

03

reproducible workflows

**teaching
data
science**

R Markdown

reproducibility

train new
analysts whose
only workflow is
a reproducible
one

pedagogy

code + output +
prose together

syntax
highlighting
FTW!

efficiency

consistent
formatting →
easier grading

key to success

knit early
and often

Git + GitHub — why?

version control

lots of mistakes
along the way,
need ability to
revert

collaboration

platform that
removes barriers
to well
documented
collaboration

accountability

transparent
commit history

early introduction

mastery
takes time,
earlier start
the better

marketability

Git + GitHub — how?

organization

one organization
per course

one repo per
student/team per
assignment

interface

via RStudio

no local git install
required since
using RStudio
Cloud

teams

for collaboration

for assigning
individual students
to repos

for graders

assessment

check
reproducibility via
clone + compile

feedback through
issues

> Your turn!

- If you don't have a GitHub account, create one now at github.com
- If you do, confirm you know your username and password by logging in at github.com
- Then, enter your name and GitHub username at rstd.io/teach-ds-form

setting up a course

1. Request educational discount: education.github.com/discount
2. Create course organization: github.com/organizations/new
3. Upgrade course organization: education.github.com/benefits
4. Invite students to organization
5. Create assignment(s)
6. Collect assignments(s)
7. Grade assignment(s)

1. request educational discount

education.github.com/discount

Which best describes your academic status? [?](#)

☐ Student ☐ Faculty

What e-mail address do you use for school?

⚡ **Pro-tip:** Selecting a school-issued email address gives you the best chance of a speedy review.

☐ mine@stat.duke.edu

✓ Duke University

☐ cetinkaya.mine@gmail.com

☐ mine.cetinkaya-rundel@ed.ac.uk

✓ University of Edinburgh

+ Add an email address

required information

When requesting the discount you will need to provide the following:


- A brief description of the purpose for the GitHub organization and how you plan to use GitHub
- Establishing connection to an academic institution by verifying with an .edu email + photo of your school id.
- Link to relevant website for the class / workshop / research group


Verification is manual and can take between a couple hours to a couple days.


2. create course organization

github.com/organizations/new

Sign up your team

**Step 1:**
Set up the organization

**Step 2:**
Invite members

**Step 3:**
Organization details

Create an organization account

Organization name

This will be your organization name on <https://github.com/>.

Billing email

We'll send receipts to this inbox.

Organization accounts allow your team to plan, build, review, and ship software — all while tracking bugs and discussing ideas.

Choose your plan

☒ **Free** \$0

Unlimited users and public repositories

☐ **Team** \$9

Starts at \$25 / month which includes your first 5 users.
Unlimited public repositories
Unlimited private repositories

per user / month

The credit card and plan you choose will be billed to the organization — not **rundel** (your user account).

3. upgrade course organization

education.github.com/benefits

Sta112-F18	Coupon already applied
sta771-f18	Coupon already applied
rstudio4edu	Upgrade
ghclass-test	Coupon already applied
ghclass-demo	Coupon already applied
rstudio-conf-2020	Upgrade
jsm19-tds-demo	Coupon already applied
+ Create an organization	

member privileges

Member repository permissions

Base permissions

Base permissions to the organization's repositories apply to all members and excludes outside collaborators. Since organization members can have permissions from multiple sources, members and collaborators who have been granted a higher level of access than the base permissions will retain their higher permission privileges.

None ▾

Organization member permissions

Admin

Members will be able to clone, pull, push, and add new collaborators to all repositories.

Write

Members will be able to clone, pull, and push all repositories.

Read

Members will be able to clone and pull all repositories.

✓ None

Members will only be able to clone and pull public repositories. To give a member additional access, you'll need to add them to teams or make them collaborators on individual repositories.

Allow Forking of private repositories

If enabled, forking is allowed on private and public repositories. If disabled, forking is only allowed on public repositories. This

member privileges

Member repository permissions

Base permissions

Base permissions to the organization's repositories apply to all members and excludes outside collaborators. Since organization members can have permissions from multiple sources, members and collaborators who have been granted a higher level of access than the base permissions will retain their higher permission privileges.

None ▾

Repository creation

If enabled, members will be able to create both public and private repositories, or private repositories only. Outside collaborators can never create repositories.

