Introduction to the Tidyverse

How to be a tidy data scientist

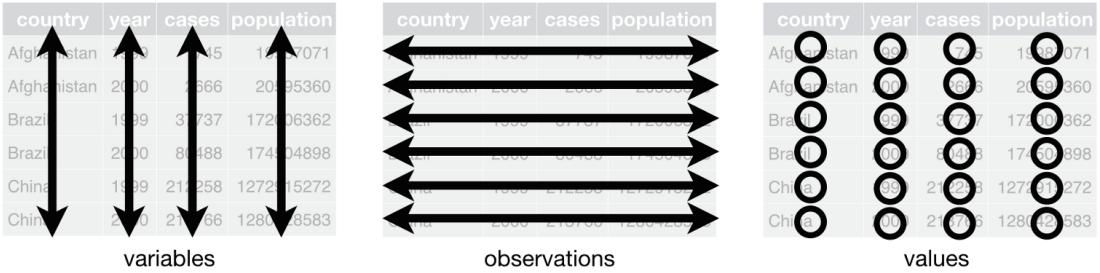
Olivier Gimenez
2019-01-14 (updated: 2019-01-24)

Tidyverse

- **Ordocosme** in 🇫🇷 with *Tidy* for "bien rangé" and *verse* for "univers"
- A collection of R 📦 developed by H. Wickham and others at Rstudio

Tidyverse

- "A framework for managing data that aims at making the cleaning and preparing steps [muuuuuuuch] easier" (Julien Barnier).
- Main characteristics of a tidy dataset:
  - each variable is a column
  - each observation is a raw
  - each value is in a different cell



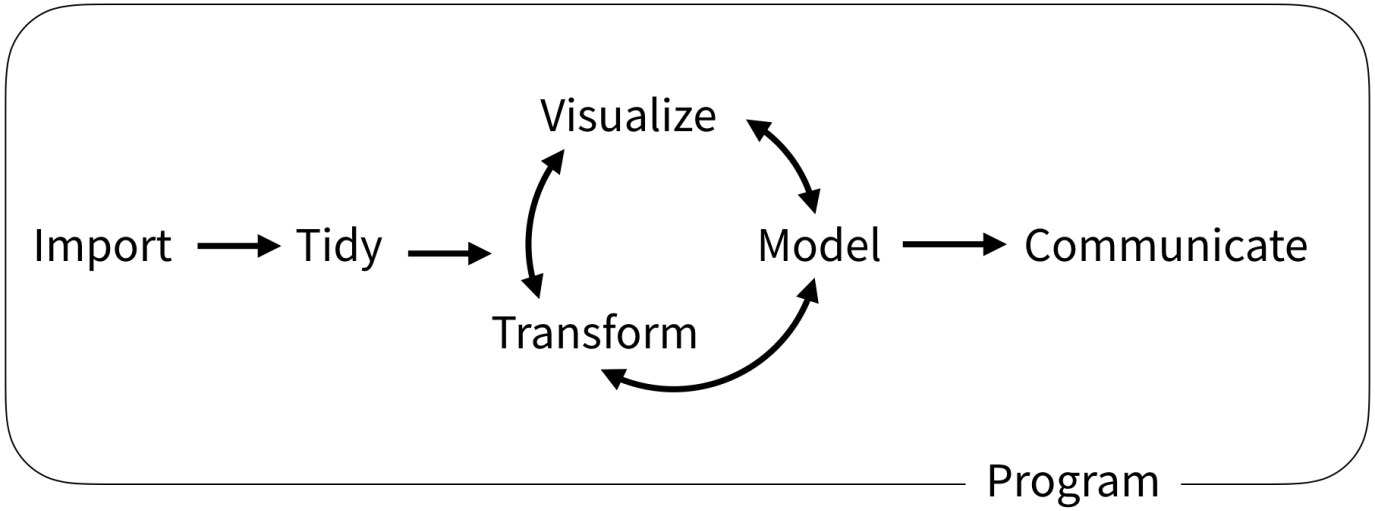
Tidyverse is a collection of R 

- `ggplot2` - visualising stuff
- `dplyr`, `tidyr` - data manipulation
- `purrr` - advanced programming
- `readr` - import data
- `tibble` - improved data.frame format
- `forcats` - working w/ factors
- `stringr` - working w/ chain of characters

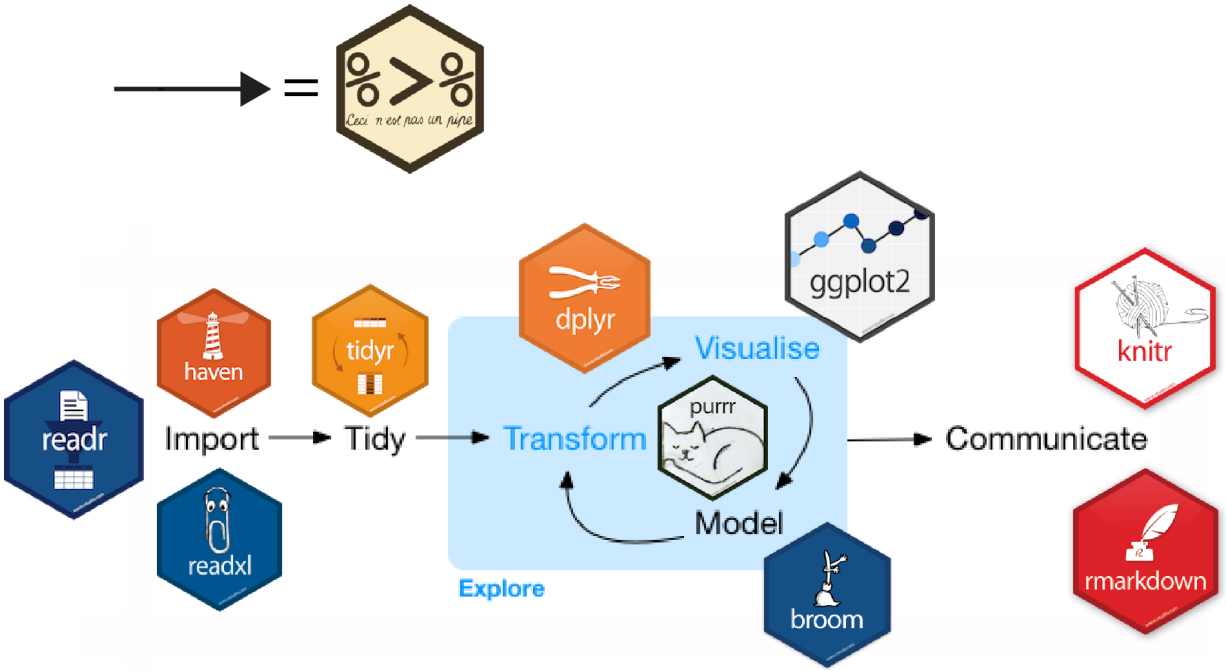
Tidyverse is a collection of R 

- `ggplot2` - visualising stuff
- `dplyr`, `tidyr` - data manipulation
- `purrr` - advanced programming
- `readr` - import data
- `tibble` - improved data.frame format
- `forcats` - working w/ factors
- `stringr` - working w/ chain of characters

Workflow in data science



Workflow in data science, with **Tidyverse**



data.frames

factors

strings

Load tidyverse

```r
#install.packages("tidyverse")
library(tidyverse)
```

```
## ── Attaching packages ──────────────────────────────────────────────────────

## ✔ ggplot2 3.1.0.9000     ✔ purrr   0.2.5
## ✔ tibble  2.0.1          ✔ dplyr   0.7.8
## ✔ tidyr   0.8.2          ✔ stringr 1.3.1
## ✔ readr   1.3.1          ✔ forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## ── Conflicts ─────────────────────────────────────────────────────────────
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Case study:

Using Twitter to predict citation rates of ecological research

Import

Import data

readr::read_csv function:

- keeps input types as is (no conversion to factor)
- creates `tibbles` instead of `data.frame`
    - no names to rows
    - allows column names with special characters
    - more clever on screen display than w/ data.frames
    - no partial matching on column names
    - warning if attempt to access unexisting column
- is daaaaaamn fast 🏎

Import data

```
citations_raw <- read_csv('https://raw.githubusercontent.com/oliviergimenez/intr
citations_raw
```

```
## # A tibble: 1,599 x 12
##    `Journal identi… `5-year journal… `Year published` Volume Issue Authors
##    <chr>                       <dbl>            <dbl>  <dbl> <chr> <chr>
##  1 Ecology Letters              16.7             2014     17 12    Morin …
##  2 Ecology Letters              16.7             2014     17 12    Jucker…
##  3 Ecology Letters              16.7             2014     17 12    Calcag…
##  4 Ecology Letters              16.7             2014     17 11    Segre …
##  5 Ecology Letters              16.7             2014     17 11    Kaufma…
##  6 Ecology Letters              16.7             2014     17 10    Nasto …
##  7 Ecology Letters              16.7             2014     17 10    Tschir…
##  8 Ecology Letters              16.7             2014     17 9     Barnec…
##  9 Ecology Letters              16.7             2014     17 9     Pinto-…
## 10 Ecology Letters              16.7             2014     17 9     Clough…
## # … with 1,589 more rows, and 6 more variables: `Collection date` <chr>,
## #   `Publication date` <chr>, `Number of tweets` <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, `Number of Web of Science
## #   citations` <dbl>
```

Tidy, transform

Rename columns

```
citations_temp <- rename(citations_raw,
       journal = 'Journal identity',
       impactfactor = '5-year journal impact factor',
       pubyear = 'Year published',
       colldate = 'Collection date',
       pubdate = 'Publication date',
       nbtweets = 'Number of tweets',
       woscitations = 'Number of Web of Science citations')
citations_temp
```

```
## # A tibble: 1,599 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <chr>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2/1/2016 9/16/2…
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2/1/2016 10/13/…
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2/1/2016 10/21/…
##  4 Ecolog…         16.7    2014     17 11    Segre … 2/1/2016 8/28/2…
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2/1/2016 8/28/2…
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2/2/2016 7/28/2…
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2/2/2016 8/6/20…
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2/2/2016 6/17/2…
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2/2/2016 6/12/2…
## 10 Ecolog…         16.7    2014     17 9     Clough… 2/2/2016 7/17/2…
## # … with 1,589 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

Create (or modify) columns

```
citations <- mutate(citations_temp, journal = as.factor(journal))
citations
```

```
## # A tibble: 1,599 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2/1/2016 9/16/2…
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2/1/2016 10/13/…
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2/1/2016 10/21/…
##  4 Ecolog…         16.7    2014     17 11    Segre … 2/1/2016 8/28/2…
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2/1/2016 8/28/2…
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2/2/2016 7/28/2…
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2/2/2016 8/6/20…
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2/2/2016 6/17/2…
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2/2/2016 6/12/2…
## 10 Ecolog…         16.7    2014     17 9     Clough… 2/2/2016 7/17/2…
## # … with 1,589 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

Create (or modify) columns

```
levels(citations$journal)
```

```
##  [1] "Animal Conservation"              "Conservation Letters"
##  [3] "Diversity and Distributions"      "Ecological Applications"
##  [5] "Ecology"                          "Ecology Letters"
##  [7] "Evolution"                        "Evolutionary Applications"
##  [9] "Fish and Fisheries"               "Functional Ecology"
## [11] "Global Change Biology"            "Global Ecology and Biogeography"
## [13] "Journal of Animal Ecology"        "Journal of Applied Ecology"
## [15] "Journal of Biogeography"          "Limnology and Oceanography"
## [17] "Mammal Review"                    "Methods in Ecology and Evolution"
## [19] "Molecular Ecology Resources"      "New Phytologist"
```

Give your code some air

Cleaner code with "pipe" operator %>%

```
citations_raw %>%
  rename(journal = 'Journal identity',
       impactfactor = '5-year journal impact factor',
       pubyear = 'Year published',
       colldate = 'Collection date',
       pubdate = 'Publication date',
       nbtweets = 'Number of tweets',
       woscitations = 'Number of Web of Science citations') %>%
  mutate(journal = as.factor(journal))
```

```
## # A tibble: 1,599 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2/1/2016 9/16/2…
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2/1/2016 10/13/…
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2/1/2016 10/21/…
##  4 Ecolog…         16.7    2014     17 11    Segre … 2/1/2016 8/28/2…
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2/1/2016 8/28/2…
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2/2/2016 7/28/2…
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2/2/2016 8/6/20…
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2/2/2016 6/17/2…
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2/2/2016 6/12/2…
## 10 Ecolog…         16.7    2014     17 9     Clough… 2/2/2016 7/17/2…
## # … with 1,589 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

