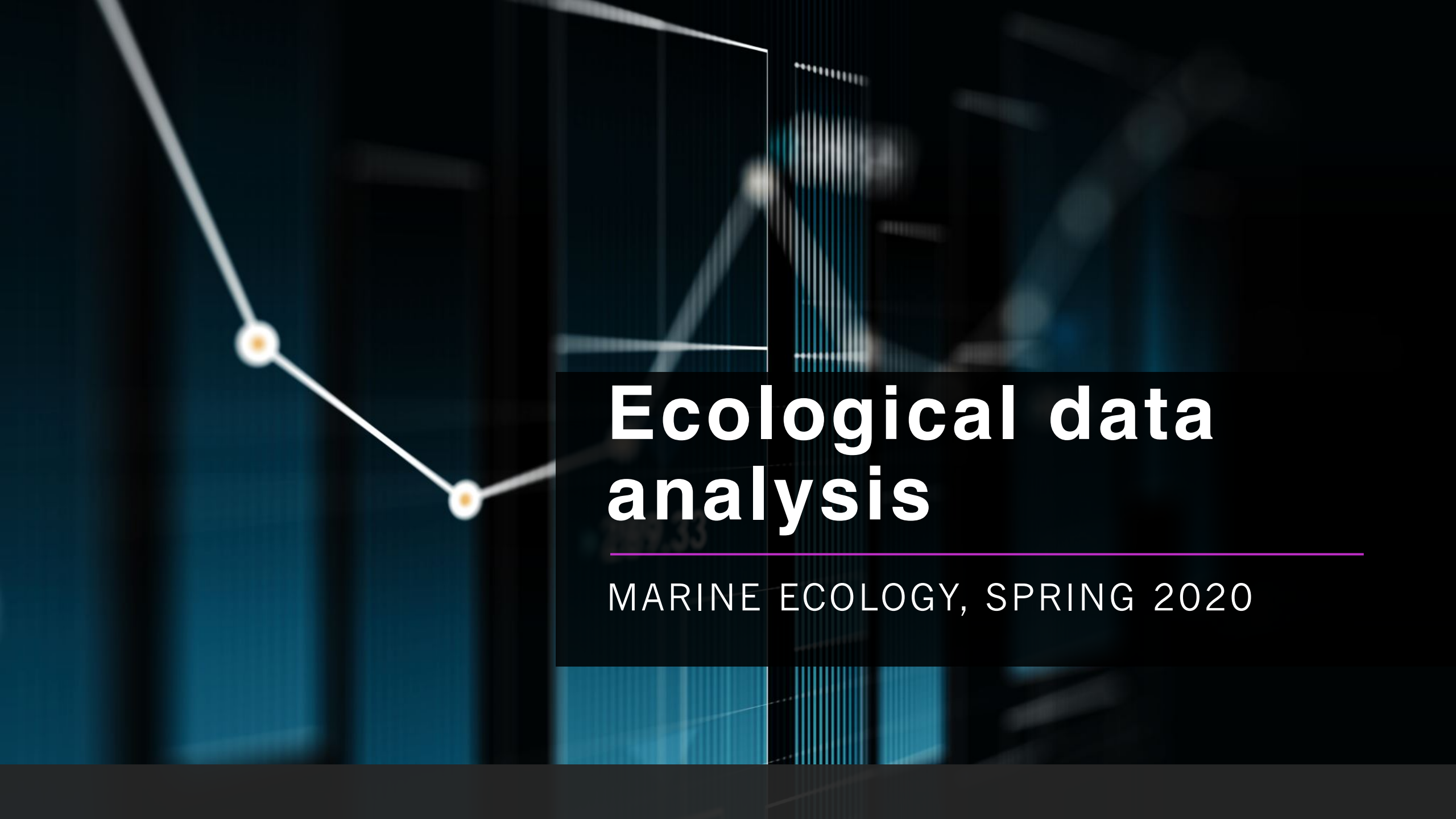


Ecological data analysis

MARINE ECOLOGY, SPRING 2020

This is a Data Expedition

- Part of the Rhodes Information Initiative
  - Duke initiative to teach students how to work with "big data”
- Data Expeditions are designed to introduce undergraduates to exploratory data analysis

https://bigdata.duke.edu/data-expeditions

By the end of this two-part series we hope you will…

1. Know what big data are
2. Understand why “big data” is such an increasingly talked-about topic in ecology and environmental science
3. Learn some methods for dealing with big data
    1. Have code you can recycle with fresh data
    2. Appreciate the importance of reproducibility data analysis
4. Be able to apply some of these methods (1) in this course with smaller datasets (i.e. your independent projects) and (2) to future work where you might need to deal with larger datasets

Data is a plural word

“THESE DATA SHOW”

“THE DATA ARE”

As environmental problems become increasingly global, researchers are talking more and more about “big data”

TO UNDERSTAND THE IMPACTS OF GLOBAL STRESSORS (E.G. CLIMATE CHANGE, INTERNATIONAL FISHING FLEETS), YOU NEED BIG, GLOBAL DATASETS

CONCEPTS AND QUESTIONS

156

Big data and the future of ecology

Stephanie E Hampton[1*], Carly A Strasser[2], Joshua J Tewksbury[3], Wendy K Gram[4], Amber E Budden[5],

CONCEPTS & THEORY

Harnessing the power of big data: infusing the scientific method with machine learning to transform ecology

Debra P. C. Peters,[1,†] Kris M. Havstad,[1] Judy Cushing,[2] Craig Tweedie,[3] Olac Fuentes,[4] and Natalia Villanueva-Rosales[4]

EDITORIAL

h big data:

data,

Big e

Jian-Lin Wang[3] · Lian-Biao Cui[1]

