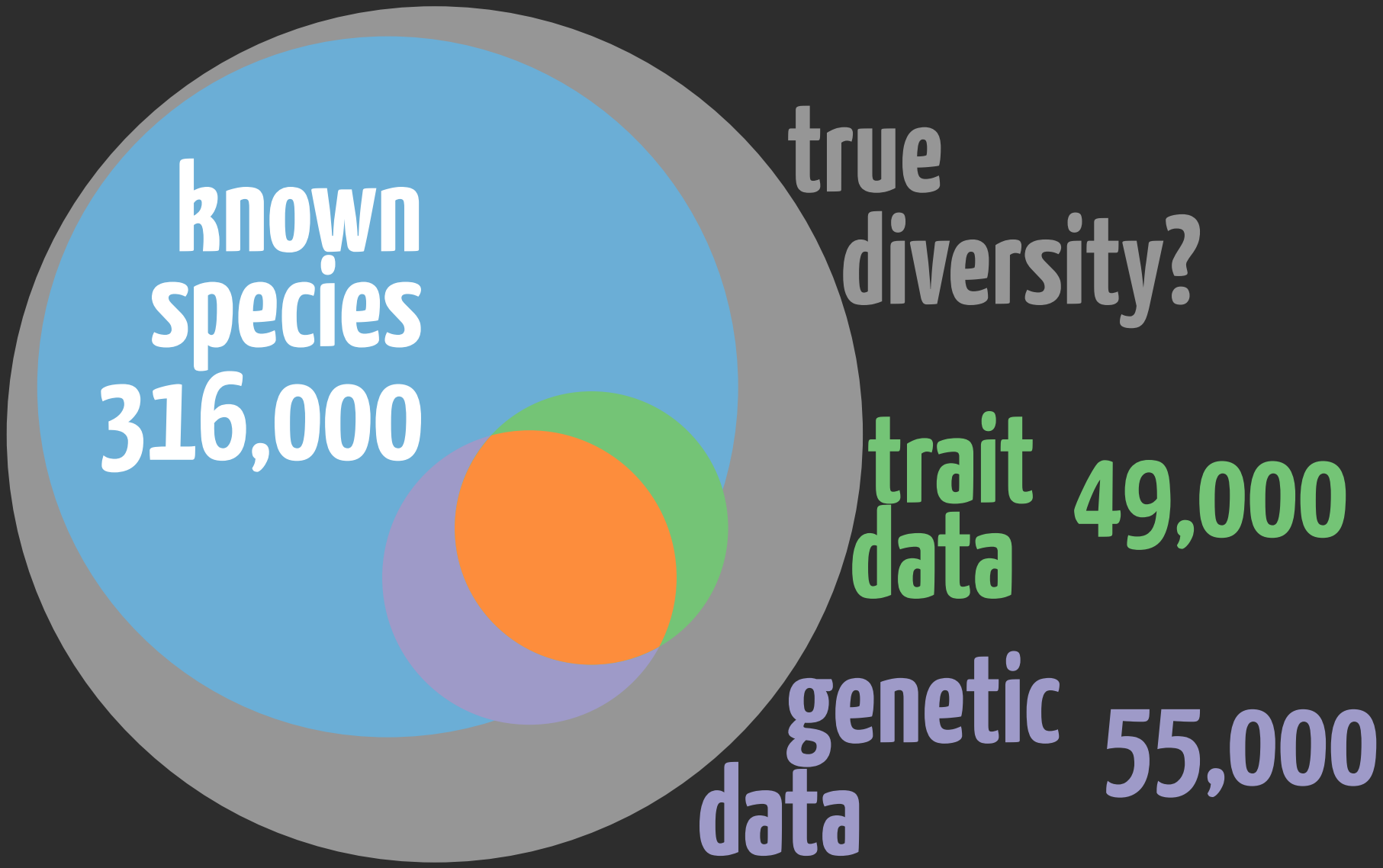

known
species
316,000
true
diversity



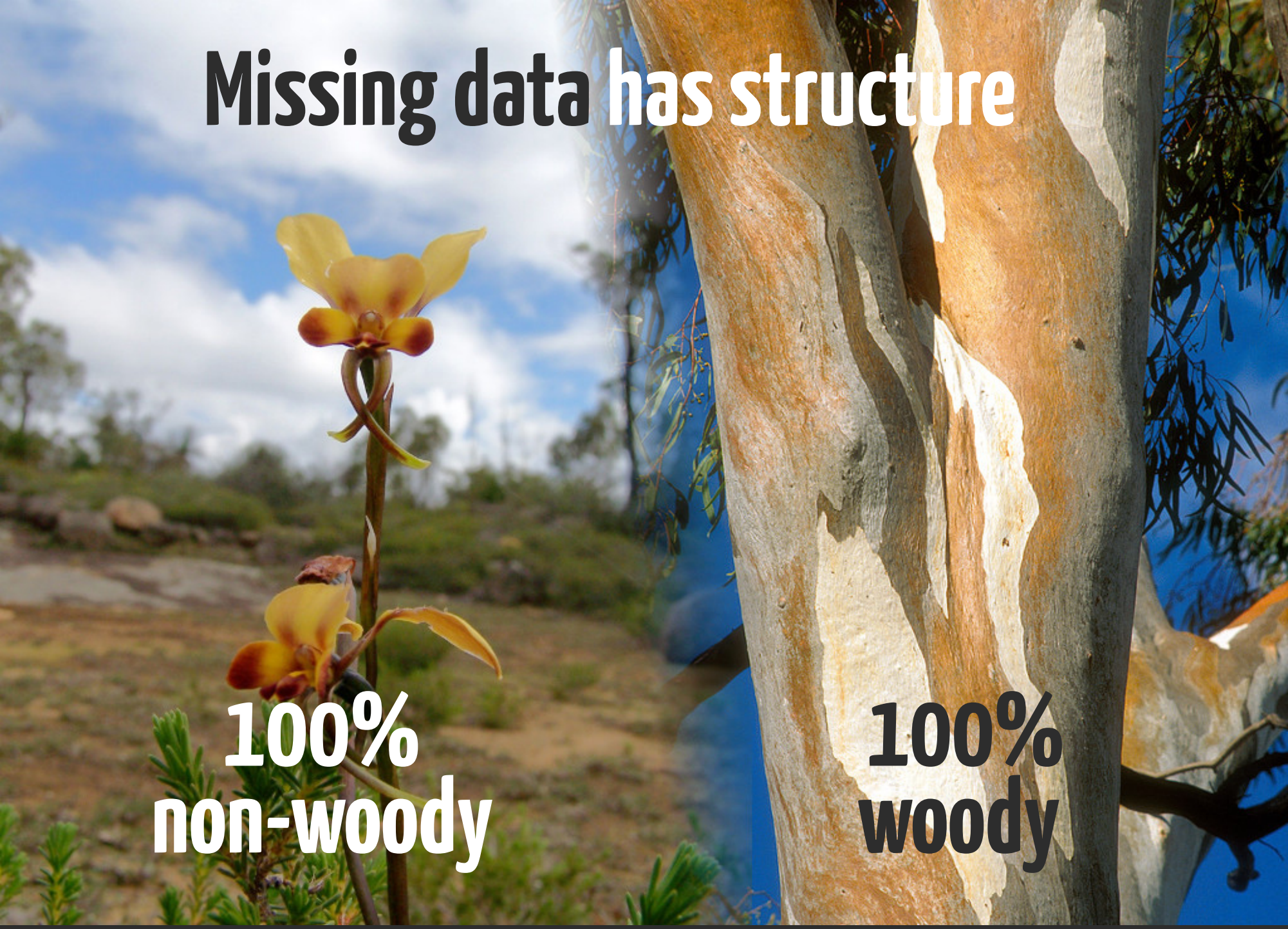
true
diversity

trait
data

genetic
data


known species 316,000
true diversity?
trait data
49,000
genetic data
55,000

Missing data has structure



Donkey orchid by Gnangarra

Red river gum by Willem van Aken

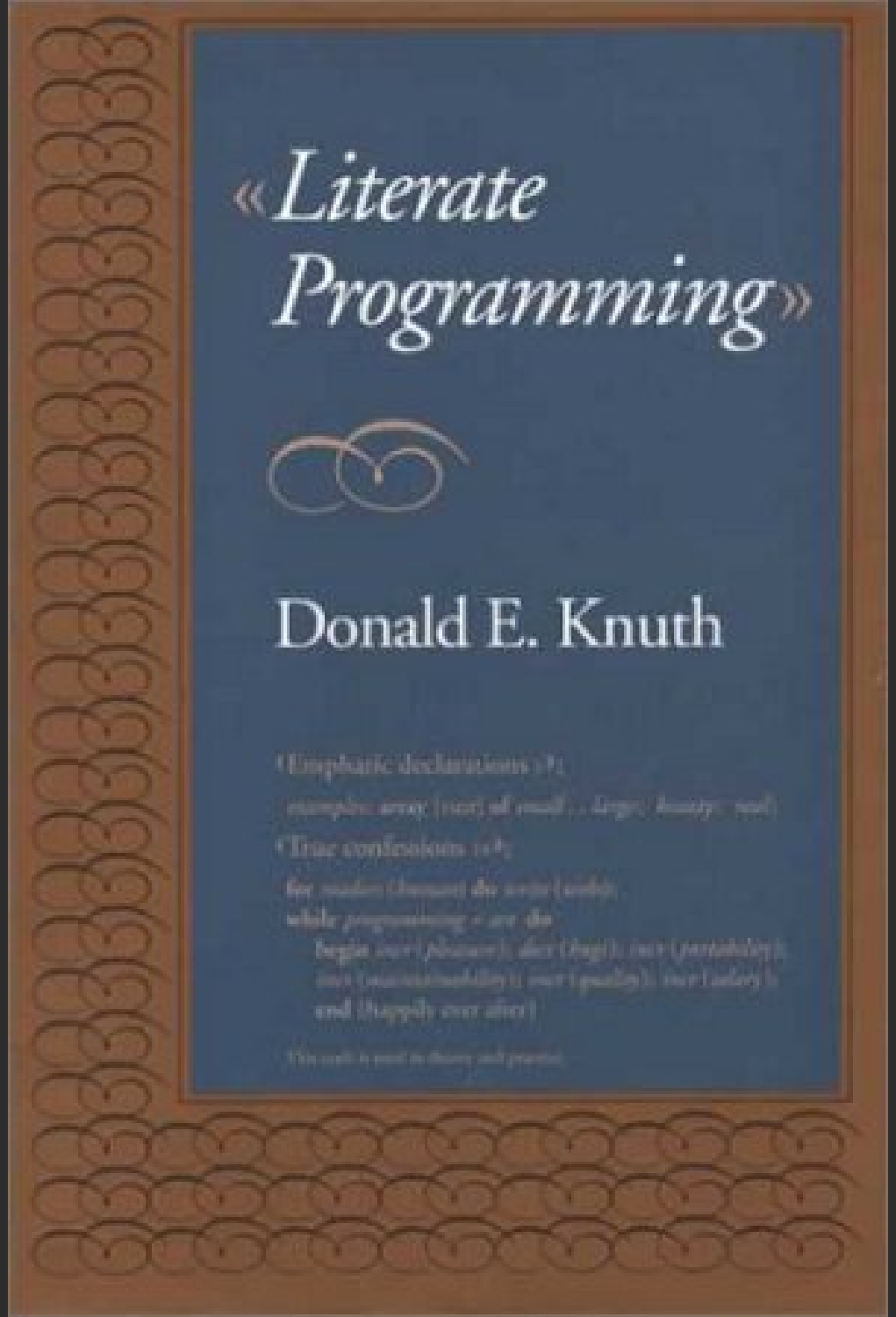
Tools we used

- knitr: what are we trying to make work?
- git: I swear it used to work
- make: it takes a while to make it work
- travis-CI: will it work elsewhere?
- packrat: will it work later?