- ☐ **Public and private repositories**
Members will be able to create public and private repositories.
- ☐ **Private repositories**
Members will be able to create only private repositories. [Why is this option disabled?](#)
- ☒ **Disabled**
Members will not be able to create public or private repositories.

Save

Repository forking

- ☐ **Allow forking of private repositories**
If enabled, forking is allowed on private and public repositories. If disabled, forking is only allowed on public repositories. This setting is also configurable per-repository.

Save

member privileges

Actions

Automate all your software workflows. Build, test, and deploy your code right from GitHub.

- ☒ **Enable local & third party Actions for this organization**
This allows all repositories to execute any Action, whether the code for the Action exists within the same repository, same organization, or a repository owned by a third party.
- ☐ **Enable local Actions only for this organization**
This allows all repositories to execute any Action as long as the code for the Action exists within the same repository.
- ☐ **Disable Actions for the organization**
This disallows any Action from running on any repository in the organization.

Save

Admin repository permissions

Repository visibility change

- ☐ **Allow members to change repository visibilities for this organization**
If enabled, members with admin permissions for the repository will be able to change repository visibility from **public** to **private**. If disabled, only organization owners can change repository visibilities.

Save

Repository deletion and transfer

- ☐ **Allow members to delete or transfer repositories for this organization**
If enabled, members with admin permissions for the repository will be able to delete or transfer **public** and **private** repositories. If disabled, only organization owners can delete or transfer repositories.

Save

member privileges

Issue deletion Beta

- ☐ **Allow members to delete issues for this organization**
If enabled, members with admin permissions for the repository will be able to delete issues.

Save

Repository Comments

- ☐ **Allow members to see comment author's profile name in private repositories**
If enabled, members will be able to see comment author's profile name in issues and pull requests for private repositories.

Save

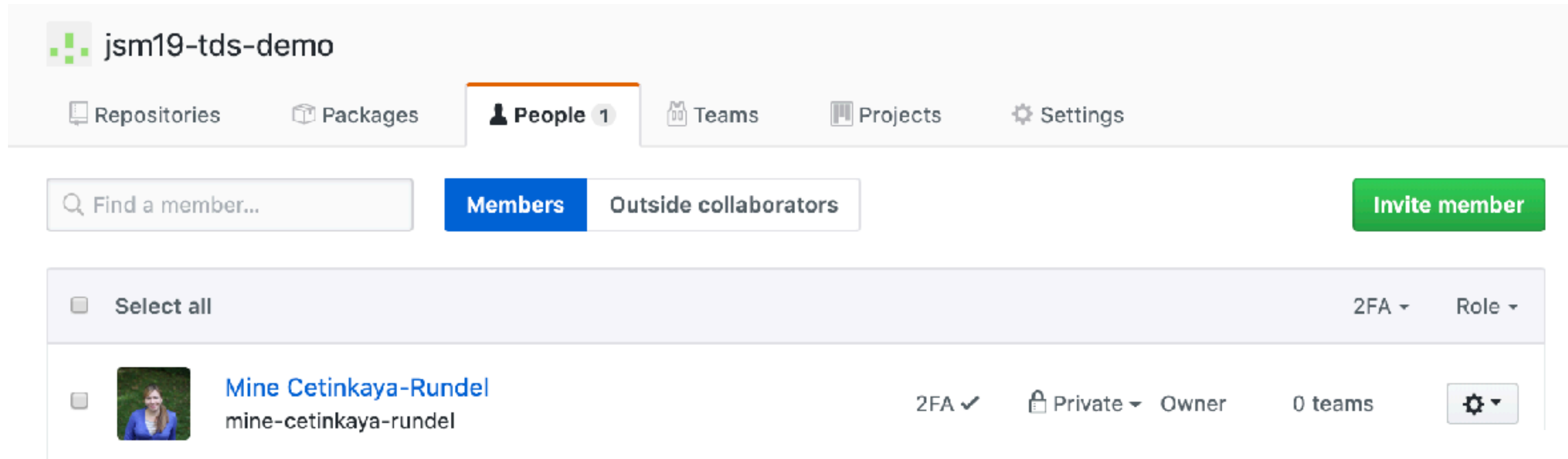
Member team permissions

Team creation rules

- ☐ **Allow members to create teams**
If enabled, any member of the organization will be able to create new teams. If disabled, only organization owners can create new teams.

