# Teaching computing using git and GitHub

TLMSCO, Sept 2020

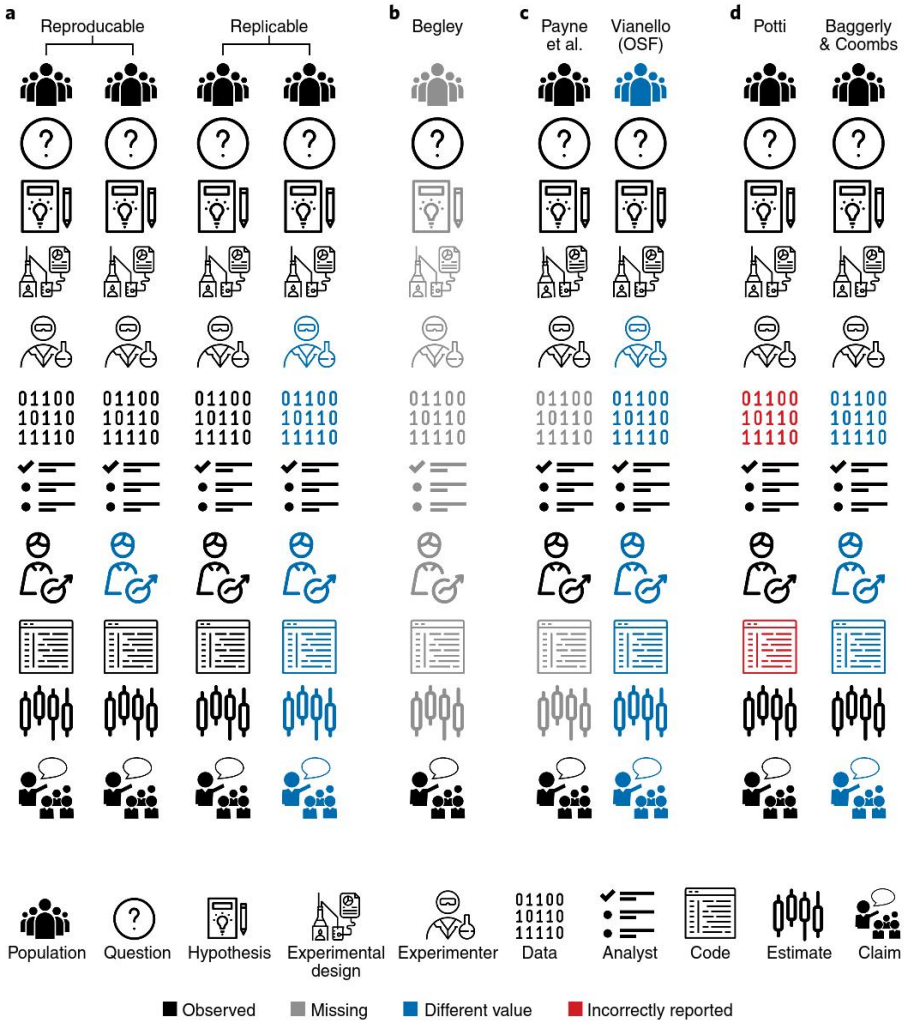
bit.ly/tlmsco_rundel

Dr. Colin Rundel

Univ of Edinburgh

# Teaching Reproducible Workflows

# Reproducible vs Replicable

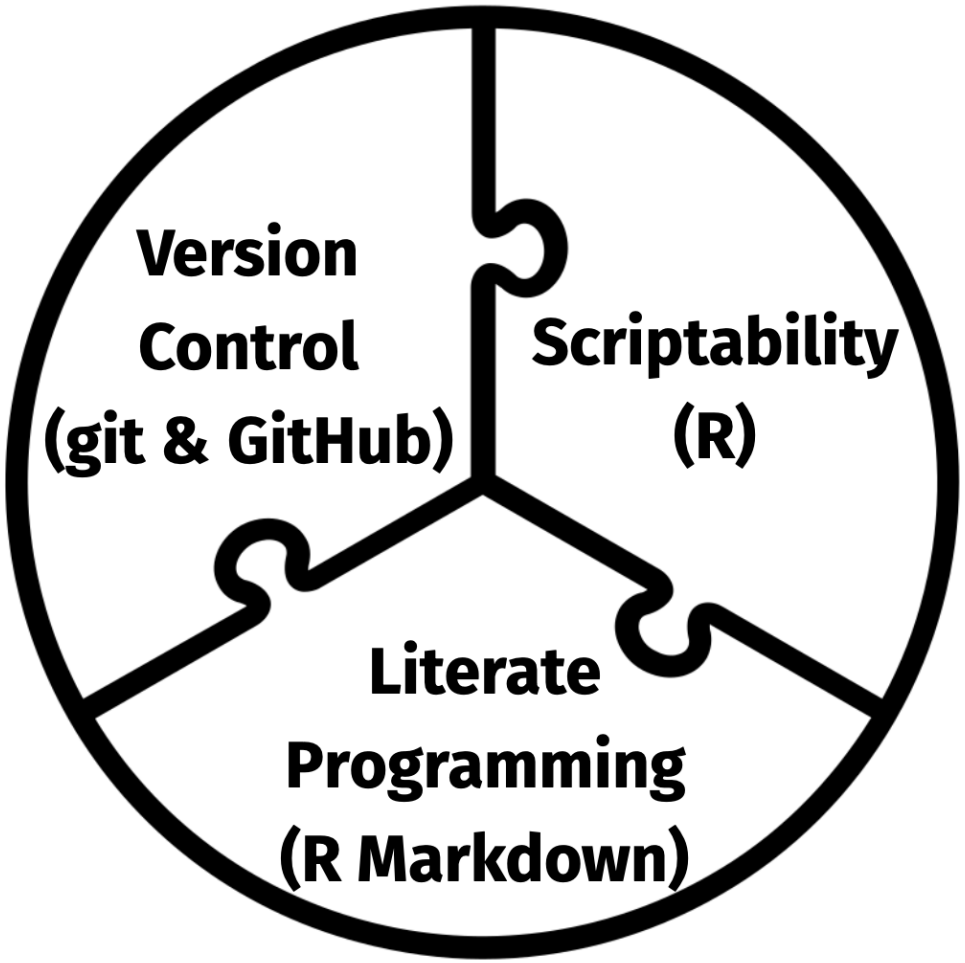



Source: Patil, Peng, Leek (2019) A visual tool for defining reproducibility and replicability. *Nature Human Behaviour*

# Reproducibility in practice

- Can you recreate the tables and figures reproducible from the code and data?
- Does the code actually do what you think it does?
- In addition to what was done, is it clear *why* it was done? (e.g. how were hyper / tuning parameters chosen?)
- Can the code be used for other data?
- Can you hand the code off to someone else and expect it to work?

# Core pieces


Version
Control
(git & GitHub)
Scriptability
(R)
Literate
Programming
(R Markdown)

