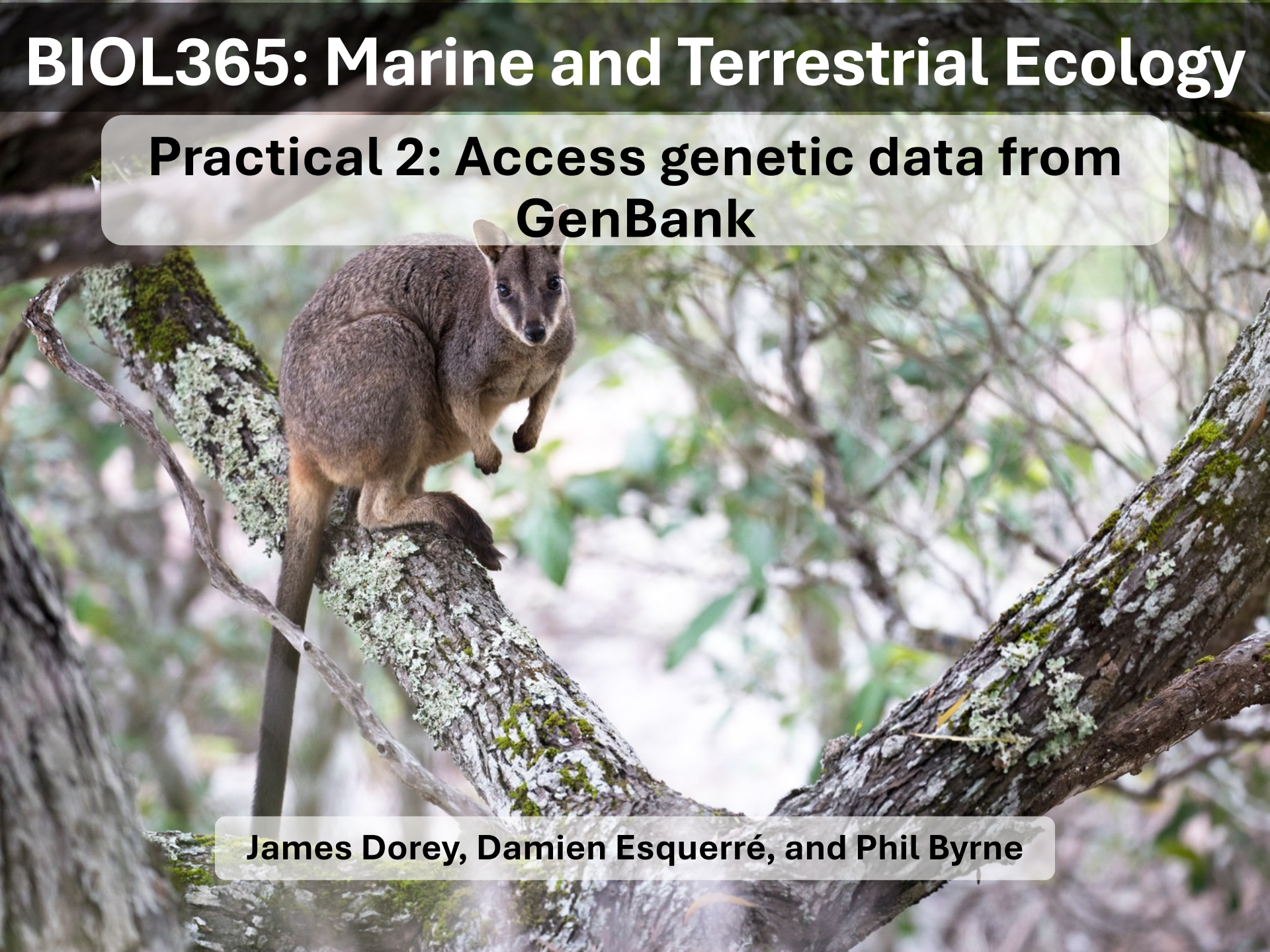
BIOL365: Marine and Terrestrial Ecology