Big Data and Industrial Ecology

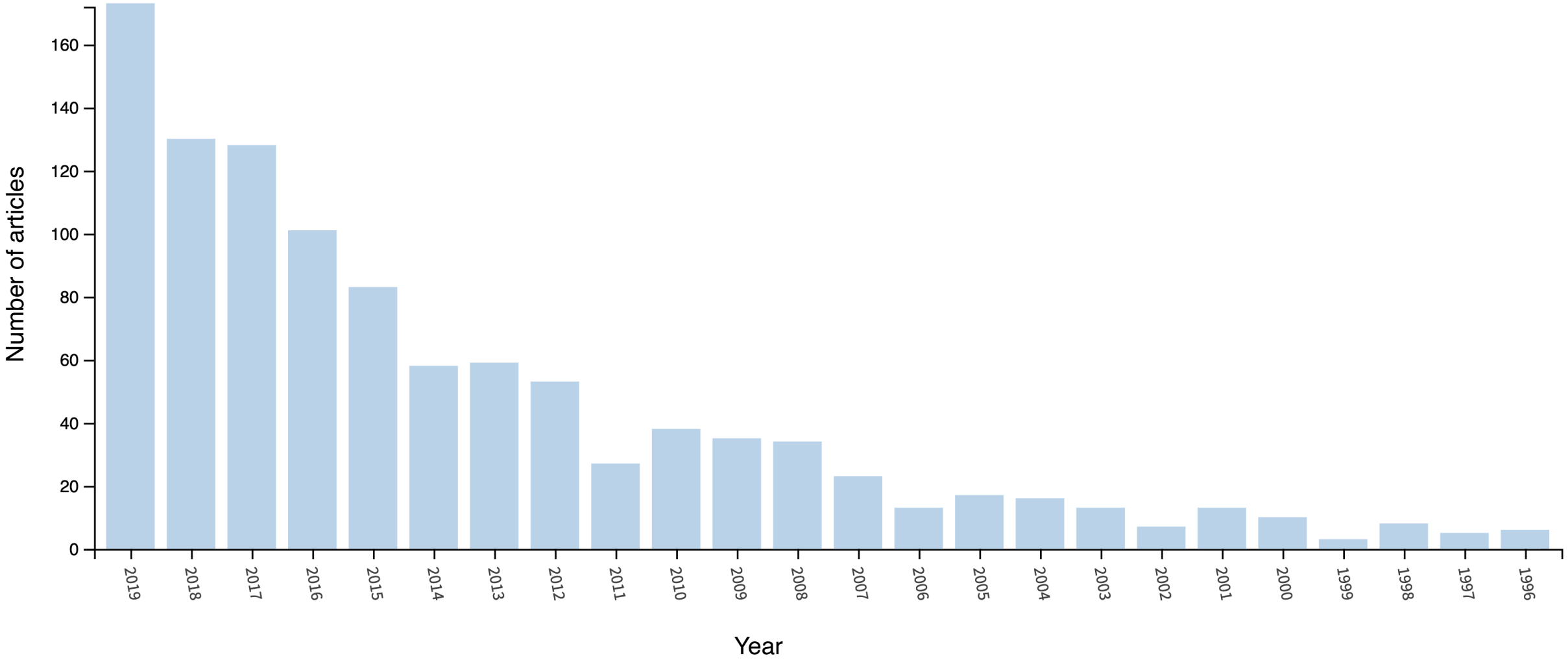
How Big Data Fast Tracked Human Mobility Research and the Lessons for Animal Movement Ecology


Michele Thums[1*], Juan Fernández-Gracia[2], Ana M. M. Sequeira[3], Víctor M. Eguíluz[2],

Carlos M. Duarte[3,4] and

Mark G. Meekan[1]

English articles with the topics *big data* and *ecology* have been increasing exponentially over the past few decades



Web of Science search January 9, 2020

What is “Big Data”?

- A lot of data. So much data you need to think of new ways to process them (think: not by hand, not in Excel, maybe not even on your local computer)

- How do you get “big data”?
  1. Combine small, fine-scale datasets to create detailed datasets
     - Downside: not necessarily geographically/temporally concurrent; can be hard to compare
     - Upside: highly-detailed
  2. Remote-sensing data sources (e.g. satellite and drone imagery)
     - Downside: Only certain variables can be measured and often those aren't even direct
     - Upside: Often span a large amount of space and time
  3. Combine sensor measurements taken over large areas and times (e.g. floats out in the ocean measuring oceanographic features, weather stations, etc.)
     - Downside: Some of the needed data may be private or scarce because instruments are expensive
     - Upside: Generally pretty standard/comparable values
  4. "Big science" initiatives that span large geographic/temporal scales and have standardized methods that can be used to answer large ecological questions
     - Citizen science programs: eBird, Nature’s Notebook, iNaturalist
     - Big government-funded initiatives: NEON (National Ecological Observatory Network), LTERs (Long-term Ecological Research sites)
     - Downside: Extremely costly, difficult to coordinate
     - Upside: Generally standardized (although some irregularities exist), intentionally designed

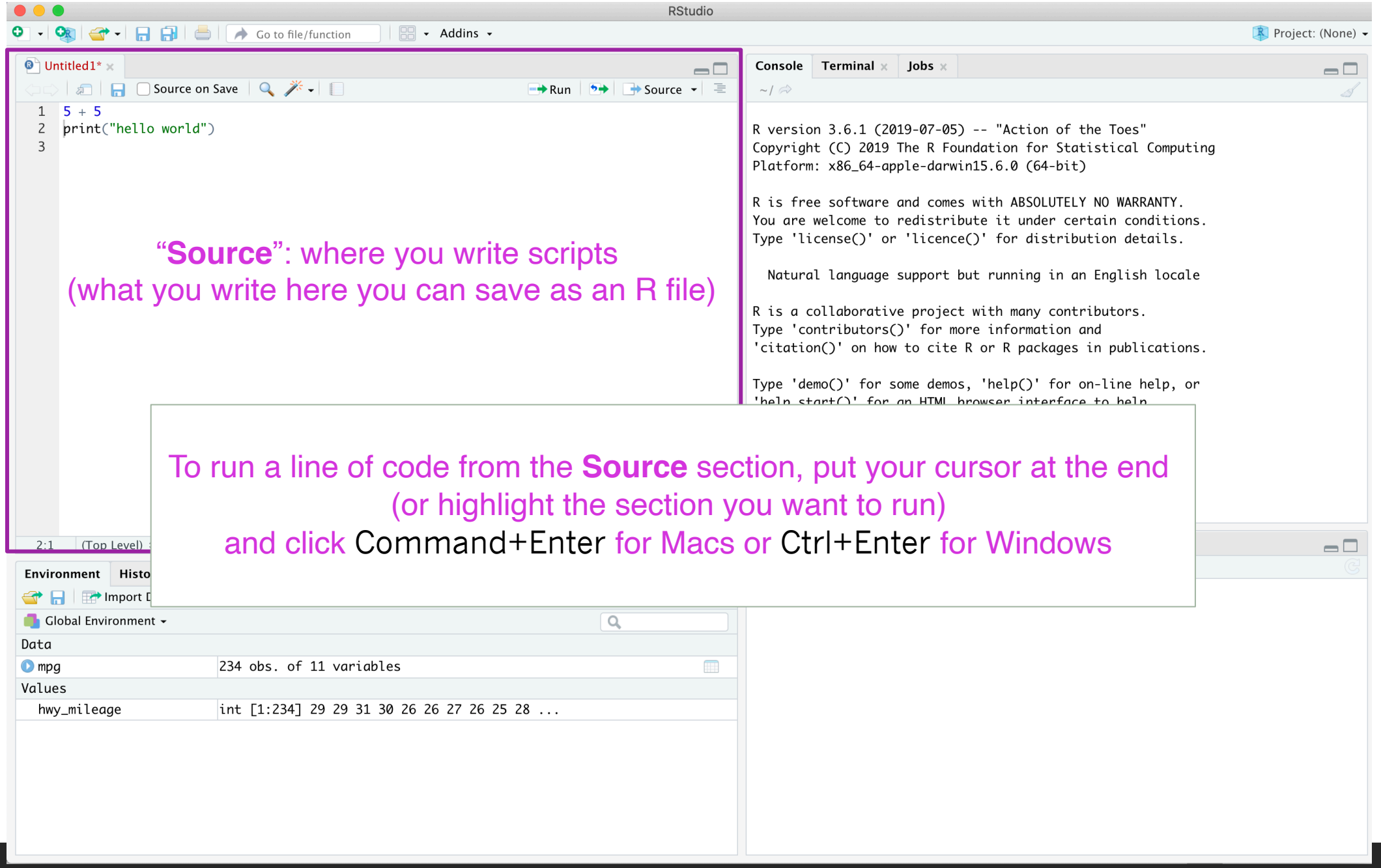
(Soranno and Schimel, 2014 *Front in Ecol and Environ*)