Literate programming
knitr

What are we trying to make work?

yihui.name/knitr



Knuth, 1984

CWEB →

Sweave →

knitr

Mix documentation & code

````
# Markdown heading
Treated as text
```{r}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

Run through knitr to run code

````
# Markdown heading
Treated as text
```r
x <- sample(10)
y <- sample(10)
cor(x, y)
# [1] 0.03030303
```
````

Render markdown to HTML

```
<h1>Markdown heading</h1>
<p>Treated as text</p>
<pre>
x <- sample(10)
y <- sample(10)
cor(x, y)
# [1] 0.03030303
</pre>
```

… or to LaTeX

```
\section{Markdown heading}
Treated as text
\begin{verbatim}
x <- sample(10)
y <- sample(10)
cor(x, y)
# [1] 0.03030303
\end{verbatim}
```

That's basically all there is to it

````
# Markdown heading
Treated as text
```{r}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

````
# Markdown heading
Treated as text
```r
x <- sample(10)
y <- sample(10)
cor(x, y)
# [1] 0.03030303
```
````

```
<h1>Markdown heading</h1>
<p>Treated as text</p>
<pre>
x <- sample(10)
y <- sample(10)
cor(x, y)
# [1] 0.03030303
</pre>
```

Graphics handled automatically

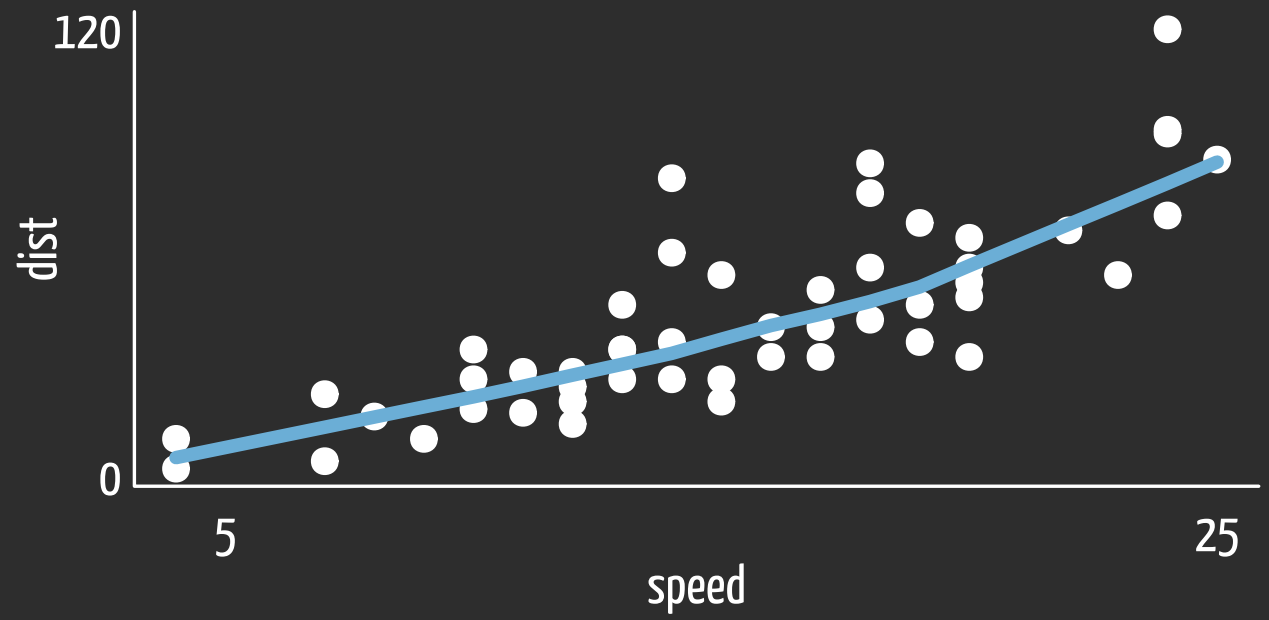
````
Here is the input data:
```{r}
plot(cars)
lines(lowess(cars), col="blue")
```
````

Graphics handled automatically

````
Here is the input data:
```r
plot(cars)
lines(lowess(cars), col="blue")
```
![plot title](figure/unnamed-chunk-2.png)
````

Graphics handled automatically

```
Here is the input data:
plot(cars)
lines(lowess(cars), col=”blue”)
```



Cache long-running computation

```
```{r, cache=TRUE}
fit <- mcmc(data)
for (x in fit[[1]]) {
  for (y in fit[[2]]) {
    for (z in fit[[3]]) {
    ...
    }
  }
}
```
```

Control what is displayed

````
# Markdown heading
Treated as text
```{r, echo=FALSE}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

Control what is displayed

````
# Markdown heading
Treated as text
```{r, echo=FALSE}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

````
# Markdown heading
Treated as text
```r
# [1] 0.03030303
```
````

Control what is displayed

````
# Markdown heading
Treated as text
```{r, results="hide"}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

Control what is displayed

````
# Markdown heading
Treated as text
```{r, results="hide"}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

````
# Markdown heading
Treated as text
```r
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

Control what is displayed

```
# Markdown heading
Treated as text
```{r, echo=FALSE,
      results="hide"}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
```

Control what is displayed

````
# Markdown heading
Treated as text
```{r, echo=FALSE,
      results="hide"}
x <- sample(10)
y <- sample(10)
cor(x, y)
```
````

```
# Markdown heading
Treated as text
```

Literate programming
knitr

Why doesn't everyone use this all the time?

How to draw an Owl.

"A fun and creative guide for beginners"

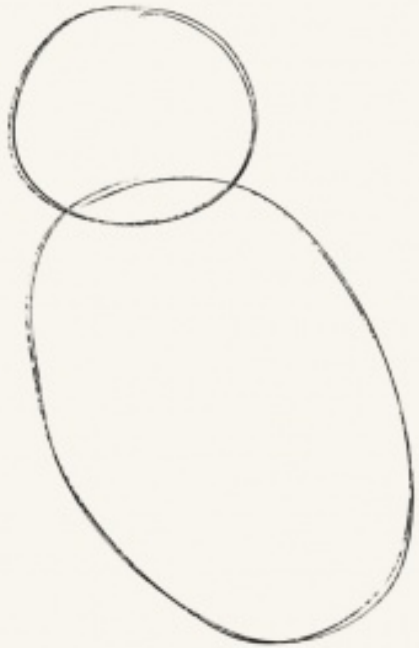


Fig 1. Draw two circles

Fig 2. Draw the rest of the damn Owl

Barriers to knitr

Barriers to knitr

Encourages overuse of global variables

Barriers to knitr

Re-running analyses because of changed punctuation gets annoying

Barriers to knitr

Requires really good editor support

Prospects for knitr

Amazing for supporting materials,
manuals, technical documentation

Examples: github.com/richfitz/reproducibility-2014/wiki

Prospects for knitr

Generate knitr files from plain R source:

`knitr::spin`

sowsear: github.com/richfitz/sowsear

Prospects for knitr

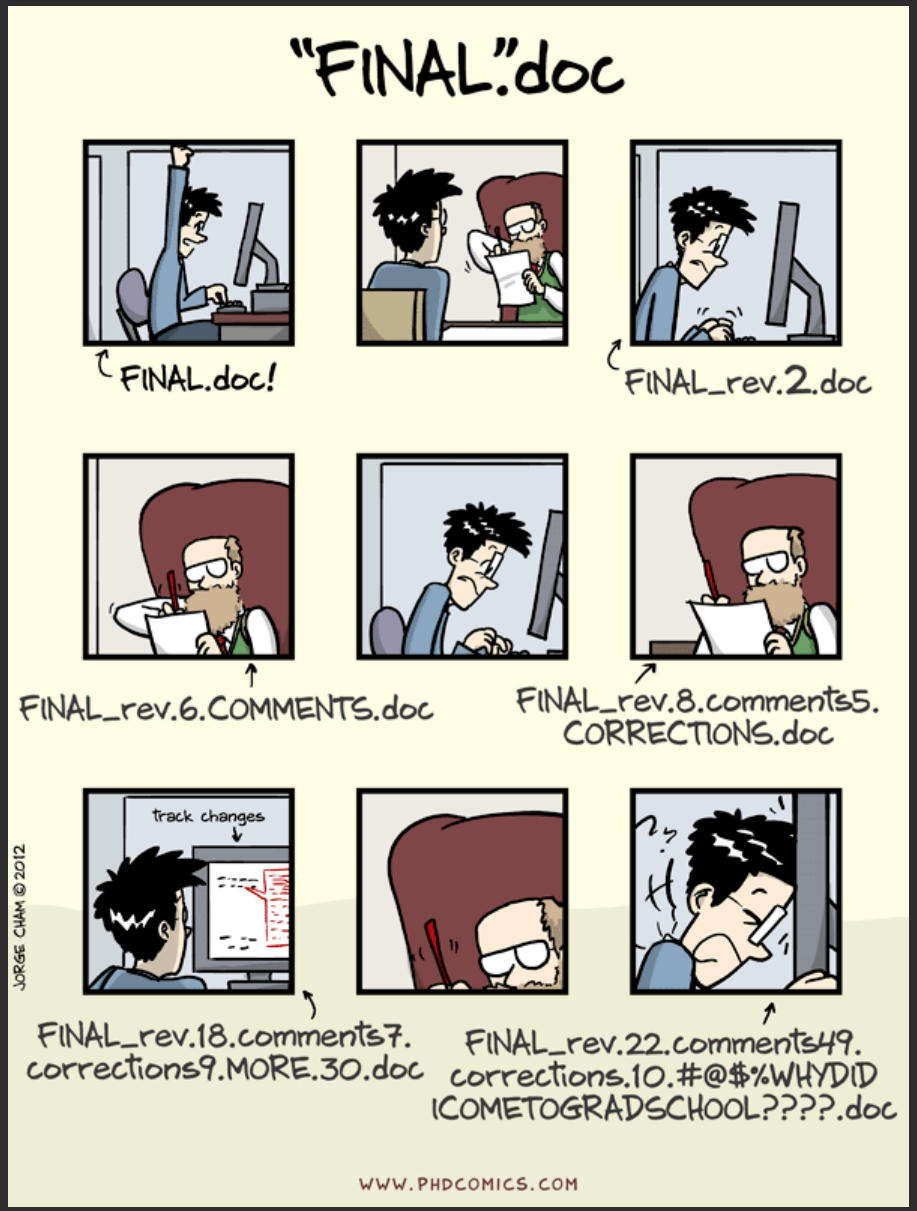
The principle holds elsewhere:
Output should be regeneratable from input

Version control
git

I swear it used to work

git-scm.com

See also
mercurial



PhD comics: phdcomics.com/comics/archive.php?comicid=1531

Store metadata

Version 1

Who
What
When
Why

... for every version

Who
What
When
Why

Who
What
When
Why

Who
What
When
Why

git add; git commit

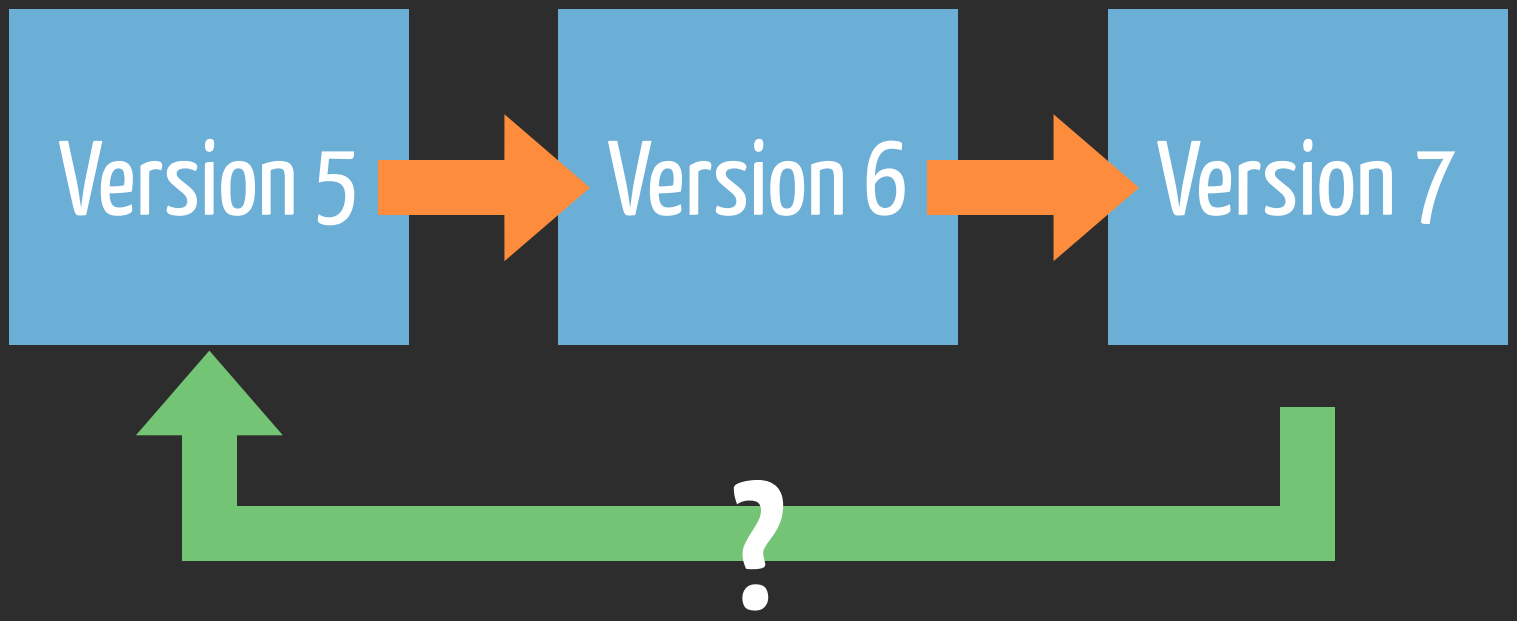
Who
What
When
Why

Who
What
When
Why

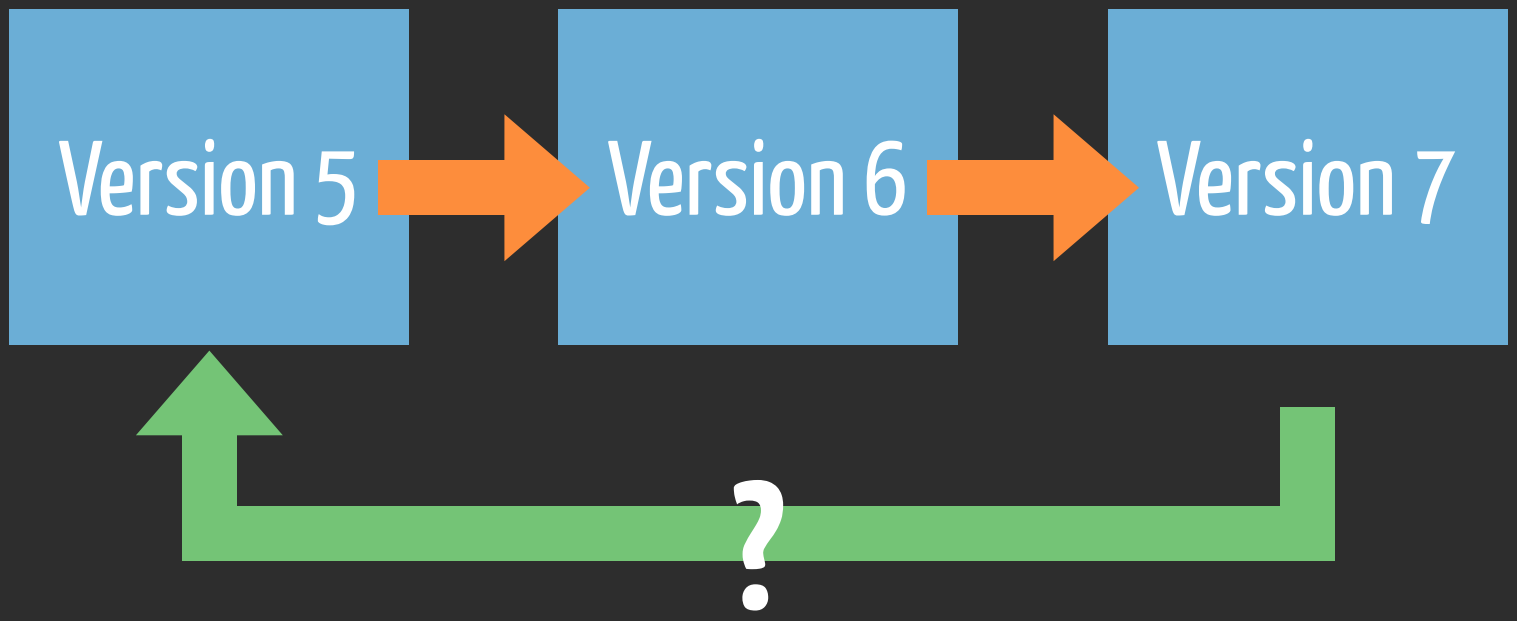
Who
What
When
Why

Query what changed

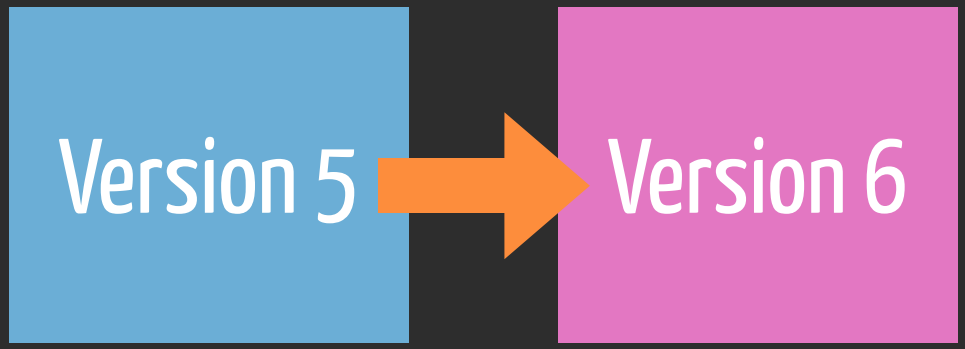
Query what changed



git diff; git log



Undo mistakes



Undo mistakes

git revert

Collaboration

R + git = nice

Collaboration

R + git + BitBucket = 

Collaboration

R + git + GitHub = 

Work on same code base

Added supplementary figure (though it's not in a separate file)
richfitz authored on Feb 17, 2013

121d561 <>

Supplementary figure with weak prior sampling
richfitz authored on Feb 17, 2013

d75ef41 <>

Minor changes to ms and bib file. Also added some figure captions tha...
mwpennell authored on Feb 17, 2013

b91ac0a <>

Tidied version of analysis code
richfitz authored on Feb 17, 2013

8107965 <>

working on the intro
Will Cornwell authored on Feb 17, 2013

6cffa61 <>

working on the abstract and introduction
Will Cornwell authored on Feb 17, 2013

369b882 <>

See what changed

167 wood-functions.R View

