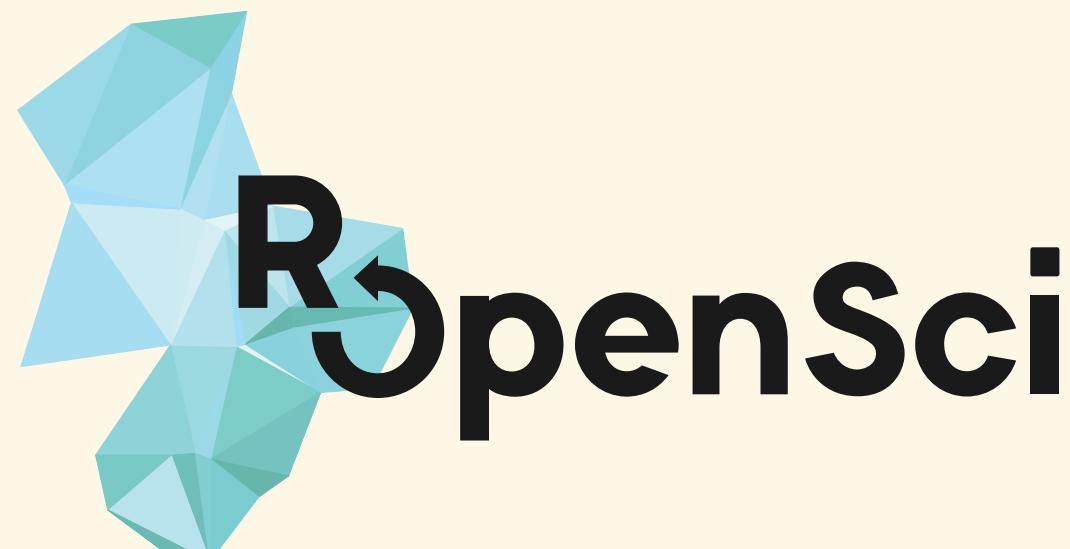


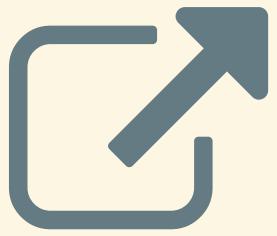
# Open Science / Research w/ R featuring rOpenSci

Scott Chamberlain (@sckottie/@ropensci)

UC Berkeley / rOpenSci



THE LEONA M. AND HARRY B.  
**HELMSLEY**  
CHARITABLE TRUST



[scottalks.info/cdc18](http://scottalks.info/cdc18)

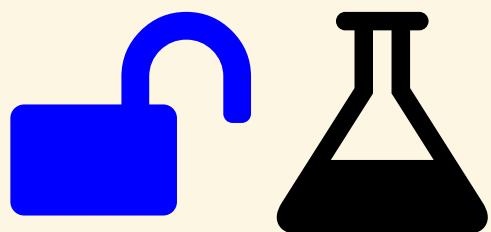
LICENSE: CC-BY 4.0

# Keyboard shortcuts: press ?

Keyboard Shortcuts

KEY	ACTION
N , SPACE	Next slide
P	Previous slide
← , H	Navigate left
→ , L	Navigate right
↑ , K	Navigate up
↓ , J	Navigate down
Home	First slide
End	Last slide

# open science/research



# Open science as a lego set



# Open science as a lego set

open science may be hard to do

but - you can work on different components

and - individual components are worth learning

# Open Data

(at least within your organization)

funders/journals often requiring this  
anyway

future self will thank you

# Versioning: code/data/text

[xkcd.com/1597](http://xkcd.com/1597)



