NTRES 6940:

Collaborative and Reproducible Data Science in R

Why are we here?

What are we going to do?

Who are we?

Getting situated with R/RStudio

Reproducible/open science

”The practice of distributing all data, software source code, and tools required to reproduce the results discussed in a research publication”

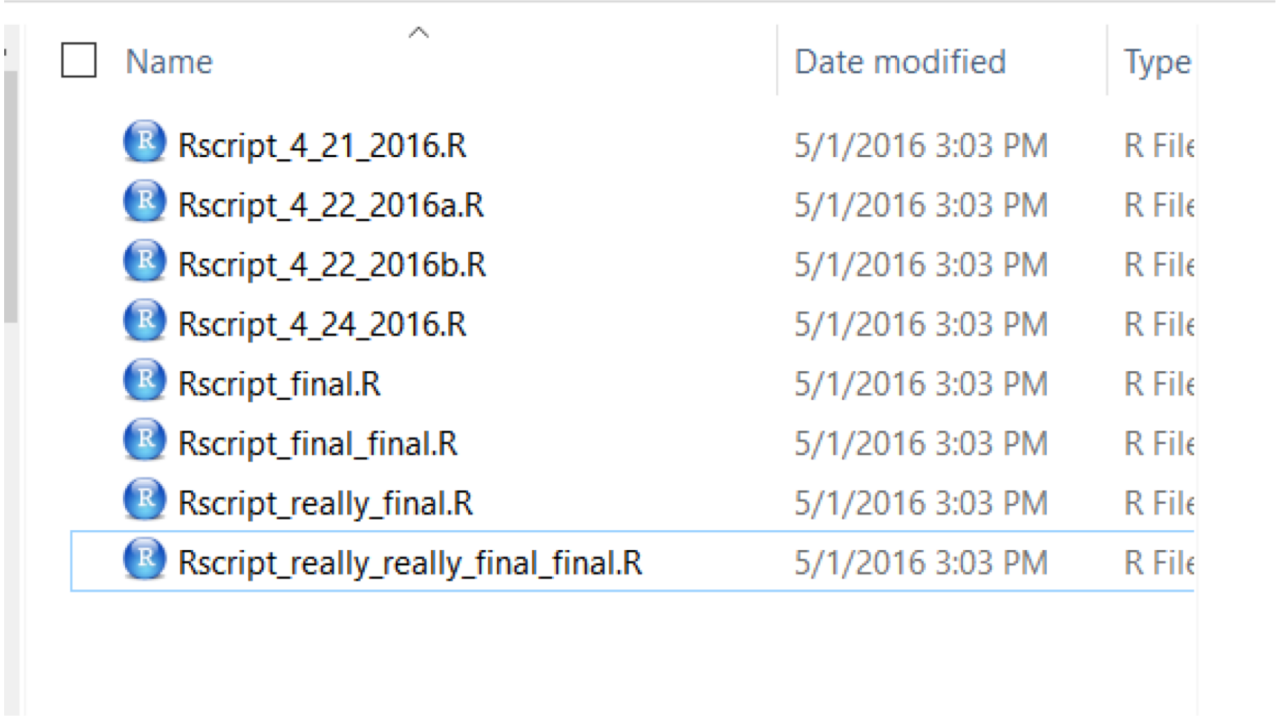
Discuss with your neighbor

- Is reproducibility the same as replicability?
- Why is there an increasing push towards open science now?
- Why should you adopt open science practices? Who will benefit and how?

Replication vs. Reproducibility

- **Replication:** the confirmation of results and conclusions from one study obtained independently in another is considered the scientific gold standard
- **Some studies can't be replicated:** too big, too costly, too time consuming, one time event, rare samples
- **Reproducibility:** minimum standard for assessing the value of scientific claims, particularly when full independent replication of a study is not feasible

Does this look familiar?



Who benefits from open science practices?

- YOU!
  - Future you will thank you
  - Increased research efficiency
- The scientific community
  - More transparency, easier to build off each other's work
- Society
  - More accurate science: errors are more likely to get detected

Tools for

better science in less time

Who are we?

