Intro to R and RStudio

FNCE5352

1/17/2023

Matt McDonald - CV

Work:

KBRA KROLL BOND RATING AGENCY

GE Capital

IBM

edgelab FRESH IDEAS. BREWED DAILY.

priceline.com

pwc

Education:

Columbia University

MS Statistics

University of Connecticut

MBA Finance

Colgate University

BA Mathematics

Career Path


Actuarial
IT/Software Development
Credit Modeling / Model Risk Management

Class Syllabus

Why Learn To Code

- In Financial Institutions, Excel currently is the common medium for analysis, BUT...
- Coding languages like R and Python have distinct advantages, AND...
- Depending on where you work, coding ability will either be an expectation, OR...
- These skills could set you apart from your peers and make you a SUPER USER

Advantages of a Code-based Analysis

- **Flexible:** No black box constraints. Access and combine data. Analyze and present it exactly as needed.
- **Iterative:** Quickly make changes and updates in response to feedback. Share updates with stakeholders
- **Reusable and extensible:** Tackle similar problems in the future. Extend to novel problems as circumstances change.
- **Inspectable:** Track changes over time, discover errors and audit the approach
- **Reproducible:** Combine with environment and package management, ensure analyses are repeatable and verifiable.

Python

Strengths

- **Versatility**: Python is a general-purpose language with a simple, readable syntax, making it suitable for a wide range of applications beyond data analysis, such as web development, automation, and software development.
- **Rich Libraries**: For data analysis and machine learning, Python offers robust libraries like Pandas, NumPy, Scikit-learn, and TensorFlow.
- **Community and Support**: Python has a vast and active community, ensuring abundant resources, tutorials, and support.
- **Integration**: Python integrates well with other technologies and can be easily embedded in applications.
- **Scalability**: It's scalable and faster for general-purpose programming.

Weaknesses

- **Statistical Analysis**: While Python has statistical capabilities, they are not as extensive and sophisticated as R's.
- **Memory Usage**: Python's memory consumption can be high for large datasets.
- **Learning Curve**: For those specifically focused on data analysis, Python might introduce unnecessary complexity due to its broad scope.

R

Strengths

- **Statistical Analysis and Visualization**: R excels in statistical analysis and visualization. It has comprehensive packages like **ggplot2** for data visualization and **lm** for regression, making it superior for statistical modeling.
- **Data Analysis Environment**: R is specifically designed for data analysis, which makes its environment and syntax more suited for statistical analysis tasks.
- **Community**: R has a strong community in academia and research, offering a wealth of packages and support for statistical analysis.
- **Integrated Development Environment (IDE)**: RStudio provides an excellent IDE for R programming, making data analysis more intuitive.

Weaknesses

- **General-Purpose Programming**: R is less versatile for general software development compared to Python.
- **Speed**: For large datasets, R can be slower than Python, especially if not properly optimized.
- **Memory Intensive**: Like Python, R can also be memory-intensive, and it traditionally keeps all data in memory.

Key Differences Between R and Python

- **Purpose and Design Philosophy**: Python is a general-purpose language designed for readability and versatility, while R is specifically designed for statistical analysis and data visualization.
- **Data Handling**: Python's Pandas library is excellent for data manipulation, offering a more programming-oriented approach, whereas R's data manipulation is deeply rooted in its statistical capabilities.
- **Machine Learning and Deep Learning**: Python is the go-to language for machine learning and deep learning due to libraries like Scikit-learn and TensorFlow.
- **Statistical Analysis**: R has a slight edge in advanced statistical modeling and tests, with a vast array of statistical packages.
- **Visualization**: R's ggplot2 is widely regarded as one of the best data visualization tools.
- **Community and Resources**: Python has a broader user base and community support, given its wide range of applications, while R's community is highly specialized in statistics and data analysis.

