

Reproducible ecology data with

R

Scott Chamberlain

@recology_ & @ropensci

Shortcuts:

M =  G = 

Interact with the talk

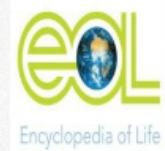
<http://bit.ly/bestalk>

Data used to be like this...



Image credits: [boat](#) and [Darwin](#)

...but now lots of accessible data on the interwebs

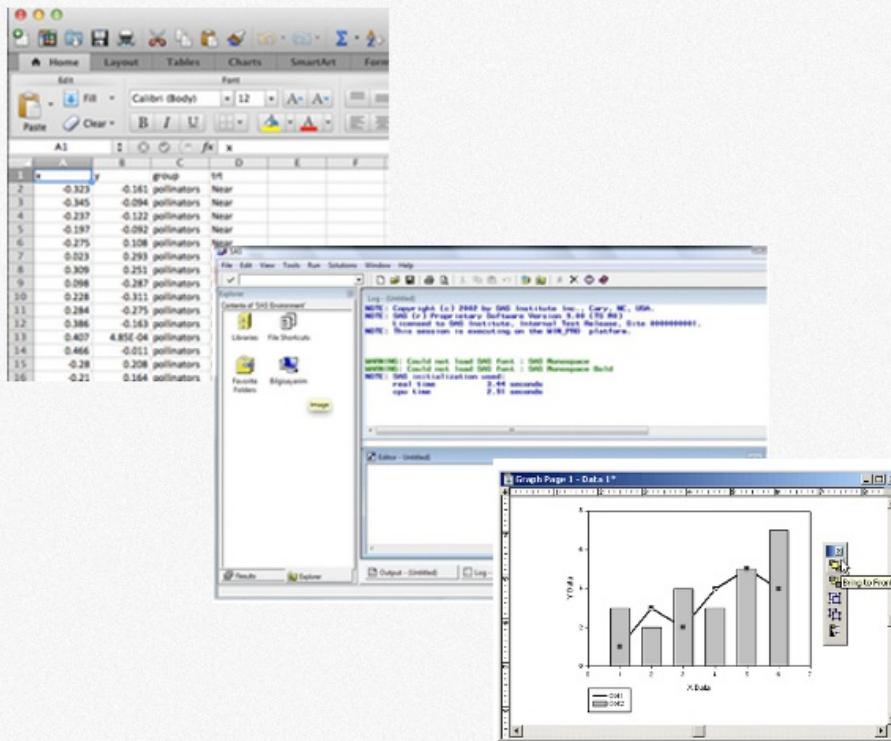


But how should we get data?

- . Programmatically**
- . Within a familiar environment to biologists**
- . Within one environment to: manipulate, visualize, analyze**

R + Data = smiles

The old way...



The new way...all in one...R in this case

Make an API call

```
library(RCurl); library(RJSONIO)
dat <- fromJSON(getURL("https://api.github.com/users/hadley/repos"))
```

Manipulate the data

```
library(plyr); library(reshape2)
dat_melt <- melt(dplyr(dat, function(x) as.data.frame(x[names(x) %in% c("name", "watchers_count",
  "forks", "open_issues")])))
```

Run some statistical model

```
lm(value ~ variable, data = dat_melt)
```

Visualize results

```
library(ggplot2)
ggplot(dat_melt, aes(name, value, colour = variable)) + geom_point() + coord_flip()
```