Save

4. invite students



The screenshot shows the GitHub interface for the organization 'jsm19-tds-demo'. The 'People' tab is selected, showing a list of members. The only member listed is 'Mine Cetinkaya-Rundel' (mine-cetinkaya-rundel), who is the Owner of the organization. The interface includes navigation tabs for Repositories, Packages, People (1), Teams, Projects, and Settings. A search bar for finding members is present, along with buttons for 'Members', 'Outside collaborators', and 'Invite member'.

Select all	2FA	Role
<input type="checkbox"/>	2FA ✓	Owner

Invite Mine's testing agent to ghclass-demo

Give them an appropriate role in the organization and add them to some teams to give access to repositories.

Role in the organization

- ☒ **Member**
Members can see all other members, and can be granted access to repositories. They can also create new teams and repositories.
- ☐ **Owner**
Owners have full administrative rights to the organization and have complete access to all repositories and teams.

automate invitations

Inviting students to the organization only needs to be done once be class, but the process gets tedious for more than a handful of students.

We have developed an R package that automates this (and other class related tasks) called **ghclass**.

```
library(devtools)  
install_github("rundel/ghclass")
```

GitHub tokens

ghclass uses the GitHub API to interact with your course organization and repos - the API verifies your identity using a personal access token which must be created and saved in such a way that **ghclass** can find and use it.

These tokens can be created here and once created should be saved to `~/.github/token` or assigned to the `GITHUB_TOKEN` environmental variable.

check tokens

If the token is found and works correctly the following code should run without error:

```
library(ghclass)  
test_github_token()
```

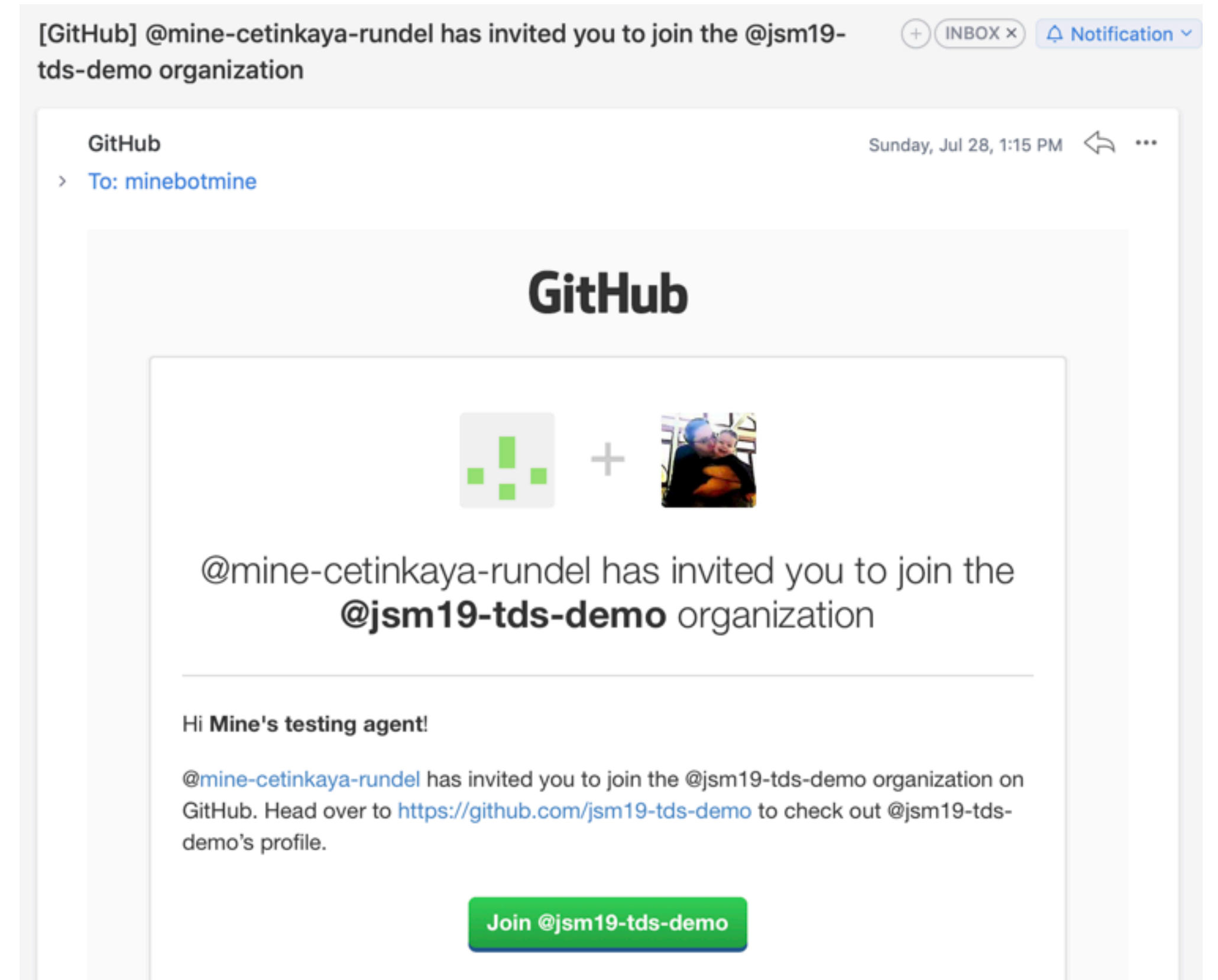
If instead the token is invalid or not found, you will see something like the following:

```
test_github_token("MADE_UP_TOKEN")  
## Error in gh("/user", .token = token): GitHub API error (401): 401 Unauthorized  
## Bad credentials
```