- Nina Overgaard Therkildsen (instructor)

- Nicolas Lou (TA)

Survey results

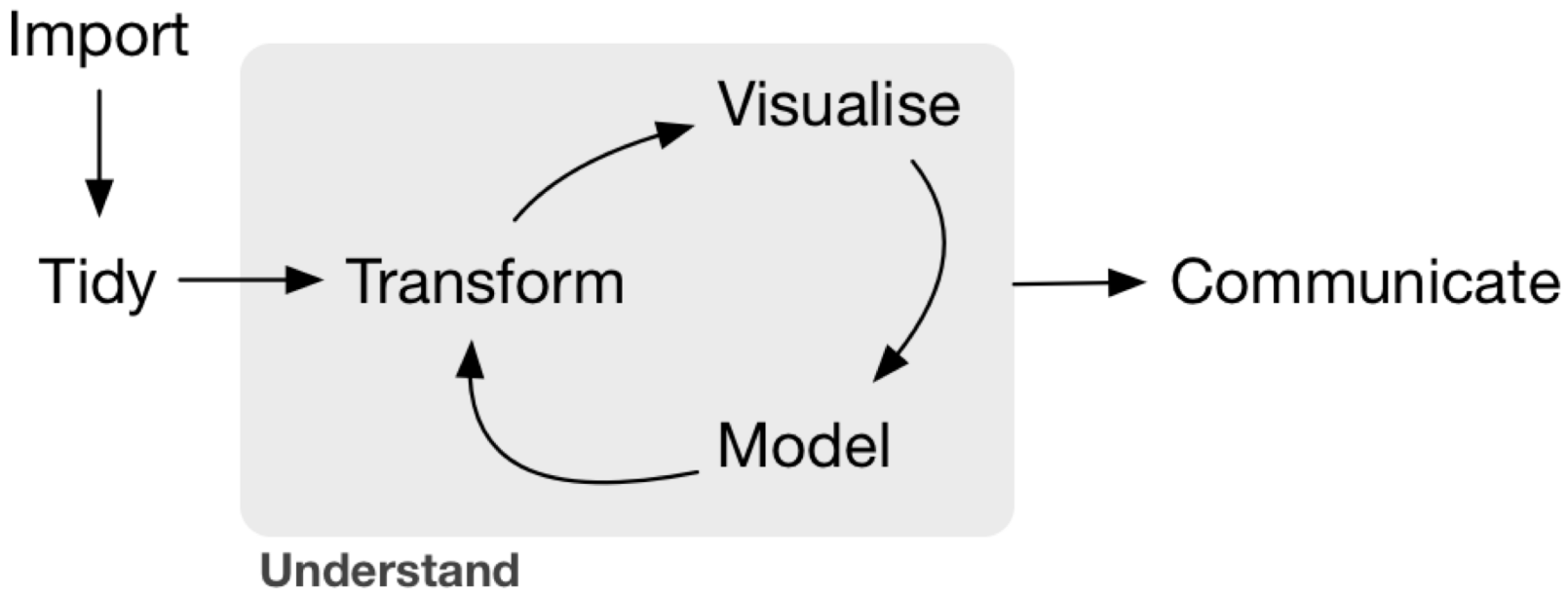
- Summarized here: https://github.com/nt246/NTRES6940-data-science/blob/master/misc/pre_course_survey_results.md

Code of conduct

Tools for

better science in less time

and with less pain


Import
Tidy
Transform
Visualise
Model
Communicate
Understand

By the end of this course, students will be able to

- Describe strategies for ensuring that their data analysis is reproducible
- Demonstrate best practices for coding and project-oriented workflows in RStudio
- Import and clean messy data files using a variety of packages and functions in R
- Subset, reorganize, and merge diverse datasets in R
- Effectively explore and visualize patterns in complex datasets with ggplot in R
- Write simple functions/programs and data analysis pipelines in R
- Automate repeated analysis tasks in R
- Track the history of file changes (version control) and collaborate effectively on scripts with others with Git and GitHub
- Use R Markdown to combine text, equations, code, tables, and figures into reports, websites, and presentations

What we will NOT cover

- Statistics and hypothesis confirmation

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights – NYTimes.com

TECHNOLOGY | For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

SHARE

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014

Monica Rogati, Jawbone's vice president for data science, with Brian Wilt, a senior data scientist.
Peter DaSilva for The New York Times