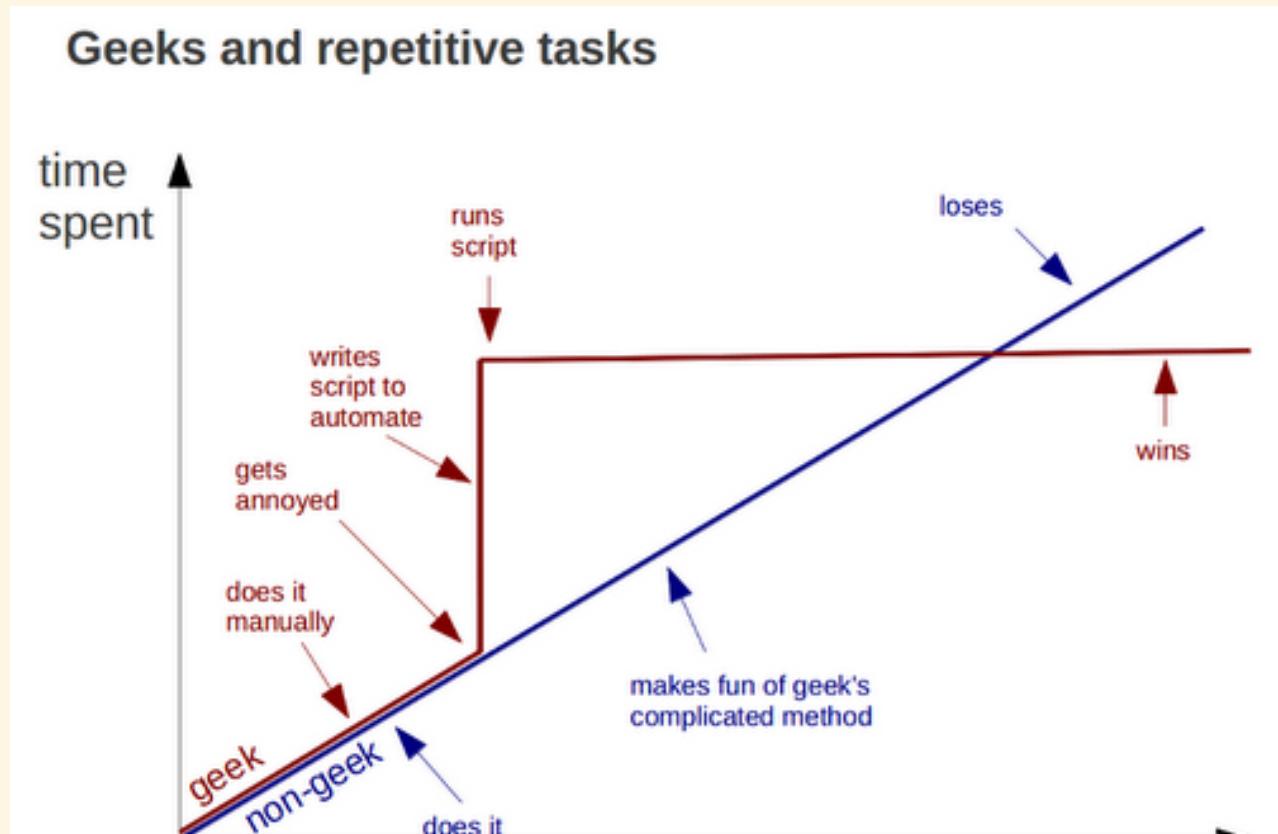
# Versioning: code/data/text

failure proofs your work

experiment freely!

makes collaboration easier

# Do all work programmatically



# Do all work programmatically

Key to reproduciblity

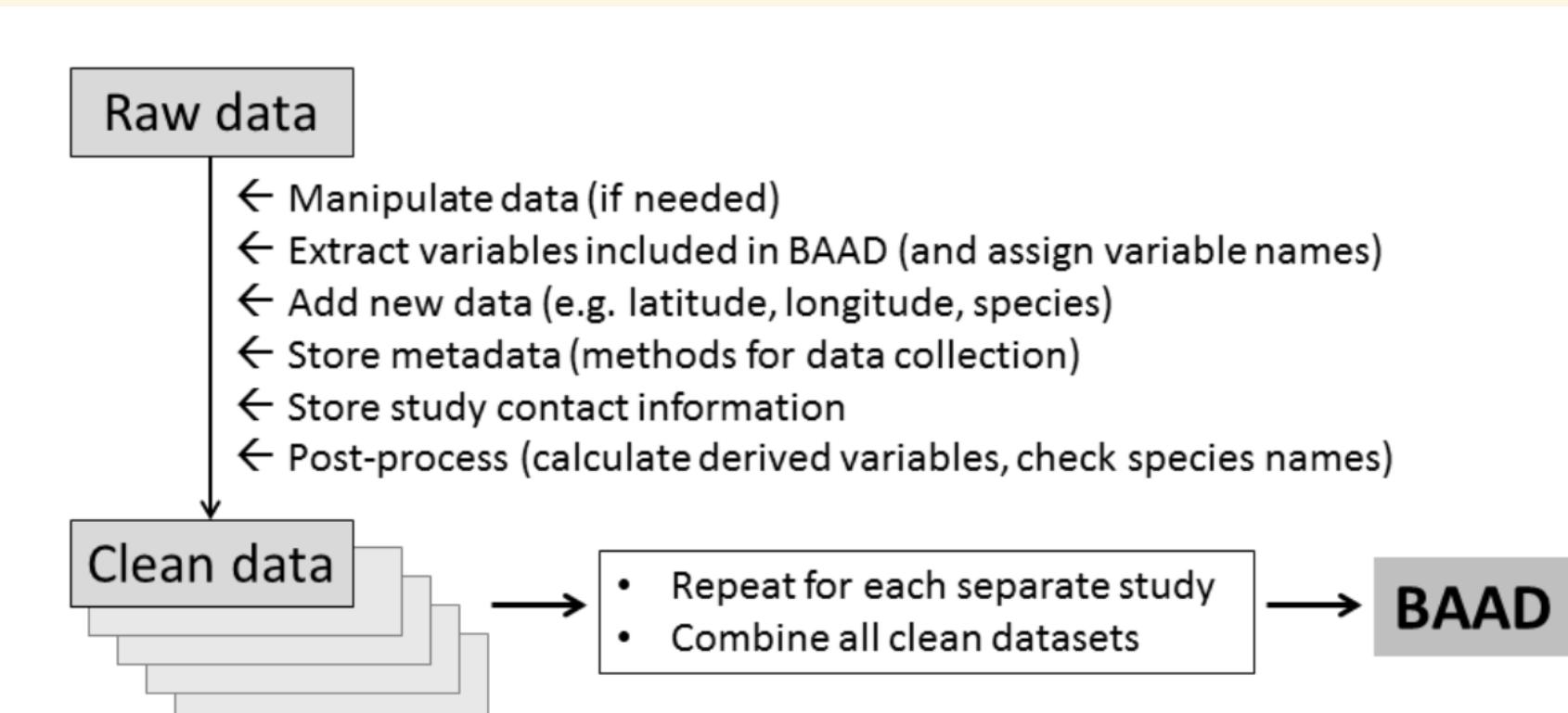
Most important person that wants to  
reproduce your work is you!