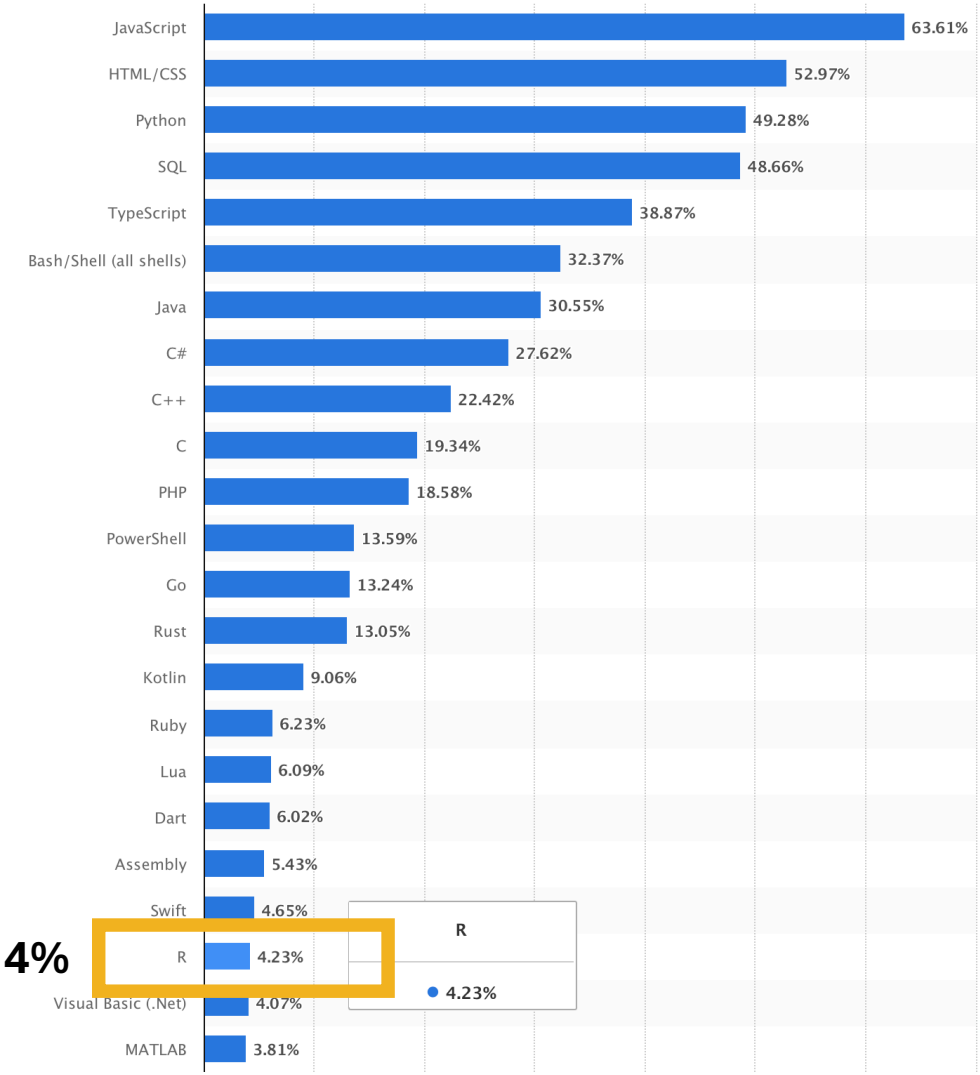
Practical 2: Access genetic data from GenBank



