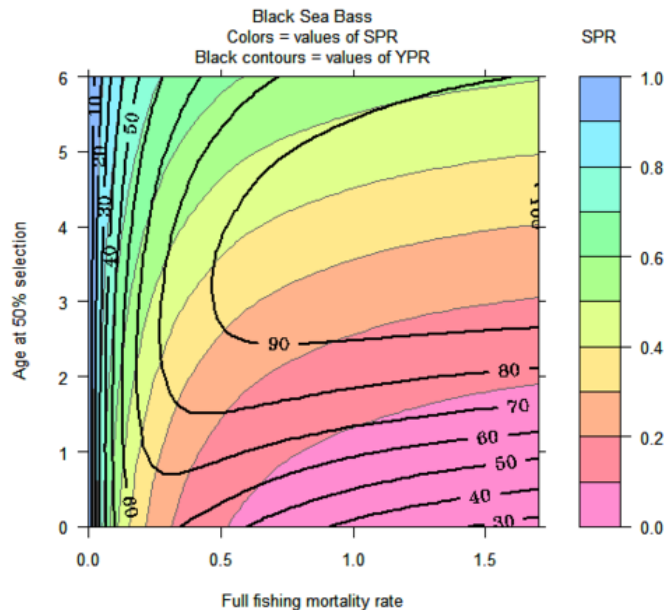
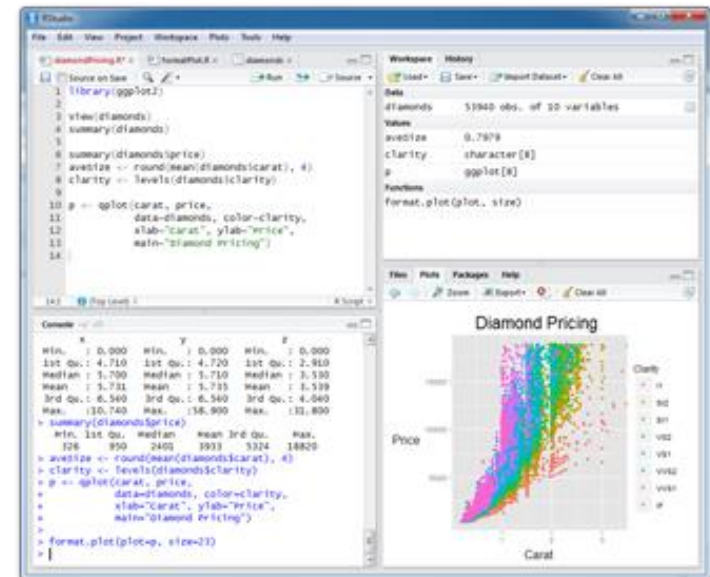


Introduction to ...for epidemiologists



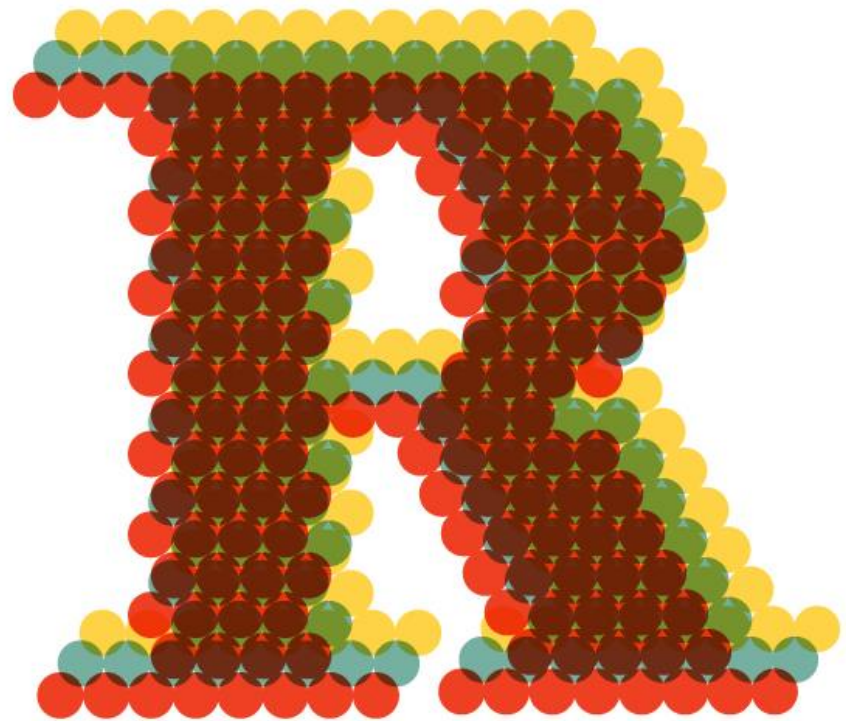
Alan Brookhart
 Xiaojuan Li
 Nathaniel MacNell
 Mike Fliss