# Do all work programmatically

you and yourself

- one week from now
- two months from now
  - & so on

# An example to shoot for



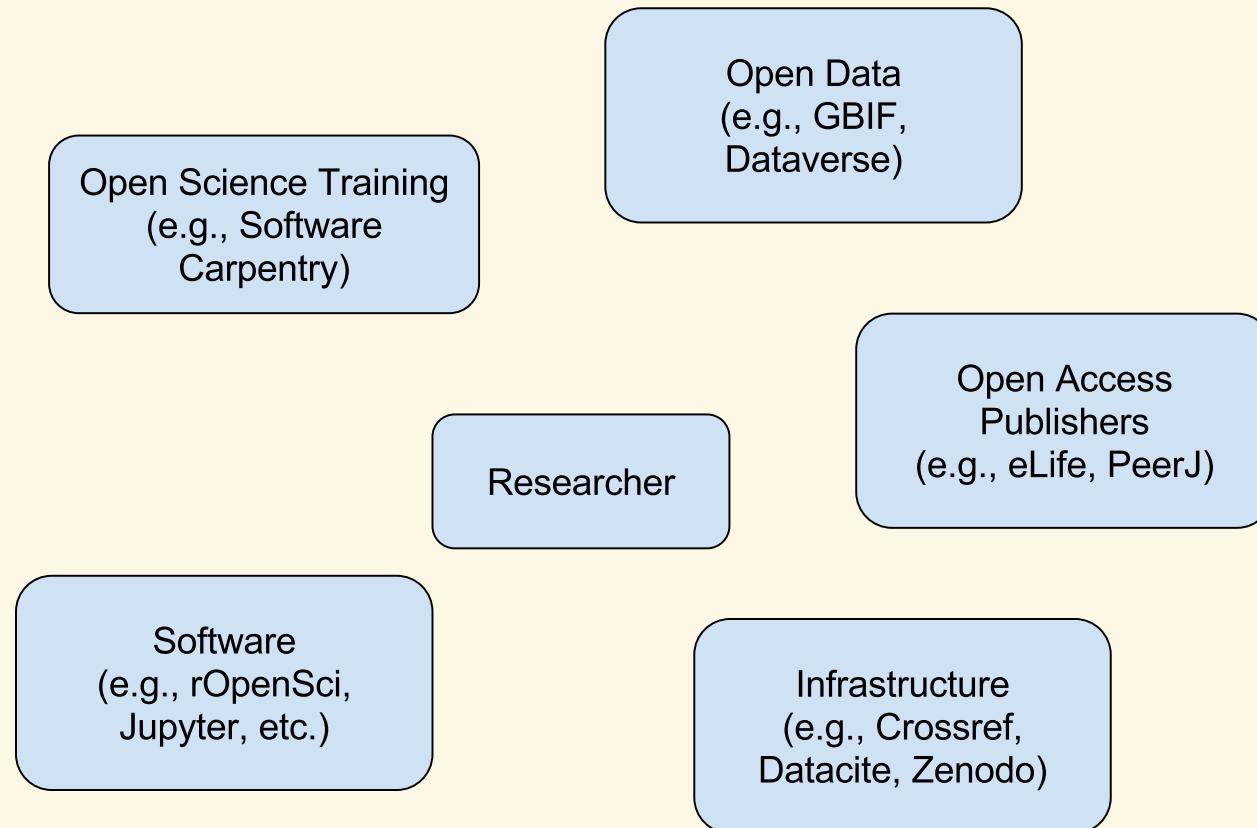
# important (higher level) scientific programming languages

