03
reproducible
workflows

teaching
data
science

> Your turn!

Welcome back!

Turn to your neighbor and discuss:

- What is one evidence-based teaching practice you have adopted in the last five years?
- What is one evidence-based practice you know about but have **not** adopted and why not?

R Markdown

reproducibility	pedagogy	efficiency	key to success
train new analysts whose only workflow is a reproducible one	code + output + prose together syntax highlighting FTW!	consistent formatting → easier grading	knit early and often

version control	collaboration	accountability	early introduction
lots of mistakes along the way, need ability to revert	platform that removes barriers to well documented collaboration	transparent commit history	mastery takes time, earlier start the better
			marketability

organization	interface	teams	assessment
one organization per course one repo per student/team per assignment	via RStudio no local git install required since using RStudio Cloud	for collaboration for assigning individual students to repos for graders	check reproducibility via clone + compile feedback through issues

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

> Your turn!

- If you don't have a GitHub account, create one now at **github.com**
- If you do, confirm your 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

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

Sign up your team

Step 1: Set up the organization	Step 2: Invite members	Step 3: Organization details

Create an organization account

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

Organization name

This will be your organization name on https://github.com/.
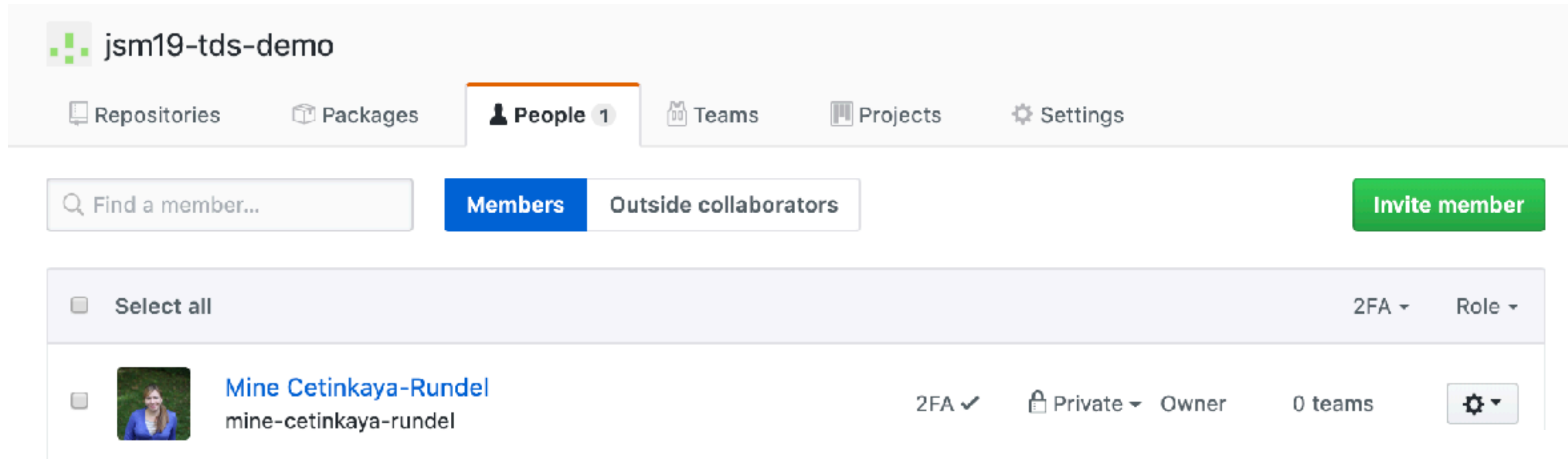
Billing email

We'll send receipts to this inbox.

