

From 50 to 150: growing in numbers while improving in quality

Anastasis Georgoulas & David Pérez-Suárez
Advanced Research Computing

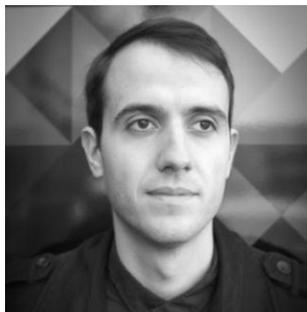
In short...

Challenges:

- Maintaining student engagement
- Coordination
- Scale of assessment

Approach:

- Flipped classroom with group work
- Tools to keep team and students on the same page



and supported by many more!

Design

- Learning Designer
- Designed collaboratively, pairing more with less experienced instructors
- Reviewed by whole team, including TAs
- Agreed principles

The screenshot shows a Learning Designer interface for a course titled "Data interaction". The course details are as follows:

- Name:** Data interaction
- Topic:** 4 hours and 30 minutes
- Learning time:** 140
- Size of class:** (blank)
- Description:** (blank)
- Mode of delivery:** Wholly online
- Aims:** Demonstrate how to read, write and visualise data from a var...
- Outcomes:** Contrast, Generate, Produce, Describe reasons for
- Editor:** rsdgroup

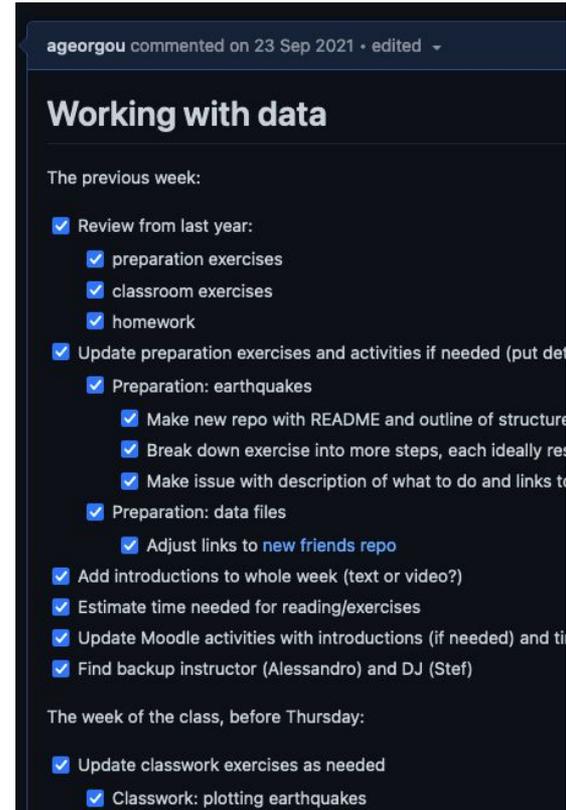
On the right side, there is a pie chart with six segments labeled Pro, Acq, Col, Dis, Inq, and Pra. Below the chart is a button that says "Turn editing on".

The main content area is divided into three columns, each representing a different activity or resource:

- Column 1: Structured data formats**
 - Read Watch Listen: 30 minutes, 1 icon, 1 icon, 1 icon
 - Read notes: Working with files, Internet, CSV, Structured datafiles
 - Practice: 20 minutes, 1 icon, 1 icon, 1 icon
 - Exercise on reading and writing an existing data structure in YAML and/or JSON
- Column 2: Working with a real scientific data set – Part 1 (reading, exploring and analysing)**
 - Read Watch Listen: 5 minutes, 1 icon, 1 icon, 1 icon
 - Read exercise description
 - Collaborate: 10 minutes, 2 icons, 0 icons, 0 icons
 - Understand the structure of the returned data (perhaps guided by some sample questions)
 - Produce: 30 minutes, 2 icons, 1 icon, 0 icons
 - Search through the data to find the biggest earthquake and its location
 - Produce: 15 minutes, 2 icons, 1 icon, 0 icons
 - Generate a map image for the area identified
- Column 3: Working with a real scientific data set – Pa and plotting)**
 - Read Watch Listen: 30 minutes, 1 icon, 1 icon, 1 icon
 - Read notes: Plotting, Numerical Python
 - Collaborate: 15 minutes, 2 icons, 2 icons, 1 icon
 - Agree on a design for this stage
 - Produce: 40 minutes, 2 icons, 2 icons, 1 icon
 - Implement the solution and produce a shared with the whole class
 - Discuss: 15 minutes, 140 icons, 1 icon
 - Discuss obstacles, different approaches benefits of numpy).