```
@@ -9,16 +9,22 @@ load.clean.data <- function(regenerate=FALSE) {
   ## Start by getting the woodiness information from the database
   dat <- read.forest.csv("export/speciesTraitData.csv")

-  ## Score the 633 species with no known information as NA
-  dat$woodiness[!(dat$woodiness %in% c("H", "W")) &
-                !is.na(dat$woodiness)] <- NA
-
   ## Only the columns we care about:
   dat <- data.frame(species=sub(" ", "_", dat$gs),
                     woodiness=dat$woodiness,
                     stringsAsFactors=FALSE)
+
   ## Filtered by whether or not they have woodiness information
-  dat <- dat[!is.na(dat$woodiness),]
+  to.drop.wood.NA <- is.na(dat$woodiness)
+  message(sprintf("Dropping %d species with NA woodiness values",
+                  sum(to.drop.wood.NA)))
```

See who changed it

142ec6ea » richfitz
2013-02-16
Version of analysis with str…

```
102   w <- matrix(NA, nrow(x), nrep)
103
```

27275949 » richfitz
2013-10-31
New, tidied, code.

```
104   ## A: genera with any known species
105   if (with.replacement)
106     w[ok,] <- x$W[ok] + rbinom(sum(ok), x$N[ok]-x$K[ok], x$W[ok]/x$K[ok])
107   else
108     w[ok,] <- t(sapply(which(ok), function(i)
109                        rhyper2(nrep, x$H[i], x$W[i], x$N[i])))
110
111   ## B: genera with no known species
112   n.unk <- sum(!ok)
113   w[!ok,] <- apply(w[ok,,drop=FALSE] / x$N[ok], 2, function(y)
114                    rbinom(n.unk, x$N[!ok], quantile(y, runif(n.unk))))
115
116   rownames(w) <- x$genus
117
```

beb815c6 » richfitz
2013-12-11
Generate supplementary data …

```
118   summarise.sim(w, x[c("order", "family", "genus",
119                        "W", "V", "H", "N", "K")])
```

git blame

142ec6ea » richfitz
2013-02-16
Version of analysis with str…

```
102   w <- matrix(NA, nrow(x), nrep)
103
```

27275949 » richfitz
2013-10-31
New, tidied, code.

```
104   ## A: genera with any known species
105   if (with.replacement)
106     w[ok,] <- x$W[ok] + rbinom(sum(ok), x$N[ok]-x$K[ok], x$W[ok]/x$K[ok])
107   else
108     w[ok,] <- t(sapply(which(ok), function(i)
109                        rhyper2(nrep, x$H[i], x$W[i], x$N[i])))
110
111   ## B: genera with no known species
112   n.unk <- sum(!ok)
113   w[!ok,] <- apply(w[ok,,drop=FALSE] / x$N[ok], 2, function(y)
114                    rbinom(n.unk, x$N[!ok], quantile(y, runif(n.unk))))
115
116   rownames(w) <- x$genus
117
```

beb815c6 » richfitz
2013-12-11
Generate supplementary data …