Choose your plan

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

- ● Free $0
  Unlimited users and public repositories

- ○ Team $9
  Starts at $25 / month which includes your first 5 users.   per user / month
  Unlimited public repositories
  Unlimited private repositories

3. upgrade course organization

4. invite students

member privileges

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]

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]

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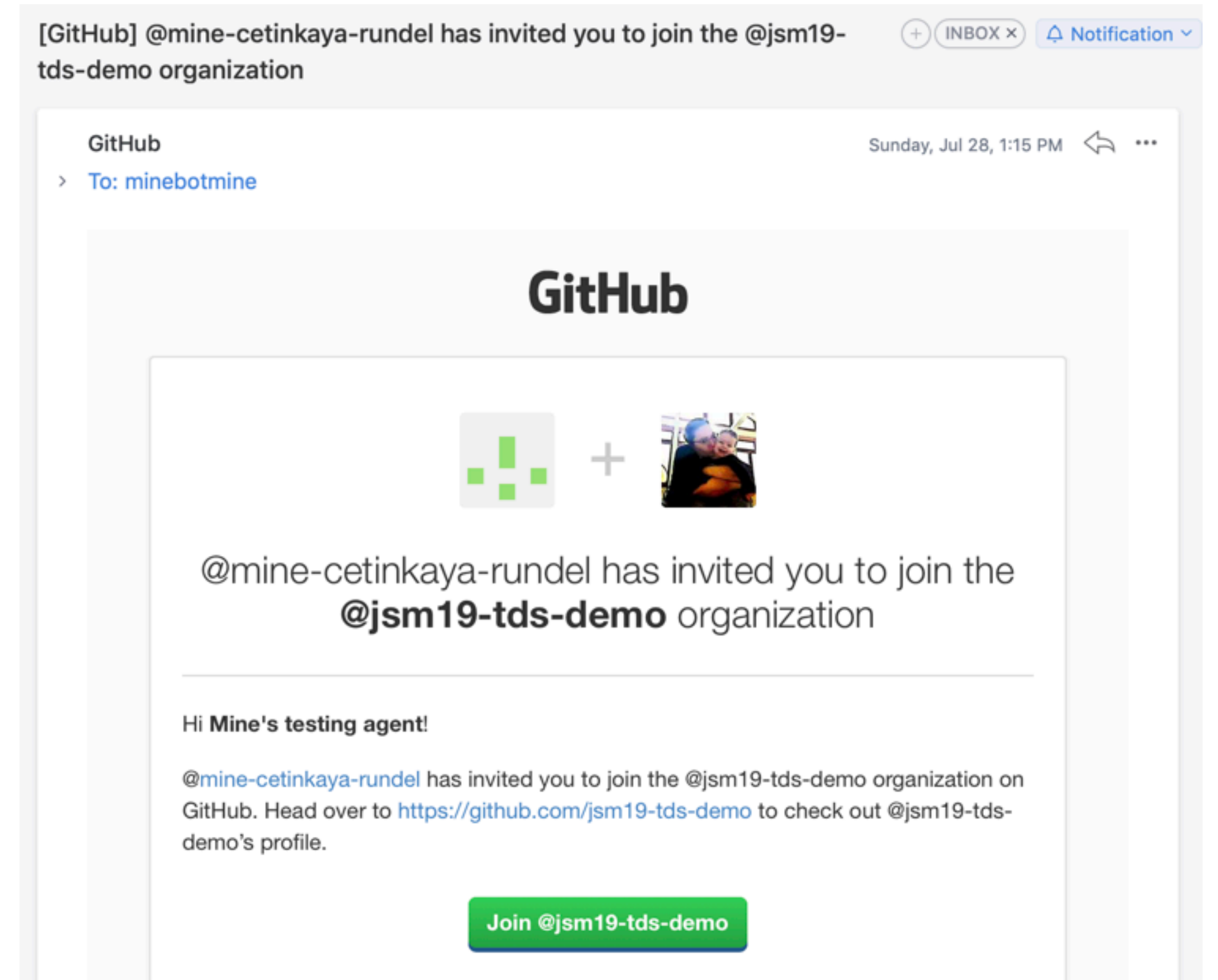
```
test_github_token("MADE_UP_TOKEN")
## Error in gh("/user", .token = token): GitHub API error (401): 401 Unauthorized
##   Bad credentials
```

invite students

```
org_invite("jsm19-tds-demo", roster$ghname)
```

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

check student status

```
org_members("jsm19-tds-demo")
```

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

```
org_pending_members("jsm19-tds-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

4.2 create assignment

```
org_create_assignment(org = "jsm19-tds-demo",
                      repo = paste0("hw-01-", roster$ghname),
                      user = roster$ghname,
                      source_repo = "jsm19-tds-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/jsm19-tds-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:

```
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 = "jsm19-tds-demo", unique(teams))
```