invite students

```
org_invite("ghclass-demo", roster$ghname)
```

✓ Invited user 'minebotmine' to org 'jsm19-tds-demo'.



check student status

```
org_members("ghclass-demo")
```

```
[1] "mine-cetinkaya-rundel"
```

```
org_pending_members("ghclass-demo")
```

```
[1] "minebotmine"
```

4. create assignments

There are a few moving parts here, so we will break it down into several steps. For each assignment we do the following:

1. Create a template repository that contains starter documents for an assignment
2. Create assignment

4.1 create template repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner Repository name *

 / hw-01 

Great repository names are short and memorable. Need inspiration? How about [crispy-sniffle](#)?

Description (optional)

A demo homework assignment

- ☐ **Public**
Anyone can see this repository. You choose who can commit.
- ☒ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

- ☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **None** 

Create repository


[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

A demo homework assignment [Edit](#)


[Manage topics](#)

[2 commits](#) [1 branch](#) [0 releases](#) [1 contributor](#)

[Branch: master](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

 mine-cetinkaya-rundel New assignment! Latest commit 3b2e371 39 seconds ago

.gitignore	New assignment!	39 seconds ago
README.md	New assignment!	39 seconds ago
hw-01-dinosaur.Rmd	New assignment!	39 seconds ago
hw-01.Rproj	New assignment!	39 seconds ago

[README.md](#) 

hw-01

Due 2019-07-24

A demo homework assignment

4.2 create assignment

```
org_create_assignment(org = "ghclass-demo",  
                      repo = paste0("hw-01-", roster$ghname),  
                      user = roster$ghname,  
                      source_repo = "ghclass-demo/hw-01")
```

- ✓ Created repo 'jsm19-tds-demo/hw-01-minebotmine'.
- ✓ Added user 'minebotmine' to repo 'jsm19-tds-demo/hw-01-minebotmine'.
- ✓ Cloned 'jsm19-tds-demo/hw-01'.
- ✓ Pushed (mirror) 'hw-01' to repo 'jsm19-tds-demo/hw-01-minebotmine'.
- ✓ Removed local copy of 'jsm19-tds-demo/hw-01'

> Your turn!

First, I will create your repositories...

You're the student:

- Go to github.com/ghclass-demo and locate your HW 01 repository.
- Create a new project from GitHub in the RStudio Cloud workspace for this workshop: rstd.io/teach-ds-cloud
- In the **Console**, run the following:

```
library(usethis)
use_git_config(user.name = "Jane", user.email =
  "jane@example.org")
```

- Make changes to the Rmd file, stage, commit, push

create teams

GitHub supports the creation of teams within an organization, these teams can then be assigned a shared repository.

