

rOpenSci Data Packages

Scott Chamberlain

 sckottie

Why Data Packages?

- Reduce duplicated effort by each researcher
- One best way to get data XYZ
- Reduced user error
- Allow researchers to focus on the science

Data Packages: Caveats

- User base (# of people) for data pkgs small relative to utilities
 - ... attracts fewer contributors
 - ... & all the carry on effects of above
- Pkg can get out of sync with data source's API
- Risk leaving out metadata/context

rOpenSci Data Packages

- Biological occurrences
- Taxonomy
- Data utilities

find more: ropensci.org/packages

Occurrence Data Packages

- [spocc](#) - Biodiversity data toolbelt
- [rgbif](#) - GBIF data
- [ecoengine](#) - Berkeley Ecoengine client
- [rinat](#) - iNaturalist client
- [rbison](#) - USGS BISON client
- [rebird](#) - eBird data via their API
- [auk](#) - eBird bulk data
- [rvertnet](#) - VertNet data
- [rfishbase](#) - Fishbase.org data
- [finch](#) - Handle Darwin Core

Data package patterns

Or at least patterns we strive towards ...

- HTTP requests
- Cache downloaded data for FTP/similar files
- Return `data.frame`'s: facilitates downstream processing
- Make it easy to cite data providers
- Incorporate metadata

rgbif



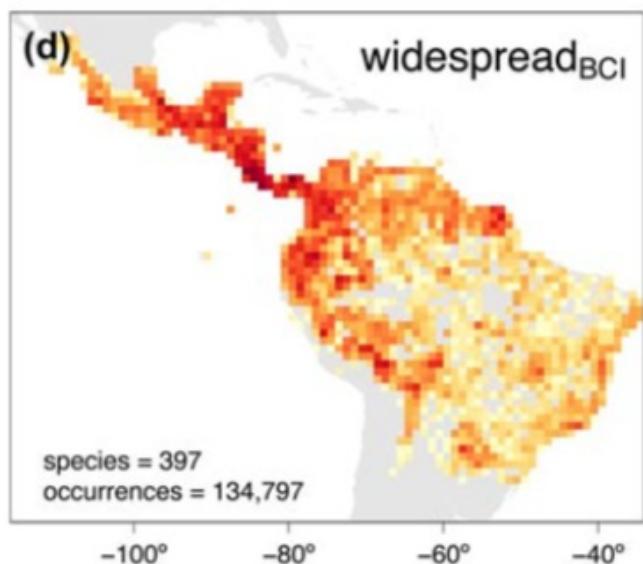
Access > 1 billion occurrence records from
GBIF

 [ropensci/rgbif](#)

- Search for species
- Download occurrence data
- Clean occurrence data
- Make maps

rgbif usage

Bemmels, et al. (2018) -- Filter-dispersal assembly of lowland Neotropical rainforests across the Andes



... records were obtained for each species ... using ... `rgbif` ... Quality controls were applied to ensure that occurrence records were correctly taxonomically identified and accurately georeferenced. In particular, records with the following issues were excluded ... invalid coordinates or geodetic datum; failed or suspicious coordinate reprojection; ... species name with no match ...

~70 more e.g.'s at <https://github.com/ropensci/roapi/blob/master/data/citations.csv>

rgbif usage: Checklist recipe

TrIAS Project - standardizing species checklist data to Darwin Core using R

├── README.md	: Description of this repository
├── LICENSE	: Repository license
├── checklist-recipe.Rproj	: RStudio project file
├── .gitignore	: Files and directories to be ignored by git
└── data	
├── raw	: Source data, input for mapping script
└── processed	: Darwin Core output of mapping script GENERATED
└── docs	: Repository website GENERATED
└── src	
├── dwc_mapping.Rmd	: Darwin Core mapping script, core functionality
├── _site.yml	: Settings to build website in /docs
└── index.Rmd	: Template for website homepage



Taxonomy Packages

- [taxa](#) - Taxonomic R classes
- [taxize](#) - Taxonomic toolbelt (remote data)
 - [taxize book](#)
- [taxizedb](#) - Leverage dumps of taxonomic database locally
- [ritis](#) - USGS's ITIS
- [rotl](#) - Open Tree of Life
- [rentrez](#) - NCBI ENTREZ taxonomy database
- [worrms](#) - WORMS marine taxonomy
- [wikitaxa](#) - Taxonomy data on Wikipedia
- [zbank](#) - ZooBank
- [rgbif](#) - GBIF taxonomy data

taxize

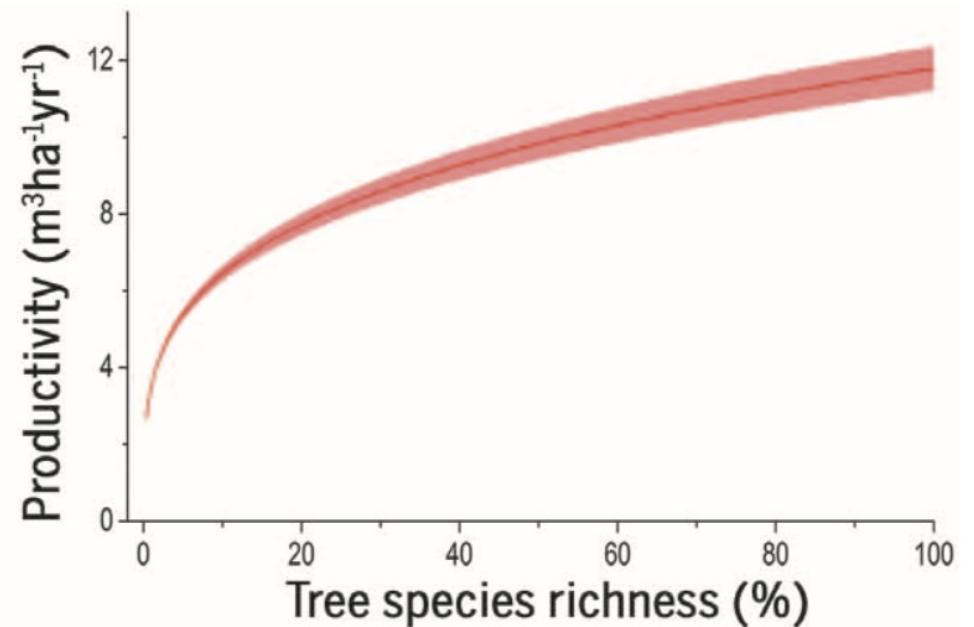
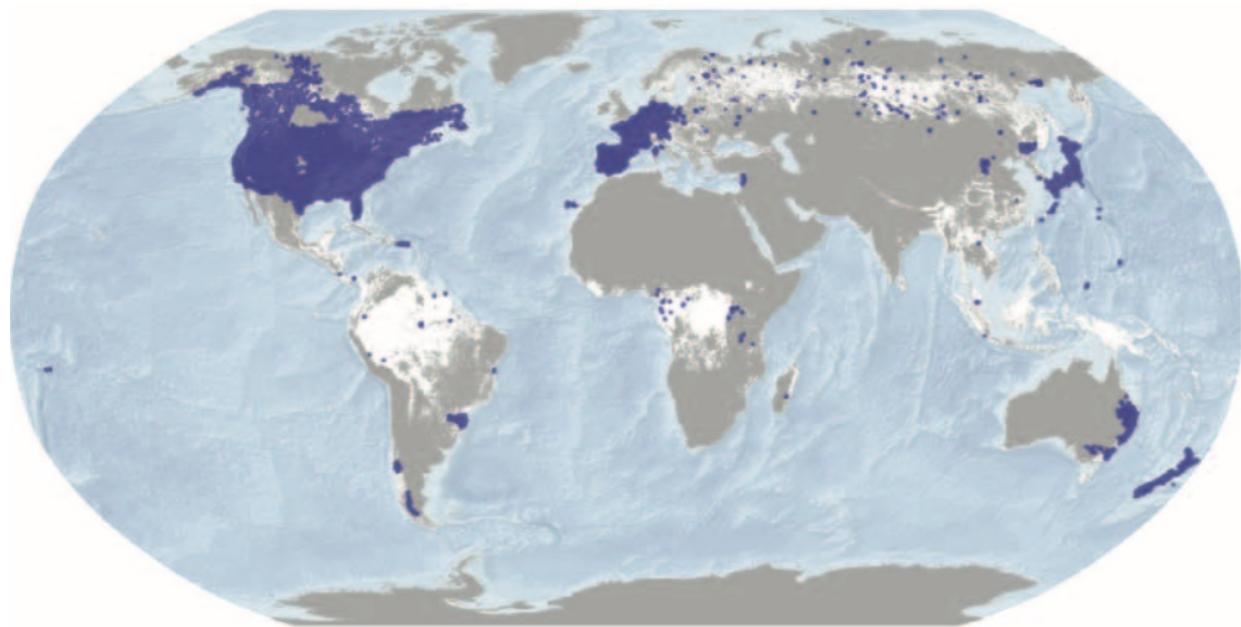
Access taxonomic data from > 20 sources



