



**ID529!**

Data Management and  
Analytic Workflows in R  
January 2023



**HARVARD**  
**T.H. CHAN**

SCHOOL OF PUBLIC HEALTH

# Modelling workflows

Import



Tidy



Model Building



Communicate

Import

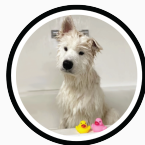


Tidy

Recoding variables



Reordering factors



Subsetting data

# Model Building

Specifying interaction terms

Comparing models

Extracting residuals

Predicting fitted values

Extracting coefficients

Computing confidence limits



Communicate

Making pretty tables

## Some goals for today

- Practice reading someone else's R code and *annotating* to help you understand what is going on
- Review and consolidate some of the concepts we've been learning about
  - > Use a *dplyr* workflow to prepare our dataset for analysis
  - > Write a quick function (using concepts of *functional programming*)
  - > Use *ggplot* to make a figure
- Fit some models
  - > `lm()` for linear regression
  - > `glm()` for generalized linear regression
  - > What arguments do these functions take?
  - > What is contained in the resulting model objects?

## Some goals for today



- Learn how to extract output of interest from model objects
  - > using base R
  - > using `broom::tidy`, `broom::augment`, and `broom::glance`
- Learn how to create some pretty tables
  - > Learn about the `gtsummary` package



- Go to <https://github.com/jarvischen01/id529-regression-models/> and download a .zip copy of the repository
- Or if you want to employ your new [git](#) skills, clone the repository and open it in a R Project in your RStudio
- Open the [id529\\_day4\\_regression\\_models.R](#) script. You can follow along, annotate, and/or run the code in your own R session.

What to prioritize in understanding the code in the example

- > Using dplyr code for data cleaning/management
- > Calling `lm()` and `glm()`
- > Using `summary()`, `coef()`, `confint()`, and `broom::tidy()` to extract and summarize coefficients.
- Writing our own function to extract coefficients and output to a tibble
- Using anova to compare models; using `predict()` and `broom::augment()` to extract predictions and residuals
- Using `broom::glance` to extract model fit statistics
- > Using `gtsummary` and `sjPlot` to generate pretty tables
- > Using `ggplot` and `sjPlot` to visualize regression output



## Take Home Messages

- Including outlines and pseudo-code in your R scripts can help you to be intentional about your coding
- Annotation is an investment that future-you (and your colleagues) will thank you for
- Don't be afraid to poke around and look inside the objects that you create.
- **Laziness can be a virtue**: when faced with a lot of repetitive tasks, can you write a function (or make use of an existing function) to automate your work and make it more efficient?
  - this is what it means to think in terms of an **efficient workflow**
- Formatting tables for publications is fiddly, so take time to find a workflow that works for you.



1. Introduction to broom <https://cran.r-project.org/web/packages/broom/vignettes/broom.html>
2. broom.mixed documentation (useful if you are fitting random effects or mixed models) [https://cran.r-project.org/web/packages/broom.mixed/vignettes/broom\\_mixed\\_intro.html](https://cran.r-project.org/web/packages/broom.mixed/vignettes/broom_mixed_intro.html)
3. gtsummary documentation <https://www.danielsjoberg.com/gtsummary/index.html>
4. sjPlot documentation <https://strengexjacke.github.io/sjPlot/index.html>
5. stargazer: another useful package for outputting pretty tables <https://cran.r-project.org/web/packages/stargazer/vignettes/stargazer.pdf>