Name object

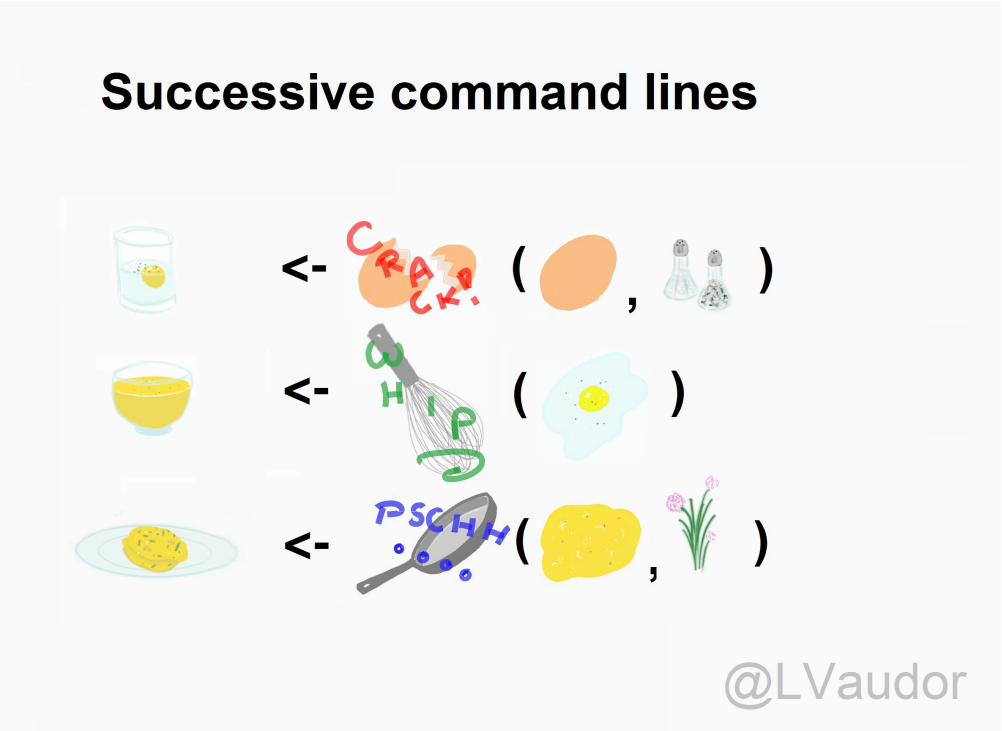
```
citations <- citations_raw %>%
  rename(journal = 'Journal identity',
       impactfactor = '5-year journal impact factor',
       pubyear = 'Year published',
       colldate = 'Collection date',
       pubdate = 'Publication date',
       nbtweets = 'Number of tweets',
       woscitations = 'Number of Web of Science citations') %>%
  mutate(journal = as.factor(journal))
```

Syntax with pipe

- Verb(Subject,Complement) replaced by Subject %>% Verb(Complement)
- No need to name unimportant intermediate variables
- Clear syntax (readability)

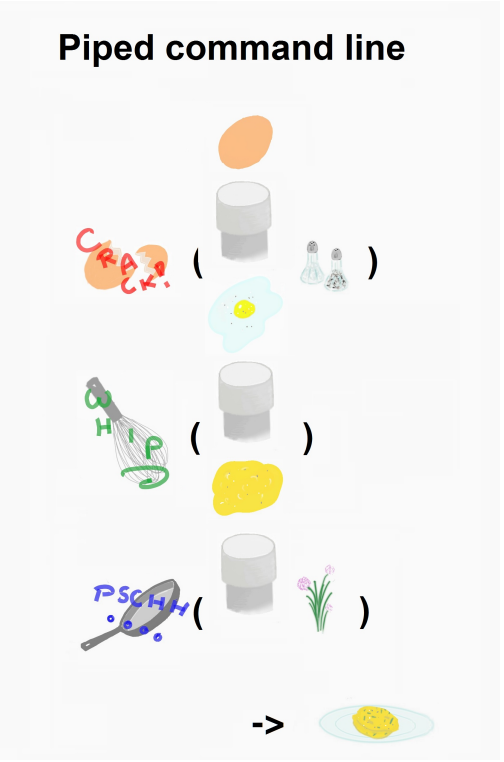
Base R from Lise Vaudor's blog

```
white_and_yolk <- crack(egg, add_seasoning)
omelette_batter <- beat(white_and_yolk)
omelette_with_chives <- cook(omelette_batter,add_chives)
```



Piping from Lise Vaudor's blog

```
egg %>%
  crack(add_seasoning) %>%
  beat() %>%
  cook(add_chives) -> omelette_with_chives
```



Tidy, transform

Select columns

```
citations %>%
  select(journal, impactfactor, nbtweets)
```

```
## # A tibble: 1,599 x 3
##    journal         impactfactor nbtweets
##    <fct>                  <dbl>    <dbl>
##  1 Ecology Letters         16.7       18
##  2 Ecology Letters         16.7       15
##  3 Ecology Letters         16.7        5
##  4 Ecology Letters         16.7        9
##  5 Ecology Letters         16.7        3
##  6 Ecology Letters         16.7       27
##  7 Ecology Letters         16.7        6
##  8 Ecology Letters         16.7       19
##  9 Ecology Letters         16.7       26
## 10 Ecology Letters         16.7       44
## # … with 1,589 more rows
```

Drop columns

```
citations %>%
  select(-Volume, -Issue, -Authors)
```

```
## # A tibble: 1,599 x 9
##    journal impactfactor pubyear colldate pubdate nbtweets `Number of user…
##    <fct>          <dbl>   <dbl> <chr>    <chr>      <dbl>            <dbl>
##  1 Ecolog…         16.7    2014 2/1/2016 9/16/2…       18               16
##  2 Ecolog…         16.7    2014 2/1/2016 10/13/…       15               12
##  3 Ecolog…         16.7    2014 2/1/2016 10/21/…        5                4
##  4 Ecolog…         16.7    2014 2/1/2016 8/28/2…        9                8
##  5 Ecolog…         16.7    2014 2/1/2016 8/28/2…        3                3
##  6 Ecolog…         16.7    2014 2/2/2016 7/28/2…       27               23
##  7 Ecolog…         16.7    2014 2/2/2016 8/6/20…        6                6
##  8 Ecolog…         16.7    2014 2/2/2016 6/17/2…       19               18
##  9 Ecolog…         16.7    2014 2/2/2016 6/12/2…       26               23
## 10 Ecolog…         16.7    2014 2/2/2016 7/17/2…       44               42
## # … with 1,589 more rows, and 2 more variables: `Twitter reach` <dbl>,
## #   woscitations <dbl>
```

Split a column in several columns

```
citations %>%
  separate(pubdate,c('month','day','year'),'/')
```

```
## # A tibble: 1,599 x 14
##    journal impactfactor pubyear Volume Issue Authors colldate month day
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr> <chr>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2/1/2016 9     16
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2/1/2016 10    13
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2/1/2016 10    21
##  4 Ecolog…         16.7    2014     17 11    Segre … 2/1/2016 8     28
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2/1/2016 8     28
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2/2/2016 7     28
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2/2/2016 8     6
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2/2/2016 6     17
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2/2/2016 6     12
## 10 Ecolog…         16.7    2014     17 9     Clough… 2/2/2016 7     17
## # … with 1,589 more rows, and 5 more variables: year <chr>,
## #   nbtweets <dbl>, `Number of users` <dbl>, `Twitter reach` <dbl>,
## #   woscitations <dbl>
```

Transform in Date format...

```r
library(lubridate)
citations %>%
  mutate(pubdate = mdy(pubdate),
         colldate = mdy(colldate))
```

```
## # A tibble: 1,599 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate   pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <date>     <date>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2016-02-01 2014-09-16
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2016-02-01 2014-10-13
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2016-02-01 2014-10-21
##  4 Ecolog…         16.7    2014     17 11    Segre … 2016-02-01 2014-08-28
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2016-02-01 2014-08-28
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2016-02-02 2014-07-28
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2016-02-02 2014-08-06
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2016-02-02 2014-06-17
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2016-02-02 2014-06-12
## 10 Ecolog…         16.7    2014     17 9     Clough… 2016-02-02 2014-07-17
## # … with 1,589 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

...for easy manipulation of dates

```r
library(lubridate)
citations %>%
  mutate(pubdate = mdy(pubdate),
         colldate = mdy(colldate),
         pubyear2 = year(pubdate))
```

```
## # A tibble: 1,599 x 13
##    journal impactfactor pubyear Volume Issue Authors colldate   pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <date>     <date>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2016-02-01 2014-09-16
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2016-02-01 2014-10-13
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2016-02-01 2014-10-21
##  4 Ecolog…         16.7    2014     17 11    Segre … 2016-02-01 2014-08-28
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2016-02-01 2014-08-28
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2016-02-02 2014-07-28
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2016-02-02 2014-08-06
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2016-02-02 2014-06-17
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2016-02-02 2014-06-12
## 10 Ecolog…         16.7    2014     17 9     Clough… 2016-02-02 2014-07-17
## # … with 1,589 more rows, and 5 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>,
## #   pubyear2 <dbl>
```

- Check out `?lubridate::lubridate` for more functions

How to join tables together?

> More #dplyr 🔧 gifs! It took me a hella long time to wrap my head around the different types of joins when I first started learning them, so here's a few examples with some excellent mini datasets from #dplyr designed specifically for this purpose! #rstats #tidyverse pic.twitter.com/G56fWmIZSq
>
> — Nic Crane (@nic_crane) 18 novembre 2018

Watch the video

Easy character manipulation

Select rows corresponding to papers with more than 3 authors

```
citations %>%
  filter(str_detect(Authors,'et al'))
```

```
## # A tibble: 1,280 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 Ecolog…         16.7    2014     17 12    Morin … 2/1/2016 9/16/2…
##  2 Ecolog…         16.7    2014     17 12    Jucker… 2/1/2016 10/13/…
##  3 Ecolog…         16.7    2014     17 12    Calcag… 2/1/2016 10/21/…
##  4 Ecolog…         16.7    2014     17 11    Segre … 2/1/2016 8/28/2…
##  5 Ecolog…         16.7    2014     17 11    Kaufma… 2/1/2016 8/28/2…
##  6 Ecolog…         16.7    2014     17 10    Nasto … 2/2/2016 7/28/2…
##  7 Ecolog…         16.7    2014     17 10    Tschir… 2/2/2016 8/6/20…
##  8 Ecolog…         16.7    2014     17 9     Barnec… 2/2/2016 6/17/2…
##  9 Ecolog…         16.7    2014     17 9     Pinto-… 2/2/2016 6/12/2…
## 10 Ecolog…         16.7    2014     17 9     Clough… 2/2/2016 7/17/2…
## # … with 1,270 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

Select rows corresponding to papers with less than 3 authors

```
citations %>%
  filter(!str_detect(Authors,'et al'))
```

```
## # A tibble: 319 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 Ecolog…         16.7    2014     17 6     Neutle… 2/15/20… 3/17/2…
##  2 Ecolog…         16.7    2014     17 5     Kellne… 2/15/20… 2/20/2…
##  3 Ecolog…         16.7    2014     17 4     Griffi… 2/15/20… 1/16/2…
##  4 Ecolog…         16.7    2014     17 3     Gremer… 2/15/20… 1/17/2…
##  5 Ecolog…         16.7    2014     17 2     Cavier… 2/15/20… 10/17/…
##  6 Ecolog…         16.7    2014     17 2     Haegma… 2/15/20… 12/5/2…
##  7 Ecolog…         16.7    2013     16 12    Kearney 2/15/20… 10/1/2…
##  8 Ecolog…         16.7    2013     16 9     Locey … 2/15/20… 7/15/2…
##  9 Ecolog…         16.7    2013     16 8     Quinte… 2/15/20… 6/26/2…
## 10 Ecolog…         16.7    2013     16 3     Lesser… 2/15/20… 12/22/…
## # … with 309 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

Select rows corresponding to papers with less than 3 authors in journal with IF < 5