- Resolve misspelled/etc. names
- Search for names
- Taxonomic classifications
- Fetch all taxa up- and down-stream
- Fetch taxonomic synonyms
- Common names to taxonomic and vice versa

taxize usage

Liang, J., et al. (2016) -- Positive biodiversity-productivity relationship predominant in global forests



there were ... 8,737 species ... We verified all ... names against 60 taxonomic data-bases, including NCBI, GRIN Taxonomy for Plants, Tropicos–Missouri Botanical Garden, and the International Plant Names Index, using the ‘taxize’ package in R

~90 more e.g.'s at <https://github.com/ropensci/roapi/blob/master/data/citations.csv>

Utility Packages

- `jqr` - R client for `jq`, the JSON processor
- `jsonld` - JSON linked data
- `rerddap` - ERDDAP server client
- `rdflib` - RDF pkg, wrapped around Redland
- `assertr` - Assertions for analysis pipelines

assertr usage: eg

Brain Somatic Mosaicism Network - Validate BSMN Grant Data

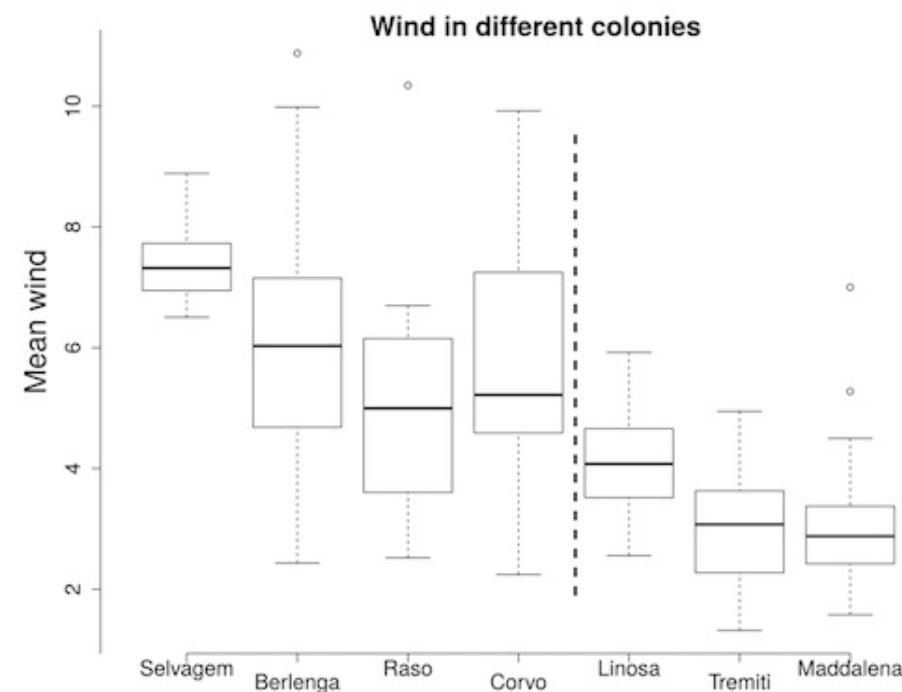
The screenshot shows a GitHub repository page for 'bsmn / grantdatavalidator'. The repository name is at the top left, followed by a 'Watch' button (4), a 'Star' button (0), and a 'Fork' button (0). Below the repository name are navigation links: 'Code' (selected), 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', and 'Insights'. A summary section below the navigation bar shows '9 commits', '1 branch', '0 releases', and '1 contributor'. The main content area displays a block of R code:

```
data %>%
  assertr::chain_start() %>%
  assertr::verify(nrow(data) == 3) %>%
  assertr::verify(assertr::is_uniq(ndc_short_name)) %>%
  assertr::verify(assertr::not_na(grant)) %>%
  assertr::verify(dplyr::n_distinct(grant) == 1) %>%
  assertr::verify(ndc_short_name %in% expected_ndc_short_names) %>%
  assertr::chain_end() %>%
  tibble::as_tibble()
```

 assertr

rerddap usage: eg

Abolaffio, J., et al. (2018) -- Olfactory-cued navigation in shearwaters: linking movement patterns to mechanisms



wind data were downloaded from the NOAA web site from the *rerddapp* package for R

 [rerddap](#)

Data integration: Steps

- Start with a species list
- Clean names with `taxize`
- Get occurrence data with `rgbif`
- Clean occurrence data w/ `rgbif`, `scrubr` or `CoordinateCleaner`
- `assertr` to check data
- Map with `mapr`

Data integration: code

```
# read in species list
spp <- read.csv("spp_list.txt", header = TRUE,
  stringsAsFactors = FALSE)$bad
# resolve names: fix misspellings
spp2 <- taxize::gnr_resolve(spp, data_source_ids = 11,
  canonical = TRUE)$matched_name2
# fetch GBIF occurrence data
dat <- rgbif::occ_data(scientificName = spp2, limit = 300)
# remove data with issues: COUNTRY_MISMATCH & COORDINATE_ROUNDED
dat <- rgbif::occ_issues(dat, -cum, -cdround)
# make a single data.frame
dat <- dplyr::bind_rows(lapply(dat, "[[", "data"))
# remove records with incomplete lat/lon data
dat <- scrubr::coord_incomplete(dat)
```