*Reproducible,
repeatable,
analytic workflow*

ROpenSci



Development team



Carl Boettiger



Karthik Ram



Scott Chamberlain



Edmund Hart



Advisory team



Duncan Temple Lang
UC Davis



Hadley Wickham
RStudio



JJ Allaire
RStudio



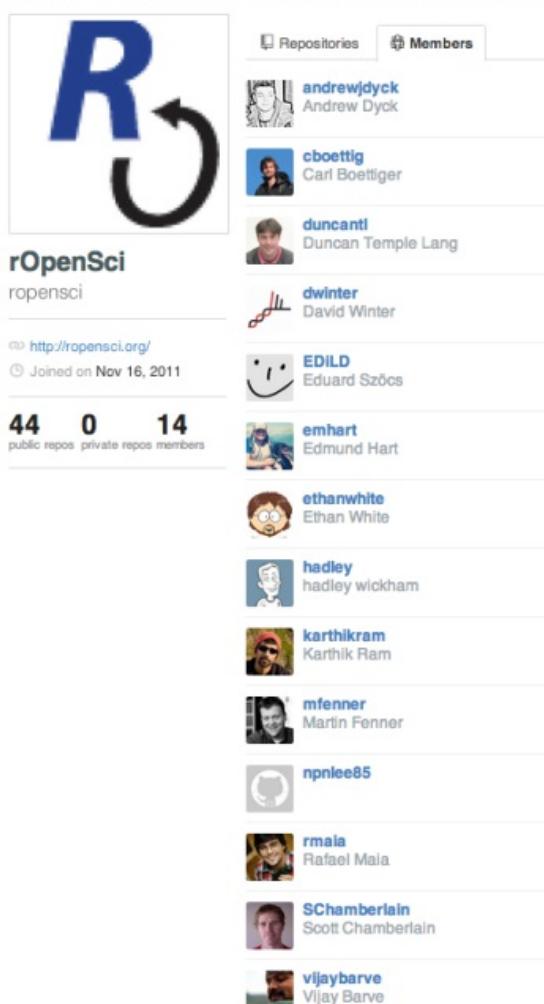
Bertram
Ludascher
UC Davis



Matt Jones
DataONE/NCEAS



But many people contribute:



The screenshot shows the GitHub organization page for `rOpenSci`. The page features a large logo consisting of a blue letter `R` with a black arrow pointing clockwise through it. Below the logo, the organization's name `rOpenSci` and handle `ropensci` are displayed. A link to the website `http://ropensci.org/` and a note indicating the account was joined on Nov 16, 2011, are also present. Key statistics shown are 44 public repos, 0 private repos, and 14 members. On the right side, there is a list of 14 members, each with a small profile picture, their GitHub handle, and their full name. The members listed are: andrewdyck (Andrew Dyck), cboettig (Carl Boettiger), duncanti (Duncan Temple Lang), dwinter (David Winter), EDILD (Eduard Szőcs), emhart (Edmund Hart), ethanwhite (Ethan White), hadley (hadley wickham), karthikram (Karthik Ram), mfenner (Martin Fenner), npnlee85 (Natalie Lee), rmaia (Rafael Maia), SChamberlain (Scott Chamberlain), and vijaybarve (Vijay Barve).

Member Handle	Full Name
andrewdyck	Andrew Dyck
cboettig	Carl Boettiger
duncanti	Duncan Temple Lang
dwinter	David Winter
EDILD	Eduard Szőcs
emhart	Edmund Hart
ethanwhite	Ethan White
hadley	hadley wickham
karthikram	Karthik Ram
mfenner	Martin Fenner
npnlee85	Natalie Lee
rmaia	Rafael Maia
SChamberlain	Scott Chamberlain
vijaybarve	Vijay Barve

rOpenSci packages

<http://ropensci.org/packages/index.html>

The screenshot shows the rOpenSci Packages website. At the top, there's a navigation bar with tabs for "Data packages" (which is selected), "Literature packages", and "Hybrid packages". Below the navigation, the page title is "rOpenSci Packages" with a subtitle "Complete list of all packages". The main content area is titled "Data Packages" and describes them as "Packages that connect to data repositories". A table lists seven packages:

Package	Description	Details	API
Dryad	Connects to the Dryad data repository	CRAN	✗
Mendeley	Programmatic interface to Mendeley Networks	CRAN	✓
ritis	Integrated Taxonomic Information Service	CRAN	✗
treebase	Programmatic interface to treebase	CRAN	✗
rfishbase	Programmatic interface to fishbase	CRAN	✗
flybase	Programmatic interface to flybase	CRAN	✗

rOpenSci packages

<https://github.com/ropensci>

 **ropensci (rOpenSci)** You are an owner of this Organization! [Edit ropensci's Profile](#)

URL <http://ropensci.org/> Member Since Nov 16, 2011

40 Public Repos **0** Private Repos **11** Members

Repositories (40)

