

TEACHING USING THE **GitHub** ECOSYSTEM

Colin Rundel

UseR! 2015 - Aalborg

Duke University

Department of Statistical Science

Course details:

- First offered in Fall 2014
- Core course in Masters of Statistical Science Program
- Approximately 30 Students
 - 2/3 MSS & MSEM, 1/3 other MS & PhD
 - Divided into teams of 3-4
 - Disparate backgrounds
- Biweekly team programming assignments
- Team final project, individual final exam

TECHNICAL LEARNING OBJECTIVES



Collaboration

Reproducible Research

- R Markdown / knitr
- GNU Make

Data in the Real World

- Messy data
- Non-flat data

Dedicated departmental server

- RStudio Server Pro
- Individual departmental accounts
- System wide install of core packages

Github Organization

- 1 private repo / team (Github Education)
- Shared public repos (e.g. examples)

Continuous Integration

- TravisCI, Wercker, Drone, etc.

ASSIGNMENTS

All assignments are turned in via github (pull the repo at the deadline)

What do we get from this?

- Forces students to use version control
- Simplifies course administration
 - Code / documentation / scaffolding all in the same place
 - Easy to grab files (pull)
 - Easy to distribute files (push)
- Searchability
- Accountability

Grading and feedback is given via pull requests

HW1 grading #2

Merged merged 6 commits into `master` from `hw1_grading` on Sep 29, 2014

Conversation 0 Commits 6 Files changed 18

rundel commented on Sep 29, 2014

Everything looks great, small changes and tidying of code and repo are included in the commits below.

Only very minor criticism is that you could interleave the code and write up a little bit more to improve the overall readability of the final document.

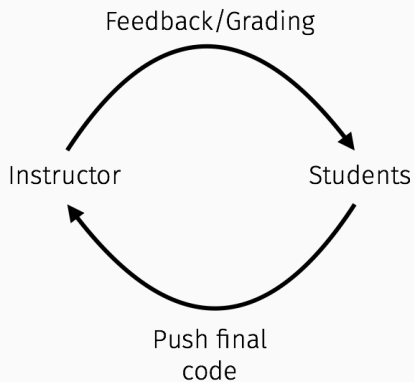
Grade: 20/20

rundel added some commits on Sep 29, 2014

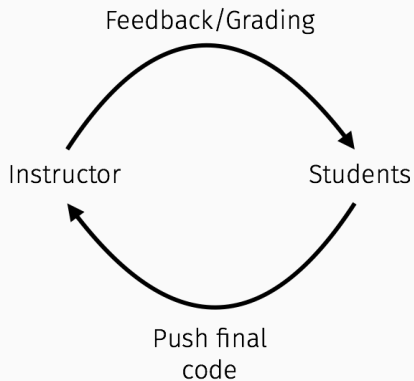
- Cleanup unneeded files and .gitignore 7cdca98
- Rearrange inclass work b9db2c9
- One more old file 0fcb4d2
- Minor comment cleanup 6473fda
- numc doesnt seem to be used elsewhere 2d0327e
- Hide library load output ca5eda1

merged commit `ec9c3fd` into `master` on Sep 29, 2014 [Revert](#)

COURSE PROCESS CARTOON

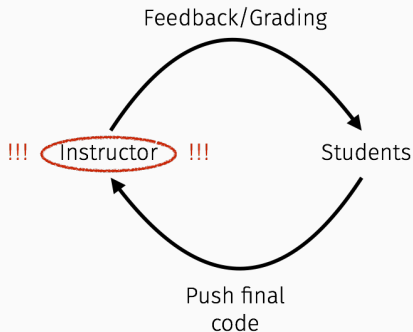


COURSE PROCESS CARTOON



Github does improve both parts of this cycle

COURSE PROCESS CARTOON



Github does improve both parts of this cycle but doesn't address the fact that *the instructor / TAs are the rate limiting step* (we don't scale well).

A PAINFULLY COMMON CONVERSATION

Student: We've submitted our HW3!

+1 Day

*Me: Your Rmd file doesn't knit, you used **setwd** with an absolute path.*

+1 Day

Student: Ok we fixed that, does it work now?

+1 Day

*Me: Nope, you used **lme4** without checking if it was installed.*

+1 Day

⋮

A SOLUTION?