```
118   summarise.sim(w, x[c("order", "family", "genus",
119                        "W", "V", "H", "N", "K")])
```

Version control
git

Why doesn't everyone use this all the time?

Barriers to git

"It is easy to shoot your foot off with git, but also easy to revert to a previous foot and merge it with your current leg."

Jack William Bell: perl.plover.com/classes/git/samples/slide034.html

Barriers to git

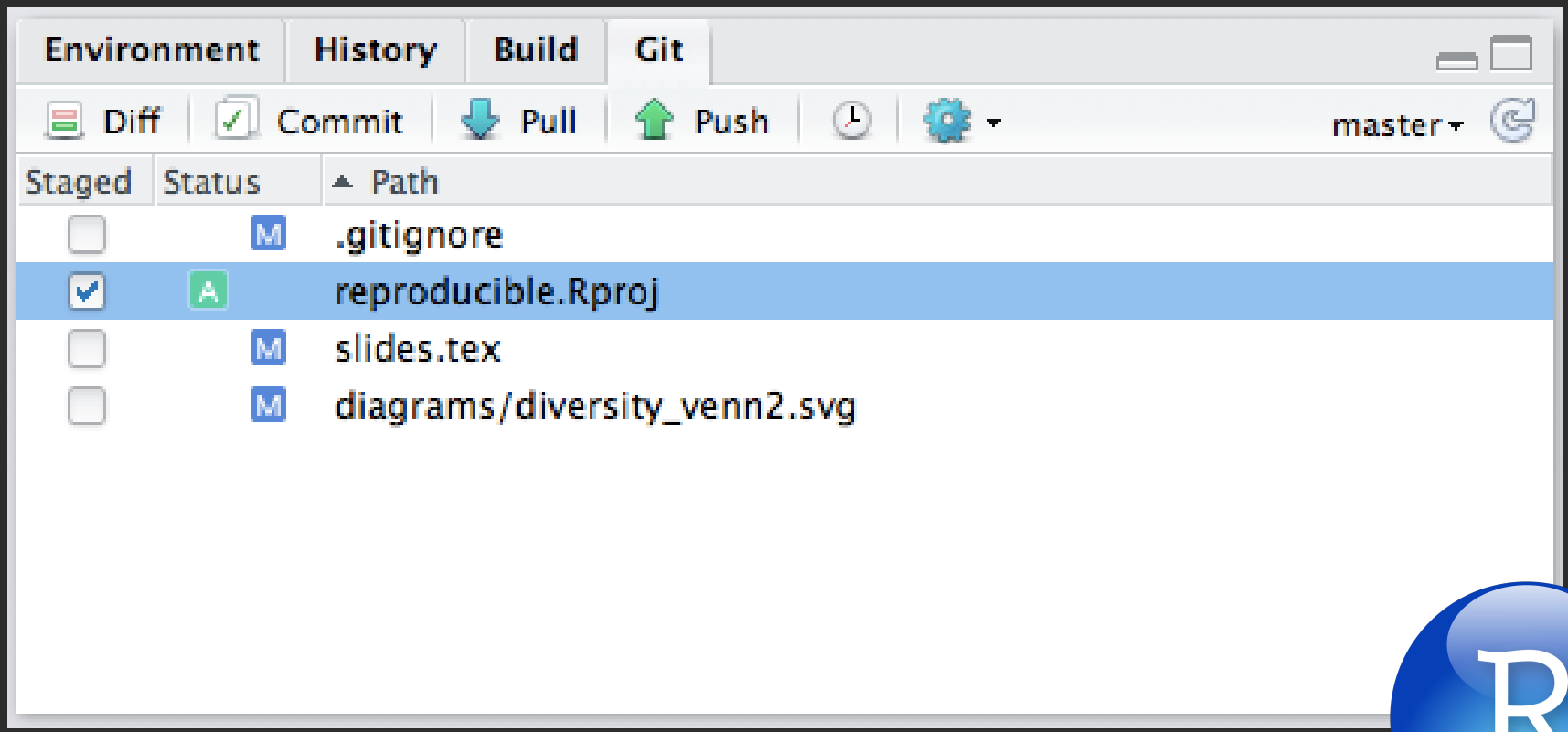
```
git rebase -s recursive -X theirs
    origin/master
```

Barriers to git

xkcd.com/1296

Barriers to git

Prospects for git