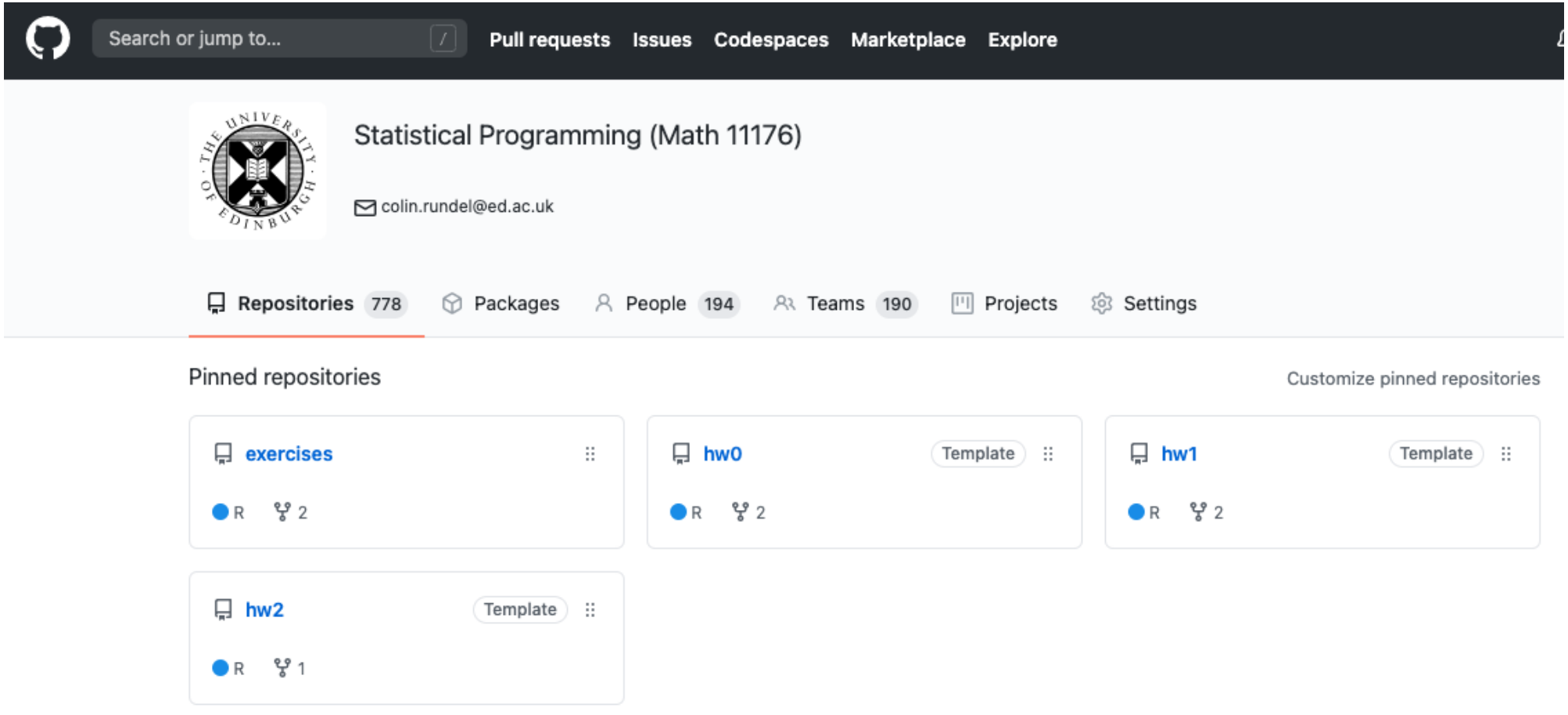
# Context

- I am the course organizer for Math 11176 - Statistical Programming
    - Course with ~200 Maths MSC students enrolled
- 100% coursework, multiple marked assignments (individual and team based)
- For each assignment we distribute:
    - Instruction document
    - Template `Rmd` for solutions
    - Data and other support files
- Need to collect:
    - Completed template `Rmd`
    - Rendered output (`pdf`, `html`, `md`, etc.)