EMAIL

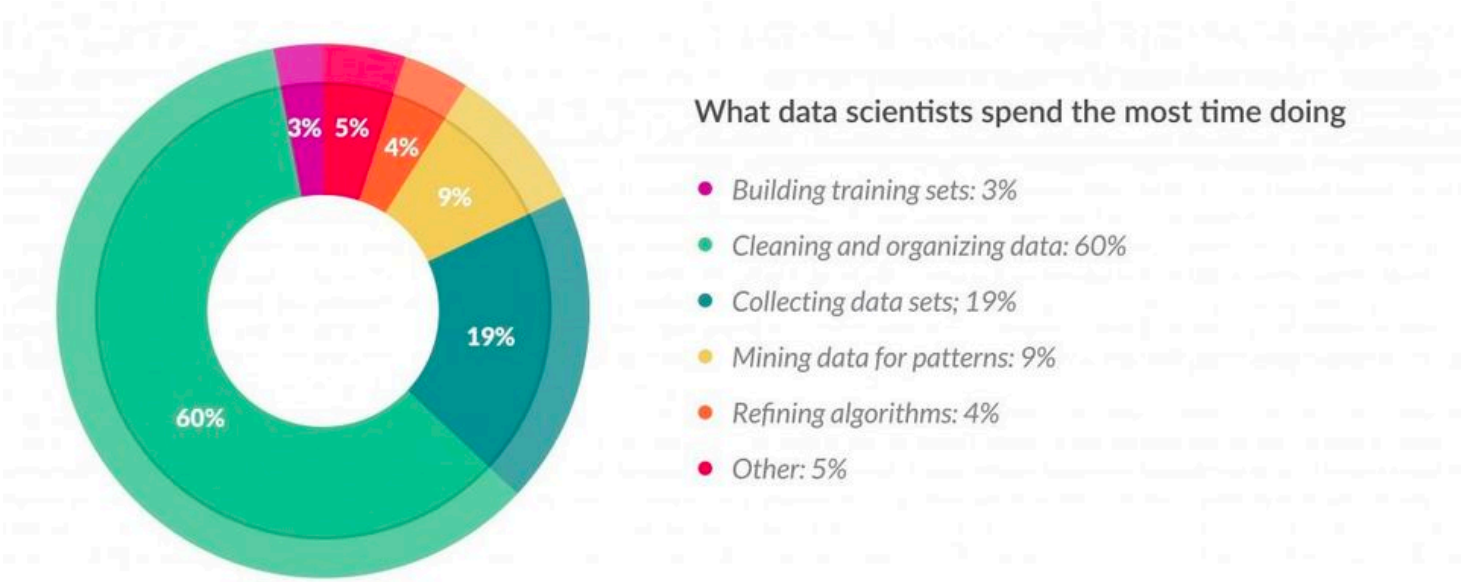
Technology revolutions come in measured, sometimes foot-dragging steps. The lab science and marketing enthusiasm tend to underestimate the

Data scientists spend 50 - 80% of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?partner=rss&emc=rss&smid=tw-nytimesscience&_r=0

Survey of 80 data scientists

https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#25167ec06f63



Goals for data wrangling

- Understand how and why to tidy data and analyze tidy data, rather than making your analyses accommodate messy data
- Appreciate how there is a lot of decision-making involved with data analysis, and a lot of creativity
- Think ahead instead of only to get a single job done now
- Increase efficiency in your science and increase reproducibility
- Facilitate collaboration with others — especially Future You!

What we will NOT cover

- Statistics and hypothesis confirmation
- Modeling and simulation
- Big data

Big data

- Before we can handle big data, we need to handle small data
  - Tools can handle 100s Mb of data (up to 1–2Gb)
  - Many big data problems are small data problems in disguise
    - Subset, subsample, summary
    - Parallel analysis on multiple independent units?

What we will NOT cover

- Statistics and hypothesis confirmation
- Modeling and simulation
- Big data
- Any other programming languages than R

Why R?

- It’s free, open source, and available on every major platform
- A massive set of packages for statistical modelling, machine learning, visualization, and importing and manipulating data
- Cutting edge tools
- Supportive and welcoming community
- Powerful tools for communicating your results