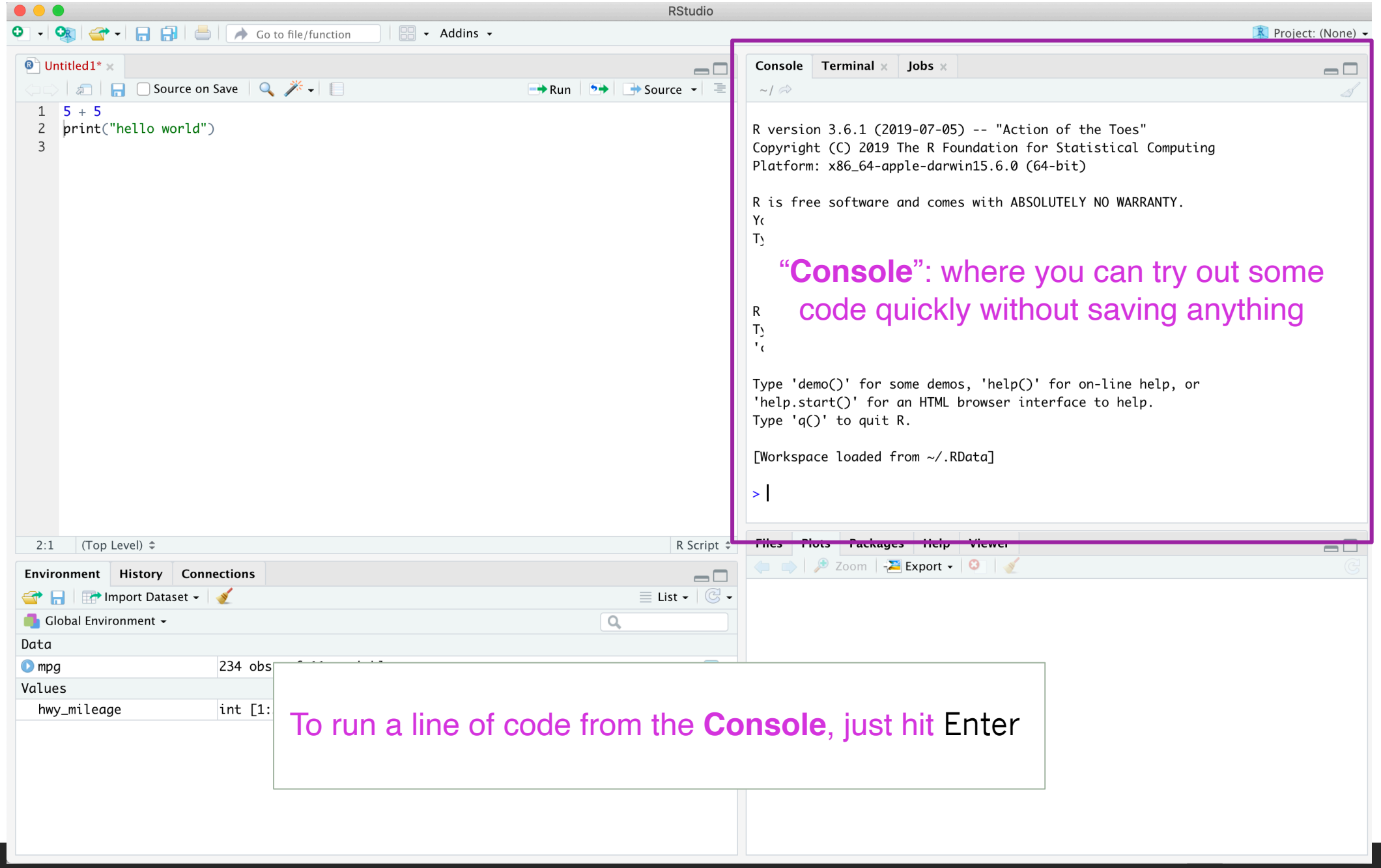
How do we deal with big data?