A Gentle Introduction to R

Install R & RStudio

- https://stat545.com/install.html

R vs Rstudio

R: Engine

RStudio: Interface

R Packages

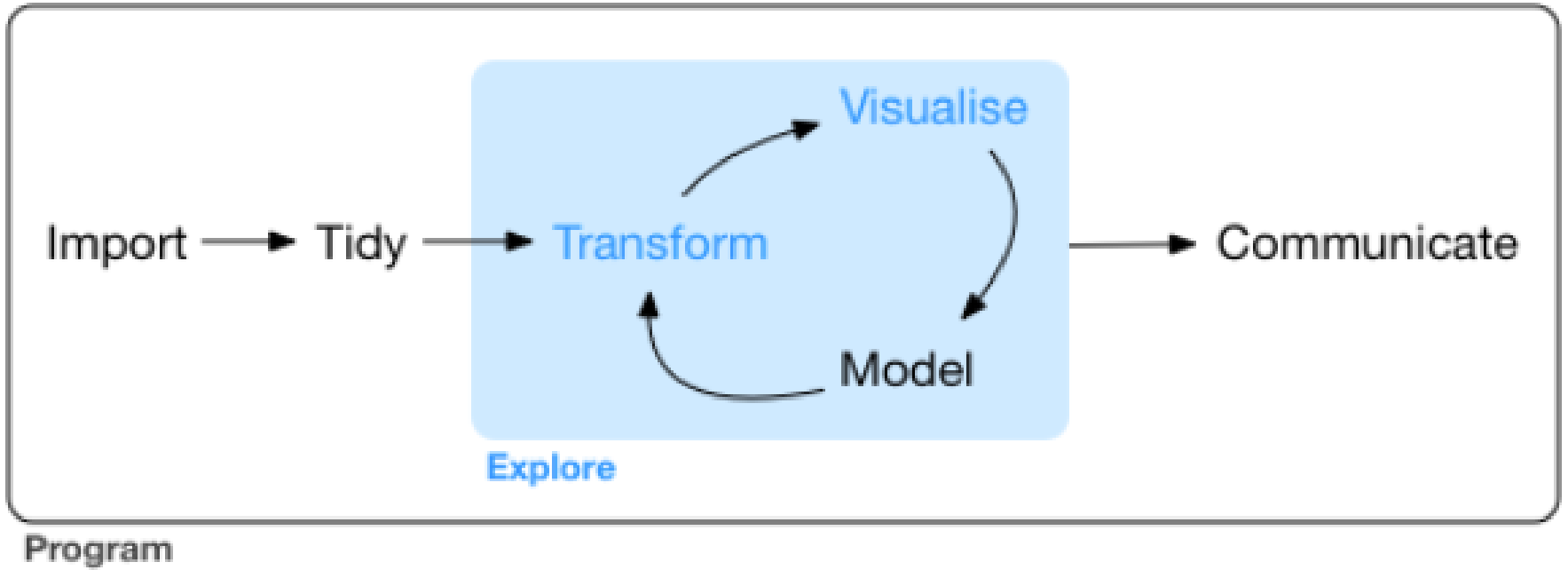
R: New phone

R Packages:
Apps you can download

GET IT ON Google Play

Download on the App Store

Modeling Workflow



Introduction to the Tidyverse

Project Based Workflow

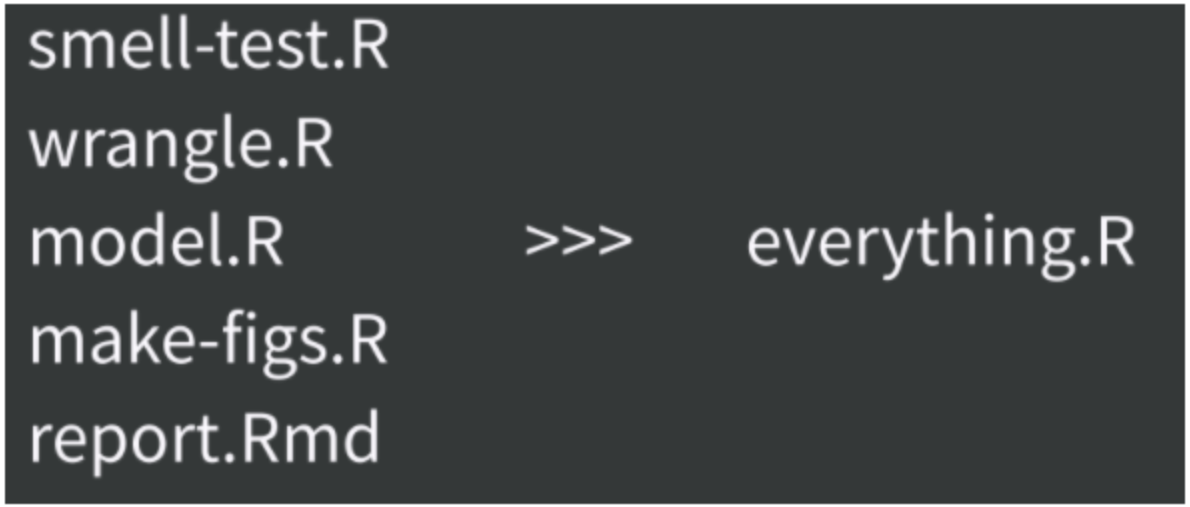
Why?