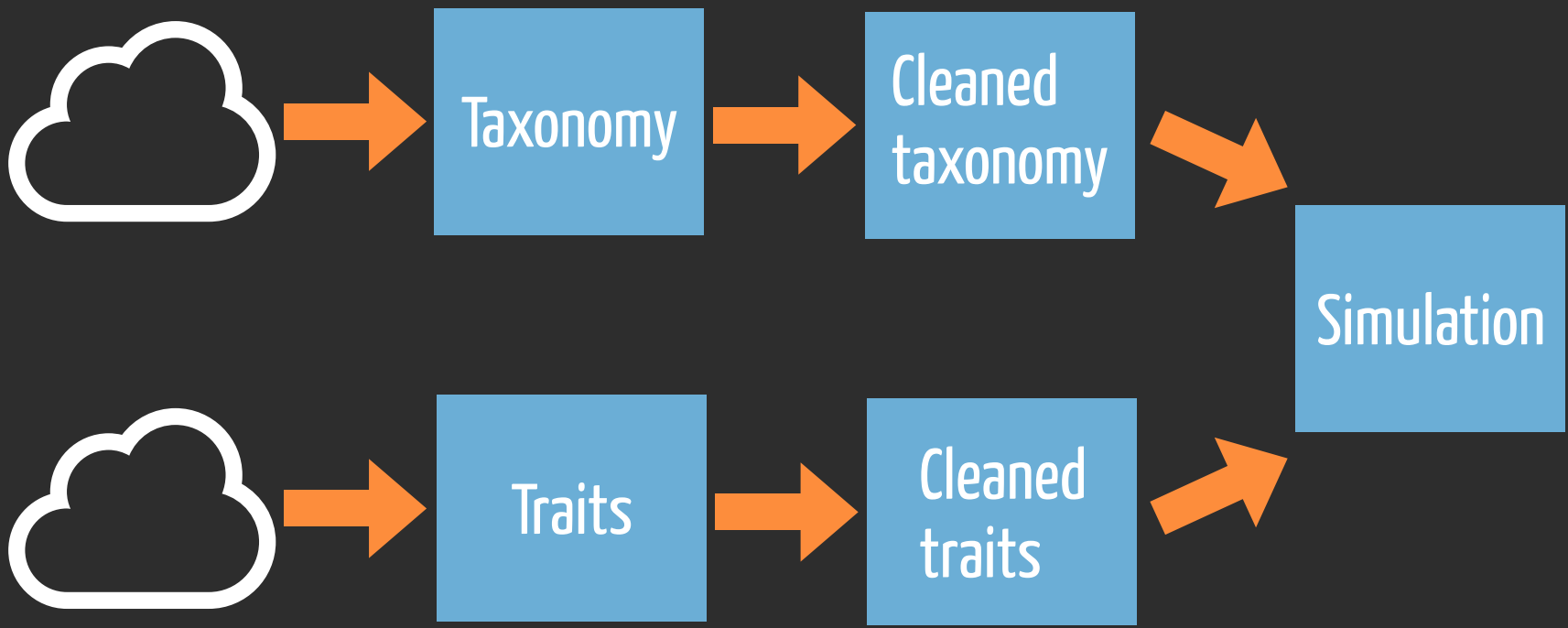


rstudio.com

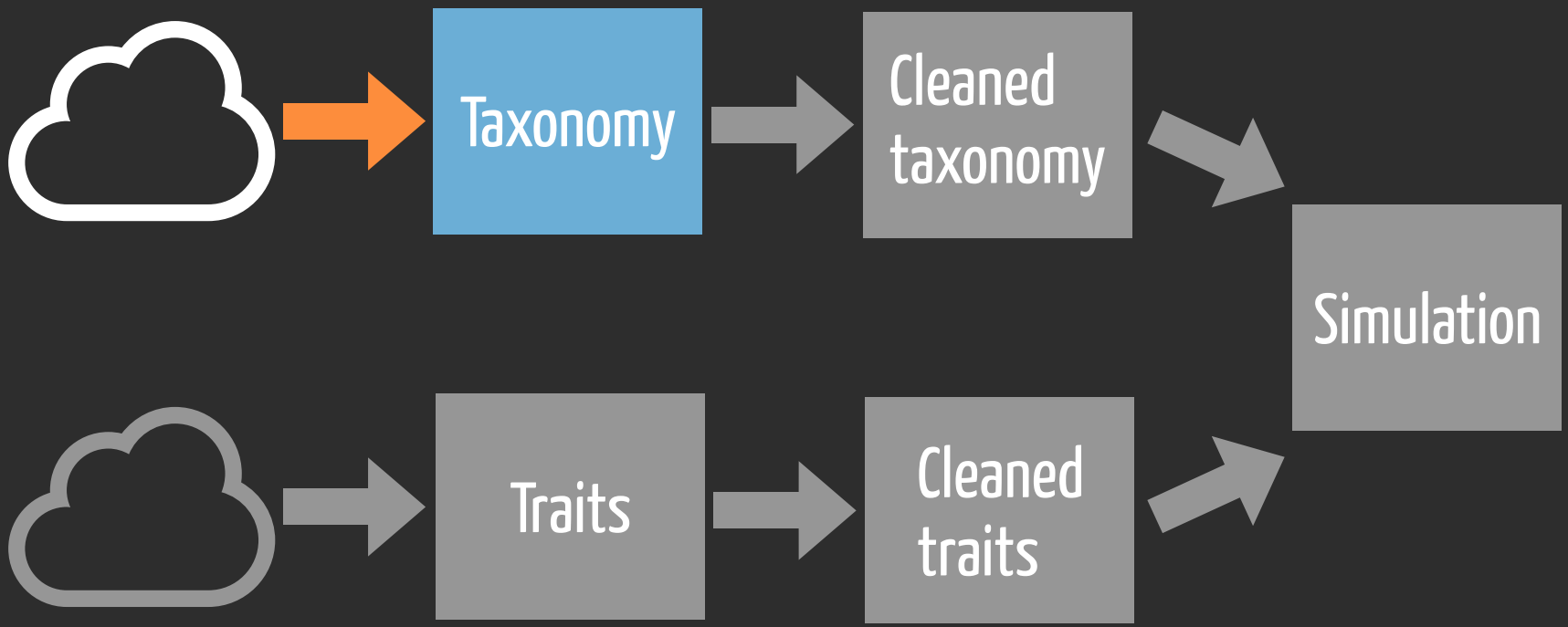
Workflows
make

It takes a while to make it work

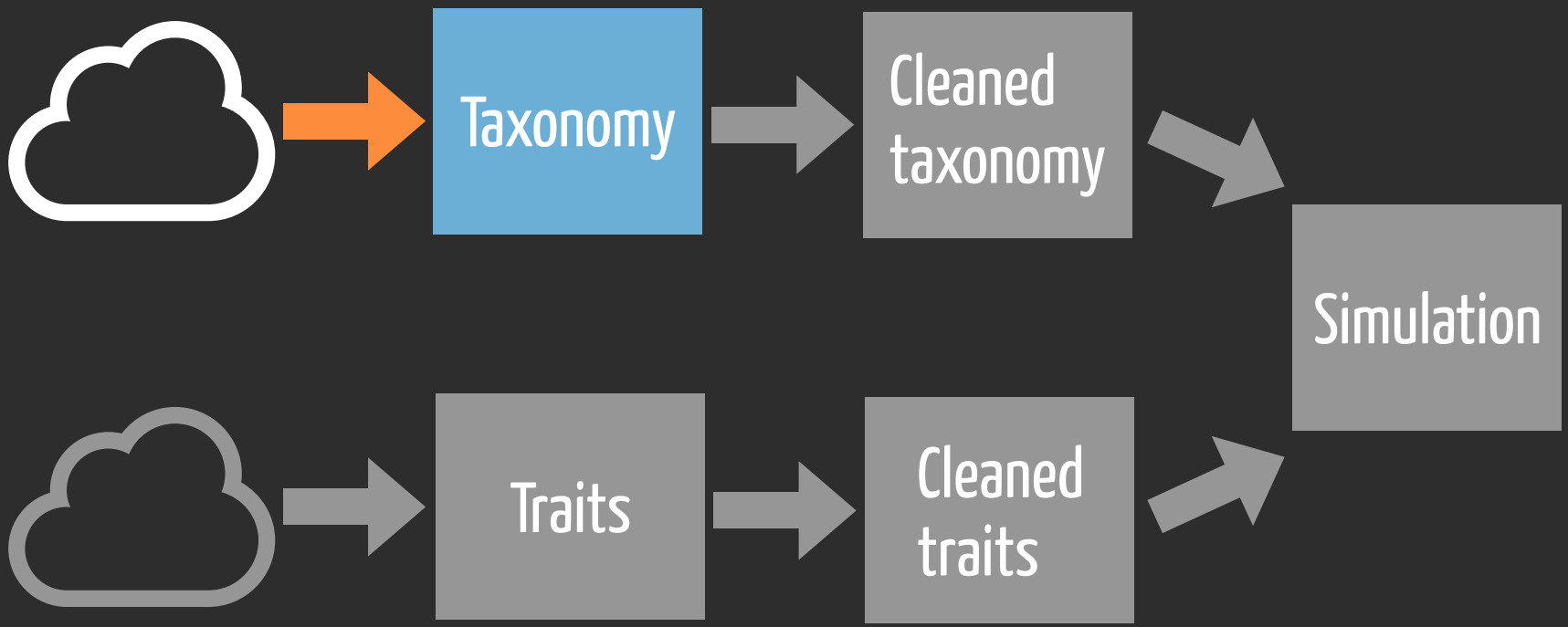
Our workflow



Download data

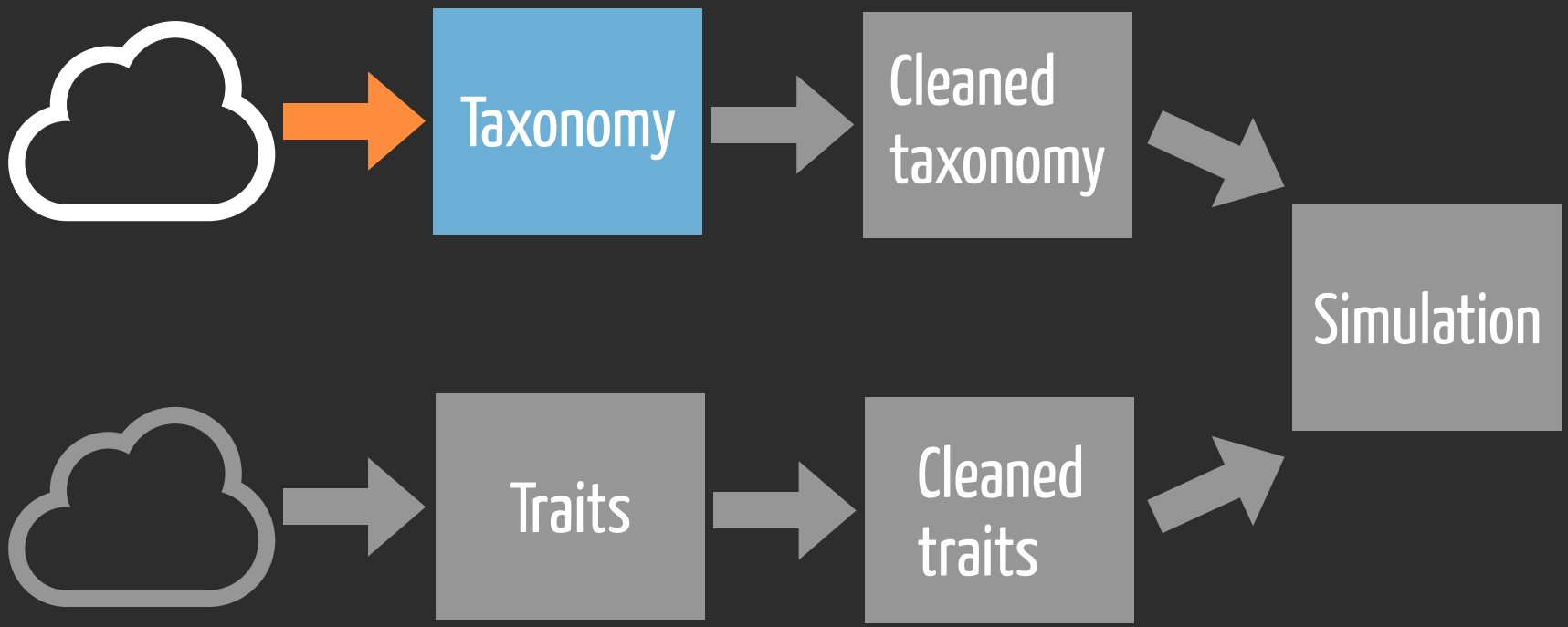


Rcurl, API access

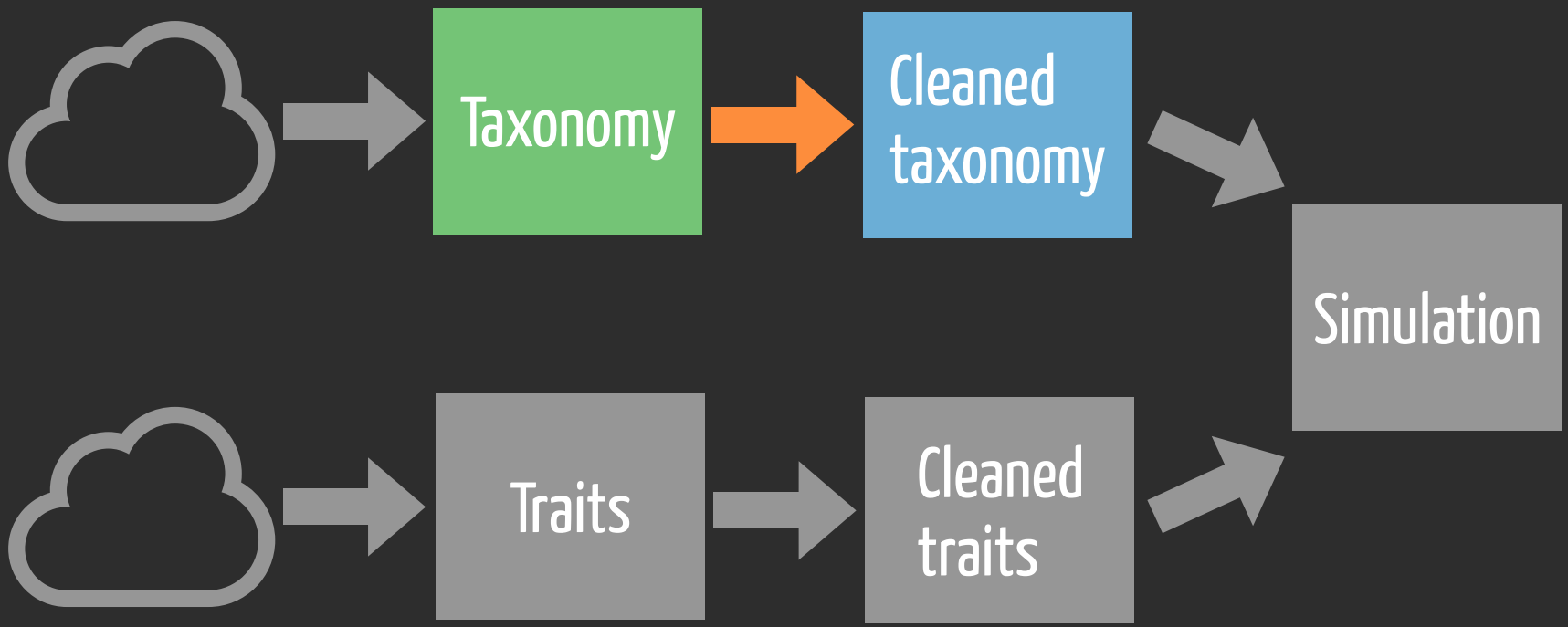


Makefile

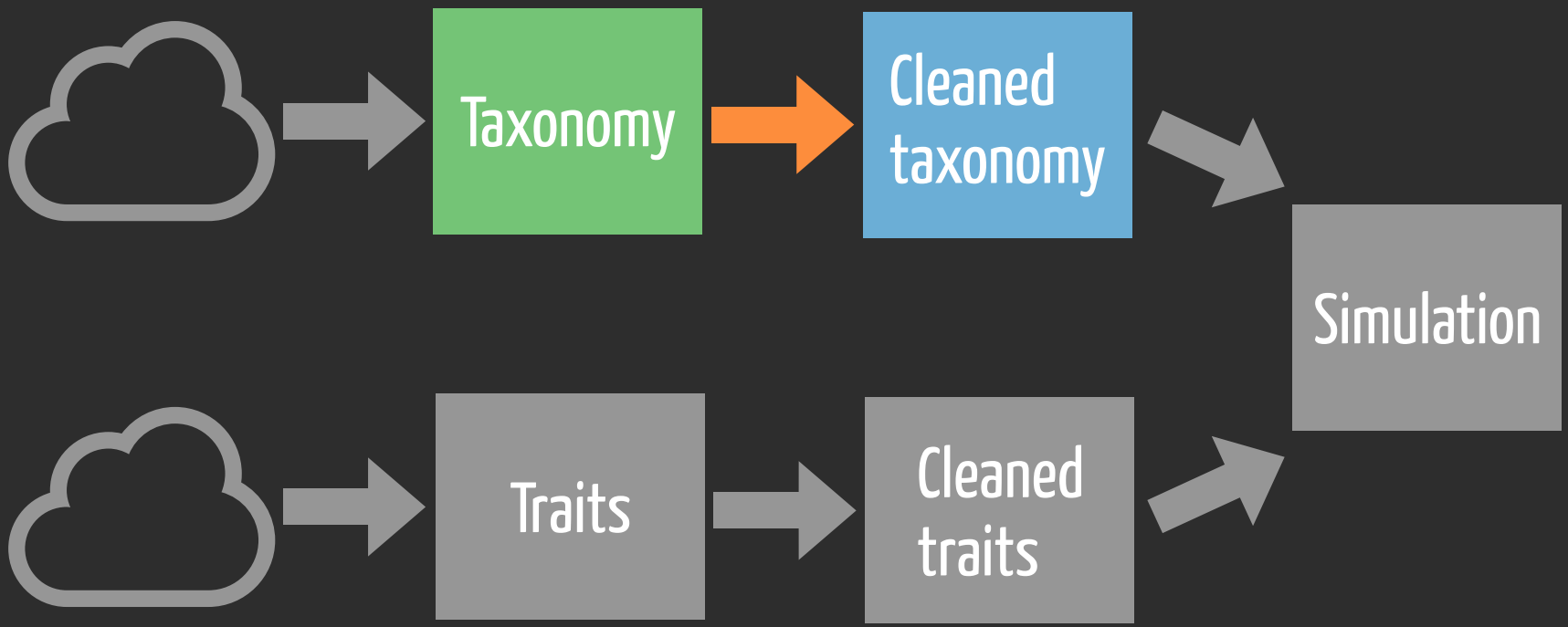
```
data/taxonomy.rds:
        Rscript download-taxonomy.R
```



Process data

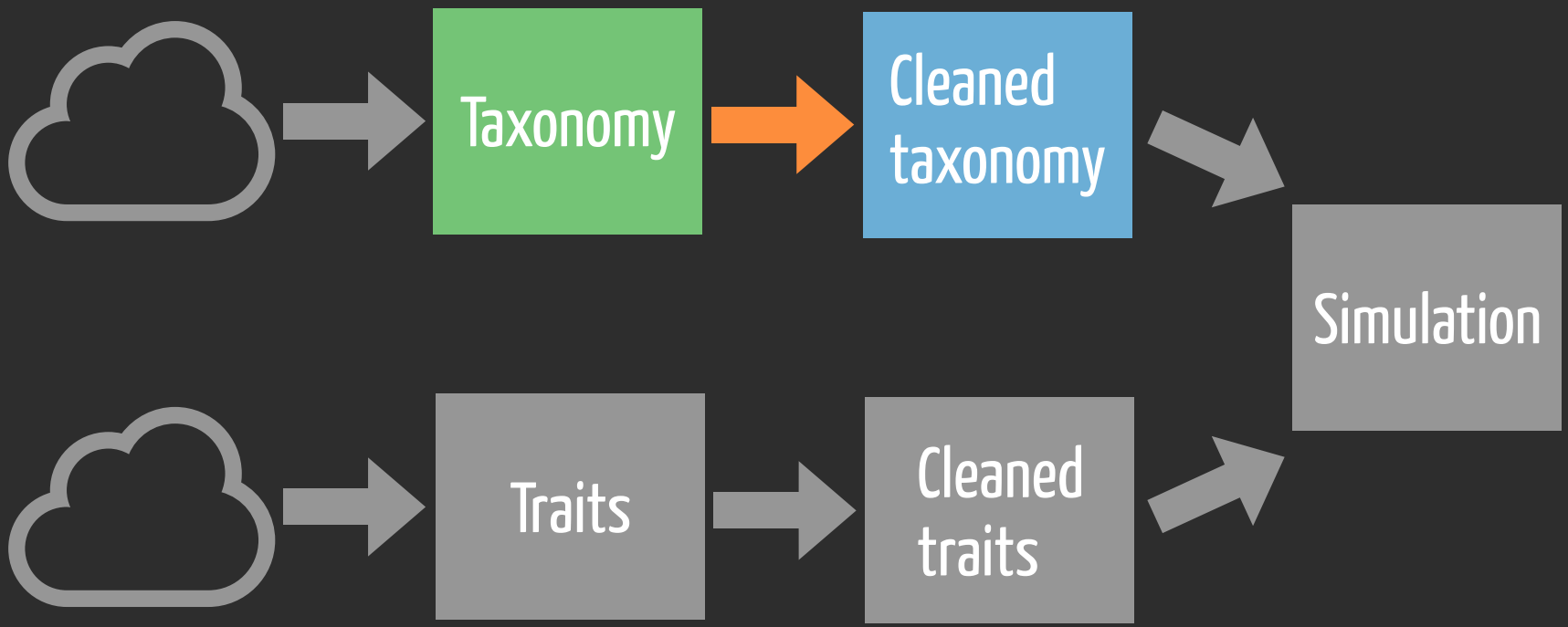


... the sausage factory

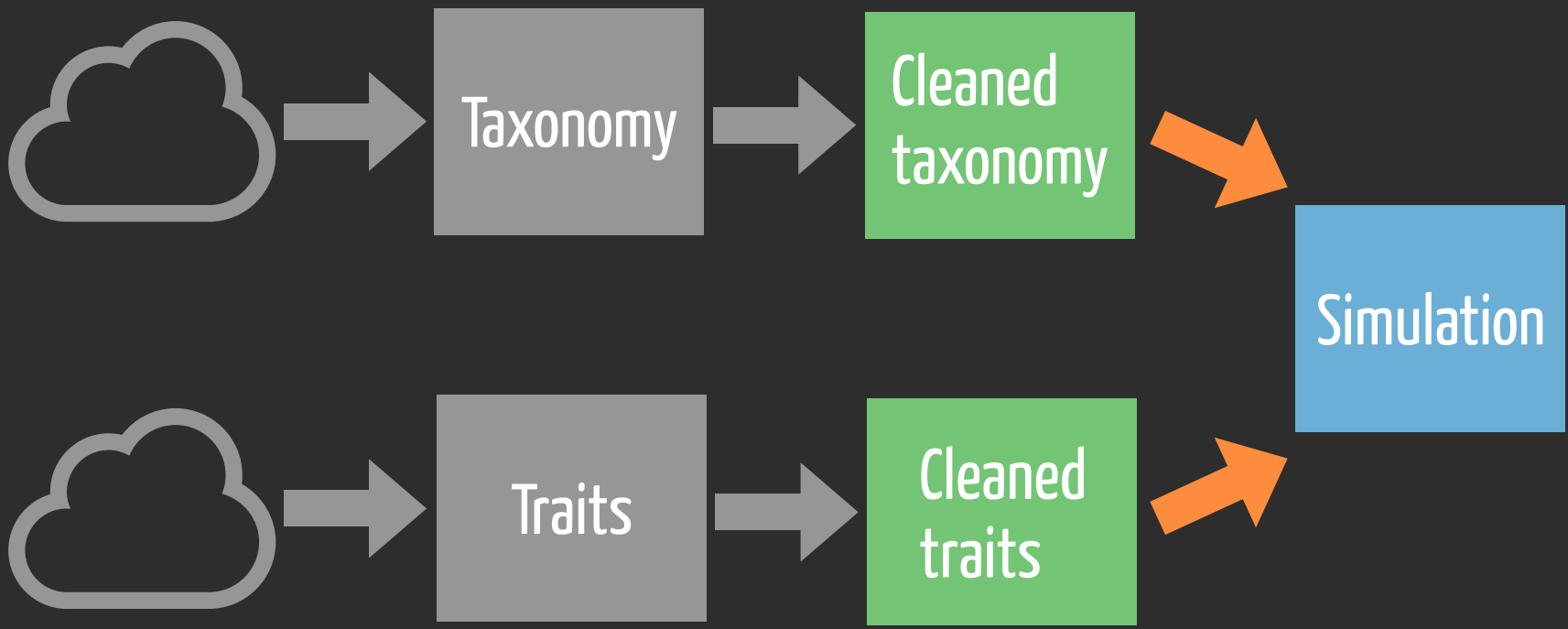


Makefile