In the limited universe of this class,

Me = CRAN

Student = Package Developer

so what we want is some interpretation of
R CMD check.

What should our R CMD check look like?

What should our R CMD check look like?



Ideally it should check that ...


- the code runs, Rmds knit
- the coding style is consistent
- the repo is tidy
- the code runs in a reasonable time frame
- the implementations are correct

We would also like this to be done automatically.

We would also like this to be done automatically.

Again we can take a lesson from the package developers.

| | | |
|--|--|---------------|
|  cran-comments.md | Update release notes | 16 days ago |
|  dplyr.Rproj | For speed, don't build vignettes when checking | 10 months ago |

 **README.md**

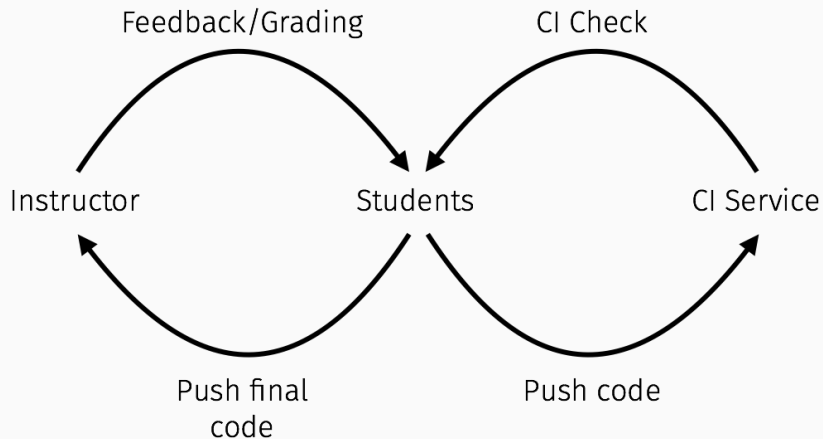
dplyr

build failing

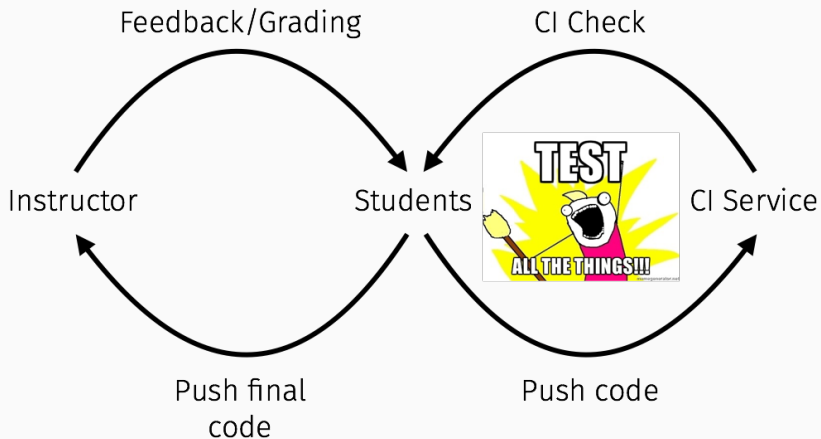
dplyr is the next iteration of plyr, focussed on tools for working with data frames (hence the `d` in the name). It has three main goals:



COURSE PROCESS CARTOON - IMPROVED



COURSE PROCESS CARTOON - IMPROVED



IMPLEMENTATION

This is currently more aspirational than reality, but the following is planned for the coming Fall 2015 semester.

Key details (subject to change):

- Adopting wercker for CI (uses Docker, steps)
- Enforced coding style via `lint`
- Enforced directory structure
- Allowed file/filetype whitelist
- `testthat` for testing implementation assignments
- automated scoring of prediction contests

CONCLUSIONS

Using github gives you a lot (for free) ...

- Version control
- Accessible web UI
- Education support
- Collaboration tools
- Search tools
- CI tools

Needs of an R programming class are very similar to the needs of the R development community

- No need to reinvent the wheel - use the existing solutions
- Teach the tools students will continue to use

QUESTIONS, COMMENTS?



rundel@gmail.com



github.com/rundel/



github.com/rundel/Presentations/



bit.ly/Sta523_2014 | bit.ly/Sta523_2015