Data: spp_list.txt

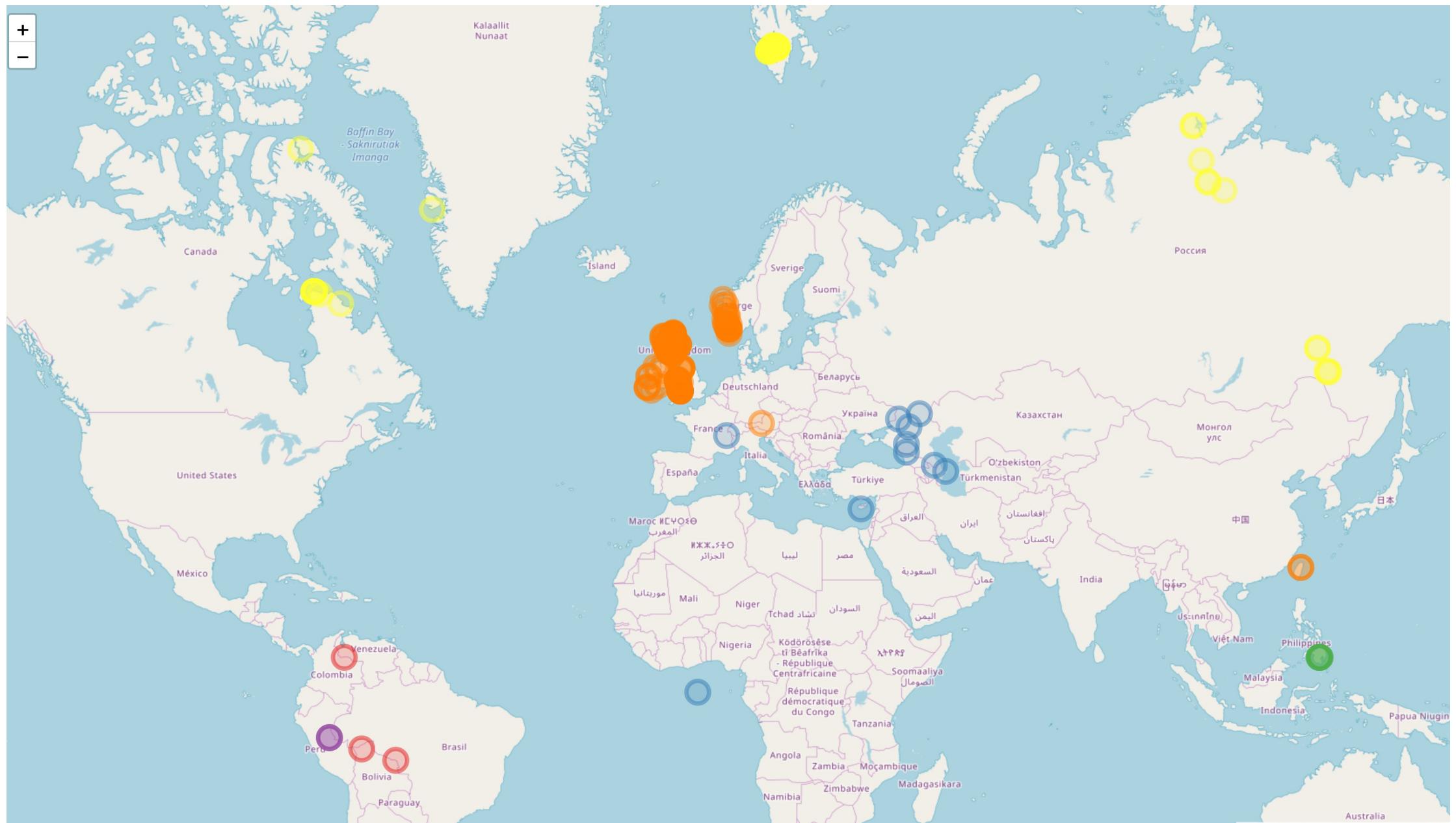
Data integration: code

Test assertions about the data

```
library(magrittr)
dat %>%
  assertr::chain_start() %>%
  # does it have more than 100 rows?
  assertr::verify(NROW(dat) > 100) %>%
  # is the key for the occurrence record unique?
  assertr::verify(assertr::is_uniq(key)) %>%
  # are there any NA's in lat/lon?
  assertr::verify(assertr::not_na(decimalLatitude)) %>%
  assertr::verify(assertr::not_na(decimalLongitude)) %>%
  assertr::chain_end() %>%
  tibble::as_tibble()
```

Data integration: code

```
mapr::map_leaflet(dat, lon = "decimalLongitude",  
    lat = "decimalLatitude")
```



Thanks!

Scott Chamberlain

 sckottie

slides: scottalks.info/dataone19

Karthik Ram

 _inundata

rOpenSci: ropensci.org

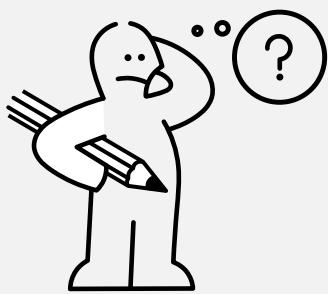
A (rough and incomplete) guide to

making your data analysis more reproducible

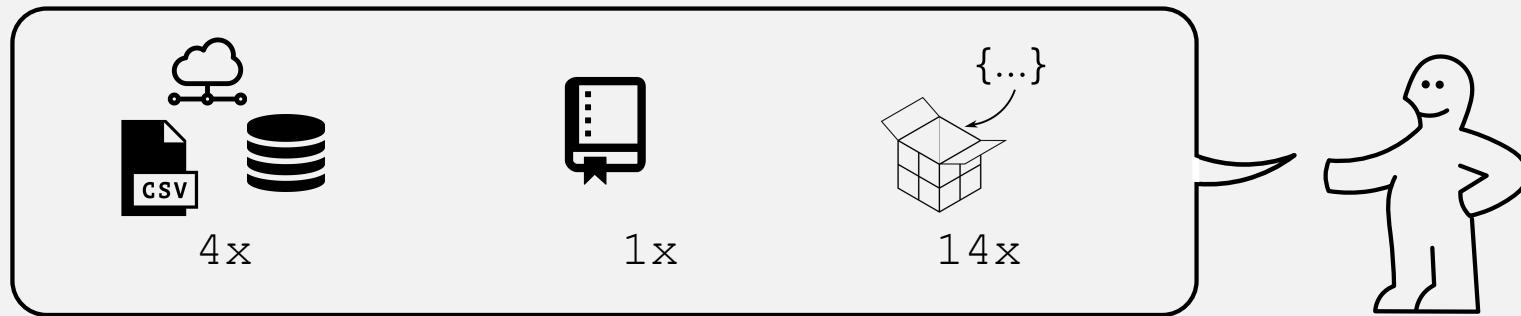
(and a bunch of other work you do)

KÖMPENDIUM

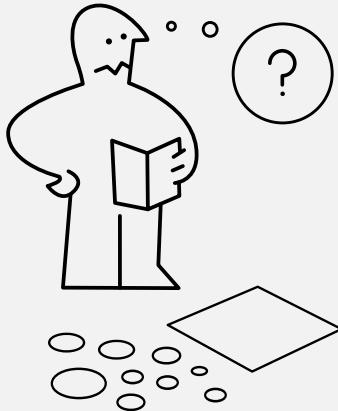
1.



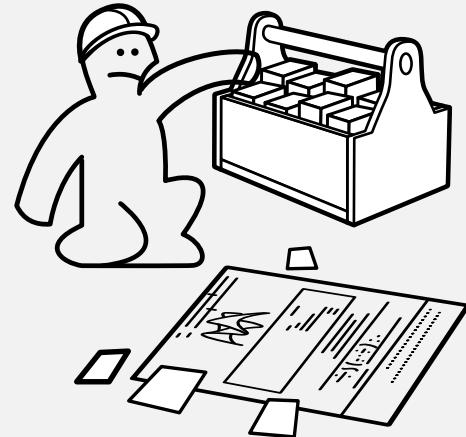
2.



3.



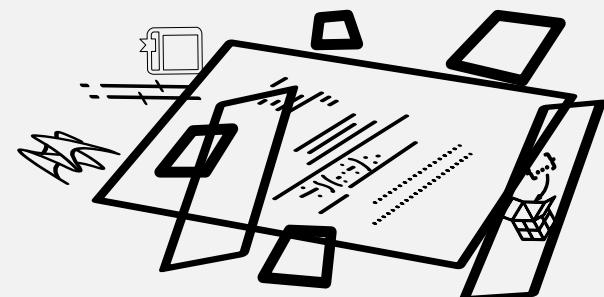
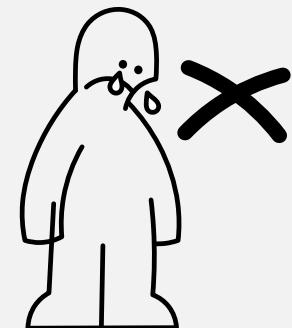
4.



5.



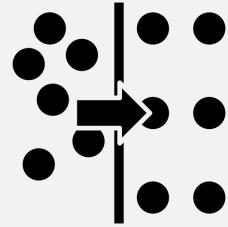
6.



Research compendia

“ . . . We introduce the concept of a **compendium** as **both a container** for the different elements that make up the **document** and its computations (i.e. **text**, **code**, **data**, . . .), and as a means for **distributing**, **managing** and **updating** the collection.