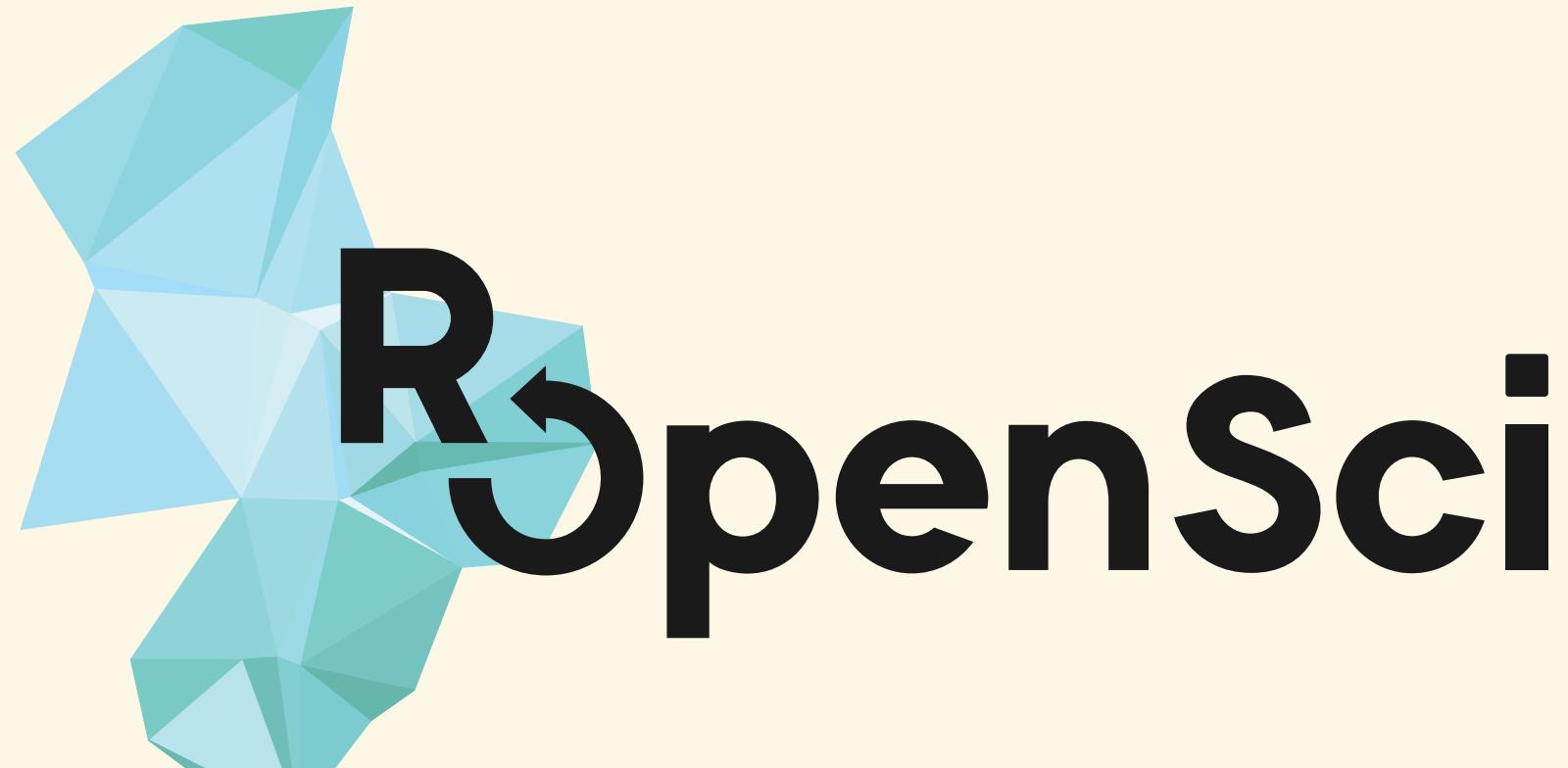


# R language

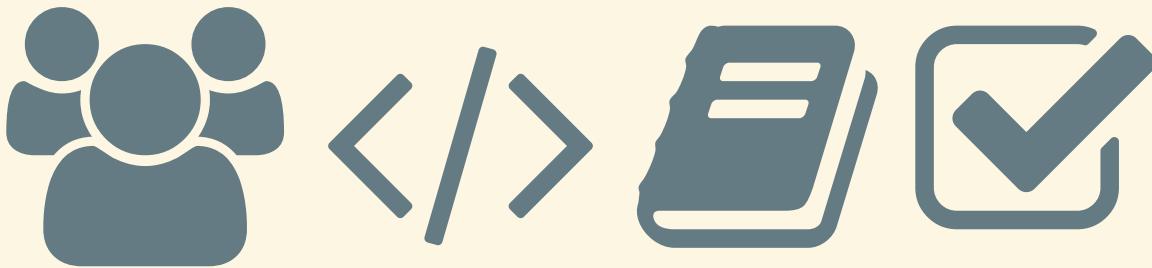
- . used widely in biology, psychology, medicine, etc.
- . rapidly growing user base, companies surrounding it
- . includes all tools for open science workflow
- . though work to be done ...

# Open science ecosystem





# rOpenSci Does



# rOpenSci Staff

[ropensci.org/about/#team](http://ropensci.org/about/#team)

- . ~5 full time
- . leadership team
- . advisory board

# Community stats

- ~ 400 code contributors
- ~ 490 Github repositories (most are R packages)
- ~ 45,000 commits
- ~ 160 published R packages on CRAN  
(another ~100 not on CRAN)

# rOpenSci Unconference

[unconf18.ropensci.org](http://unconf18.ropensci.org)

Nominations (including self) close Mar. 8th

WHO      PLAN      VENUE      SPONSORS



## rOpenSci Unconf

May 21 - 22 2018 • Seattle • WA

# What data do you use in your research?



# the research workflow

Data acquisition  +

data manipulation/analysis/viz  +

writing  +

publish 

# the research workflow

Data acquisition  +

data manipulation/analysis/viz  +

writing  +

publish 

# the research workflow

Data acquisition  +

**data manipulation/analysis/viz**   
+

writing  +

publish 

# the research workflow

Data acquisition  +

data manipulation/analysis/viz  +

writing  +

publish 

# the research workflow

Data acquisition  +

data manipulation/analysis/viz  +

writing  +

**publish** 

# rOpenSci Tools

[ropensci.org/packages](https://ropensci.org/packages)

## rOpenSci Packages

Our packages are carefully vetted, staff- and community-contributed R software tools that lower barriers to working with scientific data sources and data that support research applications on the web. Read our [blog](#) to learn how to use specific packages or contribute to their improvement. Browse our [tutorials](#) and [use cases](#).

Curious about contributing your package? See [onboarding](#) for details. We welcome [volunteers to review](#) packages submitted to our open peer review process.

**FILTERS**

All    Altmetrics    Data Publication    Tools    Visualization    Databases    Geospatial     
Web    Images Processing    Literature    Computing Infrastructure    Security    Taxonomy    CRAN /  
BIOC available