```
citations %>%
  filter(!str_detect(Authors,'et al'), impactfactor < 5)
```

```
## # A tibble: 77 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <fct>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 Molecu…         4.9     2014     14 6     Gautier 2/27/20… 5/14/2…
##  2 Molecu…         4.9     2014     14 5     Gambel… 2/27/20… 3/7/20…
##  3 Molecu…         4.9     2014     14 4     Kekkon… 2/27/20… 3/10/2…
##  4 Molecu…         4.9     2014     14 3     Bhatta… 2/27/20… 12/8/2…
##  5 Molecu…         4.9     2014     14 1     Christ… 2/28/20… 10/25/…
##  6 Molecu…         4.9     2013     13 4     Villar… 2/28/20… 5/2/20…
##  7 Molecu…         4.9     2013     13 4     Wang    2/28/20… 4/25/2…
##  8 Molecu…         4.9     2012     12 1     Joly    2/28/20… 9/7/20…
##  9 Animal…         3.21    2014     17 6     Plavsic 2/9/2016 4/17/2…
## 10 Animal…         3.21    2014     17 Supp… Knox a… 2/11/20… 11/13/…
## # … with 67 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

Convert words to lowercase

```
citations %>%
  mutate(authors_lowercase = str_to_lower(Authors)) %>%
  select(authors_lowercase)
```

```
## # A tibble: 1,599 x 1
##    authors_lowercase
##    <chr>
##  1 morin et al
##  2 jucker et al
##  3 calcagno et al
##  4 segre et al
##  5 kaufman et al
##  6 nasto et al
##  7 tschirren et al
##  8 barnechi et al
##  9 pinto-sanchez et al
## 10 clough et al
## # … with 1,589 more rows
```

Remove all spaces in journal names

```
citations %>%
  mutate(journal = str_remove_all(journal," ")) %>%
  select(journal) %>%
  unique() %>%
  head(5)
```

```
## # A tibble: 5 x 1
##   journal
##   <chr>
## 1 EcologyLetters
## 2 GlobalChangeBiology
## 3 GlobalEcologyandBiogeography
## 4 MolecularEcologyResources
## 5 DiversityandDistributions
```

Explore stringr and regular expressions

- Check out the vignette on stringr for more examples on character manipulation and pattern matching functions.
- Check out the vignette on regular expressions which are a concise and flexible tool for describing patterns in strings.

Basic exploratory data analysis

Count

```
citations %>% count(journal, sort = TRUE)
```

```
## # A tibble: 20 x 2
##    journal                                n
##    <fct>                              <int>
##  1 New Phytologist                      144
##  2 Ecology                              108
##  3 Evolution                            108
##  4 Global Change Biology                108
##  5 Global Ecology and Biogeography      108
##  6 Journal of Biogeography              108
##  7 Ecology Letters                      106
##  8 Diversity and Distributions          105
##  9 Animal Conservation                  102
## 10 Methods in Ecology and Evolution      90
## 11 Evolutionary Applications             74
## 12 Functional Ecology                    54
## 13 Journal of Animal Ecology             54
## 14 Journal of Applied Ecology            54
## 15 Limnology and Oceanography            54
## 16 Molecular Ecology Resources           54
## 17 Conservation Letters                  53
## 18 Ecological Applications               48
## 19 Fish and Fisheries                    36
## 20 Mammal Review                         31
```

Count

```
citations %>%
  count(journal, pubyear) %>%
  head()
```

```
## # A tibble: 6 x 3
##   journal               pubyear     n
##   <fct>                   <dbl> <int>
## 1 Animal Conservation      2012    18
## 2 Animal Conservation      2013    18
## 3 Animal Conservation      2014    66
## 4 Conservation Letters     2012    17
## 5 Conservation Letters     2013    18
## 6 Conservation Letters     2014    18
```

Group by variable to calculate stats

```
citations %>%
  group_by(journal) %>%
  summarise(avg_tweets = mean(nbtweets)) %>%
  head(10)
```

```
## # A tibble: 10 x 2
##    journal                      avg_tweets
##    <fct>                             <dbl>
##  1 Animal Conservation              12.4
##  2 Conservation Letters             10.2
##  3 Diversity and Distributions       1.90
##  4 Ecological Applications           2.60
##  5 Ecology                           3.10
##  6 Ecology Letters                  14.5
##  7 Evolution                         3.10
##  8 Evolutionary Applications         3.22
##  9 Fish and Fisheries                7.25
## 10 Functional Ecology                2.87
```

Order stuff

```r
citations %>%
  group_by(journal) %>%
  summarise(avg_tweets = mean(nbtweets)) %>%
  arrange(desc(avg_tweets)) %>% # decreasing order (wo desc for increasing)
  head(10)
```

```
## # A tibble: 10 x 2
##    journal                          avg_tweets
##    <fct>                                 <dbl>
##  1 Journal of Applied Ecology            18.7
##  2 Ecology Letters                       14.5
##  3 Animal Conservation                   12.4
##  4 Conservation Letters                  10.2
##  5 Methods in Ecology and Evolution       7.77
##  6 Fish and Fisheries                     7.25
##  7 Journal of Animal Ecology              5.98
##  8 Global Change Biology                  5.68
##  9 Mammal Review                          5.35
## 10 New Phytologist                        3.53
```

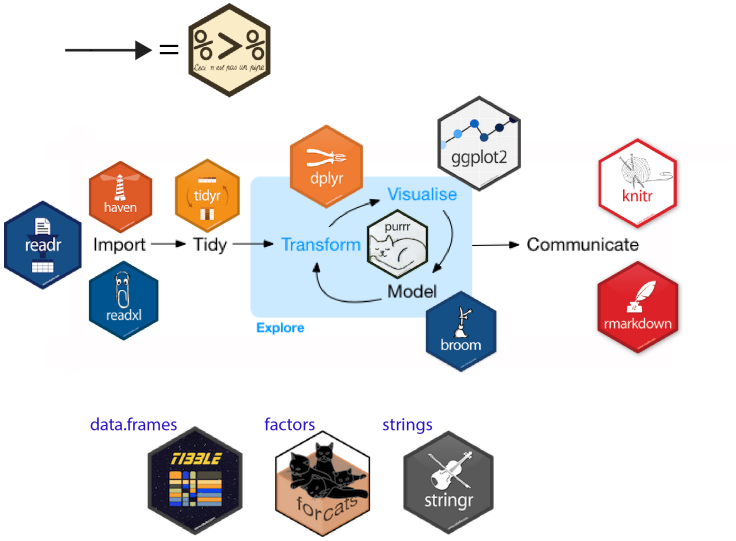
Group by variables to calculate stats

```
citations %>%
  group_by(journal, pubyear) %>%
  summarise(avg_tweets_year = mean(nbtweets))
```

```
## # A tibble: 59 x 3
## # Groups:   journal [?]
##    journal                      pubyear avg_tweets_year
##    <fct>                          <dbl>           <dbl>
##  1 Animal Conservation             2012           3.72
##  2 Animal Conservation             2013           5.83
##  3 Animal Conservation             2014          16.6
##  4 Conservation Letters            2012           6.76
##  5 Conservation Letters            2013          11.6
##  6 Conservation Letters            2014          12.2
##  7 Diversity and Distributions     2012           0.667
##  8 Diversity and Distributions     2013           1
##  9 Diversity and Distributions     2014           3.97
## 10 Ecological Applications         2012           0.917
## # … with 49 more rows
```

There are so many more things to explore

- **spread()** and **gather()** from package `tidyr` to reshape tibbles; see here for exemple
- **mutate_all()**, **select_if()** and **summarise_at()** or scoped verbs where *scoped* means that these functions operate only on a selection of variables; see here for example



Visualize

Visualization with ggplot2

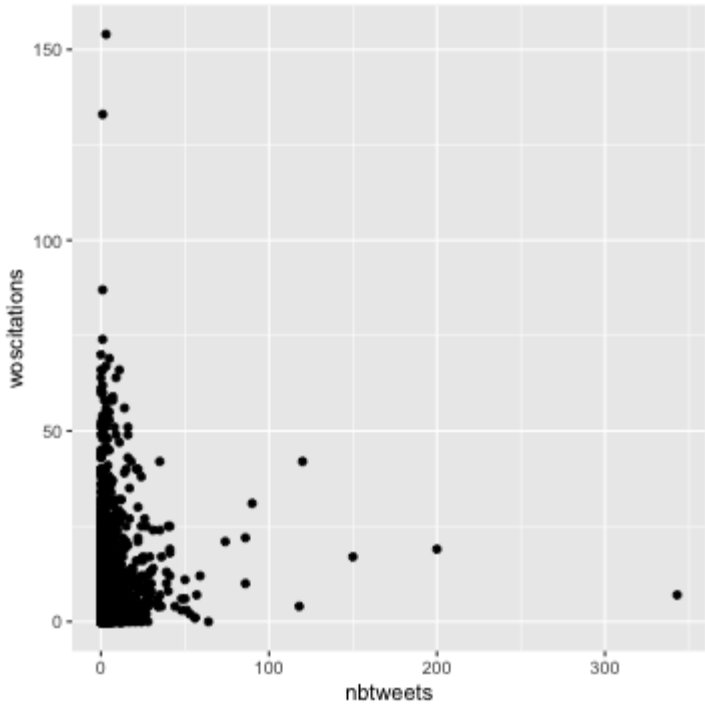
- The package ggplot2 implements a **g**rammar of **g**raphics
- Operates on data.frames or tibbles, not vectors like base R
- Explicitly differentiates between the data and its representation

The ggplot2 grammar

Grammar element	What it is
Data	The data frame being plotted
Geometrics	The geometric shape that will represent the data (e.g., point, boxplot, histogram)
Aesthetics	The aesthetics of the geometric object (e.g., color, size, shape)

Scatterplots

```
citations %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_point()
```



Scatterplots

```
citations %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_point()
```

- Pass in the data frame as your first argument

Scatterplots

```
citations %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_point()
```

- Pass in the data frame as your first argument
- Aesthetics maps the data onto plot characteristics, here x and y axes

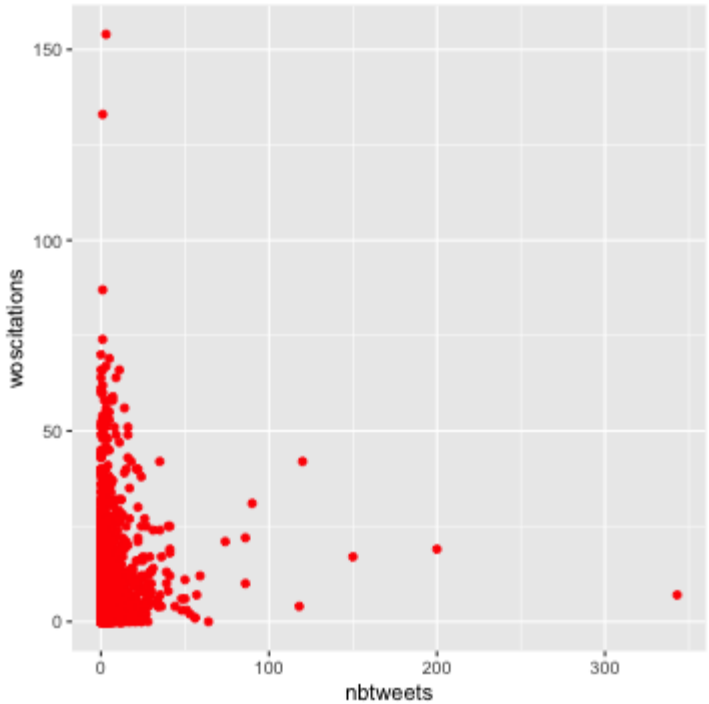
Scatterplots

```
citations %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_point()
```

- Pass in the data frame as your first argument
- Aesthetics maps the data onto plot characteristics, here x and y axes
- Display the data geometrically as points