Find a Repository...
All Repositories Public Private Sources Forks Mirrors


docs ★ 0 ⚡ 0
rOpenSci notes, etc.
Last updated an hour ago
all commits commits by owner 52 week participation


retriever Python ★ 0 ⚡ 1
Forked from weecology/retriever
Quickly download, clean up, and install ecological datasets into a database management system
Last updated 20 hours ago

Organization Members (11)

 andrewjdyck (Andrew Dyck) 10 public repos, 7 followers Conceal membership
 cboettig (Carl Boettiger) 20 public repos, 51 followers Conceal membership
 duncantl (Duncan Temple Lang) 30 public repos, 10 followers Conceal membership
 dwinter (David Winter) 6 public repos, 5 followers Conceal membership
 emhart (Edmund Hart) 8 public repos, 9 followers Conceal membership
 ethanwhite (Ethan White) 1 public repos, 7 followers Conceal membership
 hadley (hadley wickham) 59 public repos, 487 followers Conceal membership
 karthikram (Karthik Ram) 6 public repos, 16 followers Conceal membership

Others have R pkgs to get data



Others have R pkgs to get data



**But rOpenSci is a focused effort to
build bridges to data via R**

Getting data from the web into R

- Web scraping html, xml, etc.
- Reading json, csv, txt, etc.
- Hitting an Application Programming Interface (API)
 - This is the preferred, and most common method we use
 - Some require an API key, some do not

Authentication: R and APIs - API keys

API keys can be stored in a users.Rprofile file

[Click here for help on .Rprofile files](#)

```
options(MendeleyKey = "uf5daib7wyil7ag5buc")
options(MendeleyPrivateKey = "faj2os5dyd7jop2fok6")
options(PlosApiKey = "ef3vip9yak7od3hud4g")
options(SpringerMetdataKey = "ri9hi7woc6jax4vaf8w")
```

Note: These keys aren't real.

A workflow

Taxonomic Correction:

Are our species names correct? Need higher taxonomy (e.g., family names)?

Getting data:

Occurrence record, Specimen records from museums, Citizen science data

Sharing secondary data:

Sharing data increases impact of your research

And allows others to reuse (possibly you)

Taxonomic correction

Check species names - pkg taxize w/ plantminer

```
library(taxize)
```

```
# Using plantminer
```

```
plants <- c("Myrcia lingua", "Myrcia bella", "Coffea arabica", "Bleh")  
out <- plantminer(plants)
```

```
out[, !names(out) %in% c("author", "source", "source.id")]
```

	fam	genus	sp	status	confidence	suggestion	database
1	Myrtaceae	Myrcia	lingua	NA	NA	NA	Tropicos
2	Myrtaceae	Myrcia	bella	Accepted	H	NA	The Plant List
3	Rubiaceae	Coffea	arabica	Accepted	H	NA	The Plant List
4	NA	Bleh	NA	NA	NA	Baea	NA

Check species names - taxize with theplantlist.org

```
# Using theplantlist.org
splist <- c("Poa annua", "Platanus occidentalis", "Carex abrupta", "Arctostaphylos canescens",
  "Ocimum basilicum", "Vicia faba", "Quercus kelloggii", "Lactuca serriola")
out2 <- tpl_search(taxon = splist)
out2[, names(out2) %in% c("Genus", "Species", "New.Genus", "New.Species")]
```

	Genus	Species	New.Genus	New.Species
1	Poa	annua	Poa	annua
2	Platanus	occidentalis	Platanus	occidentalis
3	Carex	abrupta	Carex	abrupta
4	Arctostaphylos	canescens	Arctostaphylos	canescens
5	Ocimum	basilicum	Ocimum	basilicum
6	Vicia	faba	Vicia	faba
7	Quercus	kelloggii	Quercus	kelloggii
8	Lactuca	serriola	Lactuca	serriola

Taxonomic problems? - taxize with ITIS

```
# Using ITIS
tsn <- get_tsn("Compositae")
data.frame(getacceptednamesfromtsn(tsn))
```

```
submittedTsn acceptedName acceptedTsn
1      35421 Asteraceae    35420
```