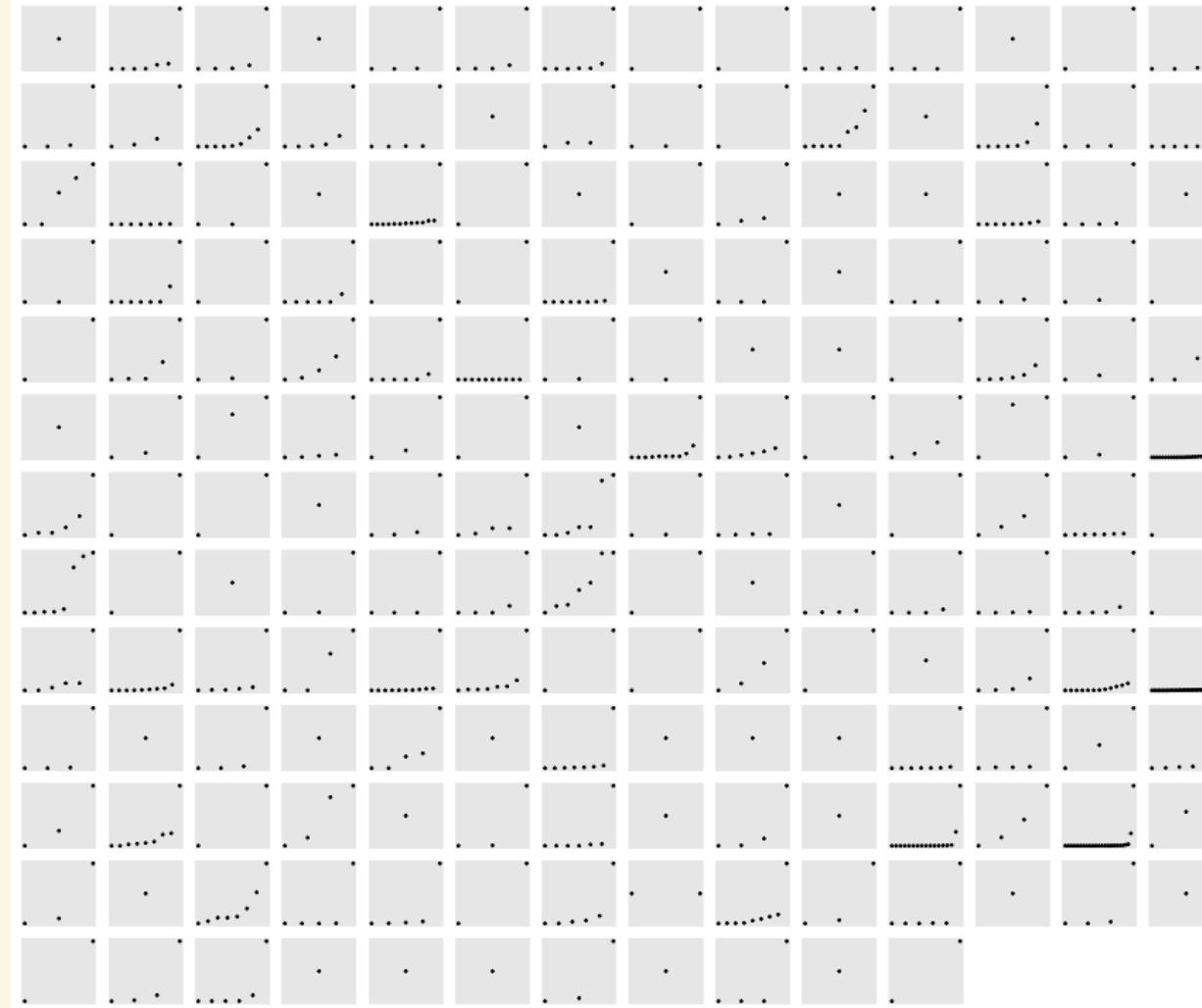
Type to search...

PACKAGE	MAINTAINER	DESCRIPTION	DETAILS
acme	Jeroen Ooms	R Client for IETF ACME Protocol	<a href="#">CRAN</a>
agent	Jeroen Ooms	Encrypted Key-Value Store for Sensitive Data	<a href="#">CRAN</a>
alm	Scott Chamberlain	R wrapper to the almetrics API platform developed by PLoS API -other publishers have built on this and work out of the box: CrossRef, Copernicus Publishers, and the Public Knowledge Project (PKP)	<a href="#">CRAN</a>
antiword	Jeroen Ooms	Extract Text from Microsoft Word Documents	<a href="#">CRAN</a>

# rOpenSci Software: some of the benefits

- reduce redundant small software efforts
- funnel effort into sustainable, well-maintained software (see lack of support for software MAINTANENCE in academia)
- bring maintainers into a community
- give otherwise isolated projects a louder voice
- hopefully we make each piece of software more sustainable

but, software sustainability is hard  
each panel is a package, each dot a person



# rOpenSci software used in

research  
within companies  
fun side projects  
journalism  
and more

here are some of the academic research uses

... usually found in methods section of papers

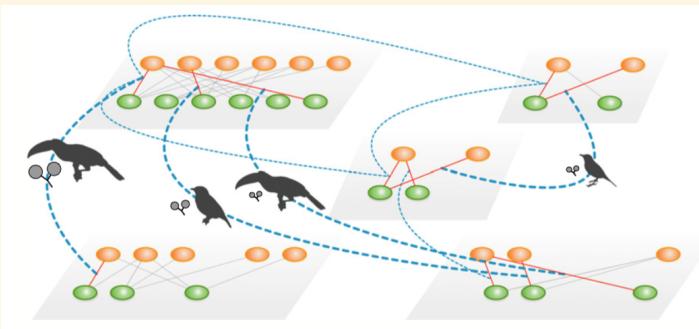
# use case 1

Claypool, K., & Patel, C. J. (2018). A transcript-wide association study in physical activity intervention implicates molecular pathways in chronic disease. BioRxiv 

*We used the **rentrez** R package to execute the query on GEO [Gene Expression Omnibus] ...*