```
processed/taxonomy.rds: data/taxonomy.rds
	Rscript cleanup-taxonomy.R
```

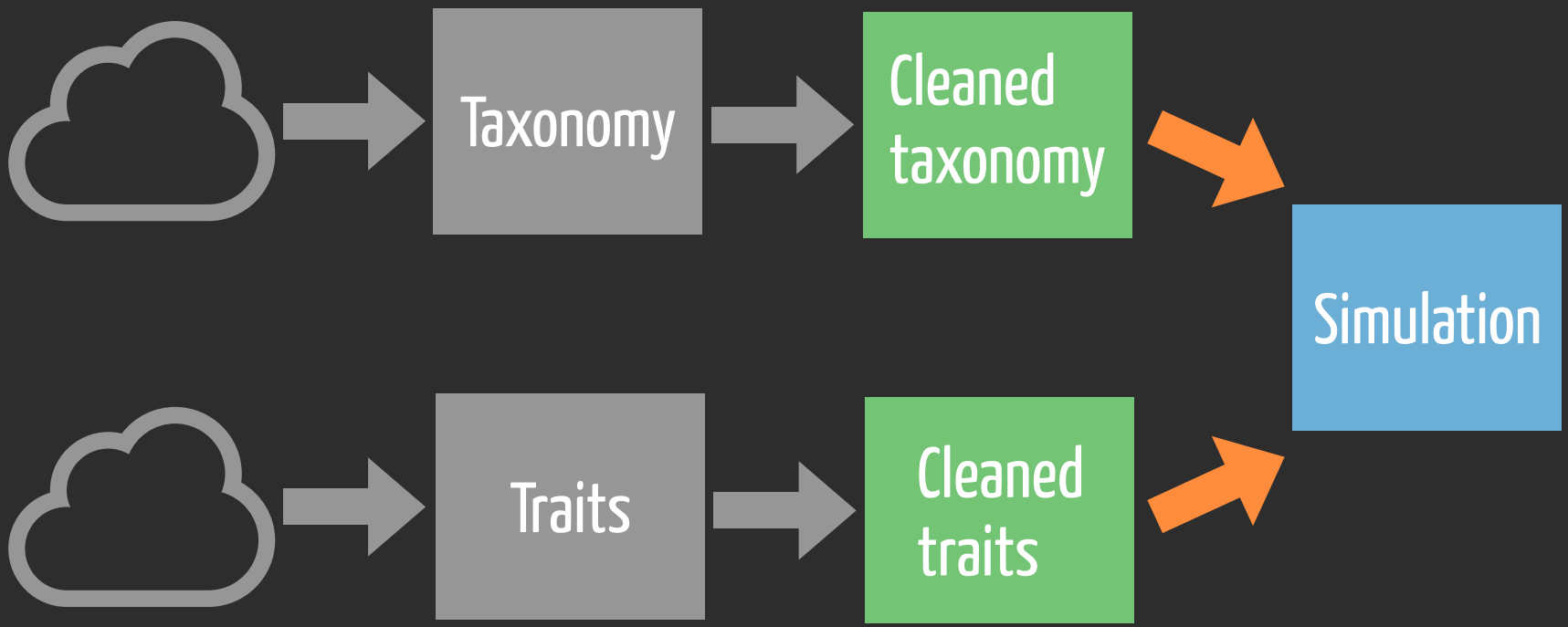


Run the actual science bit

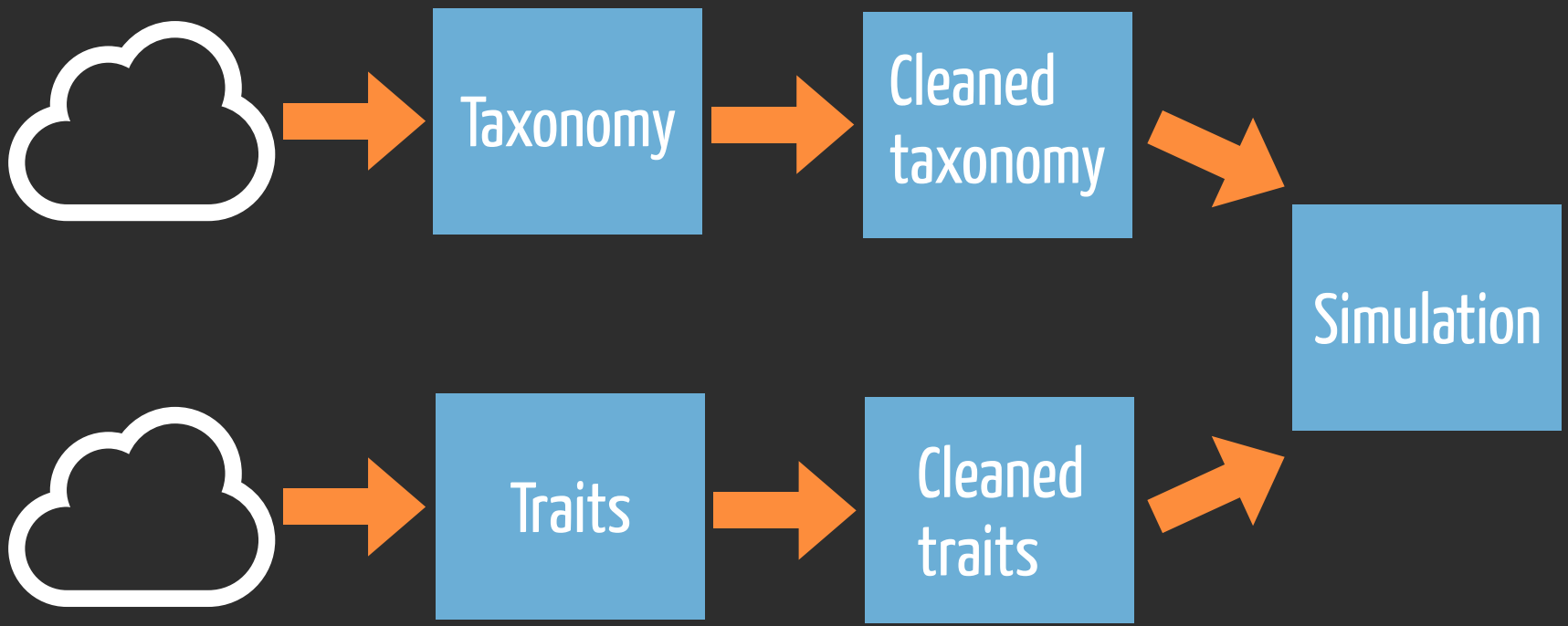


Makefile