Preparation

- GitHub issues to track work
- Share roles:
 - Instructor, backup
 - “DJ”
 - Time and attendance tracking
- Shared document (hackmd) with plan, reviewed beforehand



ageorgou commented on 23 Sep 2021 • edited

Working with data

The previous week:

- Review from last year:
 - preparation exercises
 - classroom exercises
 - homework
- Update preparation exercises and activities if needed (put details in issues)
 - Preparation: earthquakes
 - Make new repo with README and outline of structure
 - Break down exercise into more steps, each ideally resolvable
 - Make issue with description of what to do and links to resources
 - Preparation: data files
 - Adjust links to [new friends repo](#)
- Add introductions to whole week (text or video?)
- Estimate time needed for reading/exercises
- Update Moodle activities with introductions (if needed) and time
- Find backup instructor (Alessandro) and DJ (Stef)

The week of the class, before Thursday:

- Update classwork exercises as needed
 - Classwork: plotting earthquakes

Delivery

- **Activities:**
 - Programming tasks in groups
 - Discussions around code or topic
- **Tools:**
 - hackmd for keeping track
 - GitHub for description and code
 - Zoom breakout rooms (online)
 - Mentimeter for checking in and input

Poll: Data formats

Which data formats have you used and for what?
[Go to the voting page!](#)

Exercise: Plotting the quakes dataset

For this exercise, we will work with the same data to produce two relevant plots!

In smaller groups, discuss what steps you need to take, come up with a plan, and start developing your solution.

Classwork repo: [#115](#)

If you need help, you can ask for us to drop in, following [these instructions](#).

Discussion of the plotting exercise

Back in your groups from the introduction, discuss the exercise. For example:

- Could you reuse your previous work easily? Why/why not?
- Did you use any external libraries?
- What was the hardest part?

Retrospective

- Write comments during class.
- Immediately after class, discuss with the whole team.

Going well

- 14 PRs from 17 groups! that's good!!
- Very nice pace!
- .

We can improve

- Group 06 had about 50% of people not showing up(video on but no answers)/answering to being called out/responding to requests for input. Similarly, many admitted to not doing the prep.
- (Matthew) Some groups only had 1 person "online". Maybe we could move them to more active groups so they don't miss out on the group work experience.
- worried that only 35 people filled the mentimeter 😞
- 👤 Stef I regret going into tuples and sets, are they in the notes?
 - They are! (pew)
- 👤 Stef timing? or we just update the notes for timing
 - Perfect! ;)
- 👤 Alessandro I need to wait longer to "force" them to talk more.

Other

After class

Homework:

- Submit work via GitHub
- Feedback through code review (from students or instructors)
- Asynchronous discussions

Goals:

- Getting used to workflow
- Peer assessment (scalability)
- Building up to coursework

The screenshot shows a GitHub pull request interface. At the top, a comment from user **dpshelio** is displayed, dated 21 Oct 2021. The comment reads: "Great contribution! (right @ageorgou ?) This contribution still needs a README.md file under greece linking to the islands. Also, europe/README.md file needs updated to link to Greece." Below the comment is a commit history section. It shows a commit by an anonymous user (yellow bar) "added 2 commits 6 months ago". The first commit is "Added README file (gracias David)" with hash 722861a. The second commit is "Updated README.md in Europe link to Greece README" with hash 338697f, which is marked as approved with a green checkmark. Below the commit history, a green checkmark icon indicates that **dpshelio** has approved these changes on 21 Oct 2021, with a "View changes" button. At the bottom, another comment from **dpshelio** is shown, stating "dpshelio left a comment" and "Contributor", with three thumbs-up emojis below it.

Assessment

- Submittivity
 - Automated + manual marking
 - Allow testing before submission
 - Helps consistency
 - Work underway to facilitate adoption
- Group coursework
 - Include collaborative aspect, requested by student feedback
 - Preparation: fixing groups, giving feedback, tools and workflow

Autograding Testcases Expand All Close All Regrade Active Version ▾

41 / 52	Autograding Total (Without Hidden Points)	
41 / 52	Autograding Total (With Hidden Points)	
15 / 15	Uncompress, cleaning checking Git repository	Compilation Errors and/or Warnings. Show Details
	LOAD: Basic glacier data and mass-balance data	Show Details
0 / 4	ANALYSIS: Sorting by mass-balance	Show Details
2 / 2	ANALYSIS: Finding nearest glaciers	Show Details
2 / 2	ANALYSIS: Implementation of distance function	Show Details
2 / 2	ANALYSIS: Filtering by full code	Show Details

4.5 / 5 LOAD: Mass-balance data (Sheet EE) Graded by cceaag

- 0.5 You are only allowing integer values for mass-balance, which was not a specification. This can lead to unintuitive behaviour.
- 0 Matching by glacier ID is more robust than by name: it is possible (if unlikely) that two glaciers may have the same name, but the identifiers are unique.
- 0 Good idea to use a dictionary so you can quickly access a `Glacier` by its identifier.

5 / 5 LOAD: Correct handling of sub-regions for mass-balance Graded by cceaag

- 0 Full Credit

4 / 0 ANALYSIS: Sorting by mass-balance Verify Grader Graded by ccaemgr

- 4 %% Needed manual inspection but otherwise fine %%
In your implementation you have sorted the glaciers by the differences between successive mass balance measurements rather than sorting by the mass balances themselves (which represent year-on-year changes in mass). As the question wording was not entirely clear here you have not been penalised for this as your implementation otherwise appears to be correct and meet the
- 0



Learning Designer



GitHub



HackMD



Summary and lessons learned

- Takes more time than you think!
- Inspect the past to inform the future.
- Prepare students throughout the course.

“The Python and GitHub skills obtained from this module have been extremely valuable, and have also been very useful in all other modules.”

Thank you!