INTRODUCTION TO R

Where did R come from?

- Thank you Ross Ihaka and Robert Gentleman
- Created to help teach statistics
- Successor to S (also named for its co-creators)
- Comprehensive R Archive Network (CRAN) contains source code and user-created packages
- R is commonly used across many disciplines
- Object-oriented programming (OOP)

Why is R so popular?

- It's FREE
- Generally excellent documentation
- Extremely flexible
- Peer reviewed, user-created packages that do just about anything you can think of (including telling your fortune)
- **IT'S A STEP-BY-STEP DOCUMENTATION OF HOW YOU'VE DONE YOUR ANALYSIS**

How do you get started?

- Think about how you're going to share your analysis when you reach the article/report/posting stage
- Archiving data and code is a strong suggestion or requirement of many journals now
- Other forms of publication also often require documentation of your analysis
- In this class, we will use GitHub, which is often used for research in progress
- Ways to archive
  - Zenodo
  - Dryad

Version control

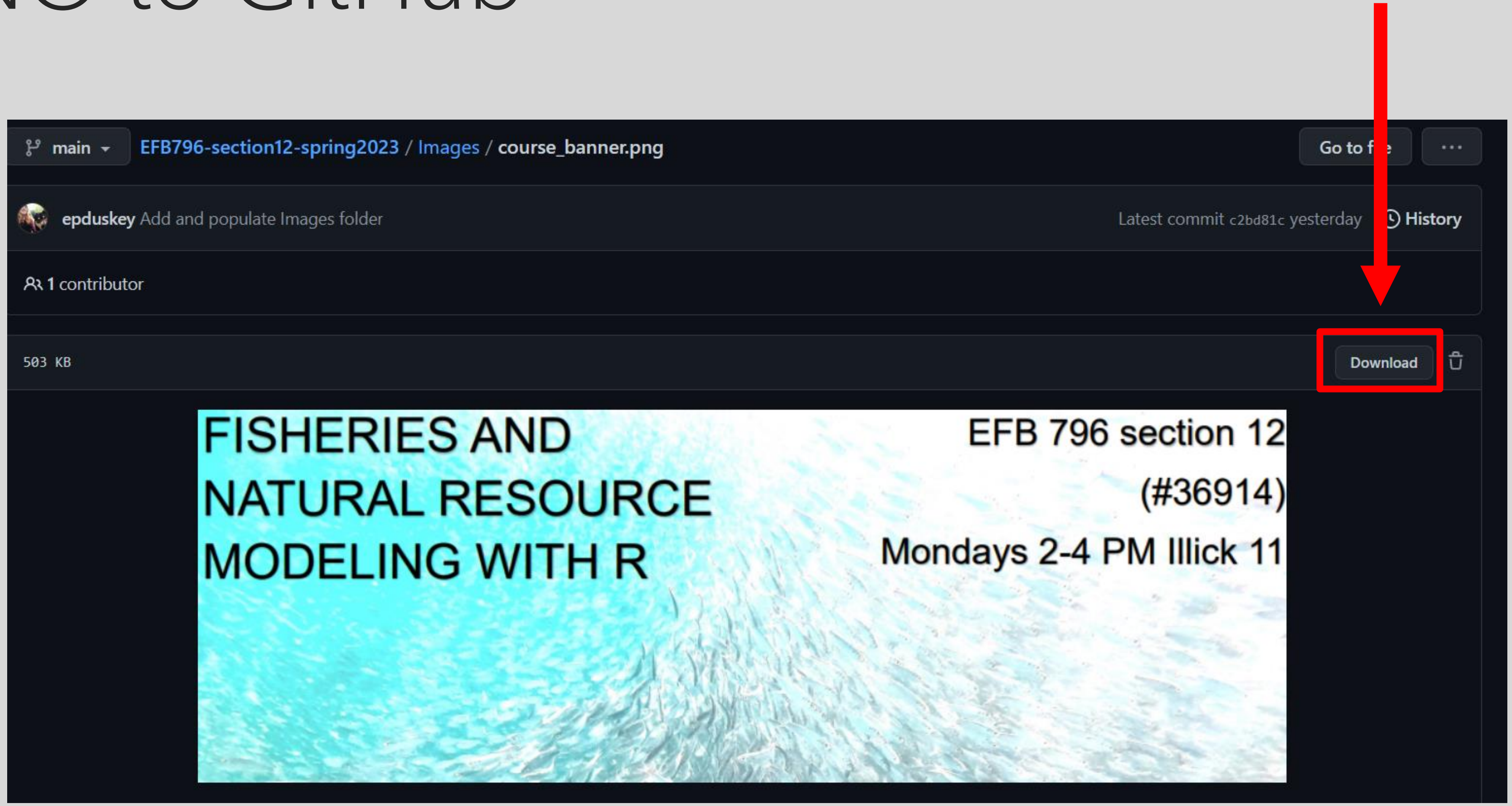
- Keeping track of changes that you and others have made to your code and data
- Sharing your analysis as it develops, either publicly or privately
- GitHub hosts many R packages, and you can follow them as they update, even make changes yourself
- You can clone* repositories** onto your own machine in R
- GitHub Tutorial: https://docs.github.com/en/get-started/quickstart/hello-world
- Connecting R and Git: https://happygitwithr.com/index.html