```
simulation.md: processed/taxonomy.rds \
               processed/traits.rds
        Rscript simulation.Rmd
```



make:
Self-documenting workflow



Workflows
make

Why doesn't everyone use this all the time?

Barriers to make

Lots of traps

Barriers to make

Command-line only, arcane tool
Comes in several incompatible flavours

Barriers to make

Currently looking for a
modern, accessible replacement

Automated testing
travis-CI

Will it work elsewhere?

travis-ci.org

CI = Continuous Integration

1. Commit changes
2. Make sure nothing breaks

CI = Continuous Integration

1. Commit changes
2. Push to GitHub

Spins up virtual machine...

```
1  Using worker: worker-linux-8-2.bb.travis-ci.org:travis-linux-10
2
3  $ export BOOTSTRAP_LATEX="1"
4  $ export GH_TOKEN=[secure]
5  $ export USE_PACKRAT=0
6  $ export CC=gcc
7  $ git clone --depth=50 --branch=master git://github.com/richfitz/wood.git   git.1
15 $ cd richfitz/wood
16 $ git checkout -qf a1c89767c03afe47d3c7bbdd676f5f8125df613e                 git.3
17 $ gcc --version
18 gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
19 Copyright (C) 2011 Free Software Foundation, Inc.
20 This is free software; see the source for copying conditions.  There is NO
21 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

...installs dependencies...

```
▶   23 $ curl -OL http://raw.github.com/craigcitro/r-travis/master/scripts/travis-tool.sh   before_install.1
▶   29 $ chmod 755 ./travis-tool.sh                                                         before_install.2
▶   30 $ ./travis-tool.sh bootstrap                                                        before_install.3
▶  526 $ ./travis-tool.sh aptget_install libgsl0-dev                                            install.1
▶  572 $ ./travis-tool.sh aptget_install fftw3-dev                                              install.2
▶  617 $ ./travis-tool.sh aptget_install texlive-humanities                                     install.3
▶  660 $ ./travis-tool.sh install_deps                                                          install.4
▶ 1007 $ ./travis-tool.sh github_package richfitz/sowsear                                       install.5
▶ 1055 $ ./travis-tool.sh github_package richfitz/diversitree                                   install.6
▶ 1534 $ make packrat-perhaps                                                                   install.7
```

...downloads and processes data...

```
1554 $ make
1555 Rscript --default-packages="datasets,utils,grDevices,graphics,stats,methods" -e "library(sowsear);
     sowsear('wood.R', 'Rmd')"
1556 Loading required package: knitr
1557 Rscript --default-packages="datasets,utils,grDevices,graphics,stats,methods" make/data-zae.R
1558 Rscript --default-packages="datasets,utils,grDevices,graphics,stats,methods" make/data-theplantlist.R
1559 Skipping Araucariaceae (gymnosperm) -- already exists
1560 Skipping Cupressaceae (gymnosperm) -- already exists
```

```
2031 Skipping Zosteraceae (angiosperm) -- already exists
2032 Skipping Zygophyllaceae (angiosperm) -- already exists
2033 Rscript --default-packages="datasets,utils,grDevices,graphics,stats,methods" make/output-woodiness.rds.R
2034 Resolving synonomy for 3037 species
2035 Dropping 8430 species not in Plant List
2036 After synonym correction, 1125 duplicated entries
```

...runs knitr ...

```
2046 Rscript --default-packages="datasets,utils,grDevices,graphics,stats,methods" -e "library(knitr);
     knitr('wood.Rmd')"
2047
2048
2049 processing file: wood.Rmd
2050   |.                                                    |   1%
2051   ordinary text without R code
2052
2053   |..                                                   |   2%
2054 label: unnamed-chunk-1 (with options)
2055 List of 2
2056  $ echo   : logi FALSE
2057  $ results: logi FALSE
2058
2059   |..                                                   |   4%
```

...& compiles manuscript.

```
2311 make -C doc
2312 make[1]: Entering directory `/home/travis/build/richfitz/wood/doc'
2313 pdflatex -interaction=nonstopmode wood-ms-supporting.tex
2314 This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009/Debian)
2315 entering extended mode
2316 (./wood-ms-supporting.tex
2317 LaTeX2e <2009/09/24>
2318 Babel <v3.81> and hyphenation patterns for english, usenglishmax, dumylang, noh
2319 yphenation, loaded.
2320 (/usr/share/texmf-texlive/tex/latex/base/article.cls
```

Configuration: .travis.yml

```
script:
    - make cache-unpack
    - make
install:
    - ./travis-tool.sh install_deps
    - ./travis-tool.sh github_package richfitz/diversitree
before_install:
    - curl -OL http://raw.github.com/craigcitro/...
    - chmod 755 ./travis-tool.sh
    - ./travis-tool.sh bootstrap
```

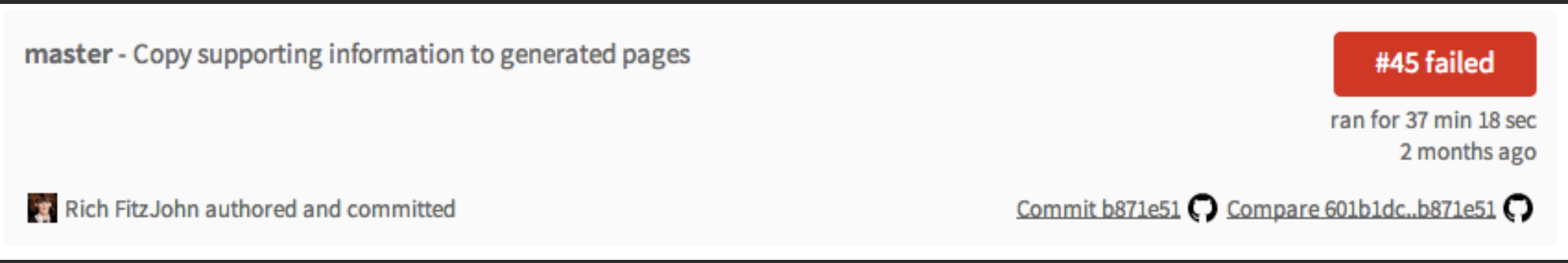
travis-CI for R: github.com/craigcitro/r-travis

Set & forget: travis never gets bored

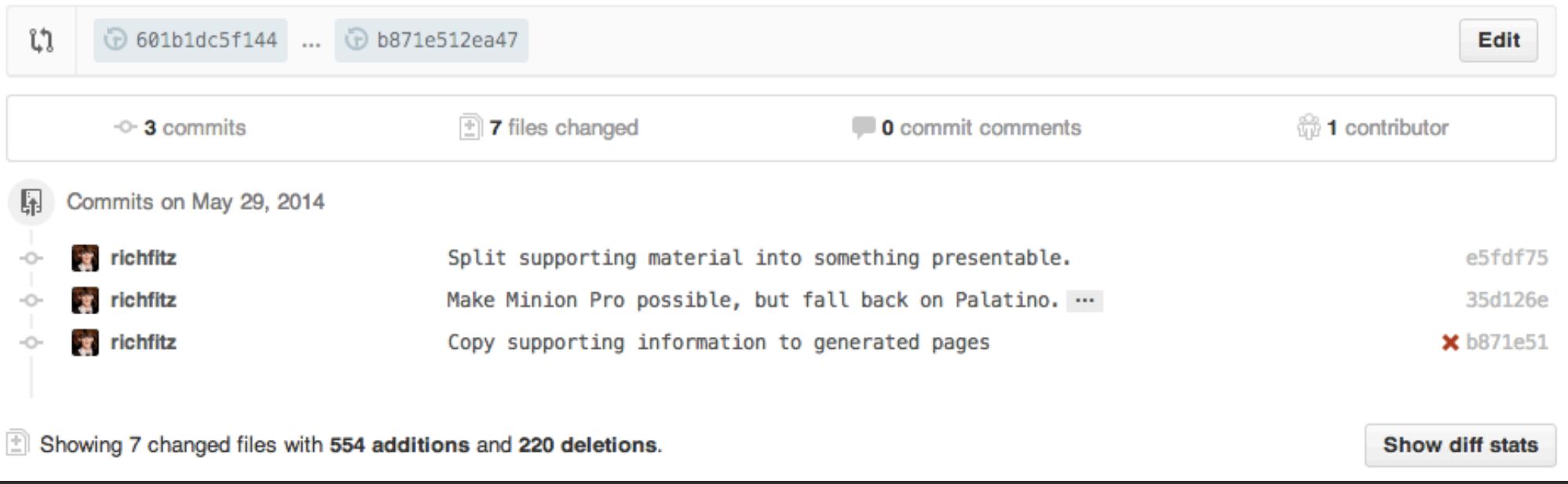
Build	Message	Commit	Duration	Finished
✓ 50	Comment from Matt	a1c8976 (master)	40 min 26 sec	2 months ago
✓ 49	Update to v1.0	6bd8393 (v1.0)	40 min 19 sec	2 months ago
✓ 48	Update to v1.0	6bd8393 (master)	44 min 30 sec	2 months ago
✓ 47	Updates to webpages.	daf3215 (master)	41 min 53 sec	2 months ago
✓ 46	Use color not xcolor (installed on travis, should fix build)	a6d539c (master)	50 min 18 sec	2 months ago
✗ 45	Copy supporting information to generated pages	b871e51 (master)	37 min 18 sec	2 months ago
✓ 44	updated zanne big tree citation	601b1dc (master)	59 min 25 sec	2 months ago
✓ 43	checked all numbers. i think everything is perfect now	5ab8413 (master)	51 min 11 sec	3 months ago
✓ 42	updating a the numbers in the text after fixing the plant list error	bd04753 (master)	39 min 41 sec	3 months ago

https://travis-ci.org/richfitz/wood/builds

Find out what/who broke the project



Find out what/who broke the project



Automated testing

travis-CI

Why doesn't everyone use this all the time?

Barriers to travis-CI

Project must already be reproducible

Barriers to travis-CI

Only for open source, or pay

Barriers to travis-CI

Ill-suited for long running jobs, sensitive data

Dependencies
packrat

Will it work later?

rstudio.github.io/packrat

See also
rbundler

Identify dependencies

```
packrat::init()
 library(ggplot2)
 require(lme4)
 assertthat::see_if(...)
```

Identify dependencies

```
packrat::init()
 Package: ggplot2
 Source: CRAN
 Version: 1.0.0
 Hash: c8bff66238347472f08b6a35608539ff
 Requires: digest, gtable, plyr...
```

… & their dependencies

```
packrat::init()
```

```
Package: plyr
Source: CRAN
Version: 1.8.1
Hash: be21bad411e628f810a92212e17b5be7
Requires: Rcpp
```

Project is now isolated from system

```
 ~/Documents/Projects/repro » R
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
...
Packrat mode on. Using library in directory:
- "/Users/rich/Projects/repro/packrat/lib"
>
```

Dependencies
packrat

Why doesn't everyone use this all the time?