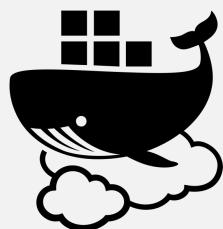
Research compendium principles



**Stick with the conventions of
your peers**

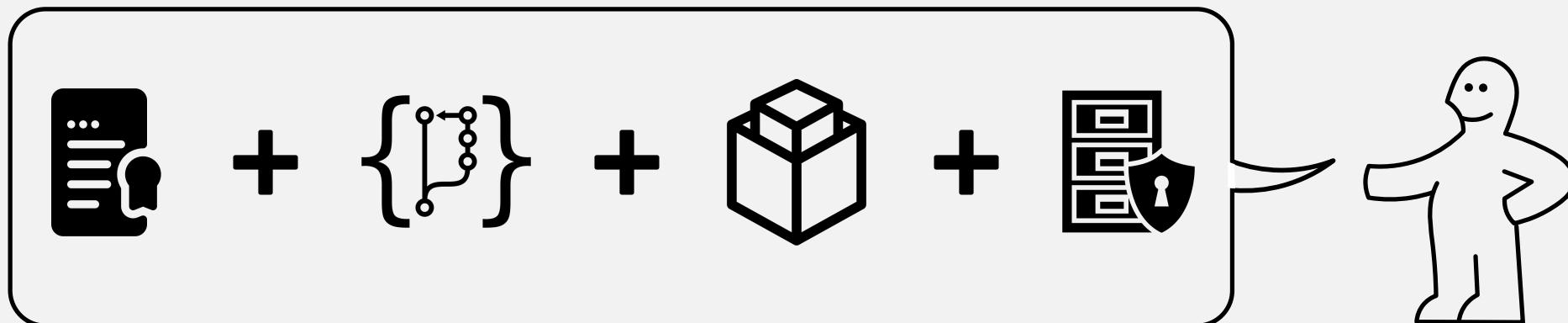


**Keep data, methods and outputs
separate**

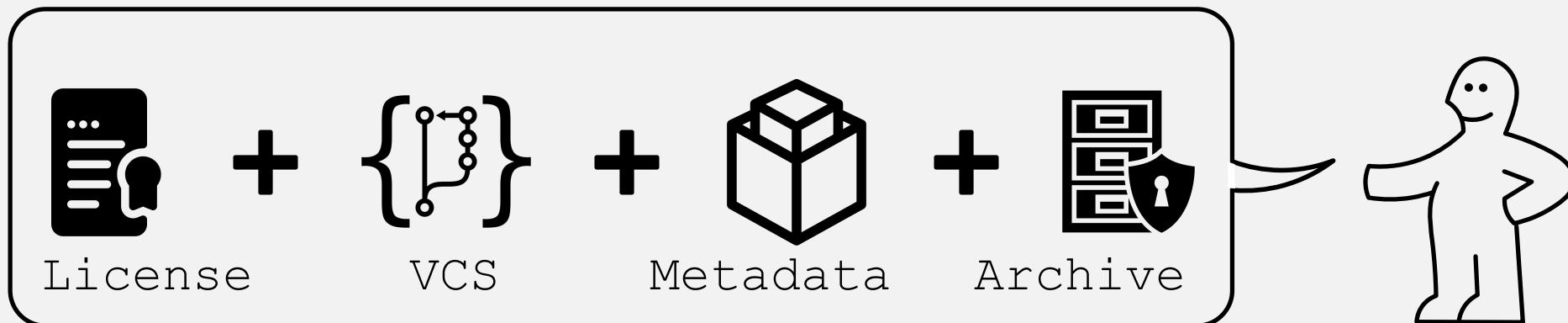


**Specify your computational
environment as clearly as you can**

Key components you'll need for sharing a compendium



Key components you'll need for sharing a compendium



**The R package structure is great
way to organize and share a
compendium!**

Package DESCRIPTION file

Package: glue

Title: Interpreted String Literals

Version: 1.3.0.9000

Authors@R: person("Jim", "Hester", email = "james.f.hester@gmail.com", role = c("aut", "cre"))

Description: An implementation of interpreted string literals, inspired by Python's Literal String Interpolation <<https://www.python.org/dev/peps/pep-0498/>> and Docstrings <<https://www.python.org/dev/peps/pep-0257/>> and Julia's Triple-Quoted String Literals <<https://docs.julialang.org/en/stable/manual/strings/#triple-quoted-string-literals>>.

Depends:

R (>= 3.1)

Imports:

methods

Suggests:

testthat,
(and many more)

License: MIT + file LICENSE

Encoding: UTF-8

LazyData: true

RoxygenNote: 6.0.1

Roxygen: list(markdown = TRUE)

URL: <https://github.com/tidyverse/glue>

BugReports: <https://github.com/tidyverse/glue/issues>

VignetteBuilder: knitr

ByteCompile: true

compendium DESCRIPTION file

Type: Compendium

Package: pomdpintro

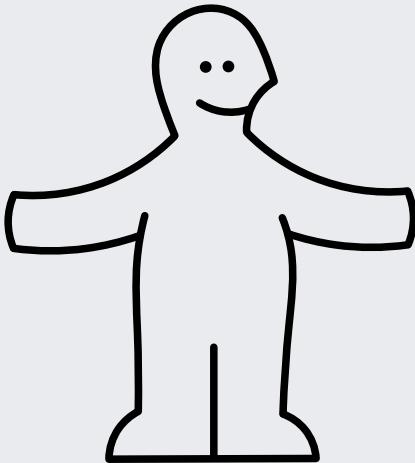
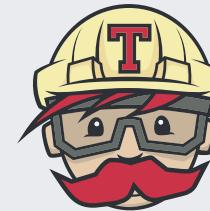
Version: 0.1.0

Depends: nimble, tidyverse, sarsop, MDPtoolbox

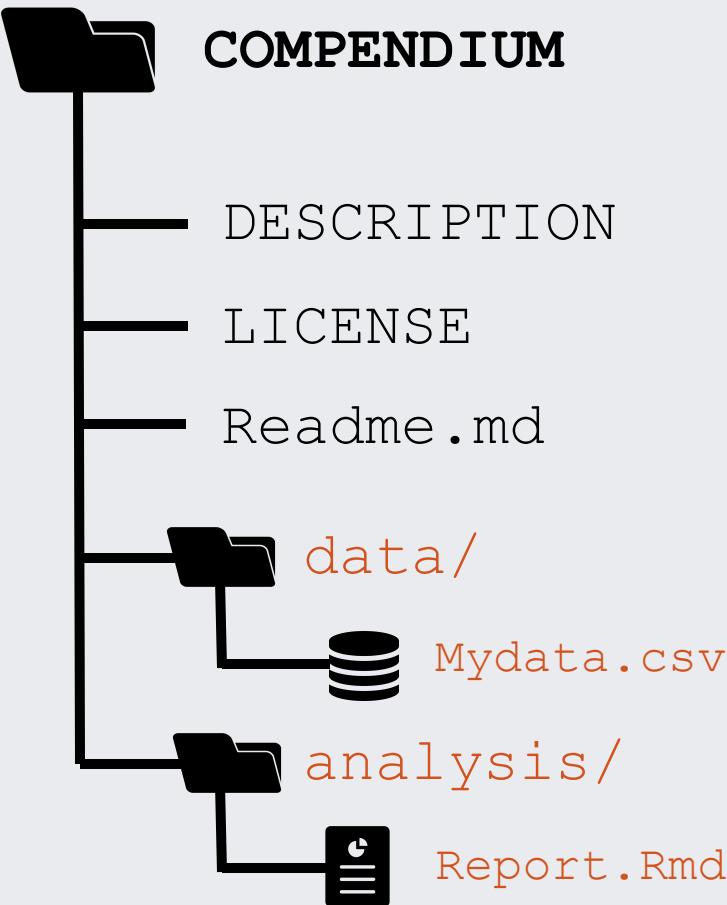
Suggests: extrafont, hrbrthemes, Cairo, ggthemes