Two options

- Download individual files
- Pros
  - Don't have to bother learning much about GitHub
- Con
  - You will have to copy/paste code into your own script files

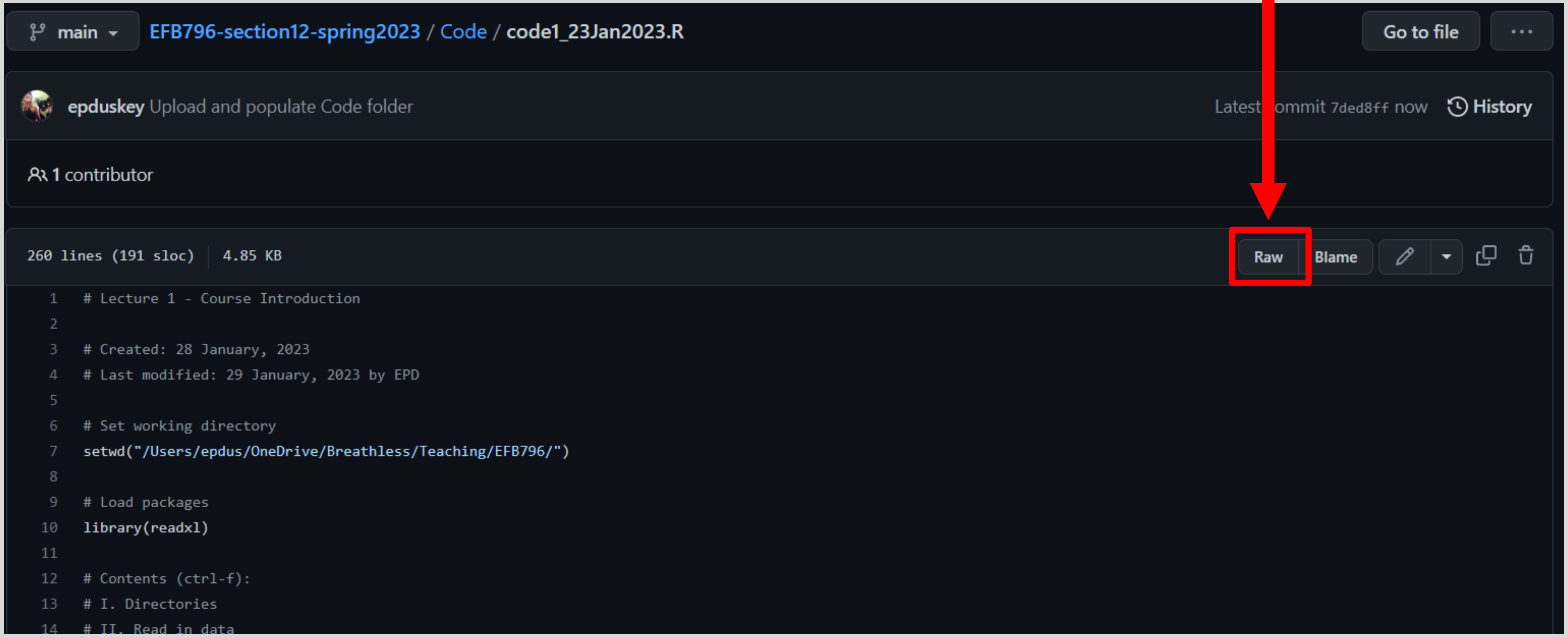
- Clone the repository
- Pros
  - Updates will appear automatically
  - You won't have to learn this stuff later when the journal you've submitted to demands compliance with open research policy
- Cons
  - Learning two things at once!