## use case 2

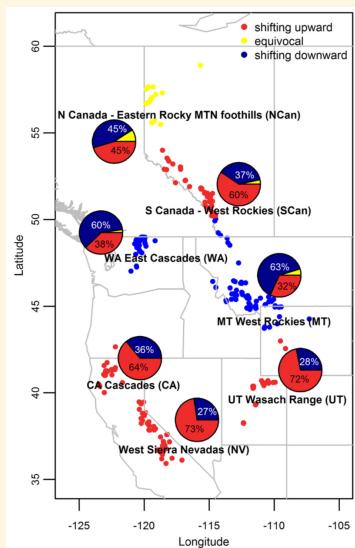
Emer, C., et al. (2018). Seed-dispersal interactions in fragmented landscapes - a metanetwork approach. Ecology Letters 



*We compiled 16 studies of BSD [bird seed dispersal]-interactions in fragments of the SE Brazilian Atlantic Forest ... We updated*

# use case 3

Harsch, M. A., & HilleRisLambers, J. (2016). Climate Warming and Seasonal Precipitation Change Interact to Limit Species Distribution Shifts across Western North America. PLOS ONE. [↗](#)



*To fill in missing elevation records and correct elevation records ... we estimated altitude ... using the GNG to SRTM data sets.*

# rOpenSci \*omics Tools

# Taxonomy

- `taxa` - Taxonomic classes and taxonomically aware data manipulation
- `taxize` - Taxonomic "toolbelt" - work w/ taxonomy web APIs
- `taxizedb` - taxize, but with local SQL databases
- `rentrez` - NCBI's Entrez services
- `biomartr` - Biomart R client
- `genbankr` - Parse GenBank files into useful objects
- `rsnps` - SNPs data retrieval

(although most omics R packages are in Bioconductor,  
rOpenSci is open to submissions!)

# Taxonomic data from >20 sources - taxize

Taxonomic hierarchies from NCBI/ITIS/COL/etc

```
library('taxize')
classification("Chironomus riparius", db = "gbif")
```

```
#> $`Chironomus riparius`
#>           name      rank      id
#> 1       Animalia kingdom      1
#> 2       Arthropoda phylum     54
#> 3       Insecta   class     216
#> 4       Diptera   order     811
#> 5 Chironomidae family    3343
#> 6       Chironomus genus 1448033
#> 7 Chironomus riparius species 1448237
```

# Taxonomic IDs

always try to move from:

- taxonomic name -- to
- taxonomic ID -- to
- whatever other data

# ENTREZ in R - `rentrez`

## Retrieve datasets from a particular organism

```
library(rentrez)
x <- entrez_search(db = "popset", term = "Latrodectus katipo[Organism]")
x_summs <- entrez_summary(db = "popset", id = x$ids)

titles <- extract_from_esummary(x_summs, "title")
COI_ids <- x$ids[c(2,6)]
COI <- entrez_fetch(db = "popset", id = COI_ids, rettype = "fasta")
substr(COI, 1, 500)
#> [1] ">AY383604.1 Latrodectus katipo isolate 1 tRNA-Leu (trnL) gene,
#> partial sequence; and NADH dehydrogenase subunit 1 (ND1) gene,
#> partial cds; mitochondrial
#> ATTACCTAAATTATCAATAATTATTAATTCAAGTTATCCCTATAATCTCTCCTAATCAGAGTCTCGT
#> TTTACACTATCTTAGAACGAAAAATTAAAGTTATATCCAAATTCTGTAAAGGGCCTAATAAAAGTGGGTTT
#> CCTTGGTATTCTCCAACCCTTAGAGACGCAATTAAACTTTCAACAAAAATCTTATTAAACCTTCTCT
#> TCTAATTTTAATTTCTACATCTCCCCGCCTATCTTAACTCTTGCTCTCCTTATTCTTCAATCA
#> TTCCTTCTTTACTTTCTCCTTATGATAACAAACATAATATCTTGATATTCTTAATCTTAT"
```

# Genomic Data Retrieval - biomartr

## Interfaces to:

- NCBI
- ENSEMBL
- ENSEMBLGENOMES
- Ensembl Biomart
- Gene Ontology

# Parsing GenBank files into semantically useful objects - `genbankr`

```
library(genbankr)
x <- system.file("sample.gbk", package="genbankr")
gb <- readGenBank(x)
#> GenBank Annotations
#> Human herpesvirus 5 strain VR1814, complete genome.

#> Accession: GU179289
#> 1 Sequence(s) with total length length: 235233
#> 174 genes
#> 170 transcripts
#> 191 exons/cds elements
#> 61 variations
#> 24 other features
genes(gb)
cds(gb)
exons(gb)
transcripts(gb)
```

# Spatial tools

# Spatial

- [geojson](#) - GeoJSON classes
- [geojsonio](#) - GeoJSON/TopoJSON input/output
- [geojsonlint](#) - lint GeoJSON
- [geoops](#) - fast C++ based GeoJSON operations
- [geonames](#) - Geonames API client
- [lawn](#) - Turf.js javascript GeoJSON operations
- [wicket](#) - Well-Known Text tools
- [wellknown](#) - WKT <-> R objects
- [rnaturalearth](#) - NaturalEarth data
- [osmplotr](#) - Open Street Maps plots
- [osmdata](#) - Open Street Maps data
- [openccg](#) - OpenCCG geocoding API

# Geospatial: Geonames data - geonames

<http://www.geonames.org/>

```
library(geonames)
```

Find a country code

```
GNcountryCode(lat = 47.03, lng = 10.2)
```

Search for nearby streets

```
GNfindNearbyStreets(lat = 37.45, lng = -122.18)
```