In this Style 10/6

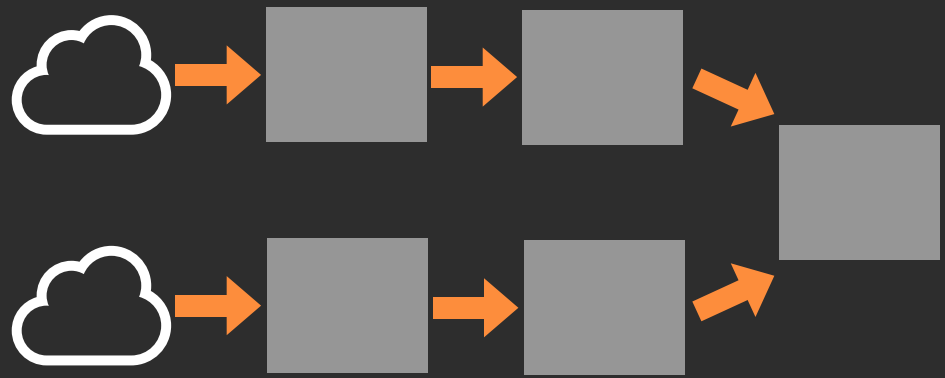
QLD grassland by Willem van Aken

QLD rainforest by Willem van Aken

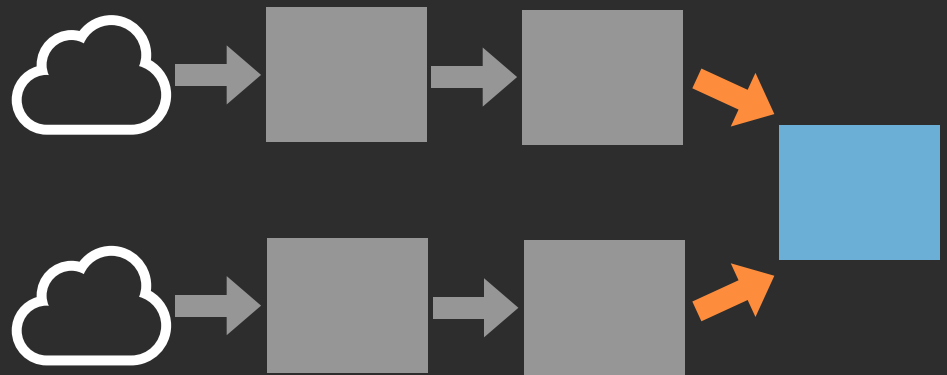
QLD grassland by Willem van Aken

QLD rainforest by Willem van Aken

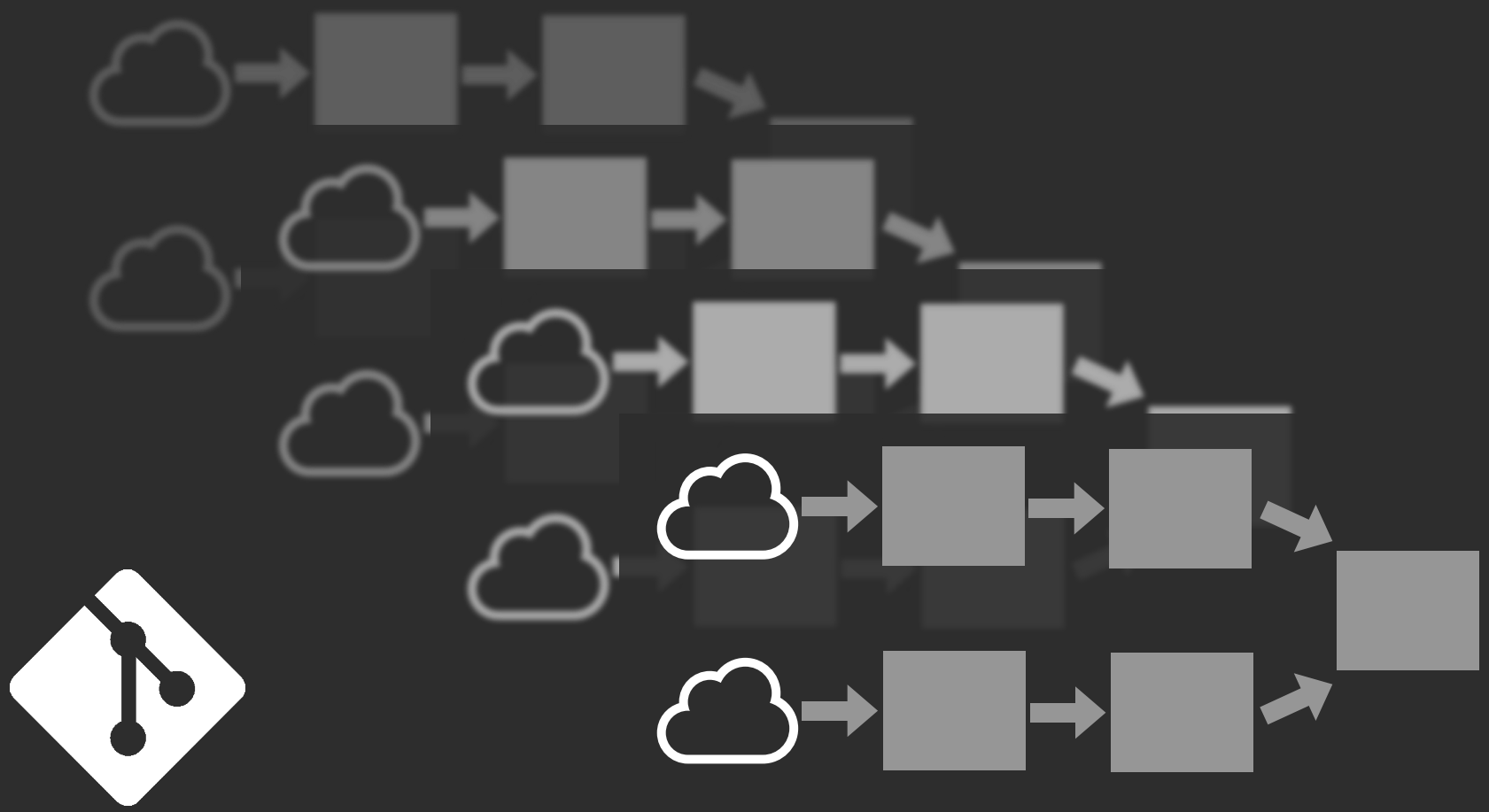
GNU make



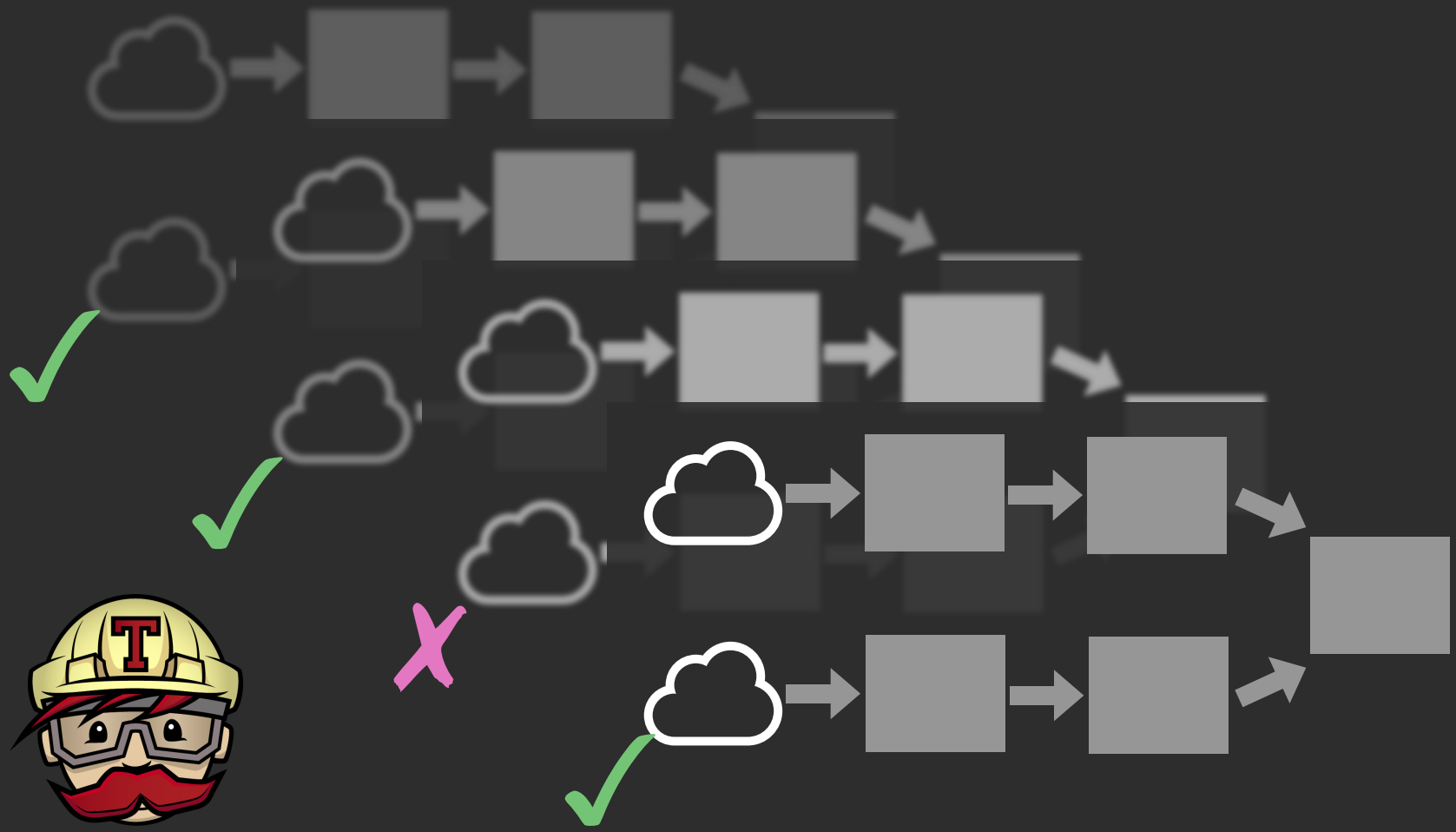
knitr



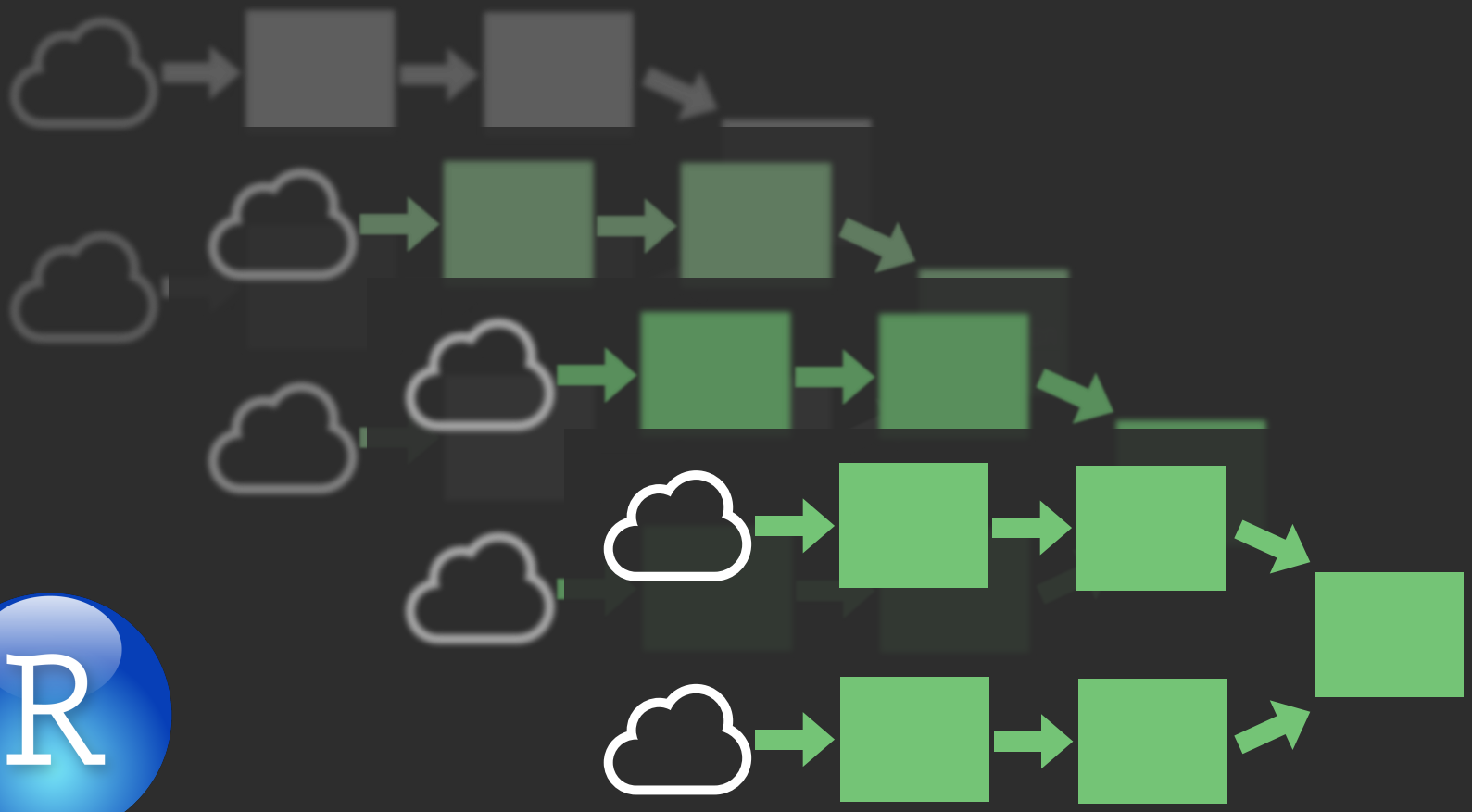
git



travis-CI



packrat



100%
reproducible

100% reproducible

```
git clone https://github.com/richfitz/wood/
cd wood
make deps all
```

...provided you have C, C++ & Fortran compilers, make, GNU scientific library, LaTeX.

100% reproducible

Probably unrealistic at the moment

Partially reproducible

It's not just good — it's good enough

Partially reproducible

Good faith effort at documenting requirements makes it much easier to pick up

How to be more reproducible

- Think about reproducibility from the start
- Avoid manual intervention
- Think about workflows, project structure
- Identify key inputs, outputs
- Run your project on a second computer

Acknowledgements


Macquarie University

University of British Columbia


Natural Sciences & Engineering Research Council of Canada

National Evolutionary Science Synthesis Center

Advice Carl Boettiger, Scott Chamberlain, Daniel Falster, Ted Hart, Sally Otto, Heather Piwowar, Karthik Ram

Design Mike Bostock: bost.ocks.org/mike/d3/workshop#0

Collaborators


Matt Pennell @mwpennell


Will Cornwell @will_cornwell


Amy Zanne @amyzanne


Dave Tank @dave_tank


Peter Stevens

Resources

Paper & analysis	richfitz.github.io/wood
This talk	github.com/richfitz/reproducibility-2014
rOpenSci	ropensci.org
Software Carpentry	software-carpentry.org
git	git-scm.com
knitr	yihui.name/knitr
travis-CI	travis-ci.org & github.com/craigcitro/r-travis