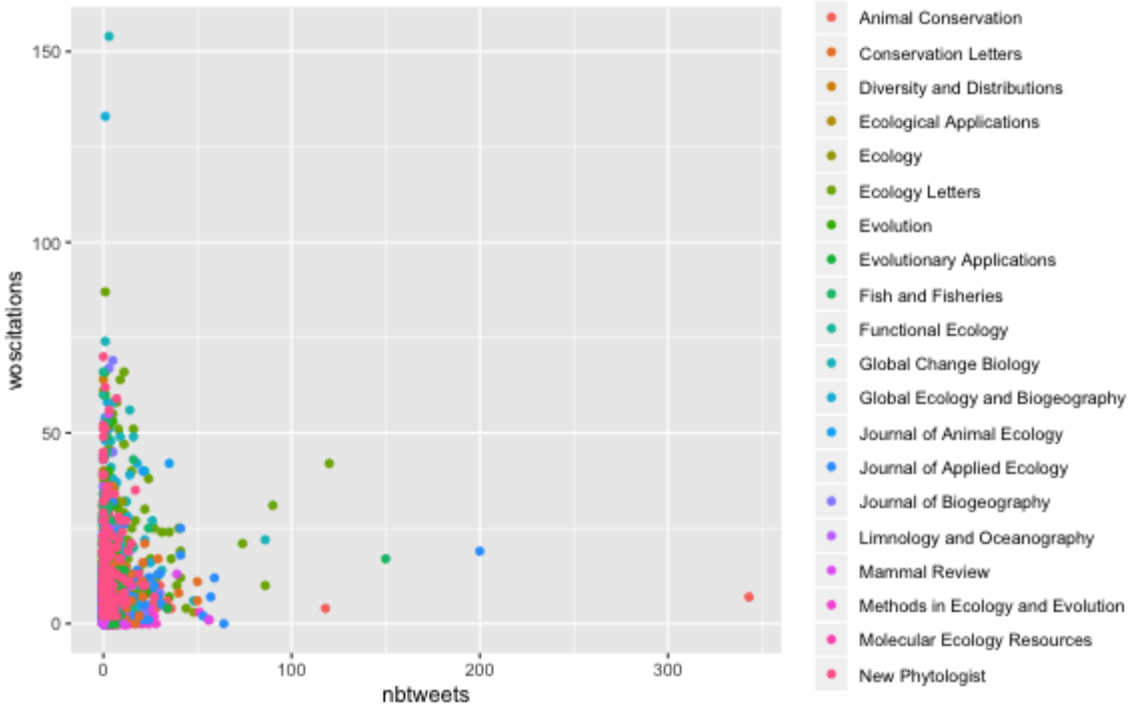
Scatterplots, with colors

```
citations %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_point(color = "red")
```



Scatterplots, with species-specific colors

```
citations %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations, color = journal) +
  geom_point()
```



- Placing color inside aesthetic maps it to the data

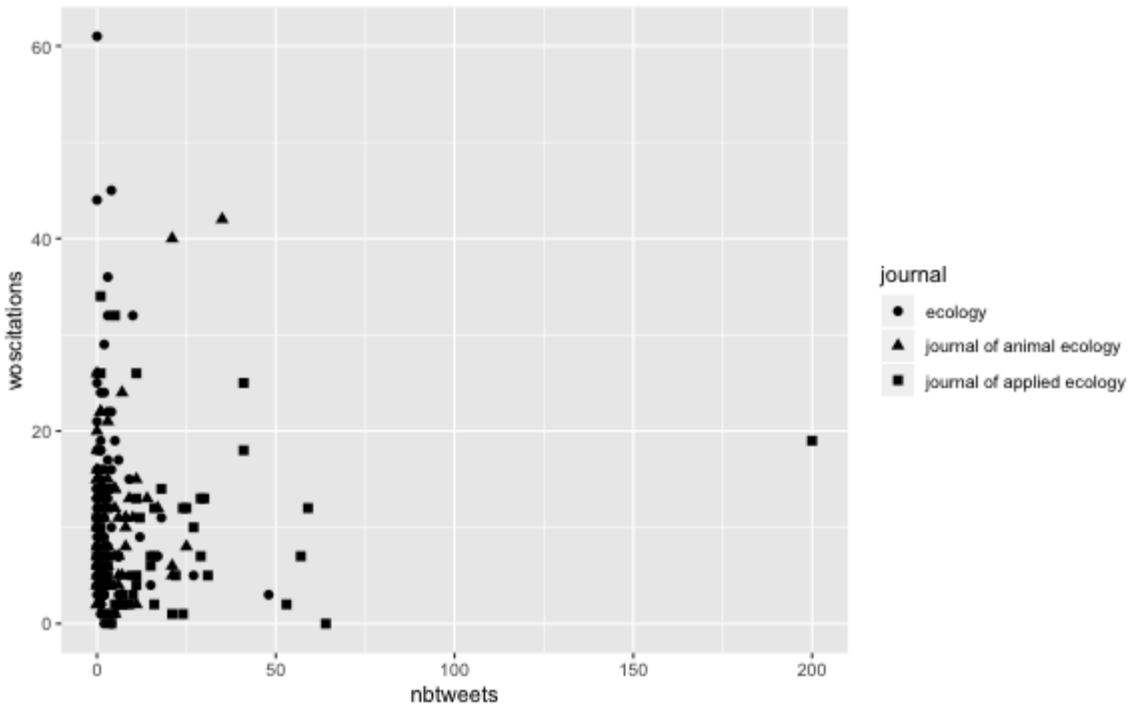
Pick a few journals

```
citations_ecology <- citations %>%
  mutate(journal = str_to_lower(journal)) %>% # all journals names lowercase
  filter(journal %in%
           c('journal of animal ecology','journal of applied ecology','ecology')
citations_ecology
```

```
## # A tibble: 216 x 12
##    journal impactfactor pubyear Volume Issue Authors colldate pubdate
##    <chr>          <dbl>   <dbl>  <dbl> <chr> <chr>   <chr>    <chr>
##  1 ecology         6.16    2014     95 12    Maglia… 3/19/20… 12/1/2…
##  2 ecology         6.16    2014     95 12    Soinen  3/19/20… 12/1/2…
##  3 ecology         6.16    2014     95 12    Graham… 3/19/20… 12/1/2…
##  4 ecology         6.16    2014     95 11    White … 3/19/20… 11/1/2…
##  5 ecology         6.16    2014     95 11    Einars… 3/19/20… 11/1/2…
##  6 ecology         6.16    2014     95 11    Haav a… 3/19/20… 11/1/2…
##  7 ecology         6.16    2014     95 10    Dodds … 3/19/20… 10/1/2…
##  8 ecology         6.16    2014     95 10    Brown … 3/19/20… 10/1/2…
##  9 ecology         6.16    2014     95 10    Wright… 3/19/20… 10/1/2…
## 10 ecology         6.16    2014     95 9     Ramahl… 3/19/20… 9/1/20…
## # … with 206 more rows, and 4 more variables: nbtweets <dbl>, `Number of
## #   users` <dbl>, `Twitter reach` <dbl>, woscitations <dbl>
```

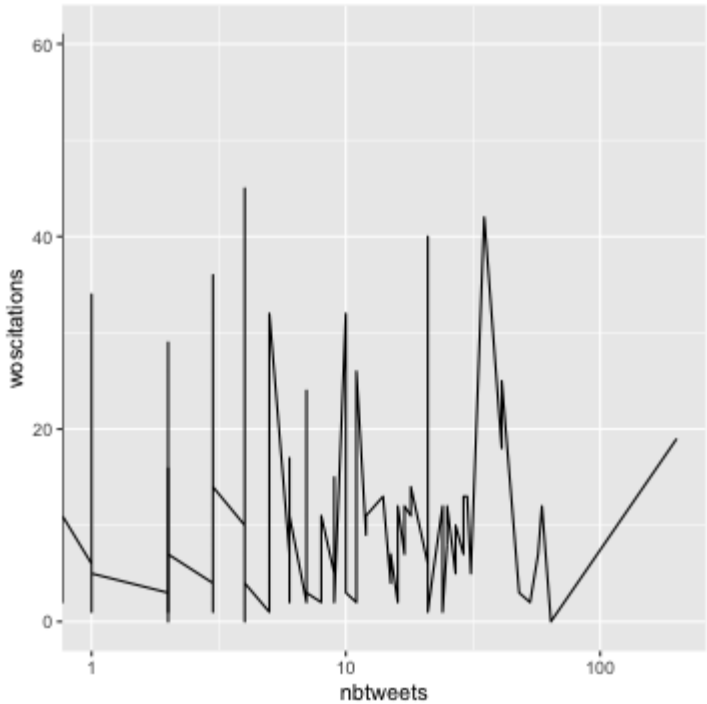
Scatterplots, with species-specific shapes

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations, shape = journal) +
  geom_point(size=2)
```



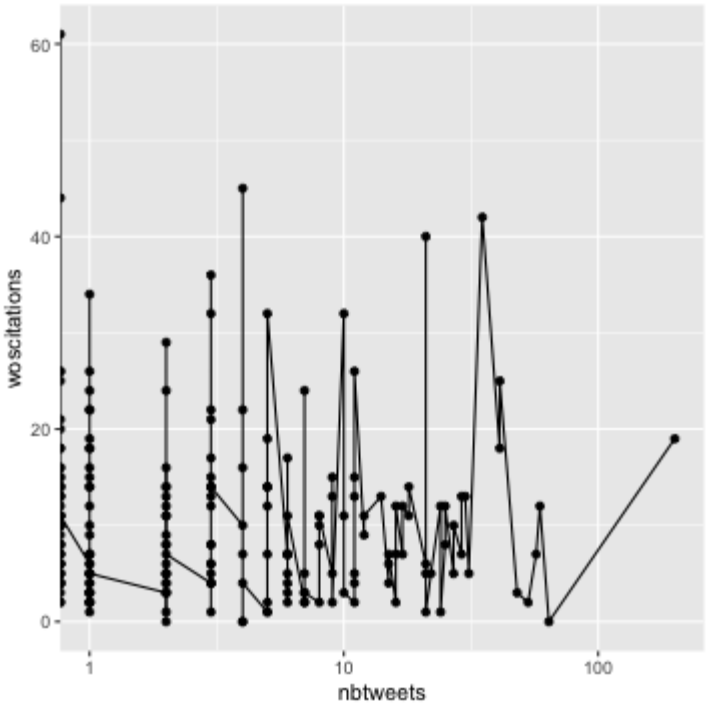
Scatterplots, lines instead of points

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_line() +
  scale_x_log10()
```



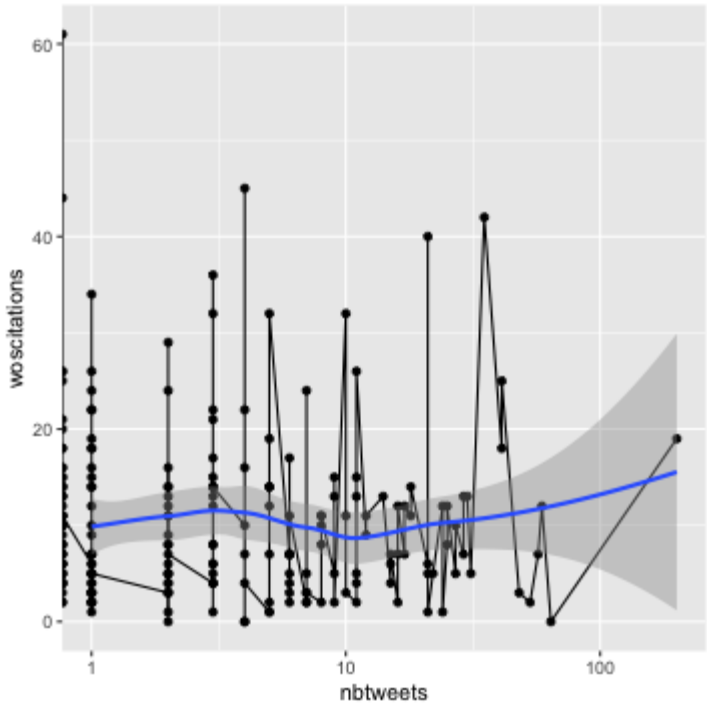
Scatterplots, add points

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_line() +
  geom_point() +
  scale_x_log10()
```



Scatterplots, add smoother

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, y = woscitations) +
  geom_line() +
  geom_point() +
  geom_smooth() +
  scale_x_log10()
```

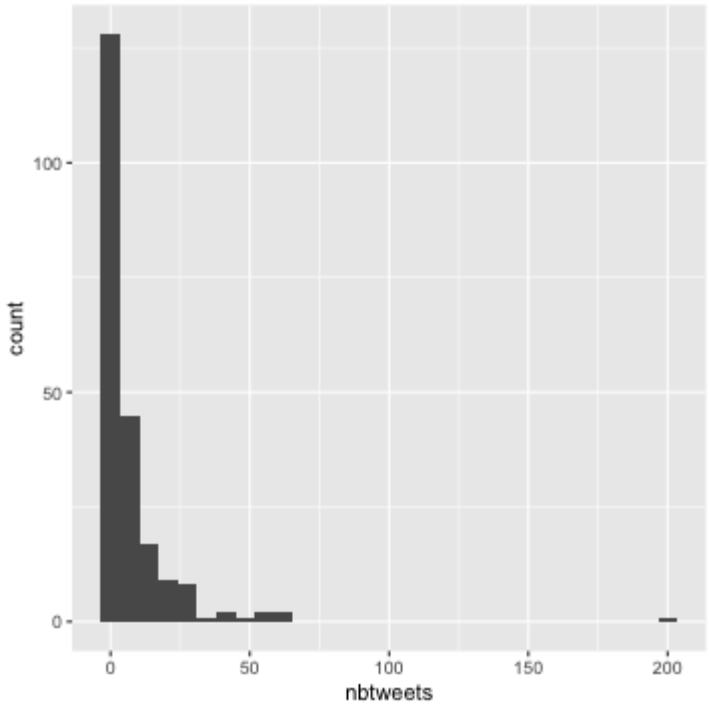


aes or not aes?