Search by place name

```
GNsearch(q = "london", maxRows = 10)
```

Postal code search

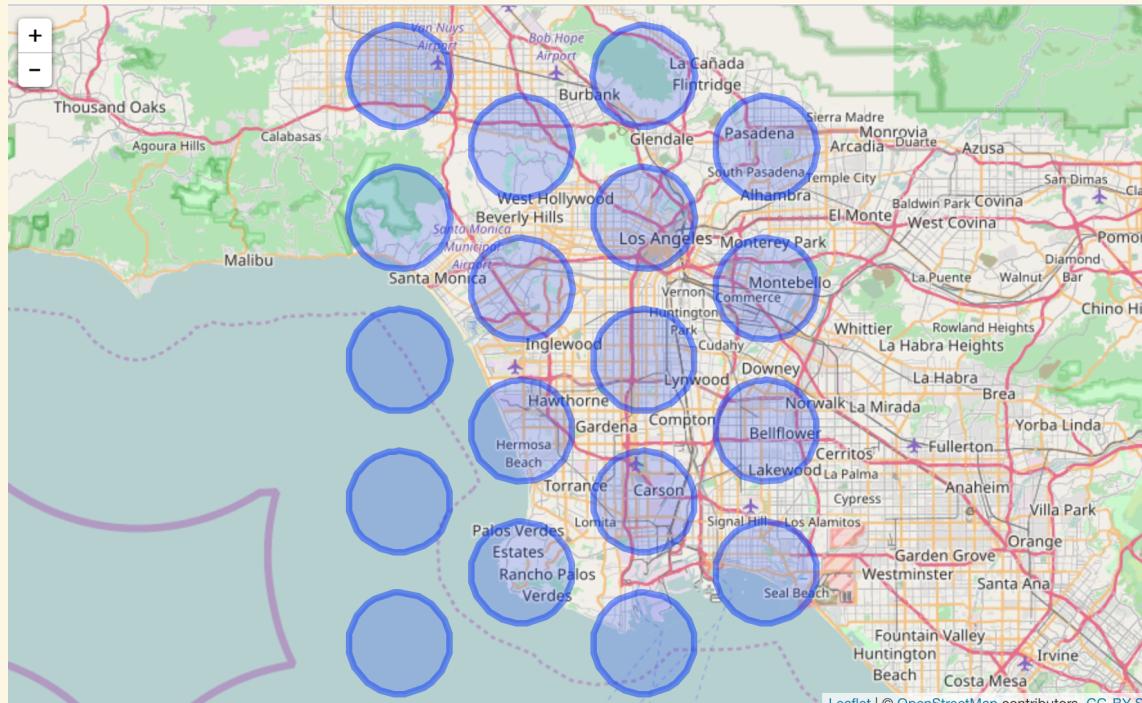
```
GNpostalCodeSearch(postalcode = 90210, country = "FI")
```

# Geospatial: conversion between data/spatial data formats - **geojsonio**

- geojson\_list - convert to GeoJSON as R list
- geojson\_json - convert to GeoJSON as JSON
- geojson\_read/geojson\_write - read/write GeoJSON from most R object types + many spatial data formats

# Geospatial: Spatial ops. w/ GeoJSON & w/o heavy dependencies - **lawn**

```
library(lawn)
bbox <- c(-118.521, 33.715, -118.145, 34.179)
lawn_hex_grid(bbox, 10, 'miles') %>%
  as_feature(hex_grid) %>%
  purrr::map(lawn_centroid) %>%
  purrr::map(lawn_circle, radius = 5) %>%
  view
```



# Climate data tools

# Climate data

- [rnoaa](#) - NOAA climate data
- [isdparsed](#) - parse NOAA Integrated Surface Data Files
- [FedData](#) - various US federal datasets (DEM's, hydrography, soil survey, climate, etc.)
- [weathercan](#) - Environment and Climate Canada data
- [getCRUCLdata](#) - CRU CL v. 2.0 Climatology Elements
- [GSODR](#) - Global Summary Daily Weather Data

# NOAA climate data - rnoaa

NCDC API



Severe weather data



Sea ice data



NOAA buoy data



Tornadoes



HOMR - Historical Observing Metadata Repository



Storm data



GHCND FTP data

# rnoaa - example

## NCDC API

```
library(rnoaa)
ncdc(datasetid = 'PRECIP_HLY', locationid = 'ZIP:28801',
      datatypeid = 'HPCP', limit = 5, token = "<YOUR_TOKEN>")
```

## GHCND FTP data

```
ids <- c("ASN0003003", "ASM00094299", "ASM00094995", "ASM00094998")
meteo_pull_monitors(ids)
```

## Tides and Currents data from COOPS

```
ops_search(station_name = 9063053, begin_date = 20150927, end_date = 20150928,
           product = "daily_mean", datum = "stnd", time_zone = "lst")
```