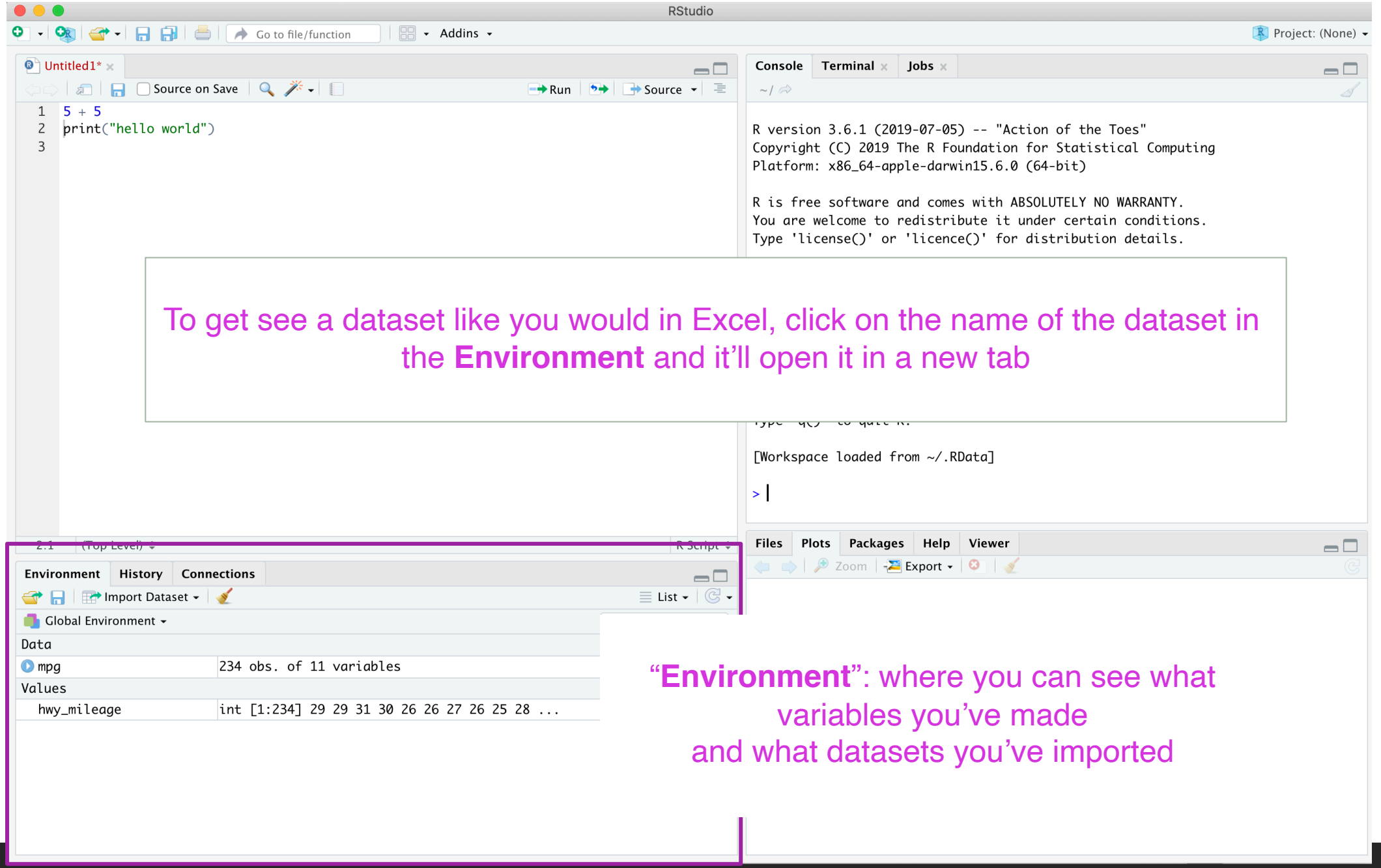
- Computers !!
  - Calculator/paper/pen → GUI-based software (Excel/JMP/Stata/etc.) → Script-based (R, Python, MATLAB, etc.)
- Why? Because most science can't be repeated
  - Need to be 100% transparent in what you're doing and be able to check settings, filters, etc.
  - Scripts provide a digital copy of what you've done with notes to you and others so you remember why you did what you did
    - Your steps are written out and anyone can run them line-by-line to see what happens
    - Now journals are starting to require publishing both your code and your original data for accountability
  - Do not manipulate original data: one wrong step and you could mess up the entire chain of analysis!
    - There are horror stories of this—don't be this person if you can help it
    - Even more extreme versions: checksum files, locking original data, etc.

R: a computing environment beloved (mostly) by ecologists everywhere

- **R** is a computer “language” or ”computing environment”
  - It has a bunch of functions built into it (like a very high-tech calculator) that allow you to manipulate, visualize, and run statistics on your data
  - You can import your data from Excel or a CSV file (basically a single sheet of an Excel file saved in a format called Comma-separated values) and then work with it in R without changing anything in the original file
- Most people use an interface called **R Studio** that makes it easier to work in R


RStudio
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 5 + 5
2 print("hello world")
3
"Source": where you write scripts
(what you write here you can save as an R file)
Console
Terminal
Jobs
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.
Natural language support but running in an English locale
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
To run a line of code from the Source section, put your cursor at the end
(or highlight the section you want to run)
and click Command+Enter for Macs or Ctrl+Enter for Windows
Environment
Global Environment
Data
mpg
234 obs. of 11 variables
Values
hwy_mileage
int [1:234] 29 29 31 30 26 26 27 26 25 28 ...


RStudio
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 5 + 5
2 print("hello world")
3
Console
Terminal
Jobs
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
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.
[Workspace loaded from ~/.RData]
>
"Console": where you can try out some code quickly without saving anything
2:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Data
mpg
234 obs
Values
hwy_mileage
int [1:
Files
Plots
Packages
Help
Viewer
Zoom
Export
To run a line of code from the Console, just hit Enter