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

```
team_invite(org = "jsm19-tds-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 = "jsm19-tds-demo",
                      repo = paste0("hw-02-", roster$team),
                      user = roster$ghname,
                      team = roster$team,
                      source_repo = "jsm19-tds-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'

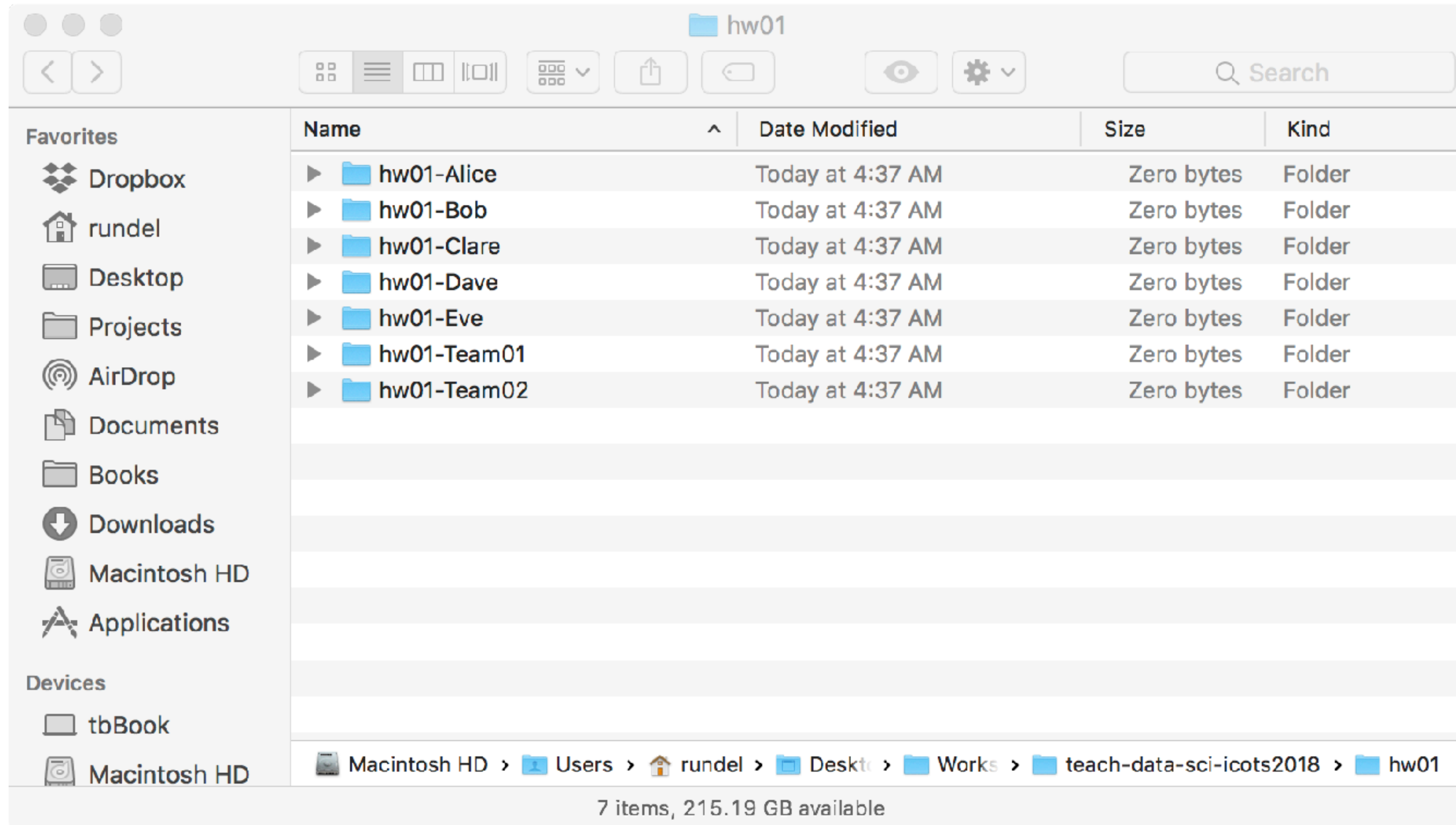# 5. collect assignments

```
hw01_repos ← org_repos(org = "jsm19-tds-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

> 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.

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.