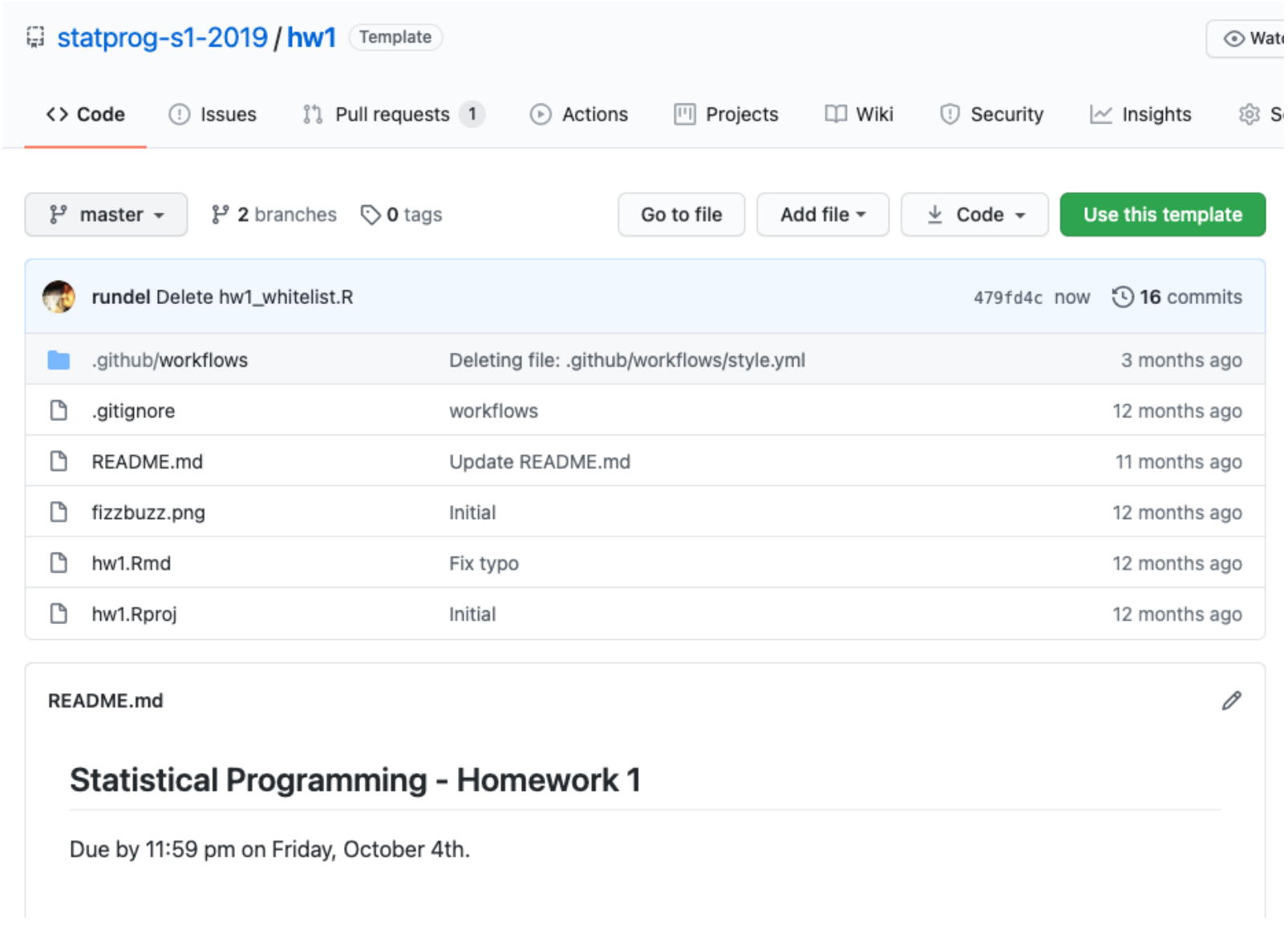
# GitHub Organization

- 1 organization / course
- Students are added (anonymously) members of the organization
- 1 template repository / assignment
- 1 private repository / assignment / (team | individual)
- Automate the distribution, collection, and feedback using GitHub's API (`ghclass`)

# GitHub Organization



# Template Example - hw1



# `ghclass`

An R package that enables instructors to automate the management of courses on GitHub.

Key features:

- Repository creation, mirroring, updating, collecting, etc.
- Organization management (members, teams, etc.)
- Summary statistics (e.g. commits) by repo or over the org
- Many other common tasks (issues, PR, etc.)