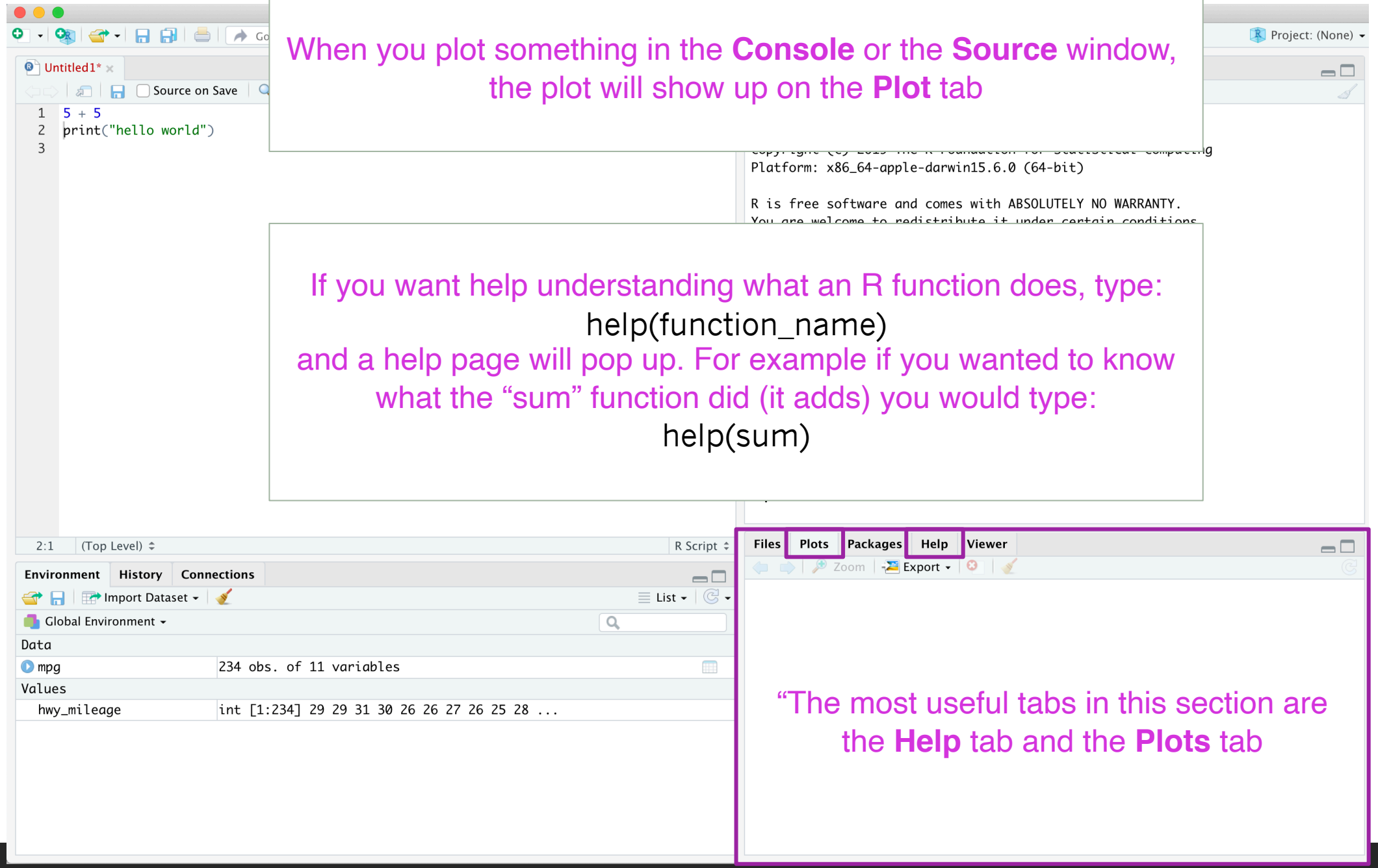


To get see a dataset like you would in Excel, click on the name of the dataset in the **Environment** and it'll open it in a new tab

"**Environment**": where you can see what variables you've made and what datasets you've imported

When you plot something in the **Console** or the **Source** window, the plot will show up on the **Plot** tab

If you want help understanding what an R function does, type:
help(function_name)
and a help page will pop up. For example if you wanted to know what the “sum” function did (it adds) you would type:
help(sum)



“The most useful tabs in this section are the **Help** tab and the **Plots** tab

sum {base} R Documentation

Sum of Vector Elements

Description

sum returns the sum of all the values present in its arguments.

Usage

```
sum(..., na.rm = FALSE)
```

Arguments

...	numeric or complex or logical vectors.
na.rm	logical. Should missing values (including NaN) be removed?

Details

This is a generic function: methods can be defined for it directly or via the Summary group generic. For this to work properly, the arguments ... should be unnamed, and dispatch is on the first argument.

If na.rm is FALSE an NA or NaN value in any of the arguments will cause a value of NA or NaN to be returned, otherwise NA and NaN values are ignored.

Logical true values are regarded as one, false values as zero. For historical reasons, NULL is accepted and treated as if it were integer(0).

Loss of accuracy can occur when summing values of different signs: this can even occur for sufficiently long integer inputs if the partial sums would cause integer overflow. Where possible extended-precision accumulators are used, typically well supported with C99 and newer, but possibly platform-dependent.

Value

There are different data types in R

- You can only do certain things with certain data types (e.g. addition is only good for numbers)
- **Characters**
  - e.g. "one", "two", "Julianna is cool"
- **Logical**
  - TRUE or FALSE (what you would get if you said 1 == 2)
- **Numerical**
  - e.g. -101, 1, 2, 3, 4, 5, 23, 156.7
- **Vectors** are a fancy word for lists (multiple things grouped together, denoted by *c(your_list_here)*). However, unlike a list, everything in the vector has to be the same type (e.g. all numbers or all characters)
  - e.g. c(1, 2, 3, 4, 5) is a vector with a length of 5
  - Since each of these are separate, it's easy to grab, say, the 4^{th} item (number 4) if you want it
- **Data frames** are like Excel tables. They have rows and columns

In R, you save everything you have to variables

- To create a variable, type what you want to call the variable, type an arrow, “<-” and then type what you want to assign to that variable

```
> x <- c("hay", "is", "for", "horses", "and", "chicken", "and", "fish")
```

- When you import your data, you’ll also assign that dataset a variable name
  - Then when you want to look at your data or do something to it (e.g. plot it), you can just use the variable name

Functions do work on your variables

- Once you have your data in one place saved to a variable, you'll want to do something with them
  - This is where functions come in
  - Some functions are simple (e.g. sum) and others are complicated (e.g. lmer)
- Most functions have the name of a function followed by parentheses
  - E.g. sum() is the function that adds multiple numbers together
  - **sum(2, 3, 4, 5)** returns **14**
    - In this case, 2, 3, 4, and 5 are the **arguments** of the function
    - Each function has different arguments that tell the function exactly what you want it to do
    - If you type help(function_name), you'll get the help page for the function, including an explanation of what each argument should be