- If we are to establish a link between the values of a variable and a graphical feature, ie a mapping, then we need an aes().
- Otherwise, the graphical feature is modified irrespective of the data, then we do not need an aes().

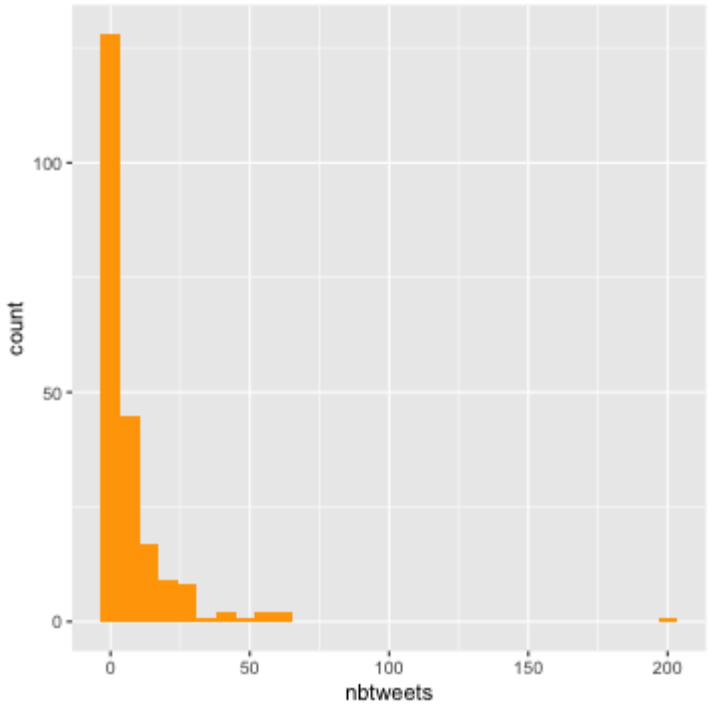
Histograms

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets) +
  geom_histogram()
```



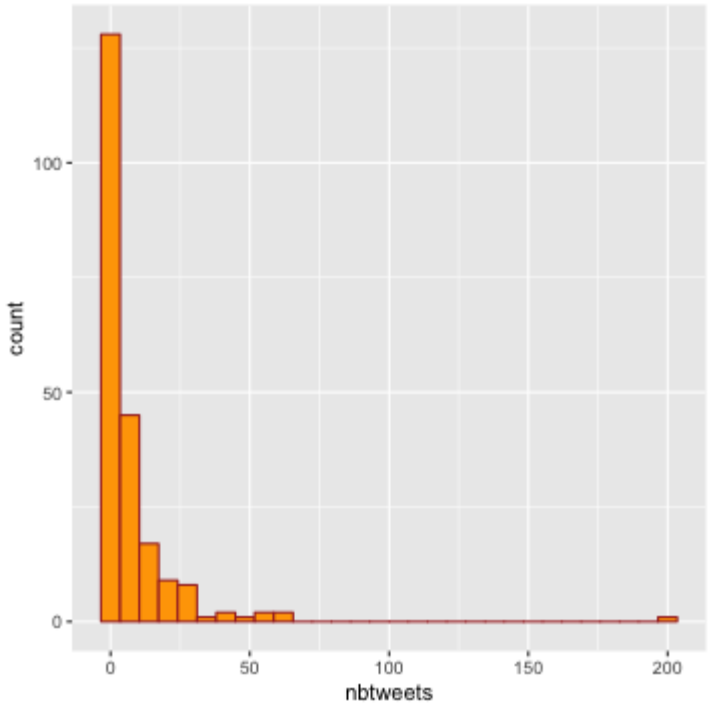
Histograms, with colors

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets) +
  geom_histogram(fill = "orange")
```



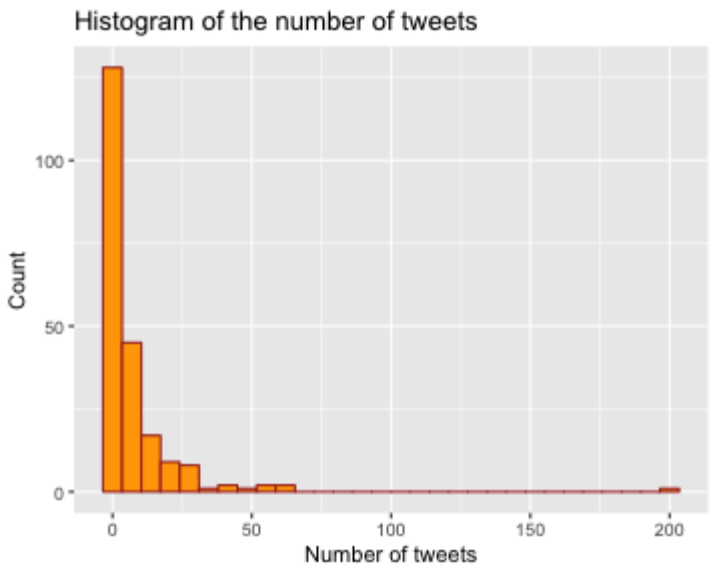
Histograms, with colors

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets) +
  geom_histogram(fill = "orange", color = "brown")
```



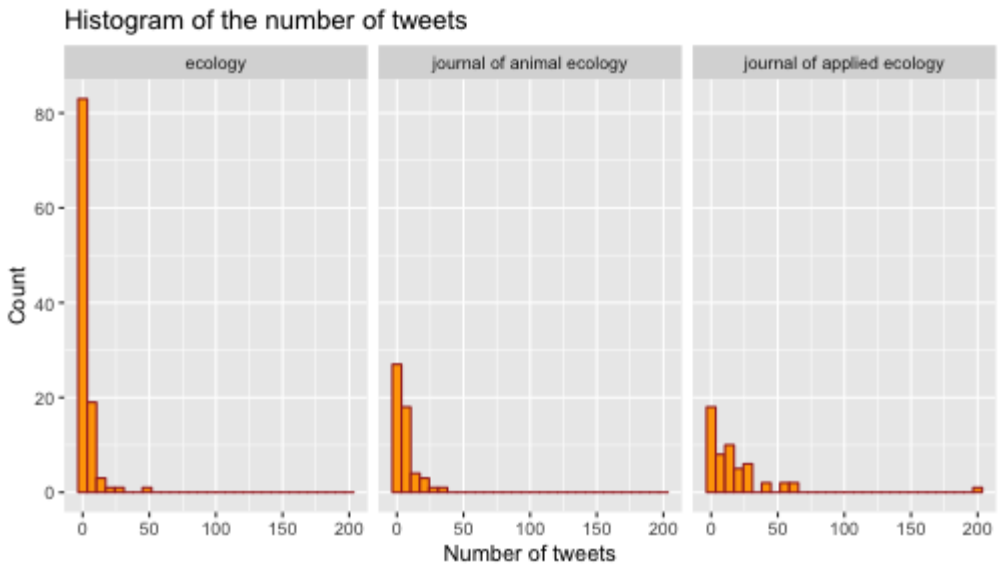
Histograms, with labels and title

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets) +
  geom_histogram(fill = "orange", color = "brown") +
  xlab("Number of tweets") +
  ylab("Count") +
  ggtitle("Histogram of the number of tweets")
```



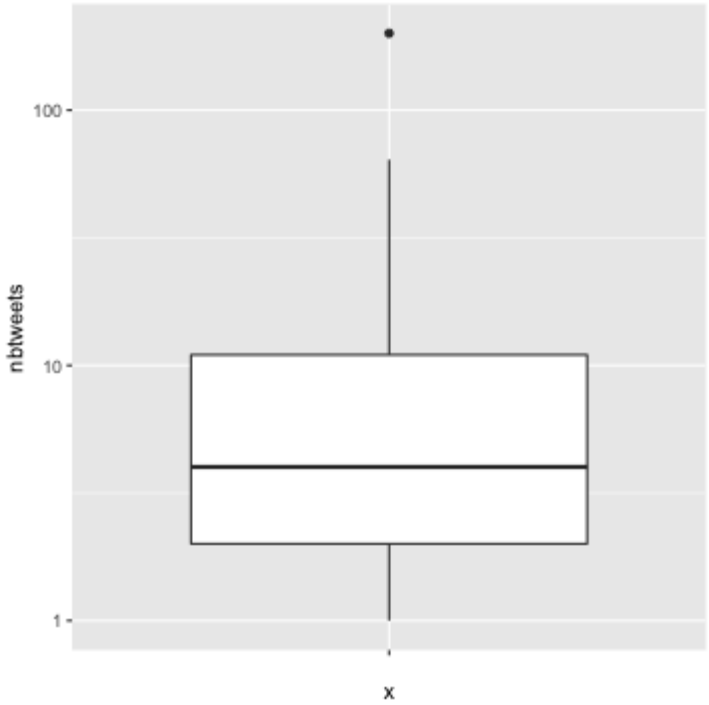
Histograms, by species

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets) +
  geom_histogram(fill = "orange", color = "brown") +
  xlab("Number of tweets") +
  ylab("Count") +
  ggtitle("Histogram of the number of tweets") +
  facet_wrap(aes(journal))
```



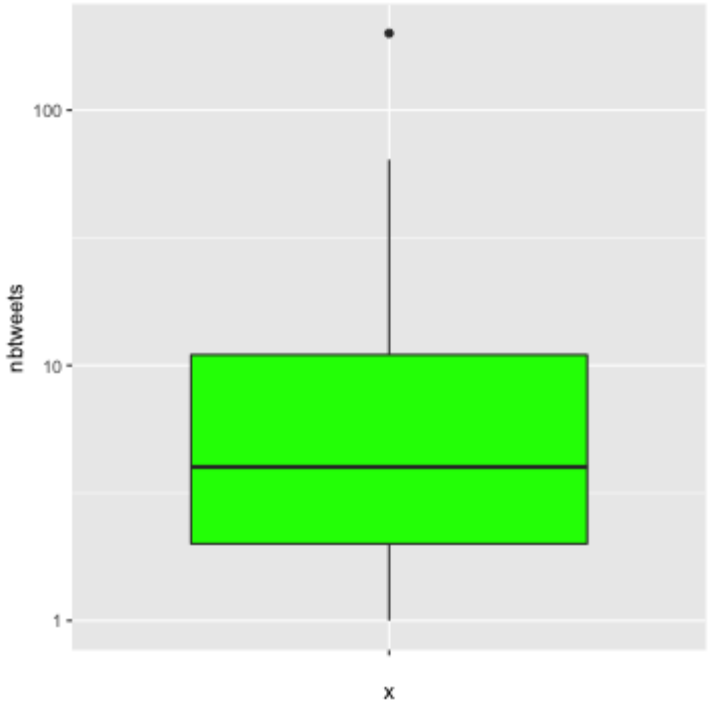
Boxplots