Remotes: boettiger-lab/sarsop

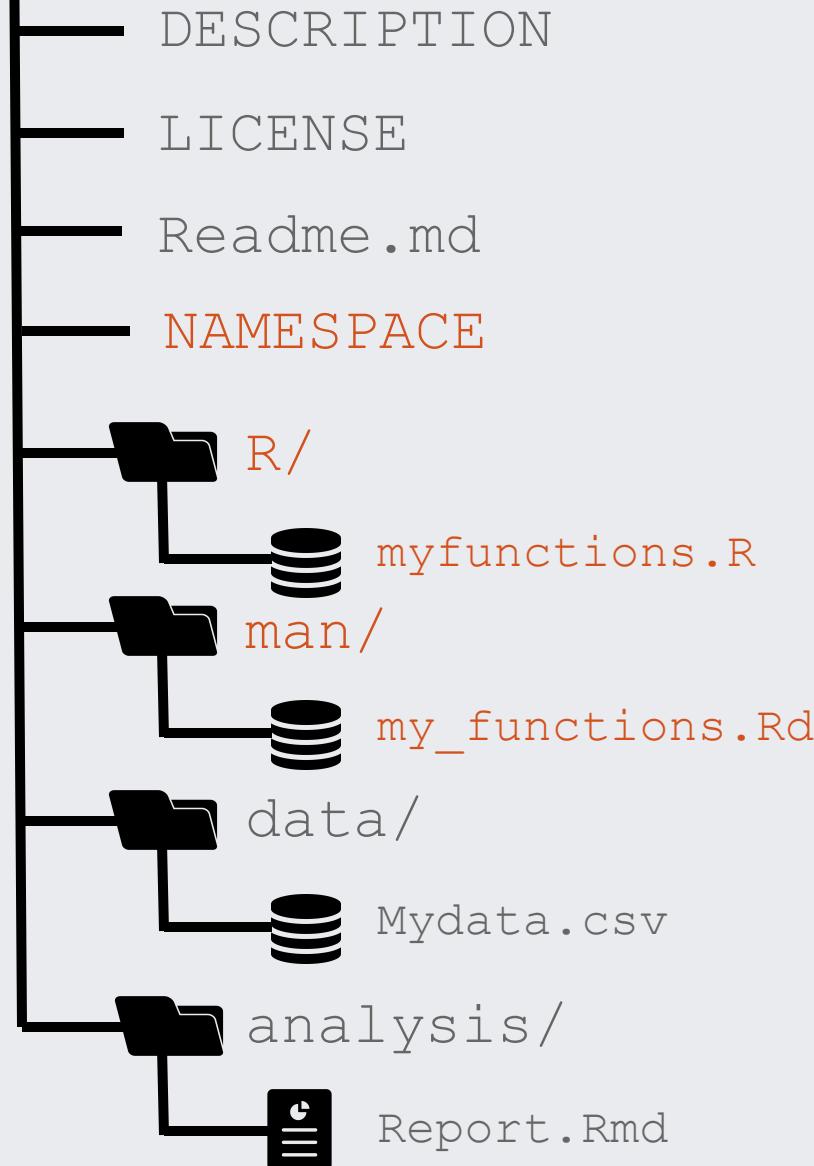
Packaging your analysis as a compendium gives you access to powerful developer tools



Small compendia

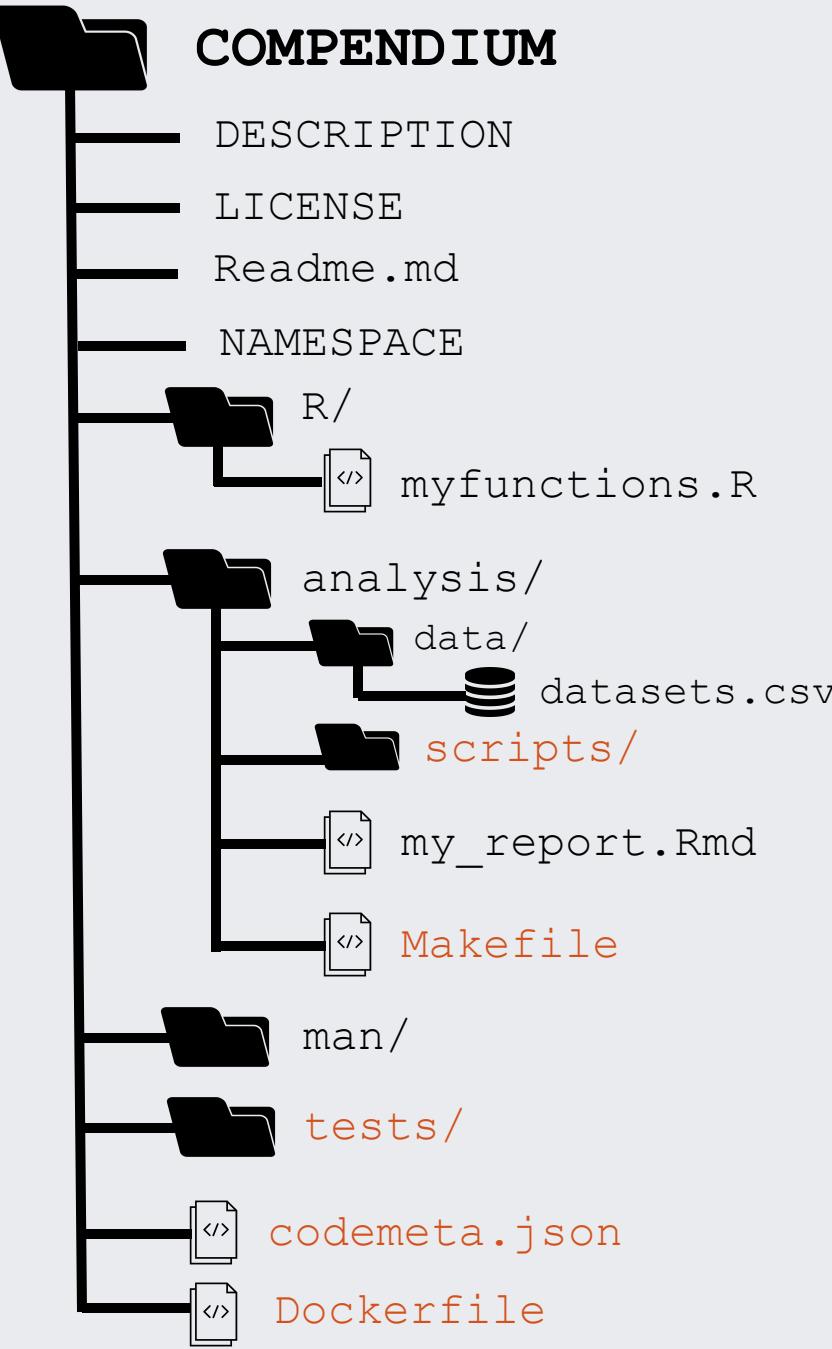


COMPENDIUM



Medium compendia

Large/complex compendia



codemeta

Add software metadata to a
repository

github.com/ropensci/codemeta

codemtar

```
{  
  "@context": [  
    "http://purl.org/codemeta/2.0",  
    "http://schema.org"  
,  
  "@type": "SoftwareSourceCode",  
  "identifier": "testthat",  
  "description": "Software testing is important, but, in part because i  
  "name": "testthat: Unit Testing for R",  
  "issueTracker": "https://github.com/r-lib/testthat/issues",  
  "datePublished": "2017-12-13 09:30:12 UTC",  
  "license": "https://spdx.org/licenses/MIT",  
  "version": "2.0.0",  
  "programmingLanguage": {  
    "@type": "ComputerLanguage",  
    "name": "R",  
    "version": "3.4.3",  
    "url": "https://r-project.org"  
  }  
}
```