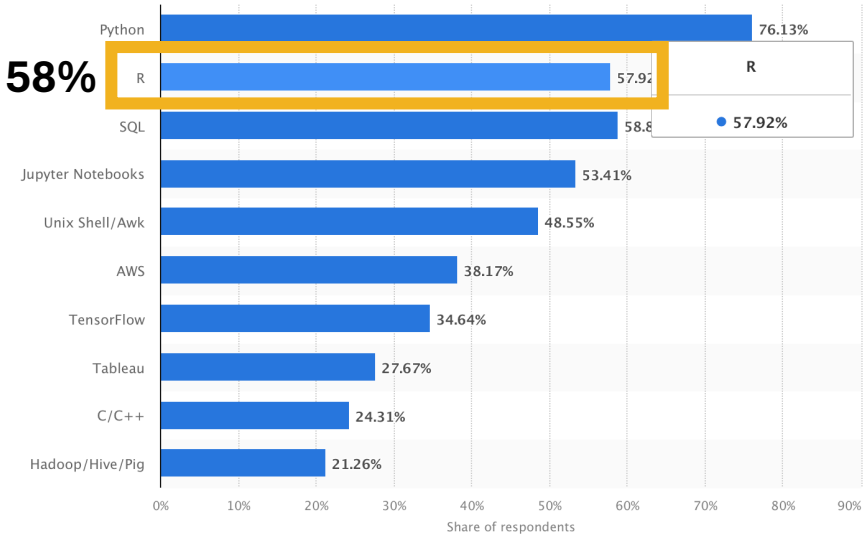
James Dorey, Damien Esquerré, and Phil Byrne

Intro to coding

Most used programming languages among **developers** worldwide as of 2023



LinkedIn's most wanted **data science** skills in United States as of April 2019



R is one of the most-commonly used **Data Science** skills

- Perhaps most common in ecology and evolution

ChatGPT says: R > Python > MATLAB > Julia

Intro to R

- S language developed in 1976 (was commercial $)
- R released (FREE) in 1995
  - stable version in 2000
- RStudio released 2011
- What makes R powerful?
  1. Open source and free
  2. Developers (>21,000 packages)
  3. A huge amount of help and support
  4. Runs on Mac, PC, Linux
  5. Reproducible

Intro to base R

- Base R (and the default packages)
  - **base**
  - compiler, datasets, grDevices, graphics, grid, methods, parallel, splines, **stats**, stats4, tcltk, tools, translations, and utils
  - This is what you get when you start R

Intro to CRAN+

- There's more to be had!
- CRAN
  - Has >21,000 packages
  - Run by volunteers
  - Very strict upload policies — **Tell me why**
- Bioconductor
  - Especially for **GENETICS** R packages
  - 3,700 packages
- GitHub
  - Wild west
  - ??? Packages
  - Great tool

base R syntax

- Comments — these are note read by R
  - **#** *text*
- Assigning variables
  - isNowThis **<-** This
  - isNowThis **=** This (usually only used in functions)
- Data types
  - Numeric (**1.23** or **9023.12**)
  - Logical (**TRUE** or **FALSE**)
  - Character (**“Sup nerds?”**)
- Vectors (strings of data types)
  - numericVector <- **c(1, 2, 3, 4, 5)**
  - logicalVector <- **c(TRUE, FALSE, FALSE)**
  - characterVector <- **c(“Sup”, “nerds”)**

base R syntax

- Lists
  - myList <- **list(name = “James”,**
    topic = “BIOL365”,
    Week = 2)
- Data frames
  - myDataFrame <- **data.frame("column1" = c(1,2,3),**
    "column2" = c("a", "b", "c"))

base R syntax

- Extracting data
- We can take an element from a vector by position
  - characterVector <- c(“Sup”, “nerds”)
  - characterVector**[2]** = “nerds”
- We can take an element from a data frame by position or name
  - myDataFrame <- data.frame("column1" = c(1,2,3),
    "column2" = c("a", "b", "c"))
  - By row myDataFrame**[1,]** = **column1** **column2**
    1 a
  - By column myDataFrame**[,1]** = 1, 2, 3
  - Or both myDataFrame**[1,1]** = 1
  - By column name myDataFrame**$**column1 = 1, 2, 3