For more details see the package website - **https://rundel.github.io/ghclass/**

# Creating a team assignment

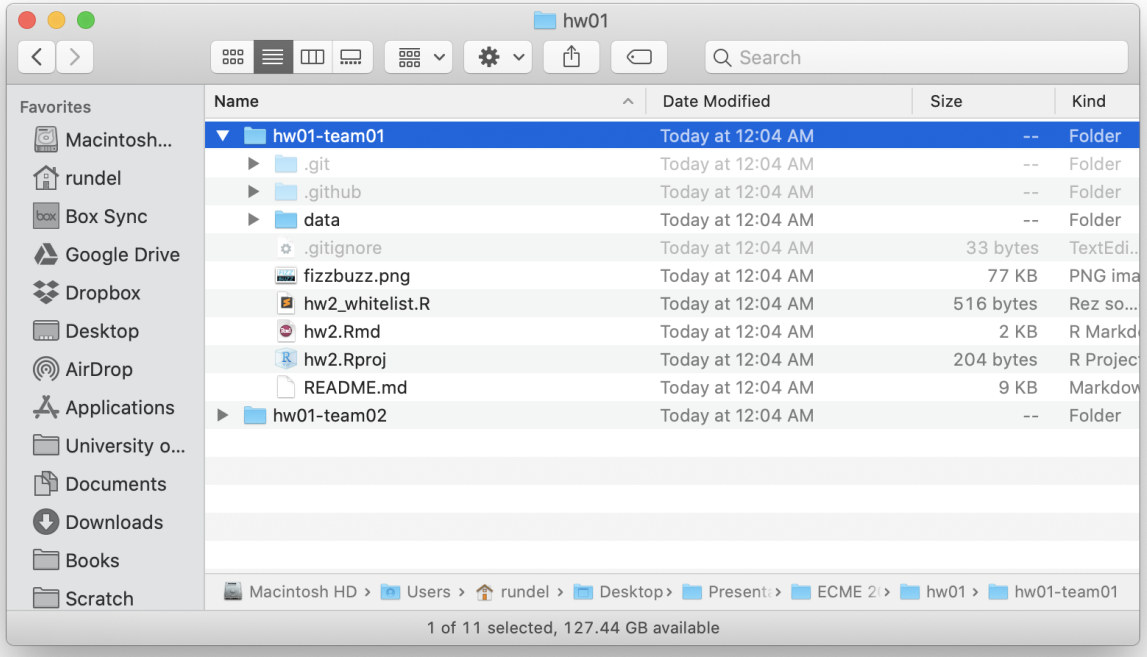
```
org_create_assignment(
  org = "ghclass-demo",
  repo = c("hw01-team01", "hw01-team01", "hw01-team02", "hw01-team02"),
  user = c("ghclass-anya", "ghclass-bruno", "ghclass-celine", "ghclass-diego"),
  team = c("hw01-team01", "hw01-team01", "hw01-team02", "hw01-team02"),
  source_repo = "statprog-s1-2019/hw1"
)
```

```
## ✓ Mirrored repo 'statprog-s1-2019/hw1' to repo 'ghclass-demo/hw01-team01'.
## ✓ Mirrored repo 'statprog-s1-2019/hw1' to repo 'ghclass-demo/hw01-team02'.
## ✓ Created team 'hw01-team01' in org 'ghclass-demo'.
## ✓ Created team 'hw01-team02' in org 'ghclass-demo'.
## ✓ Added user 'ghclass-anya' to team 'hw01-team01'.
## ✓ Added user 'ghclass-bruno' to team 'hw01-team01'.
## ✓ Added user 'ghclass-celine' to team 'hw01-team02'.
## ✓ Added user 'ghclass-diego' to team 'hw01-team02'.
## ✓ Added team 'hw01-team01' to repo 'ghclass-demo/hw01-team01' with 'push' access.
## ✓ Added team 'hw01-team02' to repo 'ghclass-demo/hw01-team02' with 'push' access.
```

# Collecting student work

```
local_repo_clone(repo = org_repos(org = "ghclass-demo", "hw01-"),
                 local_path = "hw01")
```

```
## ✓ Cloned 'ghclass-demo/hw01-team01'.
## ✓ Cloned 'ghclass-demo/hw01-team02'.
```