Data (Small → Medium)

Computing environment

Workflows

1. Data

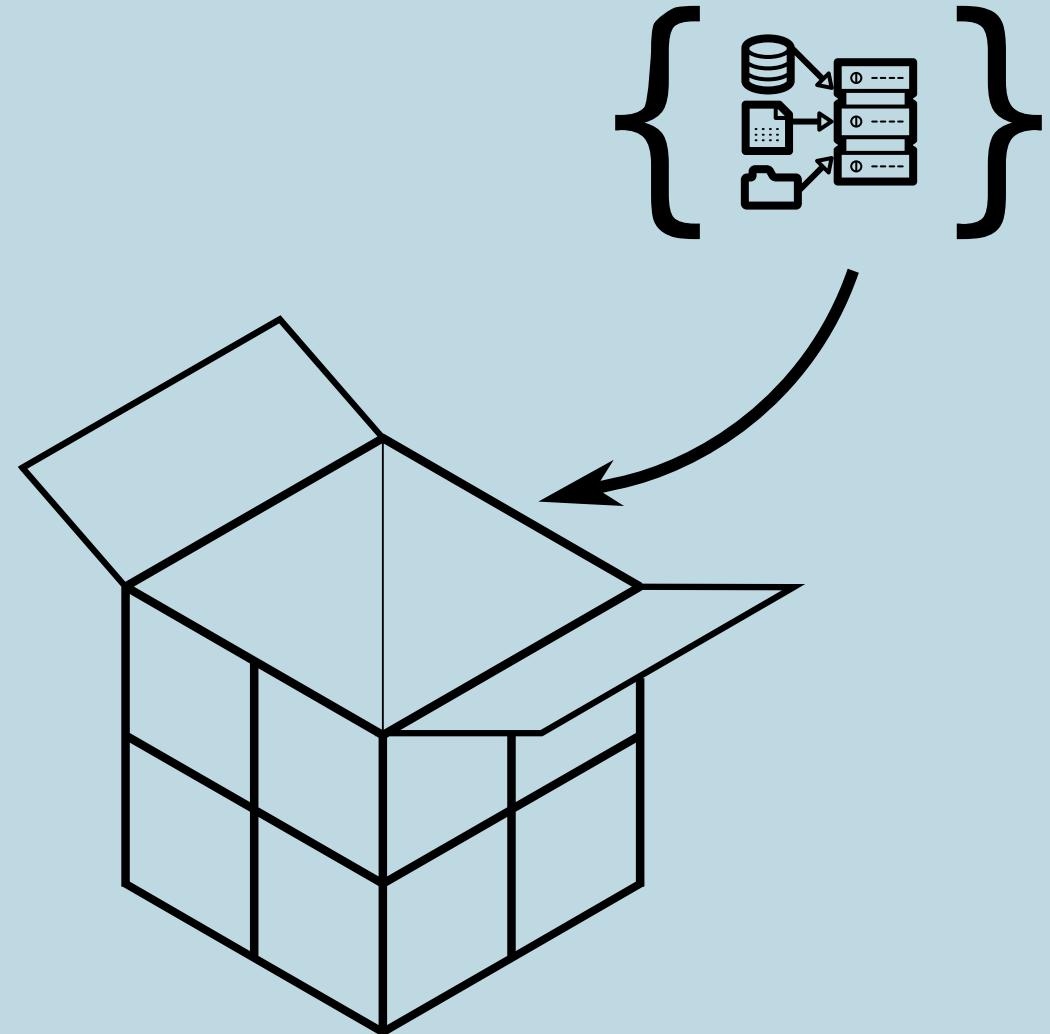
How does one manage small to
medium data in the context of a
research compendium?

Small data

Put small data inside packages,
especially if you ship a methods
package with your analysis

CRAN = < 5 mb.

37% of the 13K packages on
CRAN have some form of data.



piggyback

Attach large [data] files to
Github repositories



github.com/ropensci/piggyback

Leveraging Github releases to share medium sized files

github.com/ropensci/piggyback

```
pb_new_release("user/repo", "v0.0.5")
pb_upload("datasets.tsv.xz", "user/repo")
# Access them in your scripts with
# pb_download
```

[Releases](#)[Tags](#)[Edit release](#)[Delete](#)

Latest release

v-1.0

-o 91e9dbe

Verified

Compendium release v-1.0



karthik released this 3 minutes ago

▼ Assets 3

[mtcars.tsv.xz](#)

604 Bytes

[Source code \(zip\)](#) [Source code \(tar.gz\)](#)

Initial release of project datasets

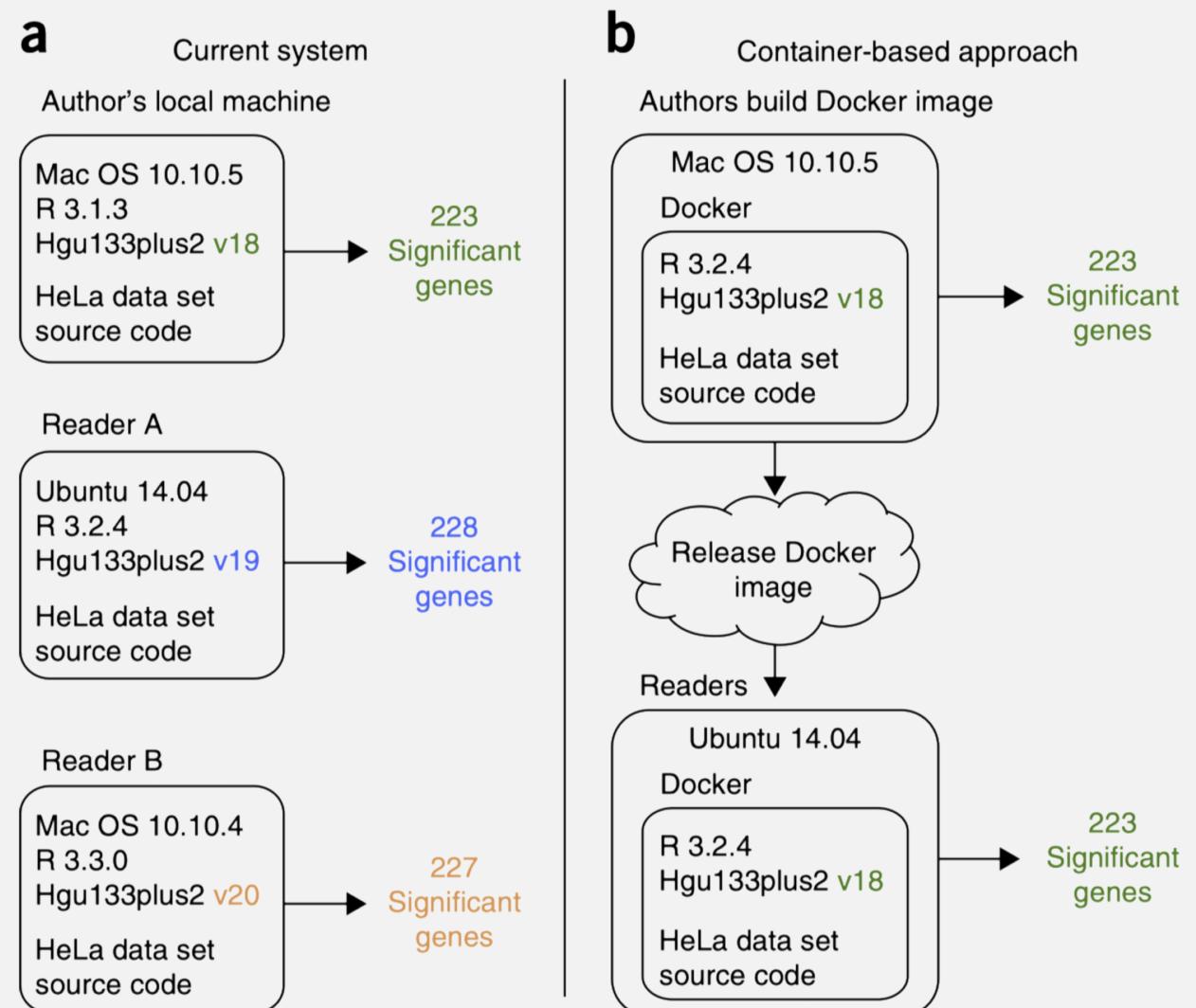
Medium data

github.com/ropensci/arkdb



2. Isolate your computing environment

It's important to isolate the computing environment so that changes in software dependencies don't break your analysis.