What we will NOT cover

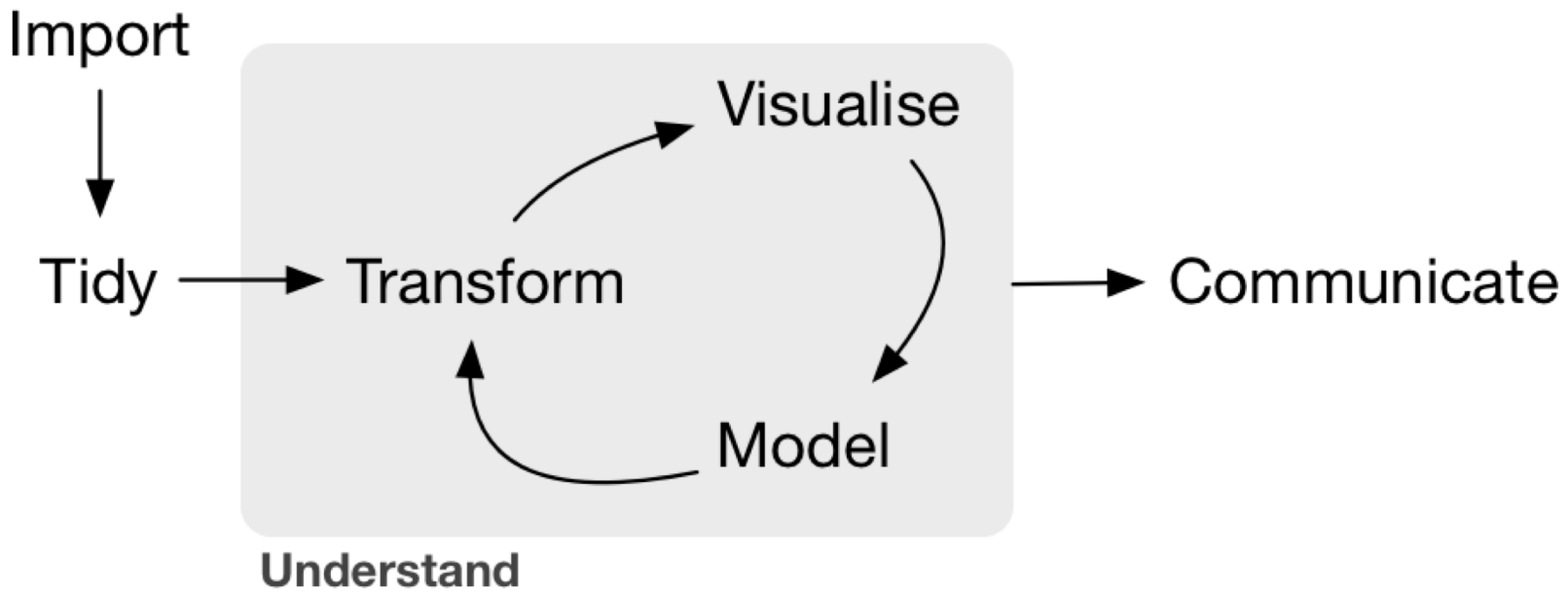
- Statistics and hypothesis confirmation
- Modeling and simulation
- Big data
- Any other programming languages than R
- Non-rectangular data (e.g. images, sounds, trees, text) or domain-specific applications