```
citations_ecology %>%
  ggplot() +
  aes(x = "", y = nbtweets) +
  geom_boxplot() +
  scale_y_log10()
```



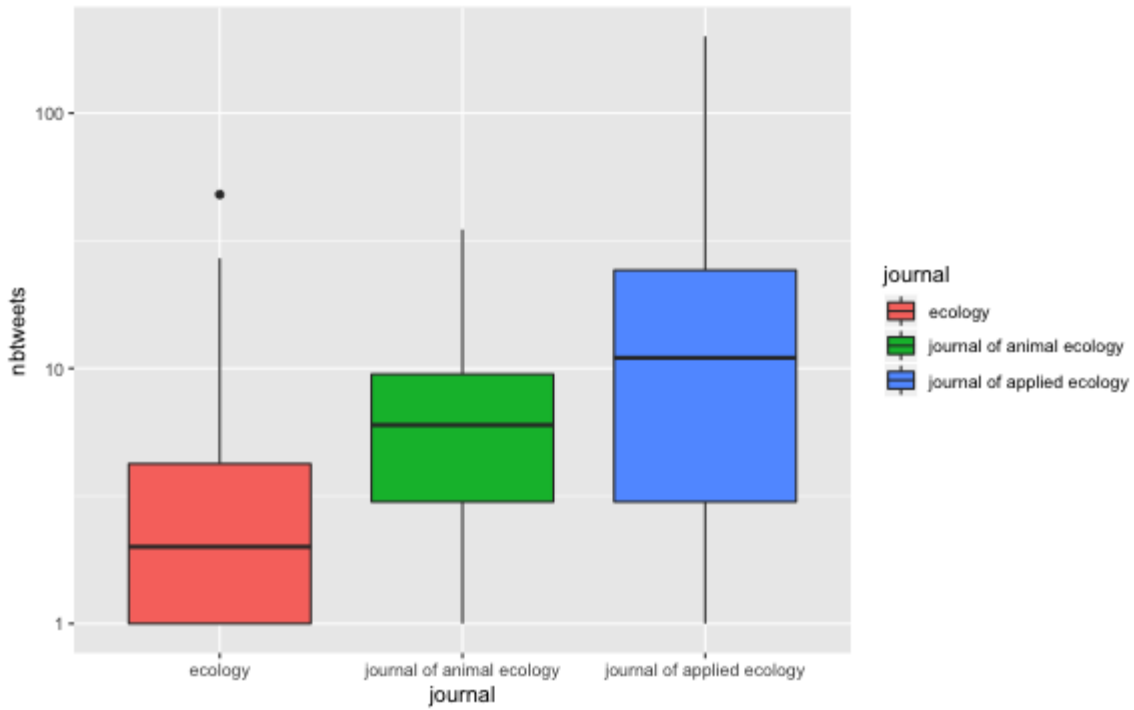
Boxplots with colors

```
citations_ecology %>%
  ggplot() +
  aes(x = "", y = nbtweets) +
  geom_boxplot(fill = "green") +
  scale_y_log10()
```



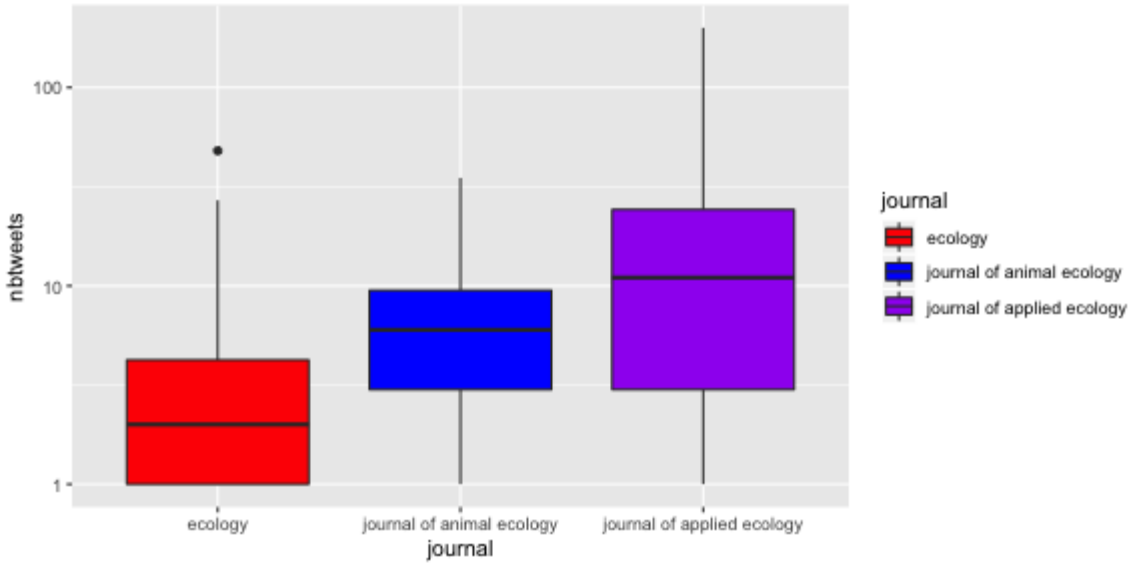
Boxplots with colors by species

```
citations_ecology %>%
  ggplot() +
  aes(x = journal, y = nbtweets, fill = journal) +
  geom_boxplot() +
  scale_y_log10()
```



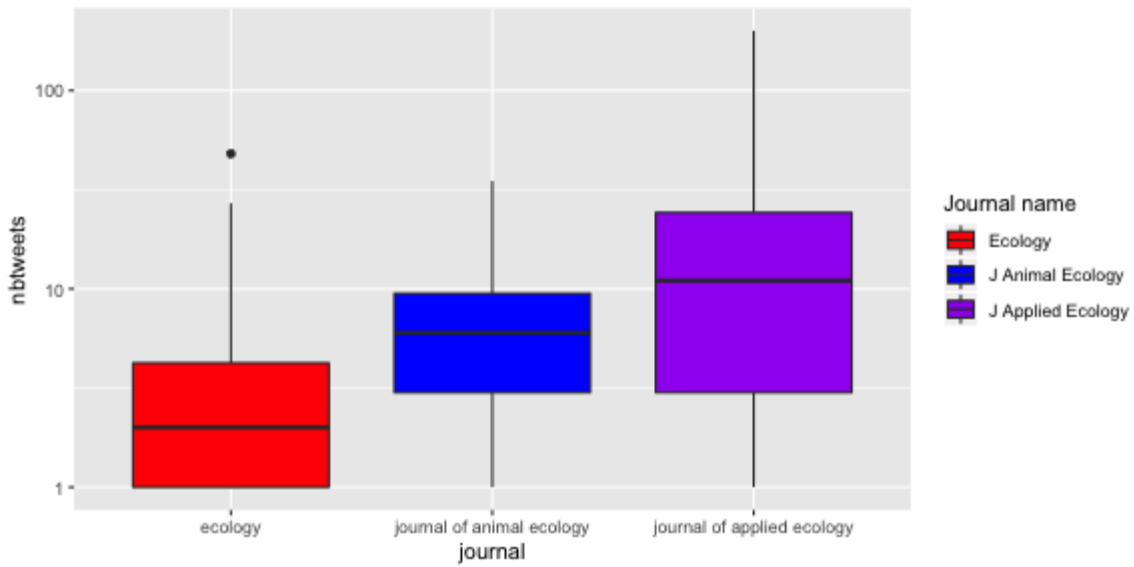
Boxplots, user-specified colors by species

```
citations_ecology %>%
  ggplot() +
  aes(x = journal, y = nbtweets, fill = journal) +
  geom_boxplot() +
  scale_y_log10() +
  scale_fill_manual(
    values=c("red", "blue", "purple"))
```



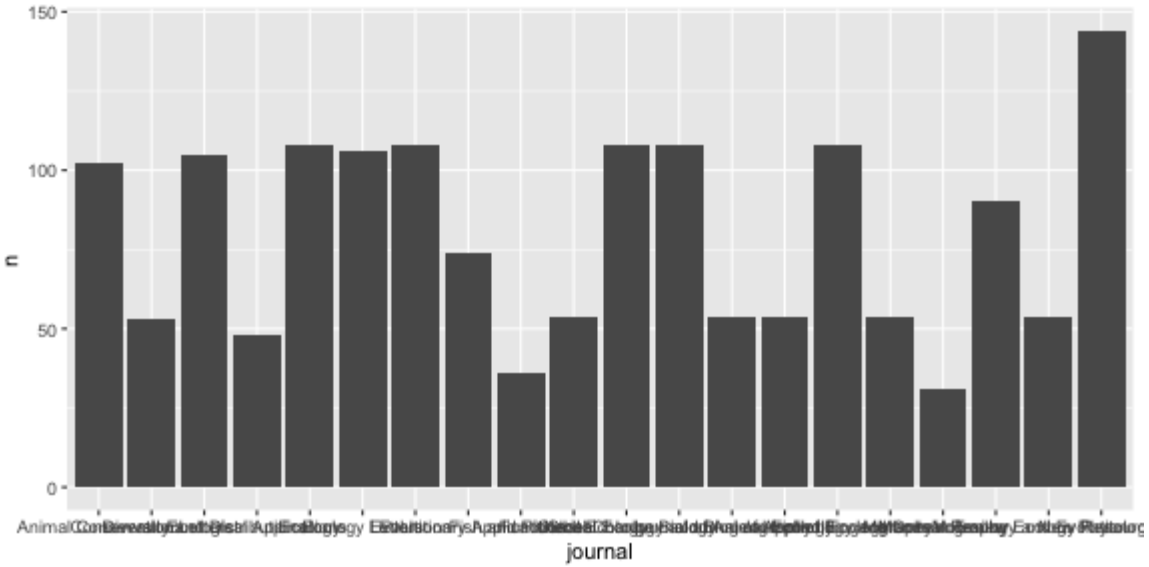
Boxplots, change legend settings

```
citations_ecology %>%
  ggplot() +
  aes(x = journal, y = nbtweets, fill = journal) +
  geom_boxplot() +
  scale_y_log10() +
  scale_fill_manual(
    values=c("red", "blue", "purple"),
    name = "Journal name",
    labels=c("Ecology", "J Animal Ecology", "J Applied Ecology"))
```



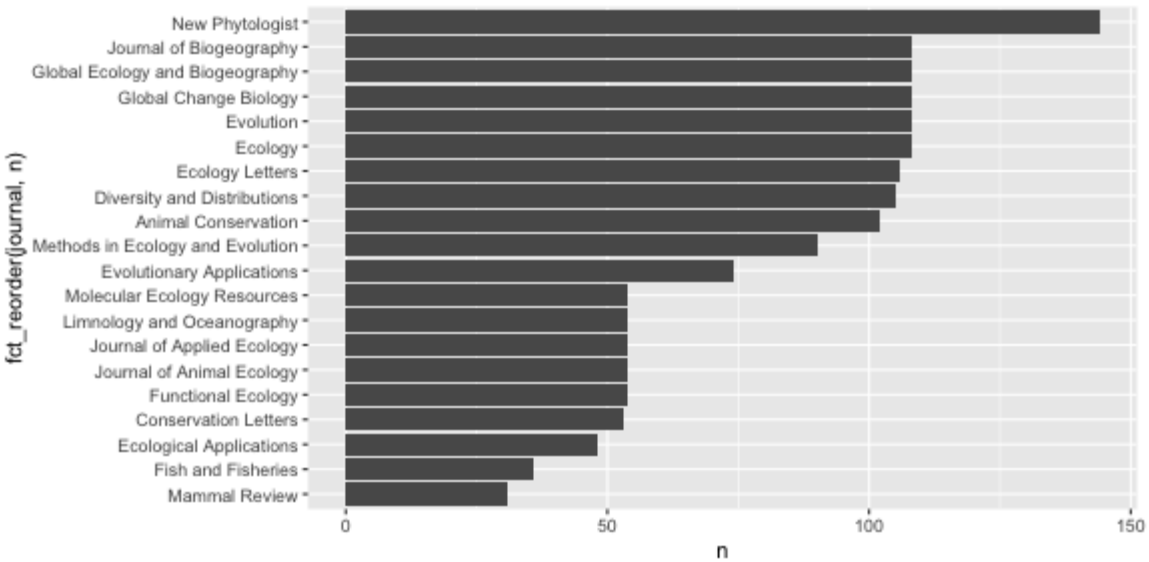
Ugly bar plots