# Contributor statistics

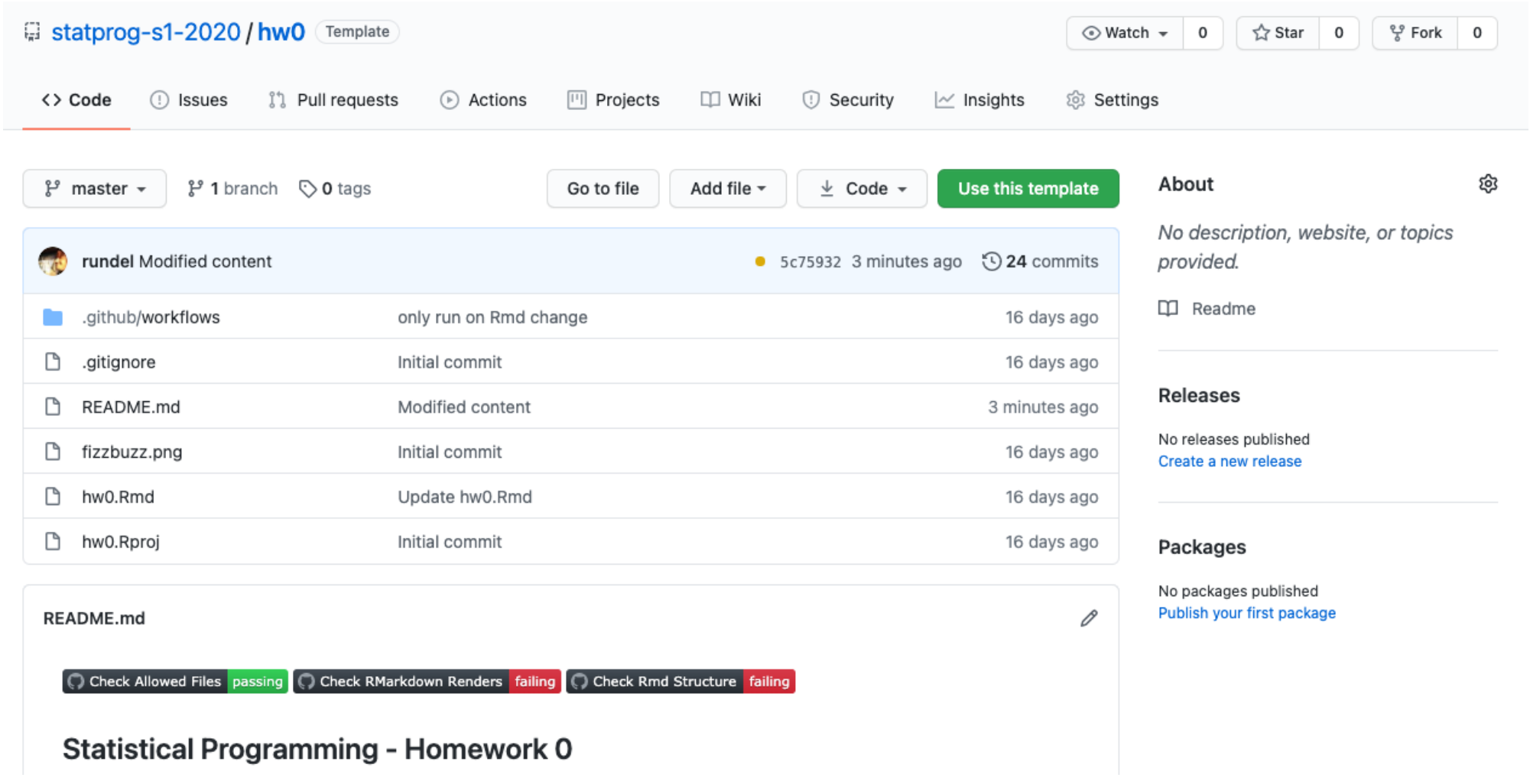
```
repo_contributors(repo = "statprog-s1-2019/hw02-lab01-team03") %>%
  mutate(username = LETTERS[1:4]) %>%
  arrange(desc(commits))
```

```
## # A tibble: 4 x 3
##   repo                                  username commits
##   <chr>                                 <chr>      <int>
## 1 statprog-s1-2019/hw02-lab01-team03 D              8
## 2 statprog-s1-2019/hw02-lab01-team03 B              5
## 3 statprog-s1-2019/hw02-lab01-team03 C              5
## 4 statprog-s1-2019/hw02-lab01-team03 A              3
```