Taxonomic problems? - taxize with ITIS

```
# Using ITIS
tsn <- get_tsn("Compositae")
data.frame(getacceptednamesfromtsn(tsn))
```

```
submittedTsn acceptedName acceptedTsn
1      35421 Asteraceae    35420
```

```
# Get names downstream
itis_downstream(tsns = 846509, downto = "Genus")
```

```
tsn parentName parentTsn  taxonName rankId rankName
1 11531        11530   Bangia   180  Genus
2 11540        11530  Porphyra   180  Genus
3 11577        11530 Porphyrella   180  Genus
4 11580        11530 Conchocelis  180  Genus
```

Get higher taxonomy names - taxize

```
library(taxize)
species <- c("Poa annua", "Abies procera", "Helianthus annuus", "Coffea arabica")
famnames <- sapply(species, tax_name, get = "family", db = "ncbi", USE.NAMES = F)
data.frame(species = species, family = famnames)
```

Get higher taxonomy names - taxize

```
library(taxize)
species <- c("Poa annua", "Abies procera", "Helianthus annuus", "Coffea arabica")
famnames <- sapply(species, tax_name, get = "family", db = "ncbi", USE.NAMES = F)
data.frame(species = species, family = famnames)
```

	species	family
1	Poa annua	Poaceae
2	Abies procera	Pinaceae
3	Helianthus annuus	Asteraceae
4	Coffea arabica	Rubiaceae

Getting data

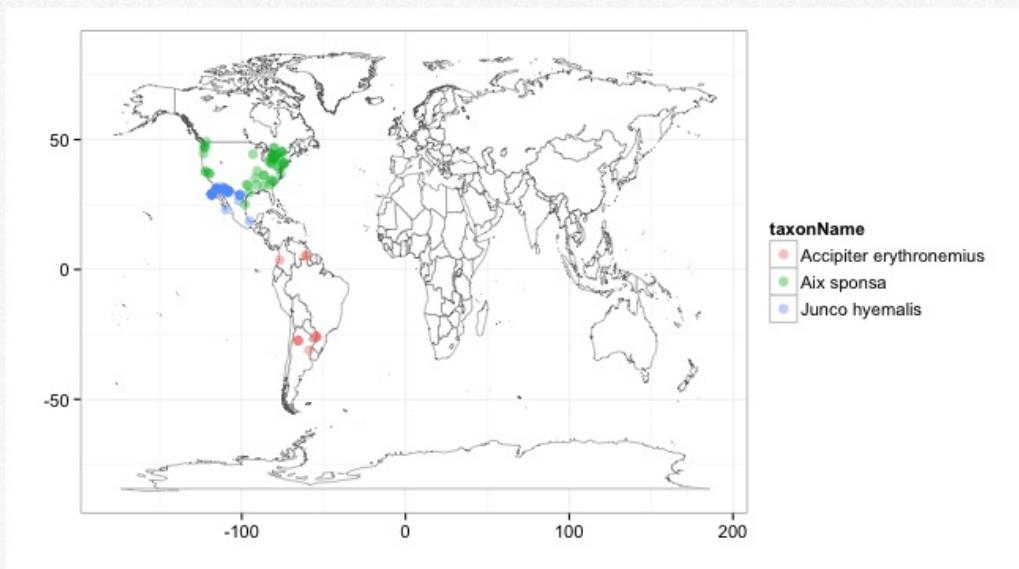
rgbif demo

Mapping biodiversity data - pkg rgbif

```
library(rgbif)
splist <- c("Accipiter erythroneurus", "Junco hyemalis", "Aix sponsa")
out <- lapply(splist, function(x) occurrence(x, coordinatestatus = T, maxresults = 50))
gbifmap(out)
```

Mapping biodiversity data - pkg rgbif

```
library(rgbif)
splist <- c("Accipiter erythonemius", "Junco hyemalis", "Aix sponsa")
out <- lapply(splist, function(x) occurrenceList(x, coordinatestatus = T, maxresults = 50))
gbifmap(out)
```