MW 10:10-11:00
 Fall 2016



Welcome & Overview

- Course logistics
 - Introductions
 - Sign up sheet
 - Syllabus review
- All about R
 - How is R different from SAS?
 - How do I install R/Rstudio?
- To do
 - Make sure R and RStudio are installed for next class
 - Fill out short survey on R experience and expectations (Assignment 0 on Sakai)



Introductions

- Please fill out the sign-in sheet (especially if you want to audit!)
- Instructor of Record: Alan Brookhart
- Facilitators: Xiaojuan Li
 Nathaniel MacNell
 Mike Fliss
- Special guests: Special topic lecturers (see course plan)

Course Approach and Format

- Course theory:
 - Designed for those familiar with SAS or another programming language.
 - Mirrors an epidemiologic analysis – not a “pure” intro to R.
 - Practical. See, try, modify, why, apply.
- Logistic goals:
 - Minimal out-of-class responsibilities. Direct relevance to your existing work.
 - Wind down assignments before the end-of-semester rush.
- Course progression:
 - Part one: Core R
 - Part two: R packages; Special topics
- Use any resources you want!
 - Internet searches, forums, books, other open courses
 - Group work on exercises is encouraged but not required (don’t just copy...)
 - Better to turn in broken/incomplete code you kind of understand (so we can help) instead of working code you don’t understand.
 - R is open and collaborative!

Student Responsibilities and Expectations

- Class exercises
 - Follow along with example code, activities, interactive exercises in R
- Homework
 - During part 1 (first half), due about every 2 weeks
 - Includes programming exercises and project activities
- Project
 - Dataset of your choice. Anything. Something you're already using.
 - A: Complete basic analysis steps (same for everyone, but with your data)
 - Included in the homework sets through the semester (graded pass/fail)
 - B: Complete analysis of your choice (using an R package)
 - Combine A & B into a research poster to show off your work
 - Due around thanksgiving – see syllabus. Include code appendix.

Why ? Important features of

- *Free*: costs nothing, runs anywhere, modify anything you want
- *Popular*: across disciplines, increasing prominence in epidemiology
- *Powerful*: do more with less (time, code, heartache)
- *Efficient*: good for big datasets, simulations, demanding calculations
- *Flexible*: do many things, in many different ways (error-checking)
- *Transparent*: you can look at how anything works, code sharing, etc.
- *Community*: package development, helpful people, fast bug iteration
- *Higher level thinking*: Avoid “card” thinking. Use abstraction and grammars

And why RStudio?

- Short answer: helps you code better and faster
- It also looks similar to the SAS interface you’re probably used to
- More on this next class

Challenges of

- *Free*: no one to sue! no centralized or official tech support.
- *Popular*: not entrenched! Resistance to change.
- *Powerful*: can require some different thinking. Obfuscated code.
- *Efficient*: thinking and coding efficiently takes work (disk v RAM?)
- *Flexible*: you can write rickety / Rube Goldberg code. Try not to.
- *Transparent*: sometimes you have to get into the guts. Can be gross.
- *Community*: Conflicts – between people, packages, syntax.
- *Higher level thinking*: have to learn it!

All that... and still VERY much worth it!



vs.



- No division of your code into PROC/DATA parts
 - No separate macro language: “macro” variables aren’t needed
 - “Modern” computer science language: functions, objects, abstraction
 - SAS output is just output. R output can be input / referenced, too.
 - SAS “gives you everything” vs. R “builds from bottom”
 - Graphical data exploration is easier in R, but takes learning
-
- R shares similarities with SAS PROC IML, STATA, Matlab, Python

Homework: Install R and RStudio

- I'll show you in class, a help guide is available on the [GitHub](#) site.
 - Install R first
- Make sure it works before you come to class.
 - Start RStudio.
 - Type `1+1` into the console and press enter.
 - If you're brave, try some expressions from high school math...