Adding a Dockerfile to your compendium



Many ways to write a Dockerfile for your R project



o2r/containerit
jupyter/repo2docker



Binder

mybinder.org

Binder is an open source project that is
designed to make it **really easy to share**
analyses that are in notebooks.



☰ README.md

Resolving the measurement uncertainty paradox in ecological management

[launch binder](#) [build](#) passing



- Authors: Milad Memarzadeh, [Carl Boettiger](#)

Contents

- [Manuscript](#): R Markdown source document for manuscript. Includes code to reproduce figures from tables generated by the analysis.
- [Appendix](#): R Markdown source documents for both appendices, containing all necessary R code to generate all

Loading repository: karthik/binder-test-fastest/master

Build logs



hide

```
--> 114a4cd0b227
Step 6/6 : RUN wget https://github.com/karthik/binder-test-fastest/raw/master/DESCRIPTION && R -e "devtools::install_deps()"
--> Running in 2dad32d8d3e2
--2019-01-13 02:04:34-- https://github.com/karthik/binder-test-fastest/raw/master/DESCRIPTION
Resolving github.com (github.com)... 192.30.253.112, 192.30.253.113
Connecting to github.com (github.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/karthik/binder-test-fastest/master/DESCRIPTION [following]
--2019-01-13 02:04:34-- https://raw.githubusercontent.com/karthik/binder-test-fastest/master/DESCRIPTION
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 773 [text/plain]
Saving to: 'DESCRIPTION.1'

OK                                     100% 22.7M=0s

2019-01-13 02:04:34 (22.7 MB/s) - 'DESCRIPTION.1' saved [773/773]
```

The screenshot shows the RStudio interface running in a web browser at hub.mybinder.org. The window has a Mac-style title bar with red, yellow, and green buttons, a refresh icon, and the URL.