- work on more than 1 thing at a time
- collaborate, communicate, distribute
- start and stop

How?

- dedicated directory
- RStudio **P**roject
- Git repo, probably syncing to a remote

Project Based Workflow



Good Enough Practices for Data Science

Box 1. Summary of practices

1. Data management
   - a Save the raw data.
   - b Ensure that raw data are backed up in more than one location.
   - c Create the data you wish to see in the world.
   - d Create analysis-friendly data.
   - e Record all the steps used to process data.
   - f Anticipate the need to use multiple tables, and use a unique identifier for every record.
   - g Submit data to a reputable DOI-issuing repository so that others can access and cite it.
2. Software
   - a Place a brief explanatory comment at the start of every program.
   - b Decompose programs into functions.
   - c Be ruthless about eliminating duplication.
   - d Always search for well-maintained software libraries that do what you need.
   - e Test libraries before relying on them.
   - f Give functions and variables meaningful names.
   - g Make dependencies and requirements explicit.
   - h Do not comment and uncomment sections of code to control a program's behavior.
   - i Provide a simple example or test data set.
   - j Submit code to a reputable DOI-issuing repository.
3. Collaboration
   - a Create an overview of your project.
   - b Create a shared "to-do" list for the project.
   - c Decide on communication strategies.
   - d Make the license explicit.
   - e Make the project citable.
4. Project organization
   - a Put each project in its own directory, which is named after the project.
   - b Put text documents associated with the project in the `doc` directory.
   - c Put raw data and metadata in a data directory and files generated during cleanup and analysis in a results directory.
   - d Put project source code in the `src` directory.
   - e Put external scripts or compiled programs in the `bin` directory.
   - f Name all files to reflect their content or function.
5. Keeping track of changes
   - a Back up (almost) everything created by a human being as soon as it is created.
   - b Keep changes small.
   - c Share changes frequently.
   - d Create, maintain, and use a checklist for saving and sharing changes to the project.
   - e Store each project in a folder that is mirrored off the researcher's working machine.
   - f Add a file called `CHANGELOG.txt` to the project's `docs` subfolder.
   - g Copy the entire project whenever a significant change has been made.
   - h Use a version control system.
6. Manuscripts
   - a Write manuscripts using online tools with rich formatting, change tracking, and reference management.
   - b Write the manuscript in a plain text format that permits version control.

Getting to Know GIT

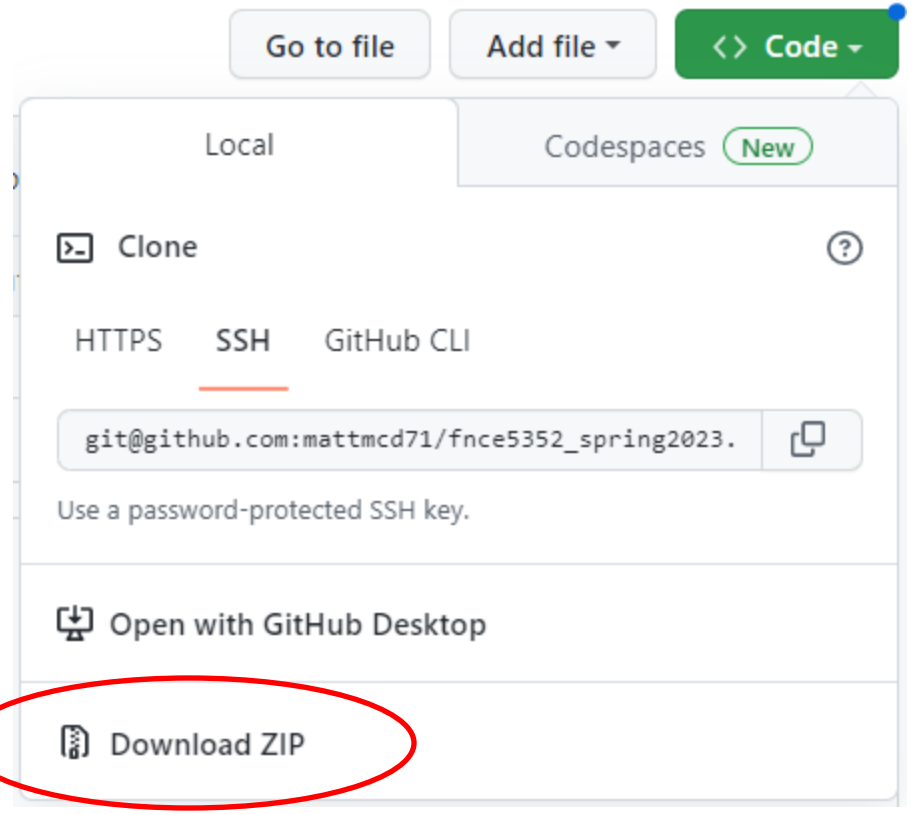
- https://happygitwithr.com/

Class Repo:

- https://github.com/mattmcd71/fnce5352_spring2023

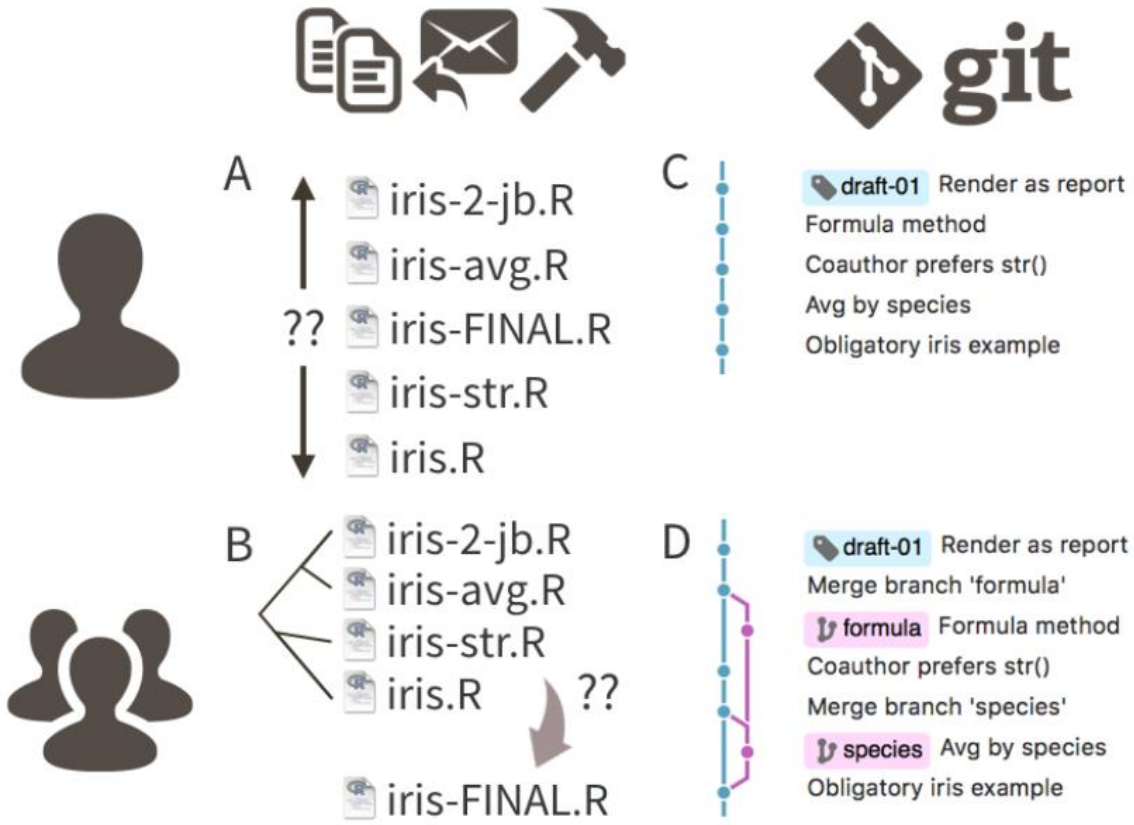
How to Access the Class GIT Repo

- Download ZIP



- Clone
  - HTTPS or SSH
  - SSH will require a setup of your SSH keys
    - See happygitwithr page for complete instructions on that

Why GIT?


A
iris-2-jb.R
iris-avg.R
??
iris-FINAL.R
iris-str.R
iris.R
B
iris-2-jb.R
iris-avg.R
iris-str.R
iris.R
??
iris-FINAL.R
C
draft-01 Render as report
Formula method
Coauthor prefers str()
Avg by species
Obligatory iris example
D
draft-01 Render as report
Merge branch 'formula'
formula Formula method
Coauthor prefers str()
Merge branch 'species'
species Avg by species
Obligatory iris example
git

GIT vs GitHub