The tidyverse

A quick note

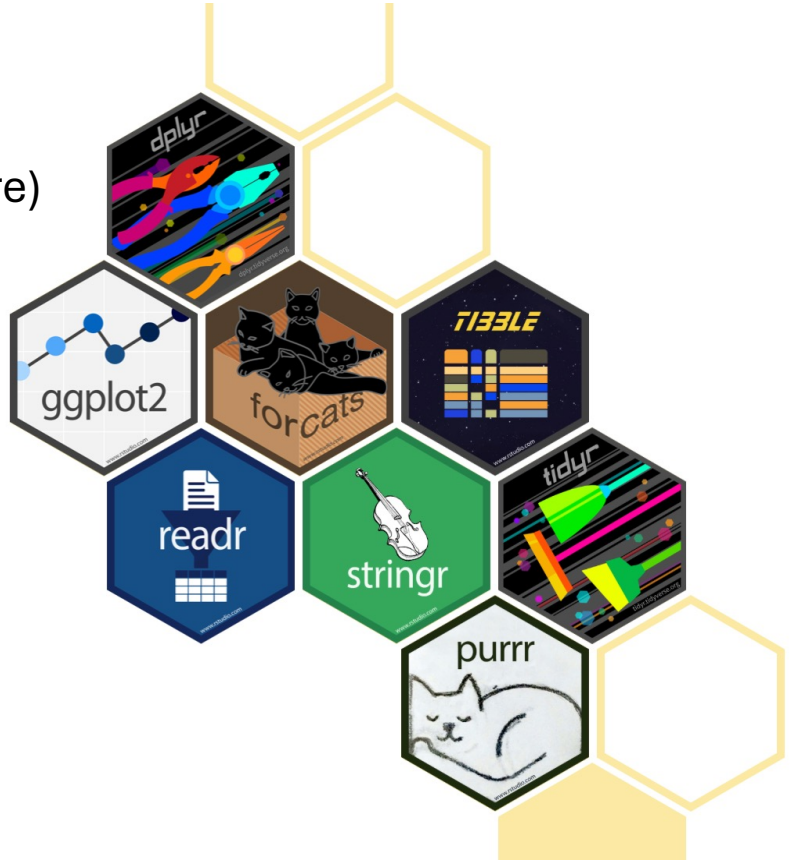
- library()
  - Makes your package active
- Package::function()
  - Negates the need for a library call
  - Makes certain that you're using the right package

```
Help on topic 'plot' was found in the following packages:

The Default Scatterplot Function
        (in package graphics in library
        /Library/Frameworks/R.framework/Versions/4.4-
        x86_64/Resources/library)
plot sf object
        (in package sf in library /Users/jamesdorey/Library/R/x86_64/4.4)
Make a map
        (in package terra in library /Users/jamesdorey/Library/R/x86_64/4.4)
Generic X-Y Plotting
        (in package base in library /Users/jamesdorey/Library/Caches/org.R-
        project.R/R/renv/sandbox/macos/R-4.4/x86_64-apple-darwin20/2edc1867)
```

Intro to the *tidyverse*

- A suite of packages for data science
  - "*The* tidyverse *is an* ***opinionated*** *collection of R packages designed for data science.*"
  - In my opinion, *tidyverse* is easier to code like a human being
  - Often, *tidyverse* is faster than base R (but, it's not the fastest R package out there)



Intro to the *tidyverse*

- As much as genetics, I want to teach you **DATA MANAGEMENT**!
- dplyr example:

	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10
7	10	18
8	10	26
9	10	34
10	11	17
11	11	28
12	12	14
13	12	20
14	12	24

...50 rows

base:

```
baseFilter <- carsTibble[carsTibble$speed >= 5, 1:2]
```

...48 rows

	speed	dist
1	7	4
2	7	22
3	8	16
4	9	10
5	10	18
6	10	26
7	10	34
8	11	17
9	11	28
10	12	14
11	12	20
12	12	24
13	12	28
14	13	26

Dplyr:

```
dplyrFilter <- dplyr::filter(carsTibble, speed > 5)
```

Dplyr + magrittr:

```
dplyrFilter <- carsTibble %>%
               dplyr::filter(speed > 5)
```