Console pane:

```
R version 3.5.1 (2018-07-02) -- "Feather Spray"  
Copyright (C) 2018 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

Environment pane:

Global Environment

Environment is empty

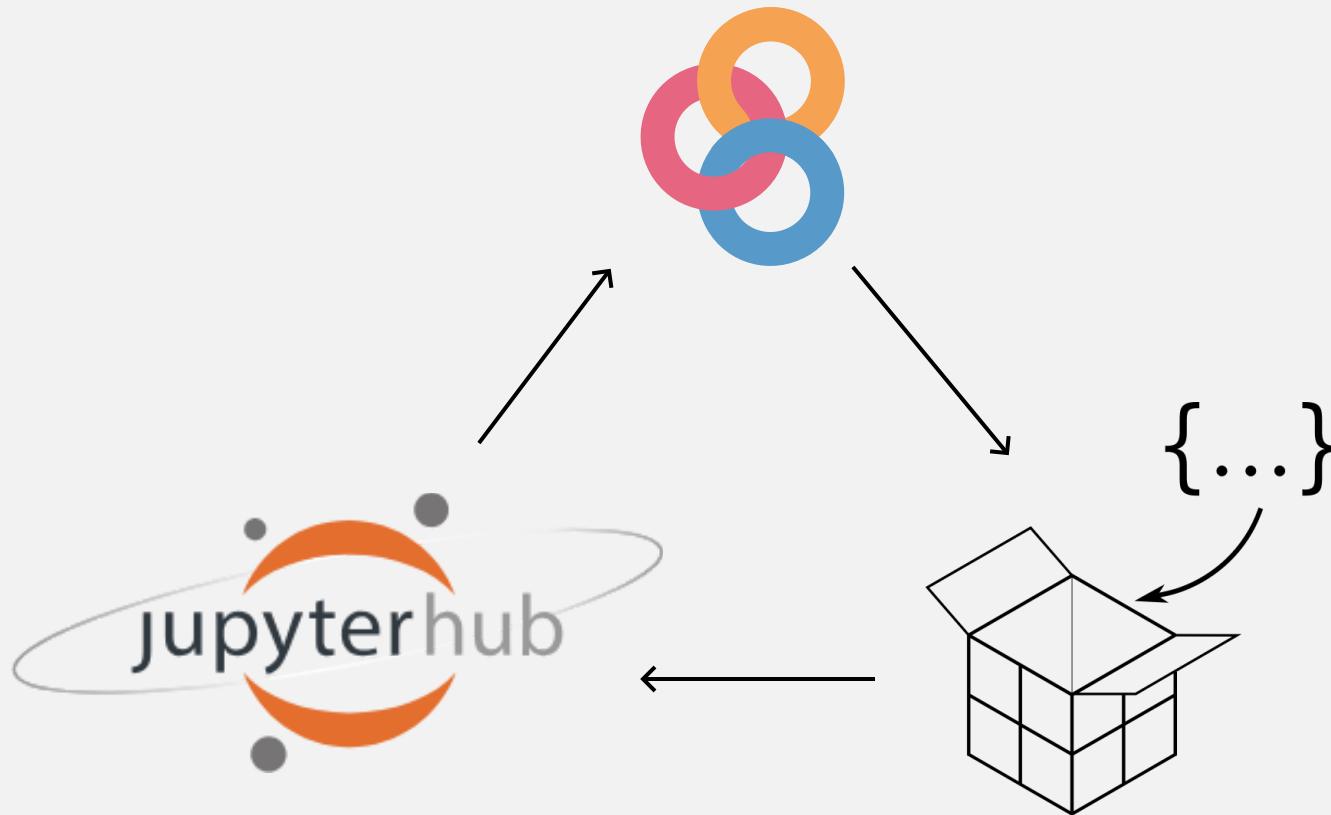
Files pane:

Name	Size	Modified
Dockerfile	248 B	Jan 8, 2019, 1:28 PM
install.R	981 B	Jan 8, 2019, 1:28 PM
kitematic		
README.md	186 B	Jan 8, 2019, 1:28 PM
tidy.R	91 B	Jan 8, 2019, 1:28 PM

Plots pane:

(No plots are currently displayed)

Git + Docker + RStudio



Setting up Binder

Branch: master ▾

New pull request



karthik Updated README

README.md

code.R

install.R

runtime.txt

r-2018-12-20

Setting up Binder

Branch: master ▾

New pull request



karthik Updated README

README.md

code.R

install.R

runtime.txt

install.packages
("ggplot2")





Build and launch a repository

GitHub repository name or URL

GitHub ▾

Git branch, tag, or commit

Path to a notebook file (optional)

File ▾

launch

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.



Copy the text below, then paste into your README to show a binder badge:  [launch](#) [binder](#)



Basic *free*

install.r

runtime.txt

apt.txt

Slow but easy
to setup.
Recommended
for beginners



launch binder

Basic

free

install.r
runtime.txt
apt.txt

Slow but easy
to setup.
Recommended
for beginners



launch binder

Premium

free

Dockerfile
install.r

Faster launch



launch binder

Basic

free

install.r
runtime.txt
apt.txt

Slow but easy
to setup.
Recommended
for beginners



launch binder

Premium

free

Dockerfile
install.r

Faster launch

Pro

free

Dockerfile
DESCRIPTION

Best for
compendia



launch binder

A fast set up binder

Dockerfile
DESCRIPTION



Pull a base image from
Rocker (e.g.
rocker:binder/latest)

rocker-project.org

The versioned stack

image	description	size
r-ver	Specify R version in docker tag. Builds on <code>debian:stable</code>	239.7MB 9 layers
rstudio	Adds rstudio	356.6MB 21 layers
tidyverse	Adds tidyverse & devtools	661.2MB 22 layers
verse	Adds tex & publishing-related packages	1GB 24 layers
geospatial	Adds geospatial libraries	1.4GB 26 layers

3. Workflow

Include a workflow to manage relationships between data output and code.

drake

general purpose workflow
manager & pipeline
toolkit for reproducibility
and high-performance
computing.



github.com/ropensci/drake

Drake: Data Frames in R for Make

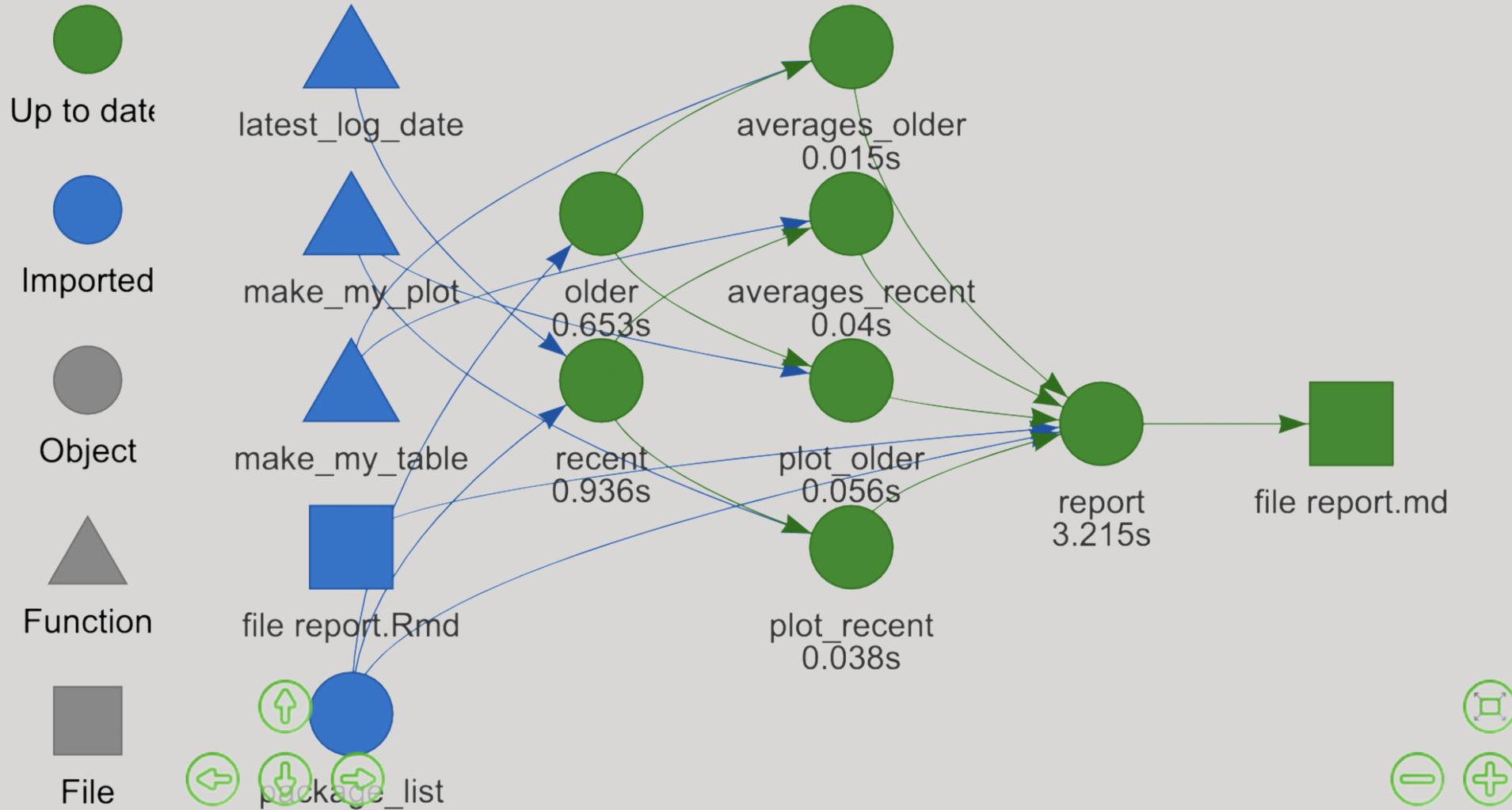
No cumbersome Makefiles

Vast arsenal of parallel computing options

Visualize dependency graph and estimate run times

Convenient organization of output.

Drake: visualize dependency graph



Take home

Leverage the R package structure
and support tools/services as
much as possible

Take home

Use modern tools to make your
compendia more accessible, but
don't forget long-term archives
and simpler formats

github.com/topics/research-compendium

data

environment

workflow

Near term

piggyback,
data packages

Binder and
friends

Drake

Long term

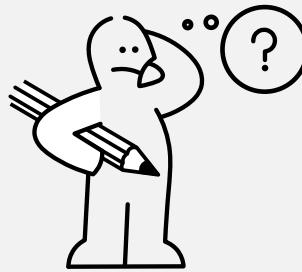
Zenodo and
friends

Dockerfile

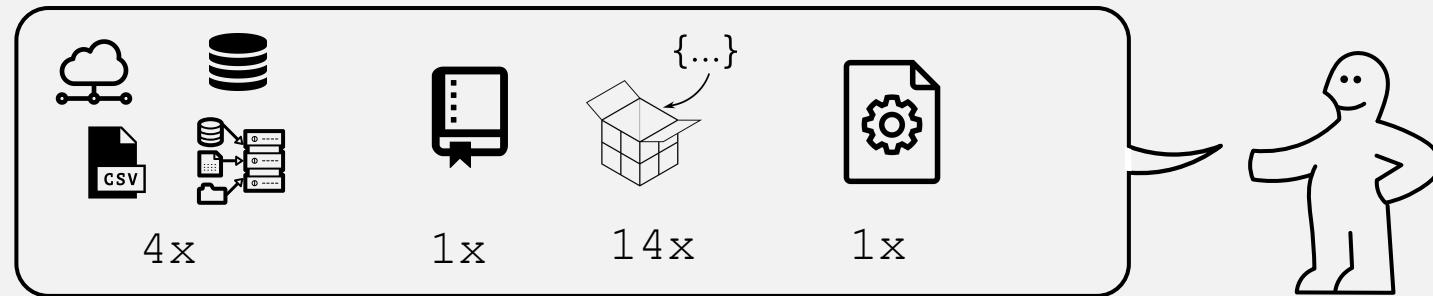
Core R tools,
Make

KÖMPENDIUM

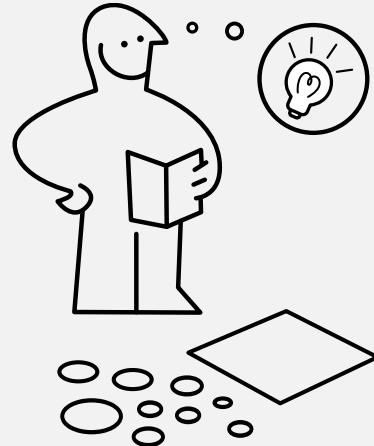
1.



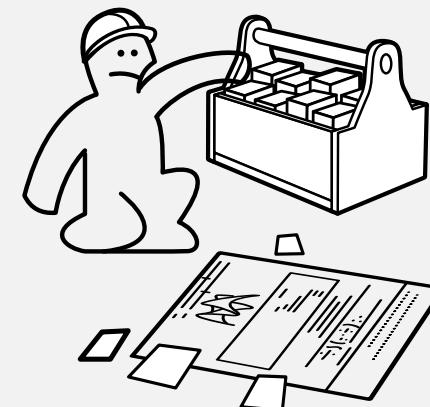
2.



3.



4.



5.