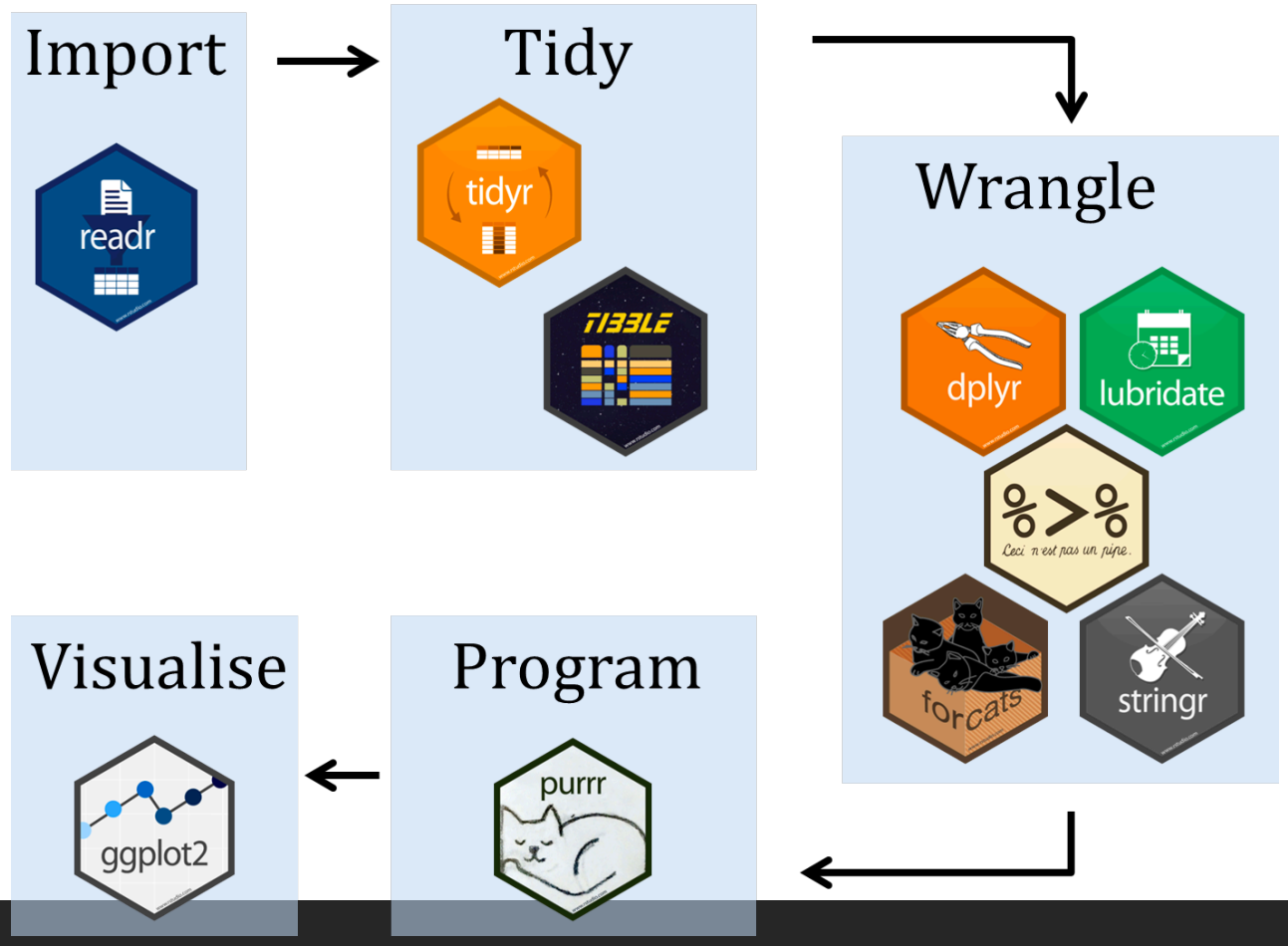
Arguments

mode	character string naming an atomic mode or "list" or "expression" or (except for vector) "any". Currently, is.vector() allows any type (see typeof) for mode, and when mode is not "any", is.vector(x, mode) is almost the same as typeof(x) == mode.
length	a non-negative integer specifying the desired length. For a long vector, i.e., length > .Machine$integer.max, it has to be of type "double". Supplying an argument of length other than one is an error.
x	an R object.

Packages are collections of functions

- "Base R" has all the basic functions the initial developers thought we needed built-in
- But as folks decide what other things they want to do with their data (other ways to move data around, new modeling techniques, etc.), they write their own functions
  - Many will group a set of functions together into a "package" that you can download off of the internet
  - There are linear modeling packages, plotting packages, ecology-specific packages, fish color packages etc.

The “Tidyverse” is a set of packages that folks use for working with data



“Happy families are all alike; every unhappy family is unhappy in its own way.” — Leo Tolstoy

“Tidy datasets are all alike, but every messy dataset is messy in its own way.” — Hadley Wickham

To be tidy, a dataset must meet three requirements:

1. Each variable is in its own column
2. Each observation is in its own row.
3. Each value is in its own cell.

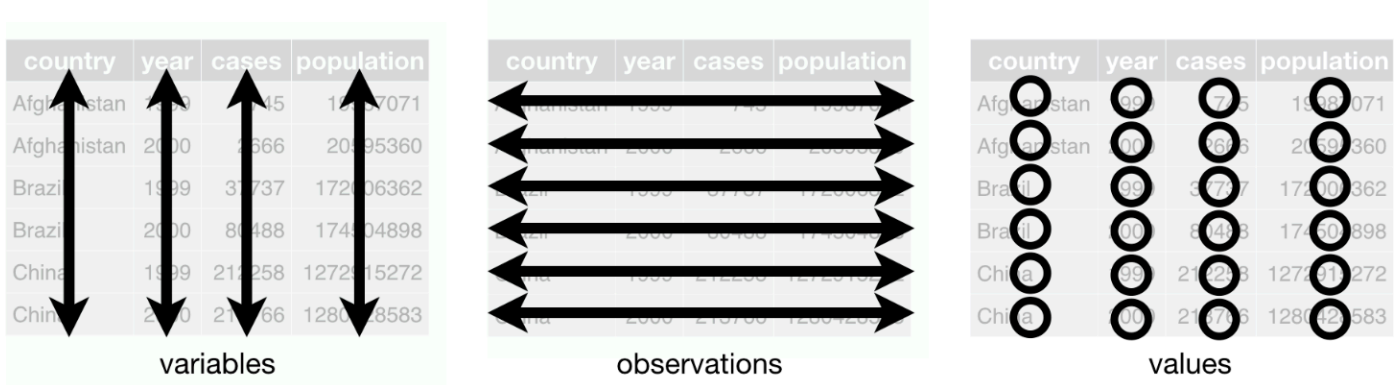


Figure 12.1: Following three rules makes a dataset tidy: variables are in columns, observations are in rows, and values are in cells.

Taken directly from: https://r4ds.had.co.nz/tidy-data.html

Tidyverse

- The tidyverse includes ggplot2 (graphics), dplyr (data manipulation), tidyr (tidy format data), tibble (better version of data frames)
- Using Tidyverse is less like memorizing a bunch of haphazard commands and more like using a few general, but useful commands that you can string together logically using "**pipes**"
  - A pipe is this: %>%
  - It funnels the output of one function into another
  - Using pipes, you can string together a bunch of simple functions to do whatever you need

R For Data Science *Cheat Sheet*

Tidyverse for Beginners

Learn More R for Data Science Interactively at www.datacamp.com

(on Sakai for your reference)

Let’s practice with some real data

Using large-scale citizen science data to look at phenological changes over time

BIG DATA IN ACTION !!

What is phenology?

