Live coding

for teaching computation

ECME, Dec 2019

bit.ly/rundel_ecme2019

Colin Rundel

Univ of Edinburgh

Context

This semester,

- **Math 11176 - Statistical Programming**
  - Required (SwDS) MSc course (10 credit)
  - 186 students (109 Stat, 50 OR, 18 ACM, 7 Other)

Previously at Duke,

- **Sta 323 - Statistical Computing**
  - Year 3 UG elective, ~40 students
- **Sta 523 - Statistical Programming**
  - Year 1 MSc required course, ~40 students
- **Sta 790-02 - Advanced Statistical Computing**
  - Year 2 MSc / PhD special topics course, ~15 students

Learning Objectives

- Teaching R as a programming language
- Reproducible computing
- Software engineering
- Collaboration

statistics course with programming

vs

programming course with statistics

In the classroom

Lectures mostly structured around a traditional format,

- Prepared slides with context on syntax, functions, etc.
- Broken up with in-class exercises (not collected or marked)
    - direct application of the topic(s) just covered
    - encouraged to work together and ask quesions
    - work through the solution as a class (live coding)

Live Coding

"the process of designing and implementing a [coding] project in front of class during lecture period" - **Paxton (2002)**

"Live-coding is an approach to teaching programming by writing actual code during class as part of the lectures. In a live-coding session, the instructor thinks aloud while writing code and the students are able to understand the process of programming by observing the thought processes of the instructor." - **Raj, et al. (2018)**

Exercise

Using the `flights` data from `nycflights13` answer the following question:

1. How many flights from New York (JFK) to Los Angeles (LAX) did each carrier have in May?

Approach 1

```
library(dplyr)
library(nycflights13)

flights %>%
  filter(origin == "JFK", dest == "LAX") %>%
  filter(month == 6) %>%
  group_by(carrier, .drop=FALSE) %>%
  summarize(n = n())
```

```
## # A tibble: 5 x 2
##   carrier     n
##   <chr>   <int>
## 1 AA        265
## 2 B6        144
## 3 DL        199
## 4 UA        170
## 5 VX        150
```

Approach 2 - Demo

Benefits

- Modeling algorithmic thinking
- Modeling problem solving process
- Modeling debugging processes / finding help
- Interactivity and experimentation
- Comparing alternative solutions

Considerations

Common issue with live coding:

- students attempting to type everything you do
- they ignore the narrative and miss the context of the problem / solution.
- non-native english speakers can have a harder time

Some additional tools for helping with these issues:

- Posting of all code produced (to LMS, GitHub, etc.)
- Recording screencasts (not just lecture capture)
- Broadcasting the code in real time

`livecode`

This an R package, in the early stages of development, for broadcasting a source code file to an audience in real time.

- Only requires a browser to connect, but also integrates with the RStudio environment
- Removes the need for students to type the code - encourage experimentation via Copy & Paste
- Includes typical IDE features like syntax highlighting
- Also includes line highlighting and messaging functionality
- Some initial trials this semester in Mine's & my course.

`livecode` demo

Future Directions

Next year Stat Progamming is likely to flip from 2 hrs lecture, 1 hr workshop to 1 hr lecture, 2 hr workshop format

- Adopting a flipped format - record and post lectures, use lecture hours for exerices / live coding
    - Entire git / GitHub component was delivered this way this year
- Trial `livecode` more extensively and collect feedback
- Balancing active / passive learning with live coding