```
citations %>%
  count(journal) %>%
  ggplot() +
  aes(x = journal, y = n) +
  geom_col()
```



Idem, with factors reordering and flipping

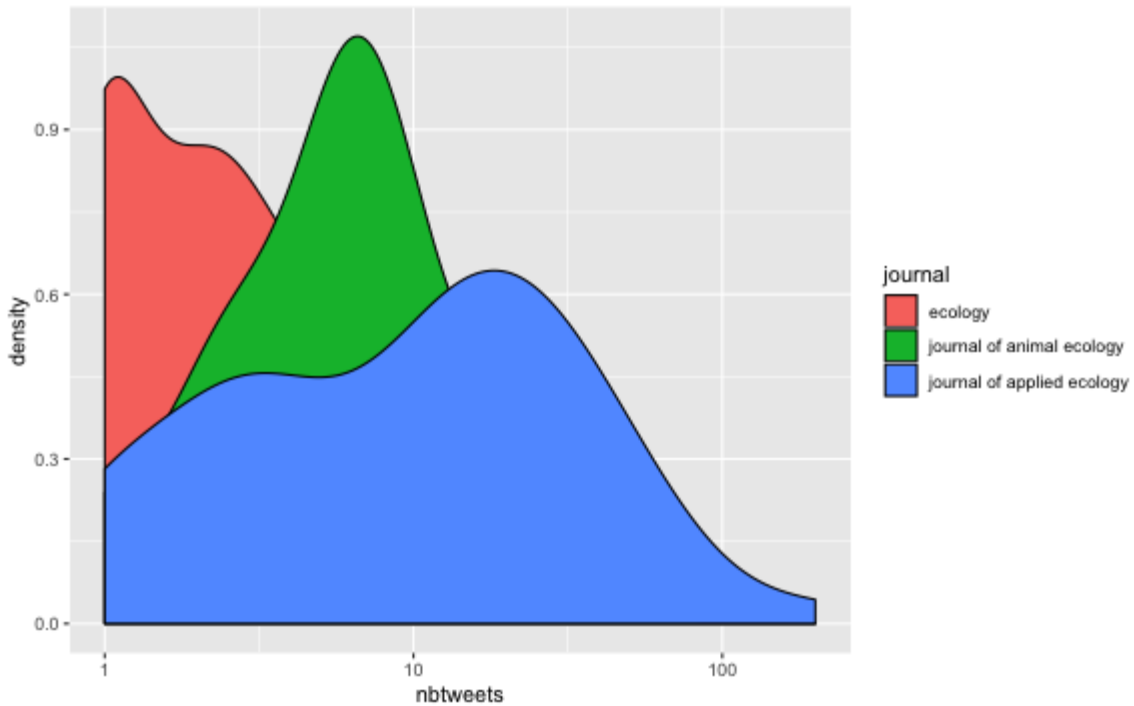
```
citations %>%
  count(journal) %>%
  ggplot() +
  aes(x = fct_reorder(journal, n), y = n) +
  geom_col() +
  coord_flip()
```



- More about how to (tidy) work with factors here and here.

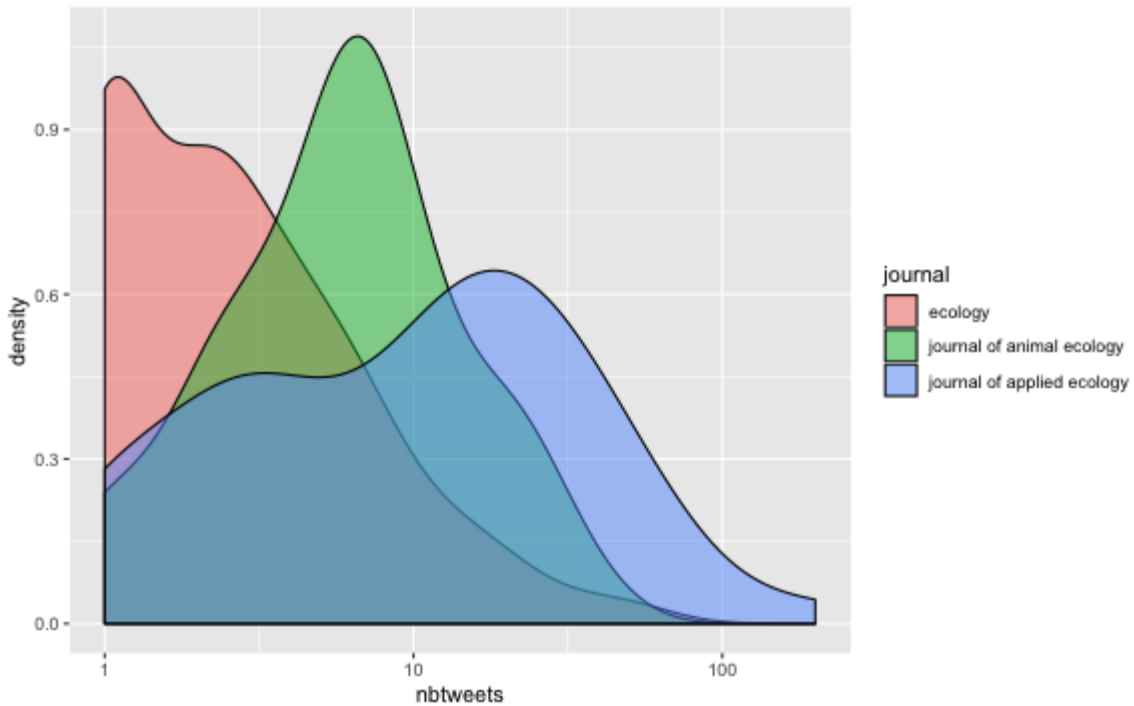
Density plots

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, fill = journal) +
  geom_density() +
  scale_x_log10()
```



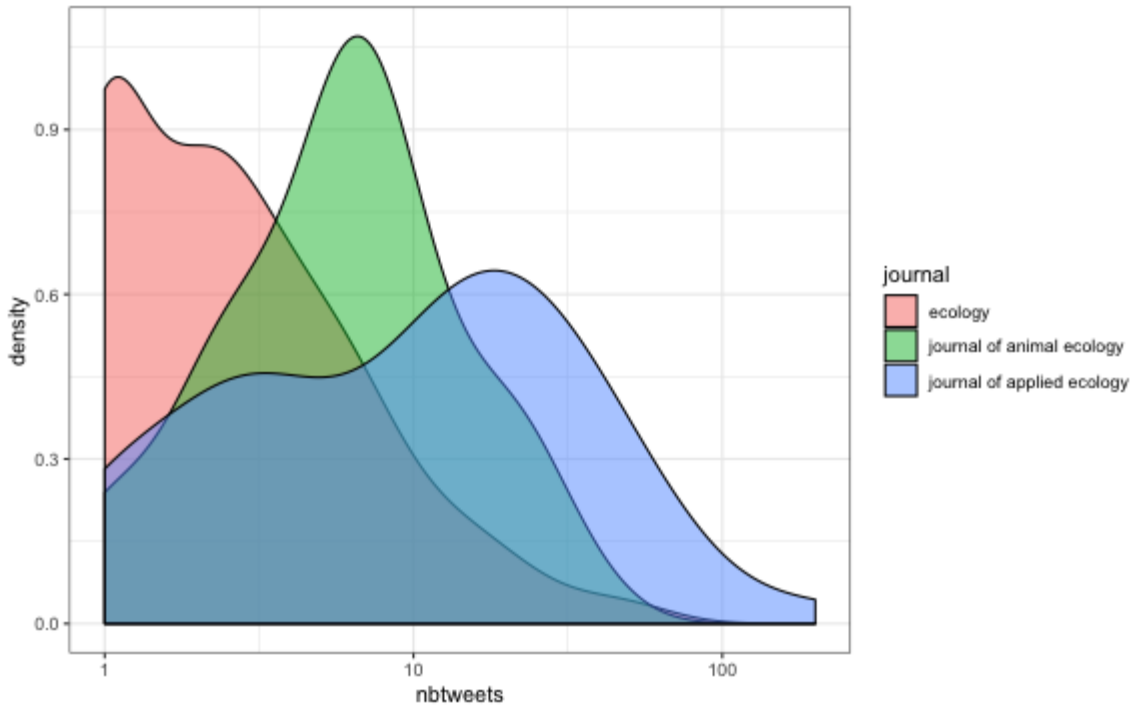
Density plots, control transparency

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, fill = journal) +
  geom_density(alpha = 0.5) +
  scale_x_log10()
```



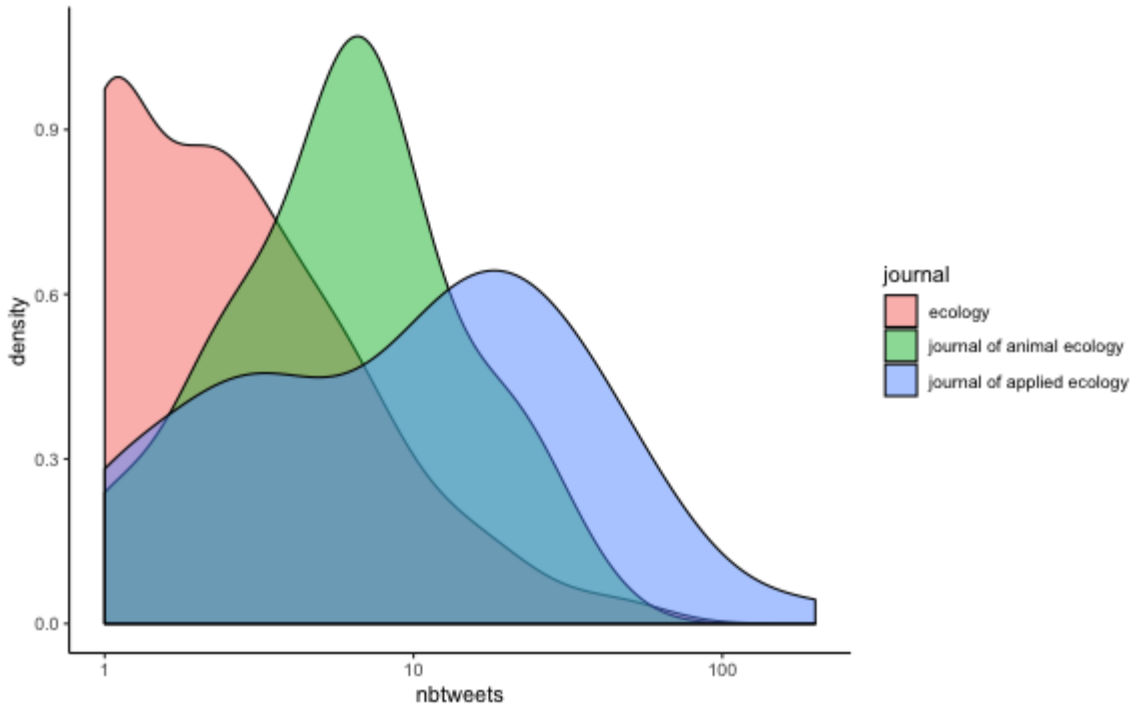
Change default background theme 1/3

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, fill = journal) +
  geom_density(alpha = 0.5) +
  scale_x_log10() +
  theme_bw()
```



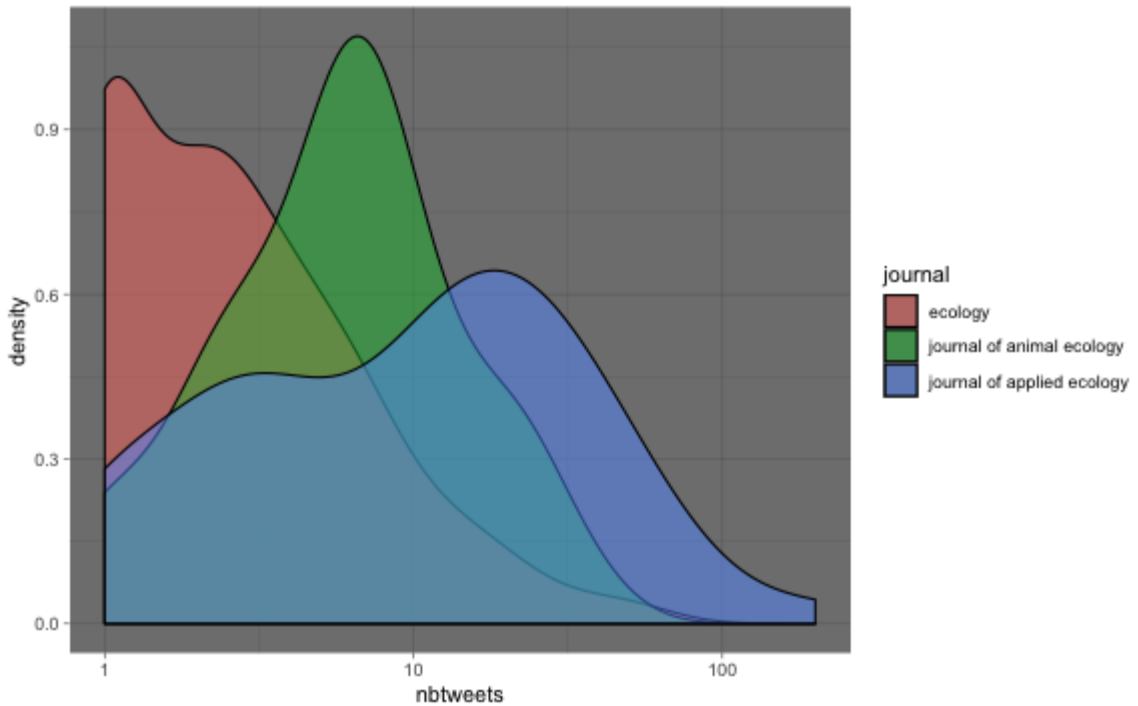
Change default background theme 2/3

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, fill = journal) +
  geom_density(alpha = 0.5) +
  scale_x_log10() +
  theme_classic()
```



Change default background theme 3/3