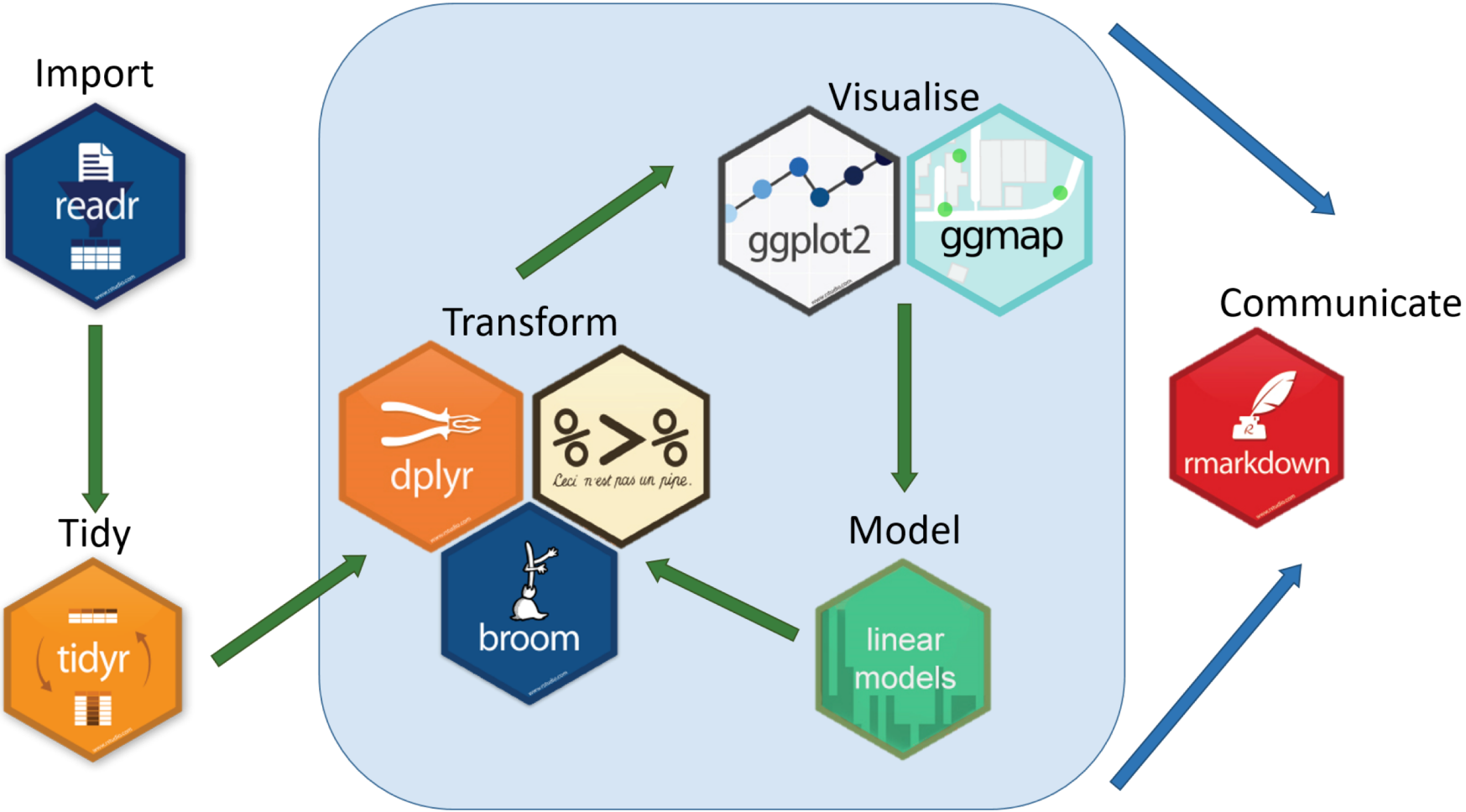
What we will NOT cover

- Statistics and hypothesis confirmation
- Modeling and simulation
- Big data
- Any other programming languages than R
- Non-rectangular data (e.g. images, sounds, trees, text) or domain-specific applications
- Base-R plotting and workflows (we will focus on the tidyverse)

The tidyverse

- An opinionated collection of R packages designed for data science
- All packages share an underlying design philosophy, grammar, and data structures


Import
Tidy
Transform
Visualise
Model
Communicate
Understand


Import
readr
Tidy
tidyr
Transform
dplyr
%>%
Leci n'est pas un pipe.
broom
Visualise
ggplot2
ggmap
Model
linear models
Communicate
rmarkdown

The tidyverse

- An opinionated collection of R packages designed for data science
- All packages share an underlying design philosophy, grammar, and data structures
- More streamlined and intuitive syntax and workflow than base R packages for most applications*

*Some would dispute that and swear by base R. We are not claiming that the tidyverse is superior to base R in all respects, only that it provides a set of very powerful tools

What we will NOT cover

- Statistics and hypothesis confirmation
- Modeling and simulation
- Big data
- Any other programming languages than R
- Non-rectangular data (e.g. images, sounds, trees, text) or domain-specific applications
- Base-R plotting and workflows (we will focus on the tidyverse)