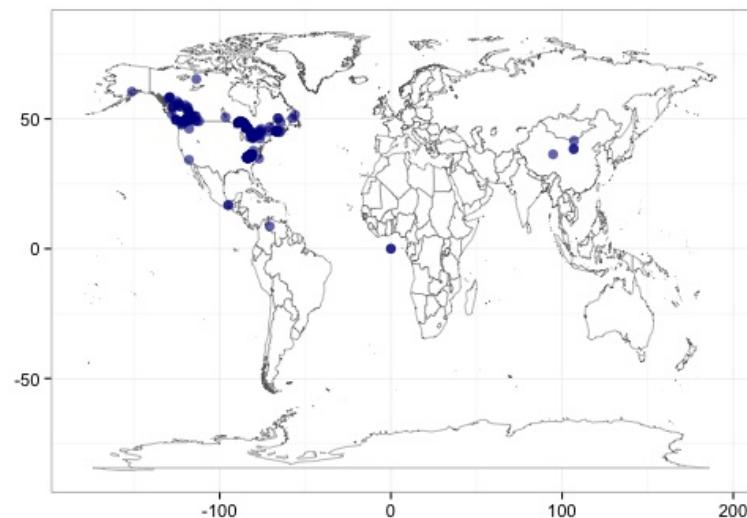
rvertnet demo

Vertebrate specimens - VertNet.org - rvertnet

```
library(rvertnet)
out <- vertoccurrence(t = "Oncorhynchus mykiss", grp = "fish", num = 300)
vertmap(input = out)
```

Vertebrate specimens - VertNet.org - rvertnet

```
library(rvertnet)
out <- vertoccurrence(t = "Oncorhynchus mykiss", grp = "fish", num = 300)
vertmap(input = out)
```



Fishbase.org - pkg rfishbase

```
library(rfishbase)
library(ggplot2)
loadCache()
reef <- which_fish("reef", "habitat", fish.data)

...code here

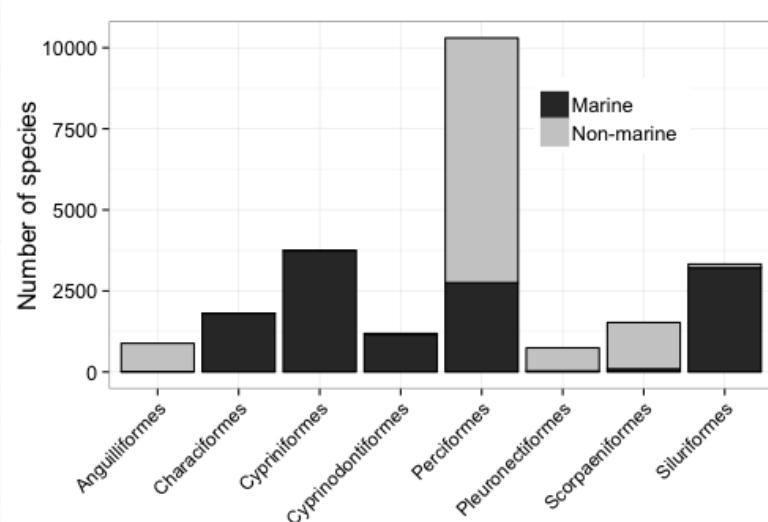
ggplot(primary_orders, aes(order, fill = marine)) + geom_bar() + # a few commands to customize appearance
geom_bar(colour = "black", show_guide = FALSE) + theme_bw(base_size = 16) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 12), legend.title = element_blank(),
        legend.justification = c(1, 0), legend.position = c(0.9, 0.6)) +
  scale_fill_grey(labels = c("Marine",
                            "Non-marine")) +
  labs(x="",y="Number of species")
```

Fishbase.org - pkg rfishbase

```
library(rfishbase)
library(ggplot2)
loadCache()
reef <- which_fish("reef", "habitat", fish.data)

...code here

ggplot(primary_orders, aes(order, fill = marine)) + geom_bar() + # a few commands to customize appearance
geom_bar(colour = "black", show_guide = FALSE) + theme_bw(base_size = 16) +
theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 12), legend.title = element_blank(),
legend.justification = c(1, 0), legend.position = c(0.9, 0.6)) +
scale_fill_grey(labels = c("Marine",
"Non-marine")) +
labs(x="",y="Number of species")
```