If NO to GitHub


main
EFB796-section12-spring2023 / Images / course_banner.png
Go to file
epduskey Add and populate Images folder
Latest commit c2bd81c yesterday
History
1 contributor
503 KB
Download
FISHERIES AND NATURAL RESOURCE MODELING WITH R
EFB 796 section 12 (#36914)
Mondays 2-4 PM Illick 11

If NO to GitHub

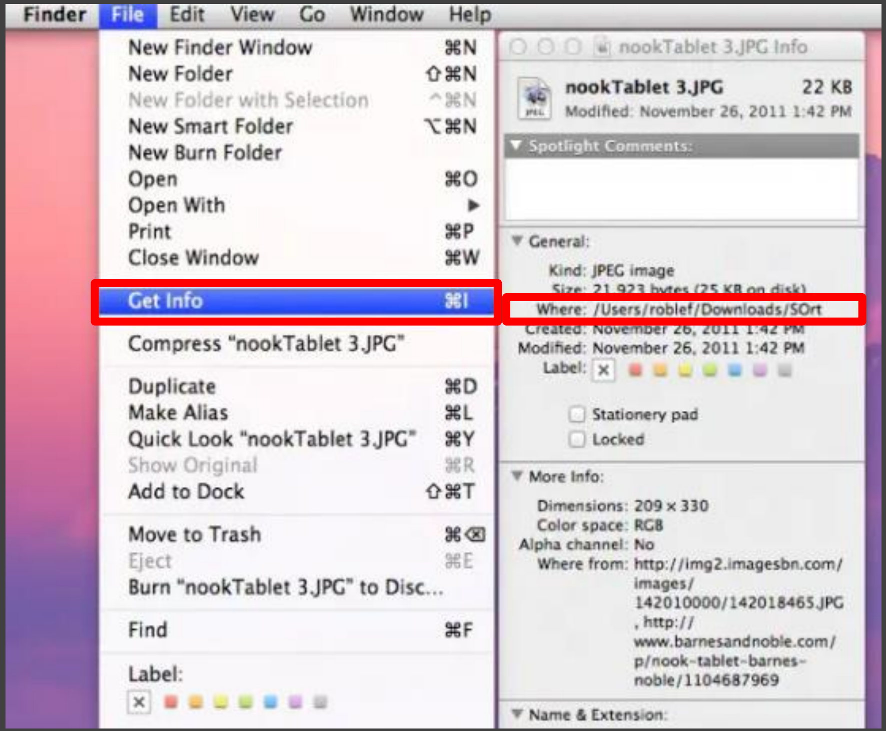
Then right-click and download OR copy/paste


main
EFB796-section12-spring2023 / Code / code1_23Jan2023.R
Go to file
epduskey Upload and populate Code folder
Latest commit 7ded8ff now
History
1 contributor
260 lines (191 sloc) | 4.85 KB
Raw
Blame

```
# Lecture 1 - Course Introduction

# Created: 28 January, 2023
# Last modified: 29 January, 2023 by EPD

# Set working directory
setwd("/Users/epdus/OneDrive/Breathless/Teaching/EFB796/")

# Load packages
library(readxl)

# Contents (ctrl-f):
# I. Directories
# II. Read in data
```