What we WILL cover

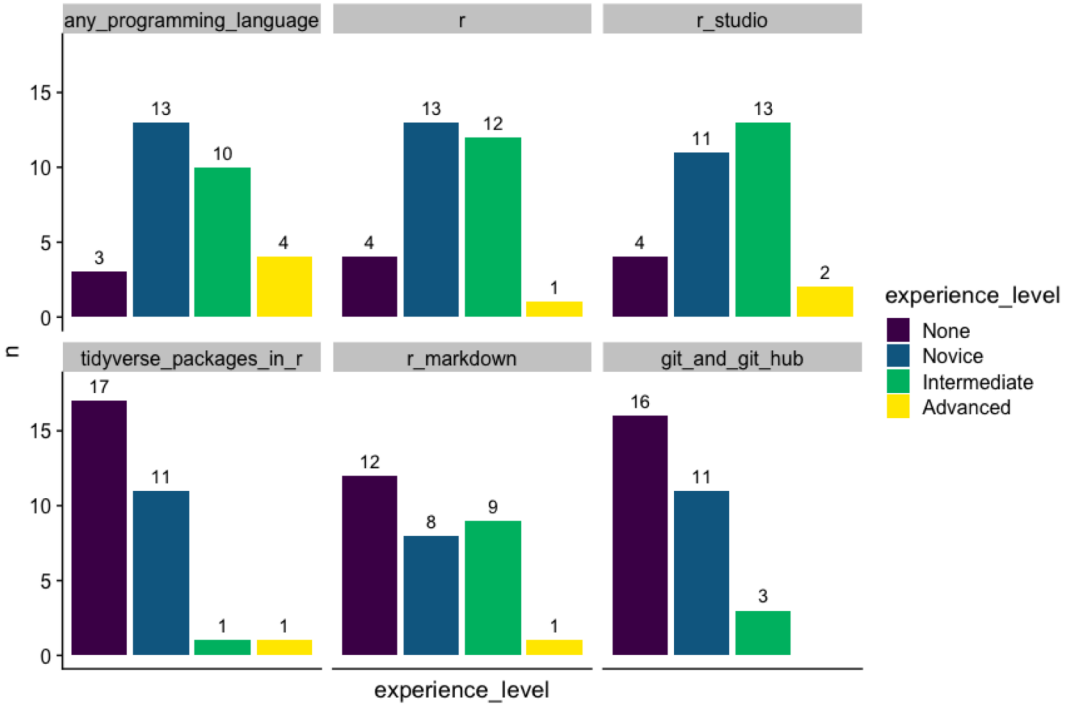
1. Coding with best practices (RStudio/tidyverse)

2. Collaborative book-keeping (Git/GitHub)

3. Reporting and communicating (RMarkdown/GitHub)

What we WILL cover

1. Coding with best practices (RStudio/tidyverse)
2. Collaborative book-keeping (Git/GitHub)
3. Reporting and communicating (RMarkdown/GitHub)



Class#	Date	Topic	Readings
1	March 2 (Mon)	Intro to the course and R/RStudio	
2	March 4 (Wed)	Markdown and GitHub	Link
3	March 9 (Mon)	The Git workflow (version control)	
4	March 11 (Wed)	Plotting with ggplot part 1	
5	March 16 (Mon)	Data wrangling part 1 (dplyr::filter, mutate, select, arrange)	
6	March 18 (Wed)	Data wrangling part 2 (dplyr::summarize, group_by)	
7	March 23 (Mon)	Plotting with ggplot part 2 + good coding practices	
8	March 25 (Wed)	Effective visualization + file I/O and tibbles	
--	March 30 (Mon)	SPRING BREAK	
--	April 1 (Wed)	SPRING BREAK	
9	April 6 (Mon)	Tidy data (what is tidy data and how to handle untidy data)	
10	April 8 (Wed)	Relational data (join functions and lookup)	
11	April 13 (Mon)	Factors + basic string manipulation + dealing with dates and times	
12	April 15 (Wed)	Writing functions in R	
13	April 20 (Mon)	Iteration (for loops and map functions)	
14	April 22 (Wed)	Review and discussion of good practices for reproducible workflows	

Course format

- Two weekly lectures – Mondays and Wednesday 4.20-5.40pm
- Optional “hacky hour” – Fridays 3-5pm
  - Come work on assignments or bring your own dataset for some end-of-week social coding
- Practice, practice, practice!

Assignments

- Weekly problem sets
  - Assigned each Wednesday, due the following Wednesday at 10pm
  - Will submit on GitHub – more instructions to follow

Evaluation

- To pass this course you must:
  - Attend all lectures (email instructor beforehand if you need to miss class)
  - Participate actively in class
  - Submit at least 6 of the 7 problem sets with demonstrated effort to complete all questions

Course communication

- We won’t be using Canvas

- Communication via Slack and GitHub

Check list

- Have you all:
  - Gotten R and RStudio working?
  - Followed the instructions for installing Git and making a GitHub account?
  - Joined the workspace on Slack
  - Added a photo to you GitHub and Slack accounts? (optional)

Introduction to R/RStudio

- RStudio IDE orientation
- Shortcuts and autocomplete
- Review of how functions work in R
  (defaults, optional vs. required, order of arguments vs. naming them, handling NAs)
- Install and load packages
- Scripts
- Home directory and RStudio projects
- Change settings to not save workspace