Openfisheries.org - pkg rfisheries

```
library(rfisheries); library(ggplot2)
countries <- country_codes()

# let's take a small subset, say 5 random countries
c_list <- countries[sample(nrow(countries), 5), ]$iso3c

# and grab landings data for these countries
results <- lapply(c_list, function(x) {
  df <- landings(country = x)
  df$country <- x
  df
}), .progress = "text")

ggplot(results, aes(year, catch, group = country, color = country)) + geom_line()
```

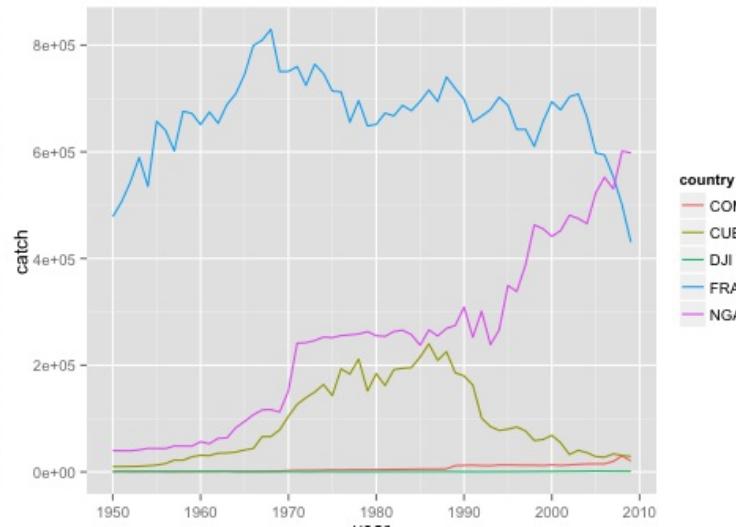
Openfisheries.org - pkg rfisheries

```
library(rfisheries); library(ggplot2)
countries <- country_codes()

# let's take a small subset, say 5 random countries
c_list <- countries[sample(nrow(countries), 5), ]$iso3c

# and grab landings data for these countries
results <- lapply(c_list, function(x) {
  df <- landings(country = x)
  df$country <- x
  df
}), .progress = "text")

ggplot(results, aes(year, catch, group = country, color = country)) + geom_line()
```



Get phylogenies Treebase.org - pkg treebase

```
library(treebase); library(ggplot2); library(reshape2)

data(treebase)
have <- have_branchlength(treebase)
branchlengths <- treebase[have]
dat <- data.frame(tips = sapply(branchlengths, Ntip), nodes = sapply(branchlengths, Nnode))
dat_m <- melt(dat)
```

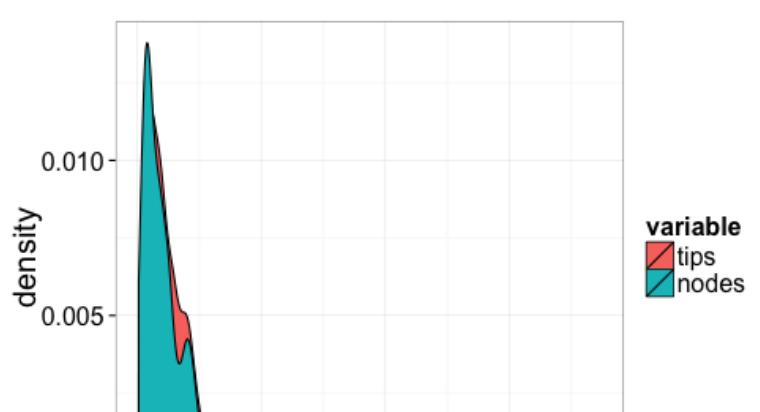
```
ggplot(dat_m[dat_m$value < 1000, ], aes(value, fill = variable)) + theme_bw(base_size = 20) +
  geom_density()
```

Get phylogenies Treebase.org - pkg treebase

```
library(treebase); library(ggplot2); library(reshape2)

data(treebase)
have <- have_branchlength(treebase)
branchlengths <- treebase[have]
dat <- data.frame(tips = sapply(branchlengths, Ntip), nodes = sapply(branchlengths, Nnode))
dat_m <- melt(dat)
```

```
ggplot(dat_m[dat_m$value < 1000, ], aes(value, fill = variable)) + theme_bw(base_size = 20) +
  geom_density()
```