We can use `ghclass` to create these teams and add students to them.

```
team_create(org = "ghclass-demo", unique(teams))
```

✓ Created team 'team4' in org 'jsm19-tds-demo'.

```
team_invite(org = "ghclass-demo",  
            user = roster$ghname,  
            team = roster$team)
```

✓ Added 'minebotmine' to team 'team4'.

create team assignments

GitHub supports the creation of teams within an organization, these teams can then be assigned a shared repository.

```
org_create_assignment(org = "ghclass-demo",  
                      repo = paste0("hw-02-", roster$team),  
                      user = roster$ghname,  
                      team = roster$team,  
                      source_repo = "ghclass-demo/hw-02")
```

- ✓ Created repo 'jsm19-tds-demo/hw-02-team4'.
- ✓ Added 'minebotmine' to team 'team4'.
- ✓ Added team 'team4' to repo 'jsm19-tds-demo/hw-02-team4'.
- ✓ Cloned 'jsm19-tds-demo/hw-02'.
- ✓ Pushed (mirror) 'hw-02' to repo 'jsm19-tds-demo/hw-02-team4'.
- ✓ Removed local copy of 'jsm19-tds-demo/hw-02'

> Your turn!

You're the student:

Make some more changes to your HW 01 and "submit" by making one last push.

Let's review the collecting process together.

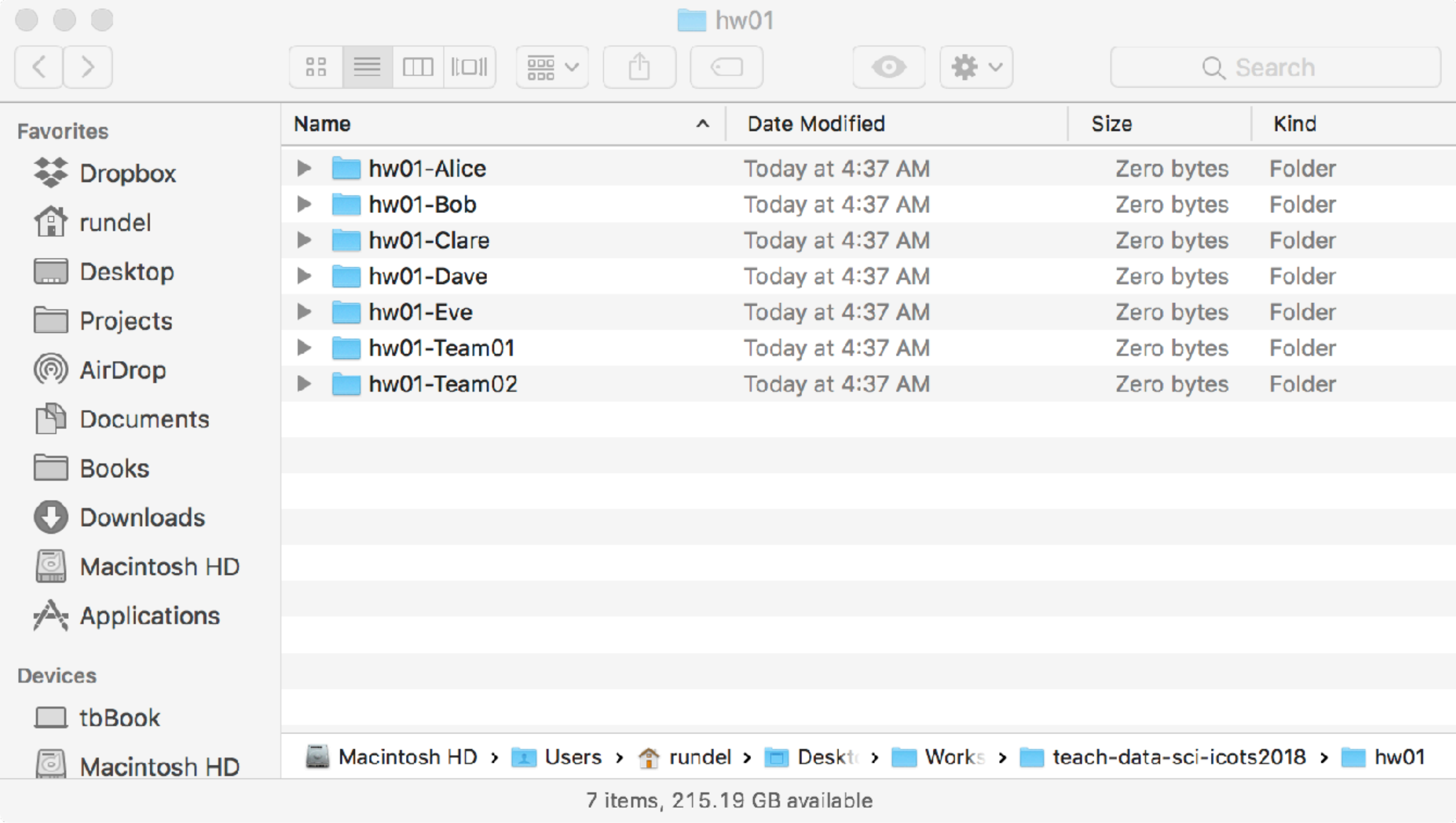
5. collect assignments

```
hw01_repos ← org_repos(org = "ghclass-demo", filter =  
"hw-01-")
```

```
local_repo_clone(repo = hw01_repos,  
                  local_path = "hw-01-collect")
```