```
citations_ecology %>%
  ggplot() +
  aes(x = nbtweets, fill = journal) +
  geom_density(alpha = 0.5) +
  scale_x_log10() +
  theme_dark()
```



More on data visualisation with ggplot2

- Portfolio of ggplot2 plots
- Top ggplot2 visualizations
- Interactive ggplot2 visualizations


tidyverse
www.rstudio.com

To dive even deeper in the tidyverse

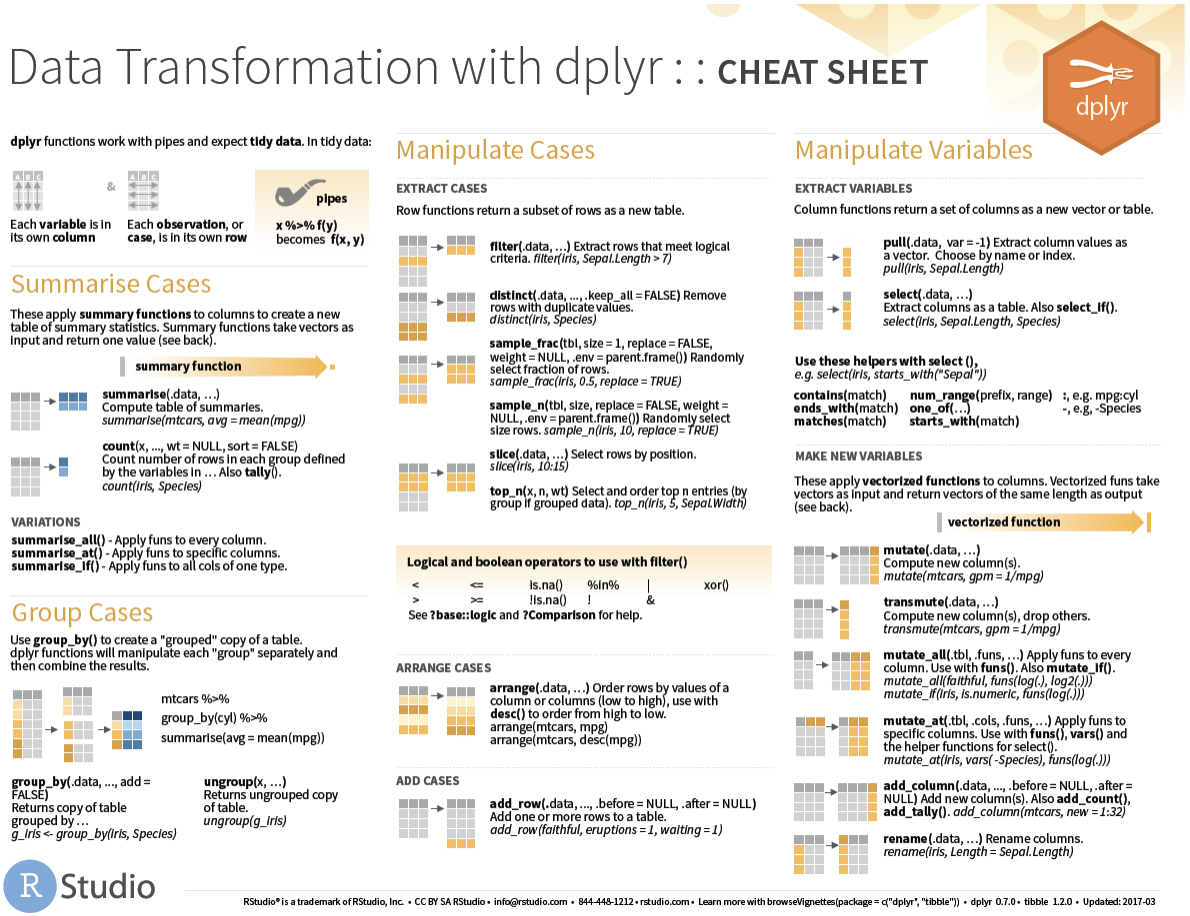
- Learn the tidyverse: books, workshops and online courses
- My selection of books:
  - R for Data Science et Advanced R
  - Introduction à R et au tidyverse
  - Fundamentals of Data visualization
  - Data Visualization: A practical introduction
- Tidy Tuesdays videos by D. Robinson chief data scientist at DataCamp
- Material of the 2-day workshop Data Science in the tidyverse held at the RStudio 2019 conference
- Material of the stat545 course on Data wrangling, exploration, and analysis with R at the University of British Columbia
- List of best R packages (with their description) on data import, wrangling and visualization

How to switch from base R to tidyverse?

Couple of notes before we start. The list below is not exhaustive (best to read package documentation for that). For instance, it doesnt cover lubridate (which covers date/time related functions), forcats (which covers everything you would want to do to factors), broom (which tidies up messy R objects), modelr (which has helper functions for creating models) or ggplot. I also use data frame and tibble interchangeably, although they are obviously different.

Base R command	Tidyverse Command	What it does and why you should use the tidyverse version	Comment
read.csv()	read_csv()	reads in a csv file, but its much faster, shows progress bar for large files, can automatically parse data types	also see read_delim(), read_tsv() and readxl::read_xlsx()
sort(), order()	arrange()	sort column(n) within a data frame	see also order_by()
mtcars$mpg = …	mutate()	modify a column	see also transmute() which drops existing variables
mtcars[,c("mpg", "am")], subset()	select(), rename()	select or rename columns	see also pull()
mtcars[mtcars$am == 1,], subset()	filter()	select rows based on a criterion	
aggregate()	summarise(), summarize(), do()	reduce grouped values to a single value	see also varaints like summarize_if()
ifelse()	if_else(), case_when()	standand vectorized if else, but stricter than base version	see also near()
unique()	distinct()	finds unique rows in a data frame, but its much, faster	
length(unique())	n_distinct()	count the number of distinct values in a vector, faster	
sample(), sample.int()	sample_n(), sample_frac()	sample n rows or a fraction of rows from a dataframe	
all.equal()	all_equal()	checks if two vectors are the same	
merge()	inner_join(), left_join()	perform joins, much faster, verbose, and row order is maintain	see also right_join(), full_join(), semi_join(), anti_join()
rbind(), cbind()	bind_rows(), bind_cols()	concatenate two dataframes along rows or columns, much faster	
x >= left & x <= right	between()	easier to read and faster implementation for larege datasets	see also near()
nrow(), sum()	tally(), count(), add_tally(), add_count()	count or sum up rows	
c()	combine()	combine into a vector	
extends base R	cumall(), cumany(), cummean()	extends base R collection of cumsum(), cumprod() etc	
mtcars$mpg[1,] etc	first(), last(), n(), top_n()	works within groups, allows you to order by another column(s) and provide defaults for missing values	
split(), aggregate()	group_by()	create a grouped data frame (tibble) to perform operations on groups	see also ungroup()
intersect(), union()	intersect(), union()	set operations, but dplyr works on data frames as well	
mtcars\$(mpg2 = c(NA, mtcars\$)mpg[1:nrow(mtcars)-1])	lead(), lag()	No equivalent command in base R, easier to read	
ifelse(…, NA)	na_if()	convert a value to NA	
switch()	recode()	change certain values in your vector	see also forcats package when dealing with factors
mtcars[3:5,]	slice()	select rows bases on row numbers	
seq_along(), quantile()	row_number(), ntile(), min_ran() etc	add rankings in various ways, much richer set of rankings supported than base r	
no easy way	complete(), expand()	expands the dataframe so that supplied columns are completely filled out	often used with nesting(), see also full_seq()
expand.grid()	crossing()	create a data frame of all possible combinations of supplied vectors	
ifelse(is.na(…), …)	drop_na(), replace_na()	drop rows with missing values or convert NAs to supplied values	see also fill(), coalesce()
some mix of paste/strsplit	separate(), unite()	separate two columns based on regex or combine two columns into one	
reshape2::dcast()	spread()	convert long (tidy) data into wide (untidy) format	
reshape2::melt()	gather()	convert wide (untidy) data into long(tidy) format	
replicate()	rerun()	run an expression n number of times	
unlist(lapply(x, [[, n))	pluck()	extract elements out of a list	
lapply(), sapply()	map(), map2()	apply a function to a set of values, working with lists	see also map_chr(), map_lgl(), map_int(), map_dbl(), map_df()
paste0()	glue()	combine two strings together, but much more powerful because it allows for expressions	

The RStudio Cheat Sheets



Data Transformation with dplyr : : CHEAT SHEET

dplyr functions work with pipes and expect **tidy data**. In tidy data:

Each **variable** is in its own **column** & Each **observation**, or **case**, is in its own **row**

pipes
x %>% f(y) becomes **f(x, y)**

Summarise Cases

These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

summary function

summarise(.data, …)
Compute table of summaries.
summarise(mtcars, avg = mean(mpg))

count(x, ..., wt = NULL, sort = FALSE)
Count number of rows in each group defined by the variables in … Also **tally**().
count(iris, Species)

VARIATIONS

summarise_all() - Apply funs to every column.
summarise_at() - Apply funs to specific columns.
summarise_if() - Apply funs to all cols of one type.

Group Cases

Use **group_by()** to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.

mtcars %>%
group_by(cyl) %>%
summarise(avg = mean(mpg))

group_by(.data, ..., add = FALSE)
Returns copy of table grouped by …
g_iris <- group_by(iris, Species)

ungroup(x, …)
Returns ungrouped copy of table.
ungroup(g_iris)

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

filter(.data, …) Extract rows that meet logical criteria. *filter(iris, Sepal.Length > 7)*

distinct(.data, ..., .keep_all = FALSE) Remove rows with duplicate values.
distinct(iris, Species)

sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame()) Randomly select fraction of rows.
sample_frac(iris, 0.5, replace = TRUE)

sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame()) Randomly select size rows. *sample_n(iris, 10, replace = TRUE)*

slice(.data, …) Select rows by position.
slice(iris, 10:15)

top_n(x, n, wt) Select and order top n entries (by group if grouped data). *top_n(iris, 5, Sepal.Width)*

Logical and boolean operators to use with filter()

<	<=	is.na()	%in%	|	xor()
>	>=	!is.na()	!	&	

See **?base::logic** and **?Comparison** for help.

ARRANGE CASES

arrange(.data, …) Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

ADD CASES

add_row(.data, …, .before = NULL, .after = NULL) Add one or more rows to a table.
add_row(faithful, eruptions = 1, waiting = 1)

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

pull(.data, var = -1) Extract column values as a vector. Choose by name or index.
pull(iris, Sepal.Length)

select(.data, …)
Extract columns as a table. Also **select_if**().
select(iris, Sepal.Length, Species)

Use these helpers with select (),
e.g. select(iris, starts_with("Sepal"))

contains(match) **num_range**(prefix, range) :, e.g. mpg:cyl
ends_with(match) **one_of**(…) -, e.g, -Species
matches(match) **starts_with**(match)

MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized funs take vectors as input and return vectors of the same length as output (see back).

vectorized function

mutate(.data, …)
Compute new column(s).
mutate(mtcars, gpm = 1/mpg)

transmute(.data, …)
Compute new column(s), drop others.
transmute(mtcars, gpm = 1/mpg)

mutate_all(.tbl, .funs, …) Apply funs to every column. Use with **funs()**. Also **mutate_if()**.
mutate_all(faithful, funs(log(.), log2(.)))
mutate_if(iris, is.numeric, funs(log(.)))

mutate_at(.tbl, .cols, .funs, …) Apply funs to specific columns. Use with **funs()**, **vars()** and the helper functions for select().
mutate_at(iris, vars(-Species), funs(log(.)))

add_column(.data, ..., .before = NULL, .after = NULL) Add new column(s). Also **add_count()**, **add_tally()**. *add_column(mtcars, new = 1:32)*

rename(.data, …) Rename columns.
rename(iris, Length = Sepal.Length)

RStudio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more with browseVignettes(package = c("dplyr", "tibble")) • dplyr 0.7.0 • tibble 1.2.0 • Updated: 2017-03

Thanks!

I created these slides with xaringan and RMarkdown using the rutgers css that I slightly modified.

Credit: I used material from Cécile Sauder, Stephanie J. Spielman and Julien Barnier.

olivier.gimenez@cefe.cnrs.fr
https://oliviergimenez.github.io/
@oaggimenez
@oliviergimenez