Sharing data

Sharing and using data - pkg rfigshare

Using Figshare's new API, it is now possible to share figures, data, and any other object generated in R directly to one's figshare account. After authenticating:

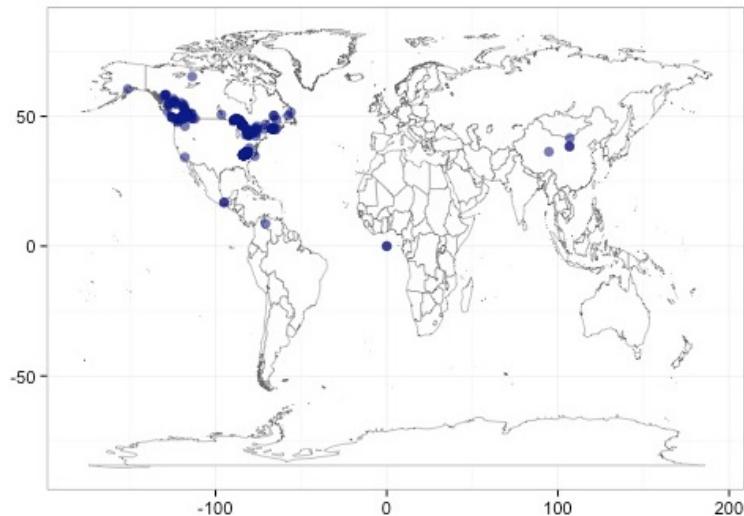
```
library(rfigshare)

# Try creating your own content:
id <- fs_new_article(title = "A Test of rfigshare", description = "testing",
                      type = "figure", authors = "Scott Chamberlain", tags = c("ecology", "openscience"),
                      categories = "Ecology", links = "http://ropensci.org", files = "figure/rvertnet.png")
```

Sharing and using data - pkg rfigshare

A Test of rfigshare

[Edit article](#)



[Preview](#)

[Download](#)

0
views

0
shares

cites
coming
soon

Categories

- [Ecology](#)

Authors

[Scott Chamberlain](#)

Tags

- [ecology](#)
- [openscience](#)

Export

- [Export to Ref. Manager](#)
- [Export to Endnote](#)
- [Export to Mendeley](#)

Share this:

No sharing options for unpublished data



Sharing and using data - (rfigshare)

The screenshot shows the figshare website interface. At the top, there is a logo with a colorful circular icon followed by the word "figshare". Below the logo is a search bar with the placeholder "My data" and a magnifying glass icon. To the right of the search bar is a red "Browse" button. The main content area features a large banner with the text "automatic analysis and publishing of data". Below the banner, there is a video player window. The video player displays a computer screen showing a file explorer window with many files listed, a browser window displaying a dashboard with various charts and graphs, and a sidebar with user statistics (1 article, 6 points). The video player has a play button in the center, and the progress bar shows it is at 02:31 of a 02:49 video.

rOpenSci tutorials

The screenshot shows a web browser displaying the [rOpenSci tutorials](http://ropensci.org/tutorials/) website. The URL is visible in the address bar. The navigation bar includes links for Ruby, Univ, J's, Lit, Email, Bills, Misc., Nets, NPR, Profs, Blogs, GistDeck, + Pocket, and Bikes. Below the navigation bar is a blue header with links for HOME, ABOUT, PACKAGES, RESOURCES, CONTRIBUTE, and CONTACT.

Tutorials

Treebase

- ⚲ Basic tree and metadata queries
- ⚲ Replicating results
- ⚲ Meta-Analysis

Taxize

- ⚲ Using the Phylomatic API
- ⚲ Search against the Integrated Taxonomic Information Service (ITIS)
- Using ITIS and Phylomatic together
- Performing taxonomic name resolution using the Taxonomic Name Resolution Service (TNRS)
- Performing taxonomic name resolution using the Global Names Resolver (GNR)
- Searching taxonomic names in Tropics
- Searching taxonomic names in uBio

ropensci.org

Please contact us if you have feedback or ideas for collaborations!

Code open-sourced on [GitHub](#)
Elsewhere [@ropensci](#) and [G+](#)

And I'm [@recology_](#)