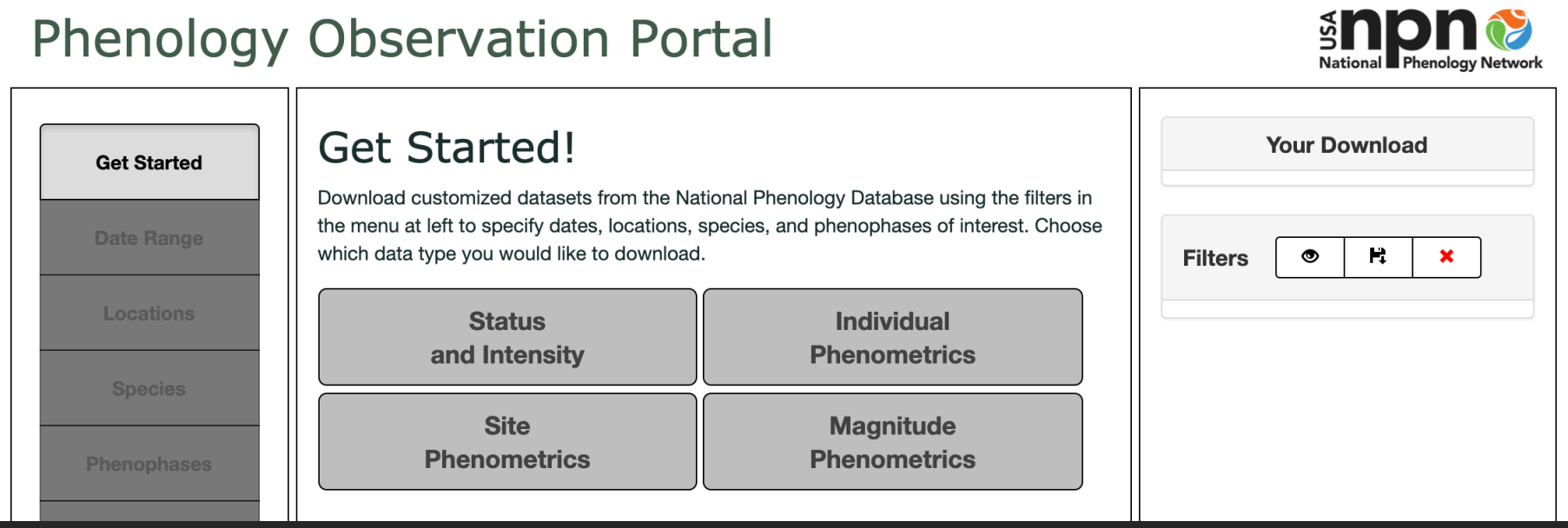
- Phenology is the study of the timing of lifecycle events
- We're seeing changes in phenology globally under changing climate conditions
  - Spring onset
  - Potential species mismatches (e.g. plant-pollinator, predator-prey)

Nation-wide phenology data

- *Nature's Notebook* is a citizen science program run by the USA National Phenology Network (USA-NPN)
- Established in 2007 with the vision of: *"providing data and information on the timing of seasonal events in plants and animals to ensure the well-being of humans, ecosystems, and natural resources"*
- The NPN's observation portal has 17.3 million phenological observations (that's Big Data)
  - 76,100 are aquatic
  - Each observation has 20 default fields, along with 21 optional satellite-derived climate fields, and 25 optional record-related fields

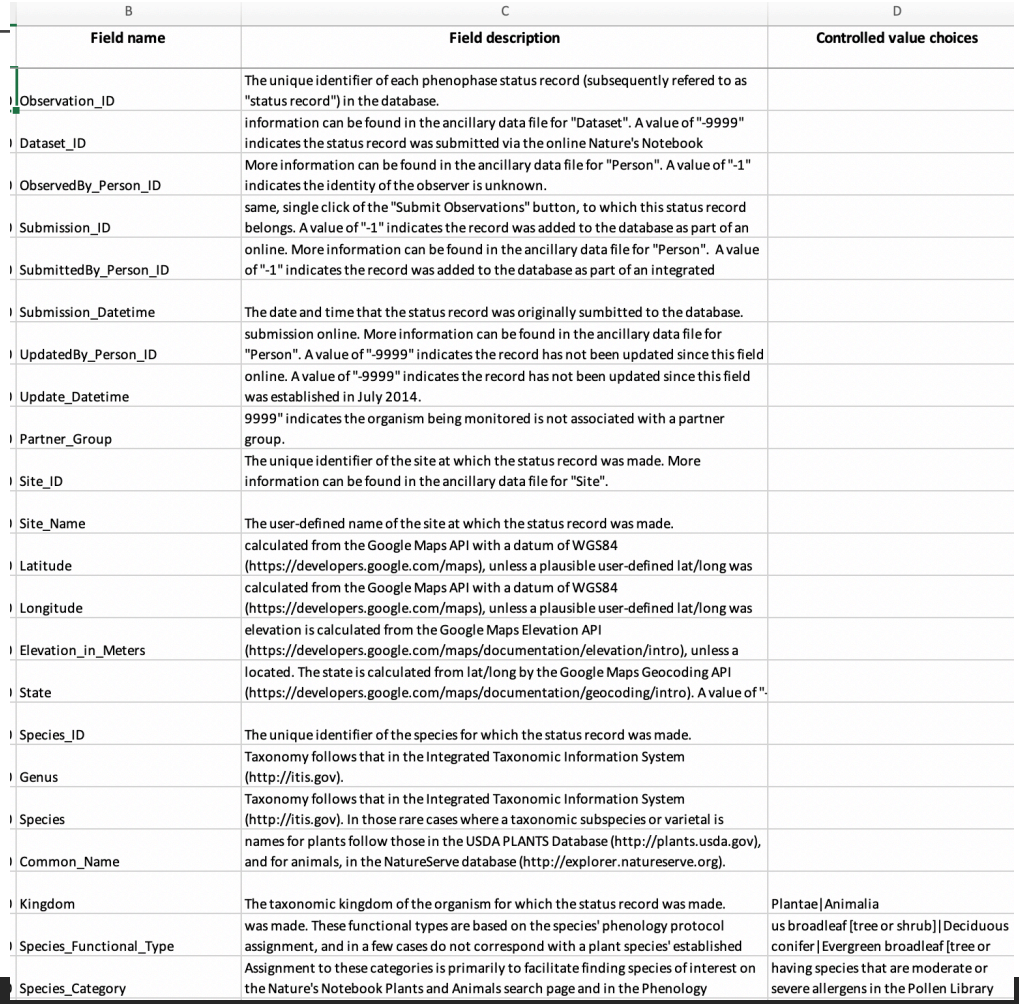
How can you access the data?