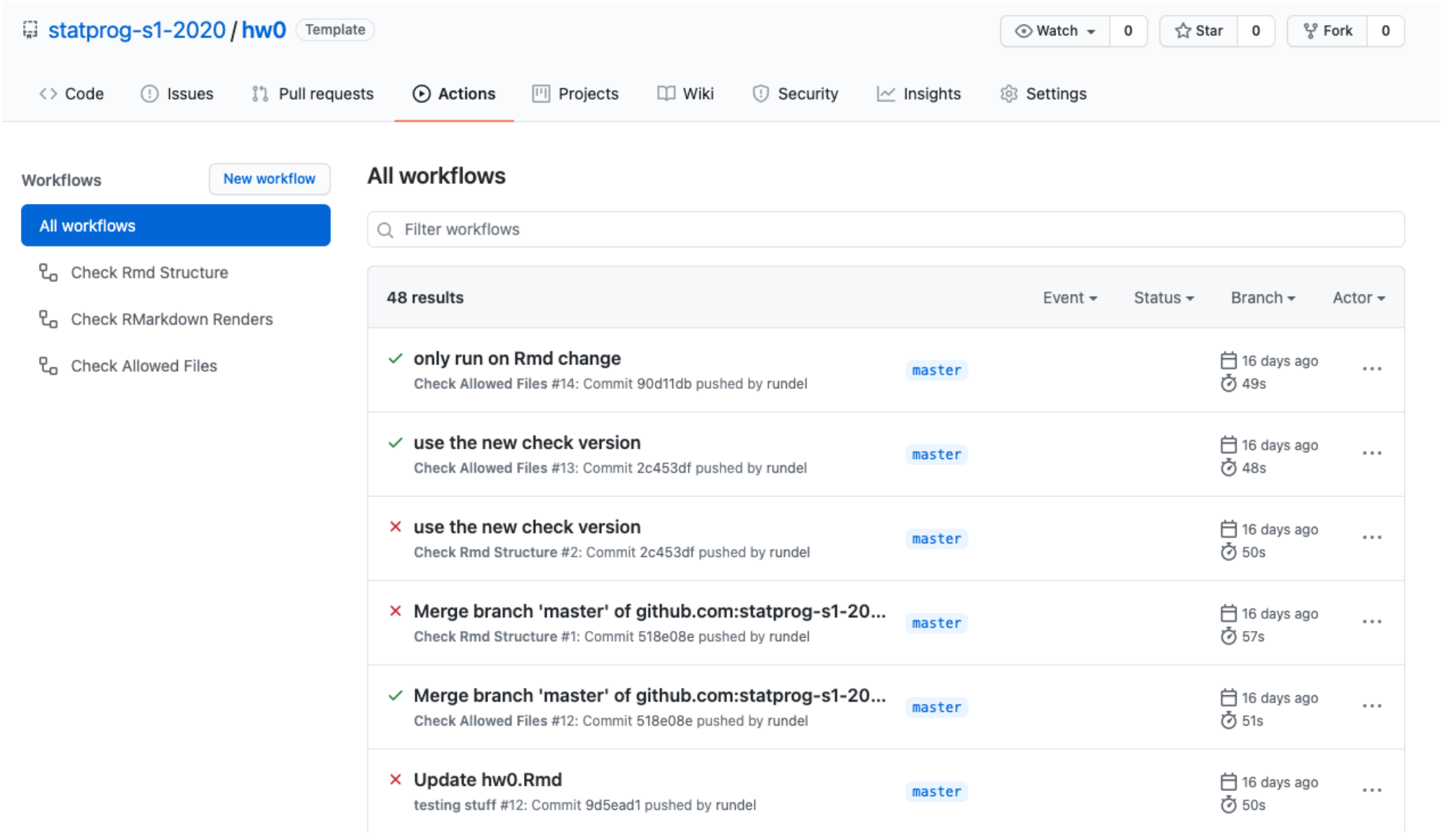
```
repo_contributors(repo = "statprog-s1-2019/hw02-lab01-team10") %>%
  mutate(username = LETTERS[12+1:5]) %>%
  arrange(desc(commits))
```

```
## # A tibble: 5 x 3
##   repo                                  username commits
##   <chr>                                 <chr>      <int>
## 1 statprog-s1-2019/hw02-lab01-team10 Q             17
## 2 statprog-s1-2019/hw02-lab01-team10 P              9
## 3 statprog-s1-2019/hw02-lab01-team10 O              5
## 4 statprog-s1-2019/hw02-lab01-team10 M              1
## 5 statprog-s1-2019/hw02-lab01-team10 N              1
```