## Local Climatological Data

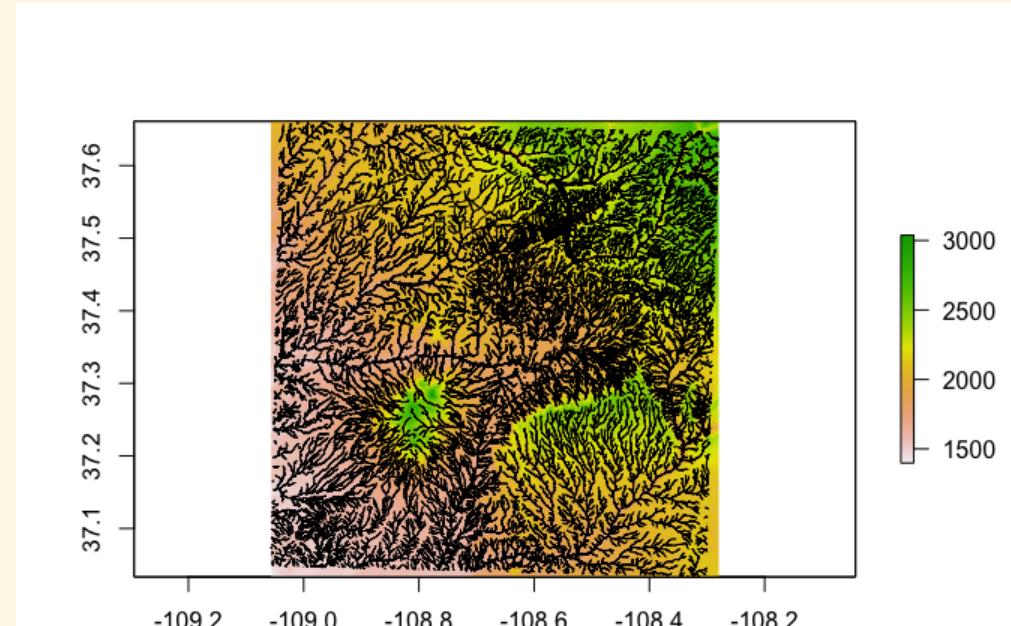
```
lcd(station = "01338099999", year = "2017")
```

# various federal datasets - FedData

plot the National Hydrography Dataset for a study area

```
library(FedData)
vepPolygon <- polygon_from_extent(
  raster::extent(672800, 740000, 4102000, 4170000),
  proj4string = "+proj=utm +datum=NAD83 +zone=12")
NED <- get_ned(template = vepPolygon, label = "VEPIIN")

NHD <- get_nhd(template = vepPolygon, label = "VEPIIN")
raster::plot(NED)
NHD %>% lapply(sp::plot, col = 'black', add = TRUE)
```



# Wrapping web APIs



# Wrapping web APIs: High level concepts

- Each pkg is a snowflake: every web API is different
- Try to cater to both beginners and power users
- Fail fast and fail well: APIs may not do it for you
- Pass on curl options! empower your users to:
  - investigate http request problems
  - set proxy options (IT often blocks certain sites/ports)
  - and more

# Defensive programming

- Fail fast
- Defend against many things
- Give users good errors

Check out my [defensive programming chapter](#)

# Basic structure of functions

DRY - don't repeat yourself

```
get_foo <- function(query, ...) {  
  cli <- crul::HttpClient$new(url = "https://foobar.com", opts = list(...))  
  res <- cli$get("/hello-world", query = list(query = query))  
  res$raise_for_status()  
  if (res$response$headers$content-type` != "application/json") stop("message")  
  jsonlite::fromJSON(res$parse("UTF-8"))  
}
```

```
#' @export  
#' @param query (character) Query terms  
#' @param ... curl options, see `curl::curl_options` for help  
#' @examples  
#' foo_bar("cellular")  
foo_bar <- function(query, ...) get_foo(query = query, ...)
```

# Example pkg wrapping web API

`ritis`: client for ITIS taxonomic data



# ritis: notes/thoughts

- imports: solrium, crul, jsonlite, data.table, tibble
- package API: fxns for REST API and Solr API
- a downside of this package possibly: a lot of functions
- return tibbles from all functions
- but `raw` JSON/XML output for those that want it
- Solr queries handled by **solrium** package

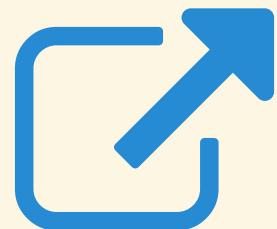
# Combining many sources into one package



# Many into one considerations

- Is it really a good idea?
- Inputs:
  - What parameters can be unified across sources?
  - Allow users to fiddle with sources specific options
    - Fail consistently across sources if possible
- Outputs:
  - What if any outputs can be combined

Many into one e.g.:  
spocc



# Many into one e.g.: spocc

- All 10 sources share common input: taxonomic names
- Pagination is similar-ish across sources (requires some source specific variable mapping)
- Geospatial search: WKT and bounding boxes then map to what source requires
- Most can toggle whether to return records that have coordinates or not
- Outputs: combine the minimum set of similar fields

# Software Review



# rOpenSci Software Review

- R package maintainer submits to [github.com/ropensci/onboarding](https://github.com/ropensci/onboarding)
- Editors determine fit or not a fit
- Editors assign reviewers
- Reviewers have ~ 3 weeks
- Reviewers and maintainer go back and forth refining pkg
- After approval, pkg moved to rOpenSci

not sure?  
pre-submission inquiry!

checkout prior presub inquiries

# What's too hard?



# talk to us



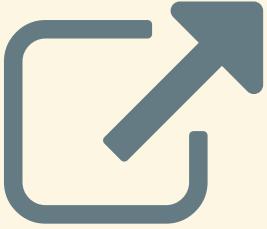
what would you like to see?

what open data is too hard to get?

discussion forum: [discuss.ropensci.org](https://discuss.ropensci.org)

submit a package/review a package:

[github.com/ropensci/onboarding](https://github.com/ropensci/onboarding)



[scottalks.info/cdc18](http://scottalks.info/cdc18)

Made w/: [reveal.js v3.2.0](#)

Some Styling: [Bootstrap v3.3.5](#)

Icons by: [FontAwesome v4.4.0](#)