Beginning with R

- General workflow
  1. Set your working directory
  2. Load the packages you need
  3. Load the data you need
  4. Manipulate and edit data
  5. Perform analysis ↔ debugging
  6. Function-ize it
  7. Clean-up and organization



nookTablet 3.JPG Info

nookTablet 3.JPG 22 KB
Modified: November 26, 2011 1:42 PM

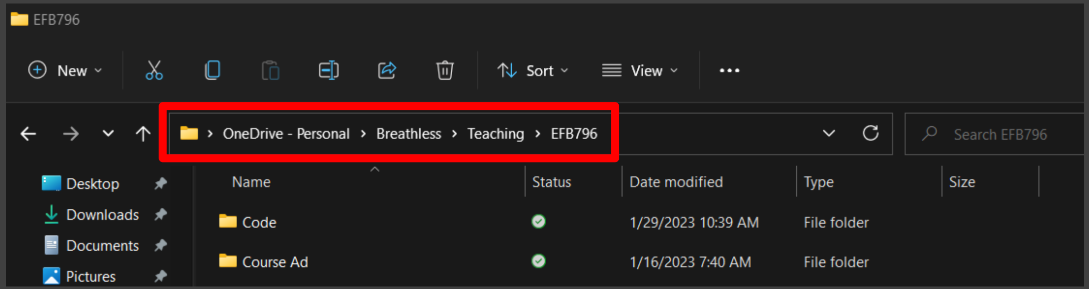
Spotlight Comments:

General:
Kind: JPEG image
Size: 21,923 bytes (25 KB on disk)
Where: /Users/roblef/Downloads/SOrt
Created: November 26, 2011 1:42 PM
Modified: November 26, 2011 1:42 PM
Label:

Stationery pad
Locked

More Info:
Dimensions: 209 × 330
Color space: RGB
Alpha channel: No
Where from: http://img2.imagesbn.com/images/142010000/142018465.JPG , http://www.barnesandnoble.com/p/nook-tablet-barnes-noble/1104687969

Name & Extension:



OneDrive - Personal › Breathless › Teaching › EFB796

Name	Status	Date modified	Type	Size
Code	✓	1/29/2023 10:39 AM	File folder	
Course Ad	✓	1/16/2023 7:40 AM	File folder	

HOW TO FIND YOUR WORKING DIRECTORY

Copy file path in file finder on both Windows and Mac

Packages in R

- Find a package
  - Search engine
  - Use a package finder package to find a package
- Use a package
  - See documentation at: https://cran.r-project.org/
  - Search for questions on stack overflow: https://stackoverflow.com/
  - ASK a question on stack overflow, but check it's not been asked before
- Examples
  - Use help() to get documentation and see reproducible examples
  - Modify the code for your own purposes

How to find data

- Repositories galore!
  - Zenodo
  - Dryad
  - Web of Science
  - Supplemental material on journal website
  - Discipline specific data such as CEFAS

Loading in data

Place data somewhere in your working directory, preferably in a sub-folder

You can also load data from online sources

If you clone a GitHub repository, the data therein will update automatically

Manipulate data

- Adding new rows or columns
  - New calculations result in new columns
  - Combining multiple identically structured data frames results in new rows
- Splitting data into multiple data frames
  - You usually do this when you want a separate data frame for each [value]
- Renaming columns or rows (usually columns)
  - Shorter names are better as they're more efficient and visually less confusing
- Subsetting data
  - There might be certain times/values/species/etc. that you are not interested in
  - You might be interested in everything, but want to look at one [value] at a time

Perform analysis

- This is where the statistics in the base and analytical packages come in
- Let the question you're asking guide the analytical method that you choose
    - Stats/math experts are handy here; we study for years and years to learn these things, so ask us!
- Sometimes you have a tool and you have to use it
    - Check documentation
    - Check stack overflow
    - Look for examples that are like your own, manipulate your data to match, and apply the code

Debugging

- Deal with error messages
- Sometimes you can ignore these, but usually not
- Copy/paste cryptic error messages into a search engine (I still do this all the time)
- Always test your code on one row/chunk of data, knowing what the outcome should be
  - If the result is as expected, it's still a good idea to check a couple more to make sure it's not a fluke

Function-ize it

Make your code more general by turning your analysis into a function

In the future, you may wish to perform a similar analysis on a different species/population/habitat/etc.

Others who wish to use your code will find it much easier

This is not a required step, but it will help