- They're free! Amazing.
- Go to the online Phenology Observation Portal (POP) on the NPN website: https://data.usanpn.org/observations/ (we'll do this now)



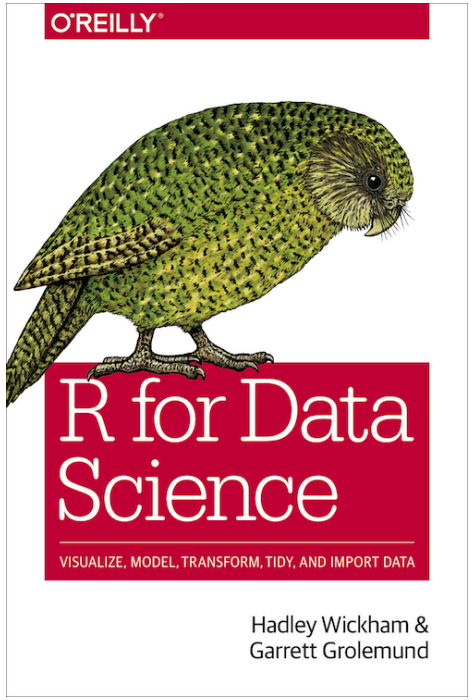
Metadata are important

- Metadata are data about data
- If you are downloading someone else's metadata, you 100%-definitely-absolutely need to read what they wrote about how the data were collected and how they're formatted
- If you collect your own data, WRITE YOUR OWN METADATA AS SOON AS YOU START ENTERING DATA—your collaborators and future self will thank you
  - Be as detailed as possible
  - Explain each field
- There are many metadata standards you can use: http://www.dcc.ac.uk/resources/metadata-standards/list

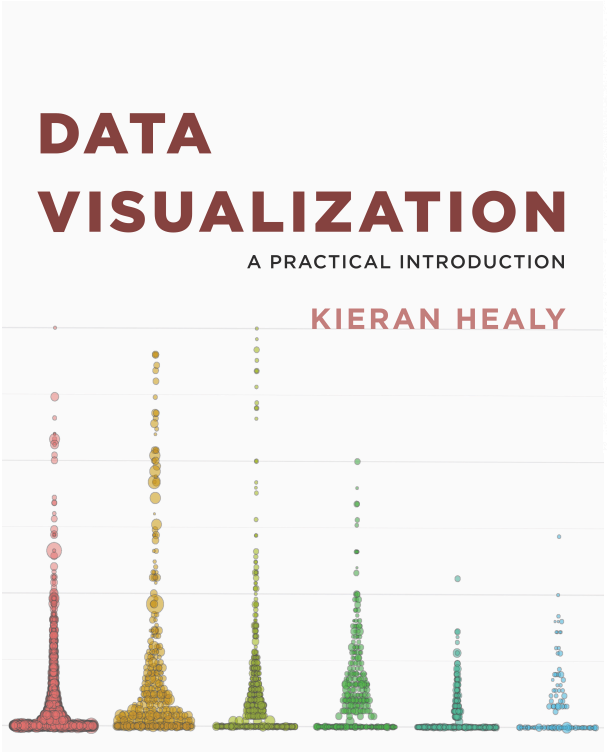


B	C	D
Field name	**Field description**	**Controlled value choices**
Observation_ID	The unique identifier of each phenophase status record (subsequently refered to as "status record") in the database.	
Dataset_ID	information can be found in the ancillary data file for "Dataset". A value of "-9999" indicates the status record was submitted via the online Nature's Notebook	
ObservedBy_Person_ID	More information can be found in the ancillary data file for "Person". A value of "-1" indicates the identity of the observer is unknown.	
Submission_ID	same, single click of the "Submit Observations" button, to which this status record belongs. A value of "-1" indicates the record was added to the database as part of an	
SubmittedBy_Person_ID	online. More information can be found in the ancillary data file for "Person". A value of "-1" indicates the record was added to the database as part of an integrated	
Submission_Datetime	The date and time that the status record was originally sumbitted to the database.	
UpdatedBy_Person_ID	submission online. More information can be found in the ancillary data file for "Person". A value of "-9999" indicates the record has not been updated since this field	
Update_Datetime	online. A value of "-9999" indicates the record has not been updated since this field was established in July 2014.	
Partner_Group	9999" indicates the organism being monitored is not associated with a partner group.	
Site_ID	The unique identifier of the site at which the status record was made. More information can be found in the ancillary data file for "Site".	
Site_Name	The user-defined name of the site at which the status record was made.	
Latitude	calculated from the Google Maps API with a datum of WGS84 (https://developers.google.com/maps), unless a plausible user-defined lat/long was	
Longitude	calculated from the Google Maps API with a datum of WGS84 (https://developers.google.com/maps), unless a plausible user-defined lat/long was	
Elevation_in_Meters	elevation is calculated from the Google Maps Elevation API (https://developers.google.com/maps/documentation/elevation/intro), unless a	
State	located. The state is calculated from lat/long by the Google Maps Geocoding API (https://developers.google.com/maps/documentation/geocoding/intro). A value of "-	
Species_ID	The unique identifier of the species for which the status record was made.	
Genus	Taxonomy follows that in the Integrated Taxonomic Information System (http://itis.gov).	
Species	Taxonomy follows that in the Integrated Taxonomic Information System (http://itis.gov). In those rare cases where a taxonomic subspecies or varietal is	
Common_Name	names for plants follow those in the USDA PLANTS Database (http://plants.usda.gov), and for animals, in the NatureServe database (http://explorer.natureserve.org).	
Kingdom	The taxonomic kingdom of the organism for which the status record was made.	Plantae|Animalia
Species_Functional_Type	was made. These functional types are based on the species' phenology protocol assignment, and in a few cases do not correspond with a plant species' established	us broadleaf [tree or shrub]|Deciduous conifer|Evergreen broadleaf [tree or
Species_Category	Assignment to these categories is primarily to facilitate finding species of interest on the Nature's Notebook Plants and Animals search page and in the Phenology	having species that are moderate or severe allergens in the Pollen Library

Have questions? Want more?



https://r4ds.had.co.nz/



https://socviz.co/

stackoverflow

Google

Coding is for everyone

Pull up R Studio

- Download **Aquatic_flower_leaves_East_coast.csv** off of Sakai
  - It's under *Resources* in the *Ecological_data_analysis* folder, which is in the *NPN_files* folder
  - You can check it out in Excel if you want/that's a more familiar way to start
- Open a new R script