✓ Cloned 'jsm19-tds-demo/hw-01-minebotmine'.

6. grade assignments



feedback - issues

Instructors (and TAs) can view all repositories within the course organization:

- You can open issues in a repository with feedback for the students.
- Use the blame view to get specific line references.
- Make sure to @ mention the student so that they are notified when an issue is opened.

Note: You might want to consider keeping points out of issues.

feedback - peer review

- Once an assignment is completed you can let other students/teams into a repository and they can provide peer review.
- Peer review is an incredibly effective learning experience for both the reviewers and the reviewees, however it does require coordination and being able to carve out sufficient time in the course schedule.

Tip: Do not solely count on peer review for feedback as some reviewers might be less diligent than others. Teams reviewing teams, as opposed to individual reviewing individuals, might address this issue partially.

feedback - pull requests

- Another option is to open pull requests for your students' work where you directly edit their work and ask them to approve the edits.
- This can be effective as students will see your corrections and review them before accepting them.
- However this also does mean that you're directly correcting their work as opposed to giving them higher level instructions on how to correct it.

styler revisions #1

Edit

 Draft mine-cetinkaya-... wants to merge 1 commit into master from styler

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1

+16 -18 



mine-cetinkaya-... commented 14 minutes ago

+  ...

This pull request contains the results of running the automated R code formatting tool styler on your repo. Styling is based on the [tidyverse R style guide](#).

Click on the commit below to see details of recommended changes. It is not necessary that your code cleanly pass these checks, but if there is a large number of significant changes suggested you should review the style guide with an eye towards potentially improving your code formatting.


@ghclass-any

  Results of running styler: ...

721c39c

Add more commits by pushing to the **styler** branch on [ghclass-demo/hw-03-ncbikecrash-ghclass-any](#).



 **This pull request is still a work in progress**
Draft pull requests cannot be merged.

Ready for review

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers



Suggestions

 ghclass-any

[Request](#)

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet


Milestone



No milestone

Notifications

[Customize](#)

 Unsubscribe

You're receiving notifications

github demo / hw-03-ncbikecrash

Code

Issues 0

Pull requests 1

Projects 0

Wiki

Security

Insights

Settings

styler revisions #1

Edit

Draft

mine-cetinkaya-... wants to merge 1 commit into master from styler

Conversation 0

Commits 1

Checks 0

Files changed 1

+16 -18

Changes from all commits

File filter...

Jump to...

Review changes

Results of running styler:

Styling 1 files:
hw-03-ncbikecrash.Rmd

Status	Count	Legend
✓	0	File unchanged.
i	1	File changed.
x	0	Styling threw an error.

Please review the changes carefully!

styler (#1)

mine-cetinkaya-rundel

committed 16 minutes ago

commit 721c39cf8b6c0e6b3bc50f2918747415910b980e

34

hw-03-ncbikecrash.Rmd

@@ -7,7 +7,7 @@ output: github_document

7

8

9

10

``r load-packages, message=FALSE}

load packages

-library(tidyverse)

7

8

9

10

``r load-packages, message=FALSE}

load packages

+library(tidyverse)

Git + GitHub — lessons learned

if you plan on using git in class, start on day one, don't wait until the "right time"

first assignment should be individual, not team based to avoid merge conflicts

remind students on that future projects should go on GitHub with PI approval

students need to remember to pull before starting work

impossible (?) to avoid shell intervention every once in a while