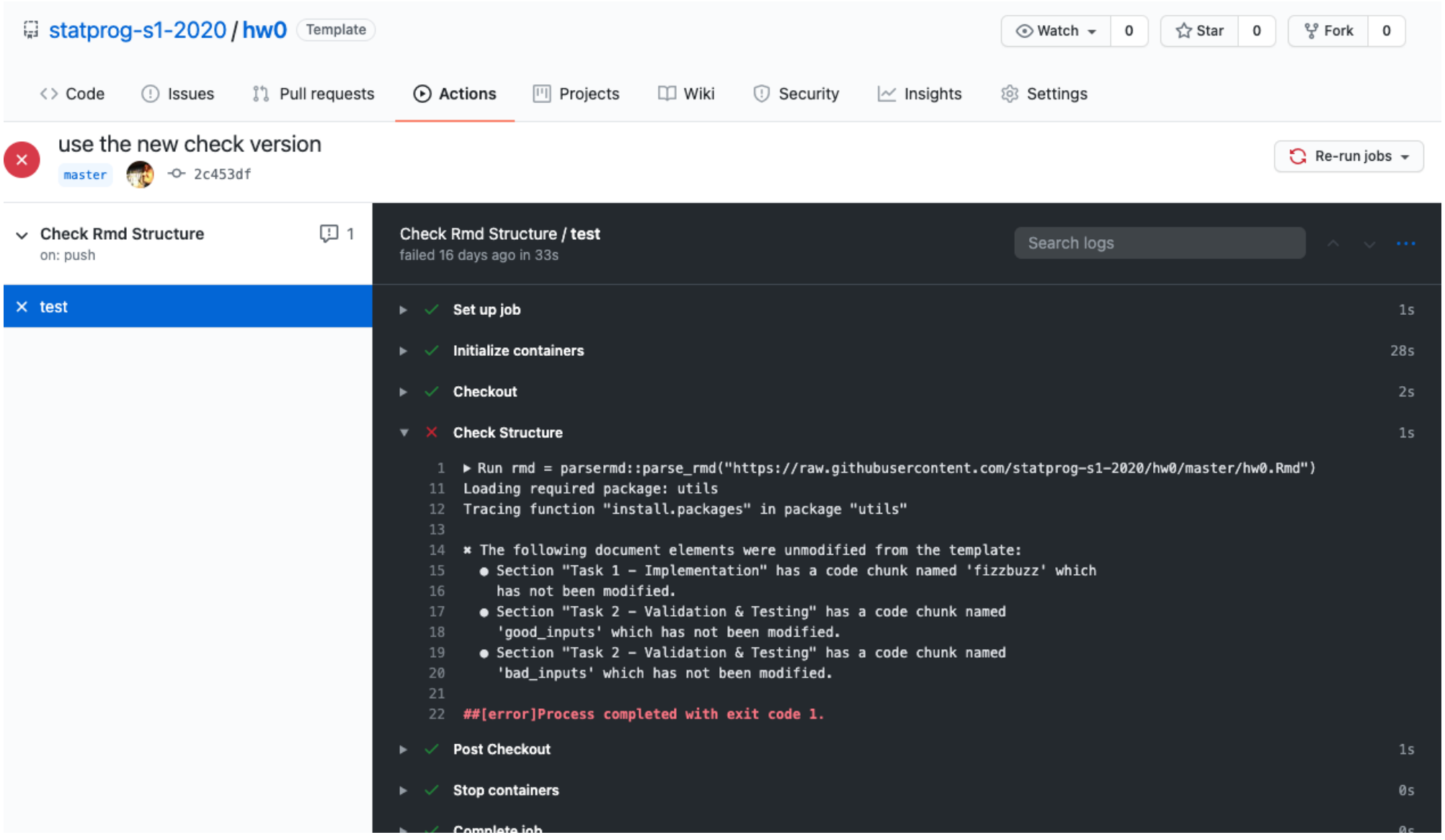
# Automated feedback



# Automated feedback



# Automated feedback



# Related ongoing work

- Peer evaluation (Mine Cetinkaya-Rundel and Therese Anders)
- Simplifying the automated feedback process:
  - `checklist` - R package for simplifying automated checks
    **https://github.com/rundel/checklist**
  - `parsermd` - R package for programmatic interaction with R markdown documents
    **https://rundel.github.io/parsermd/**

# Additional Resources

- **Happy Git and GitHyb for the useR**
  Jenny Bryan, Jim Hester

- **Excuse me, do you have a moment to talk about version control?**
  Jenny Bryan (2018), *The American Statistician*.

- **Using GitHub Classroom To Teach Statistics**
  Jacob Fiksel, Leah Jager, Johannna Hardin, and Margaret Taub (2019), *Journal of Statistics Education*.

- **Implementing version control with Git as a learning objective in statistics courses**
  Matthew Beckman, Mine Çetinkaya-Rundel, Nicholas Horton, Colin Rundel, Adam Sullivan, Maria Tackett (2020), *Journal of Statistics Education (in review)*

- **Teaching Statistics and Data Science Online Workshops**
  Mine Çetinkaya-Rundel, Colin Rundel (2020), *Centre for Statistics Online Workshop*

# Thank you!


rundel@gmail.com
colin.rundel@ed.ac.uk

@rundel

bit.ly/tlmsco_rundel

statprog-s1-2019.github.io/