Department of Earth, Ocean and Atmospheric Sciences

Future-ready computing & quantitative skills; opensource solutions in Earth Science courses

Tara Ivanochko (PI), Phil Austin (Lead), Francis Jones (STLF)

3-year TLEF: *Opensource Computing for Earth Sciences Education* (OCESE)

Project goals

- Introduce **opensource computing facilities** to enhance quantitative & computing learning in EOSC, ENVR, ATSC courses;
- Establish sustainable local & institutional **infrastructure** and **workflows**;
- **Support faculty** to transition courses or course components.

jupyter {book} jupyterhub

Contributors *so far*:
11 Faculty: T. Ivanochko (P.I.), P. Austin (Lead), F. Jones (STLF), C. Johnson, V. Radic, A. Ameli, R. Merrill, S. Waterhouse, M. Maldonado, K. Orians, S. Sutherland
12 Students: J. Byer, B. Chang, M. Colclough, Y. Egorovo, Y. Kuzmenko, A. Loeppky, J. McFarlane, D. Platonov, C. Rodell, F. Rossmann, Y. Su, H. Umashankar.

Opensource computing facilities we are working on

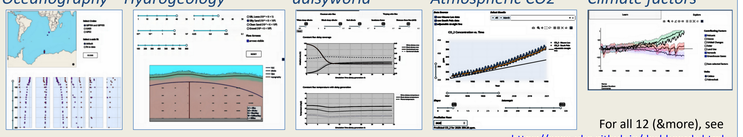
- **Jupyter notebooks**, including auto-grading, question management, and other plugins
- **Open & interactive textbooks** and related resources (e.g. Jupyter Books)
- **Question** and assessment tactics
- **Interactive dashboards** to explore Quantitative Earth Science concepts & data sets.
- **Hubs** and cloud-computing processes and workflows.
- **Collaborative development** steps with colleagues in UBC & beyond.

Open Education Resources (OER) being produced

OER Interactive learning resources

- **Jupyter Notebooks** now in: EOSC211, EOSC213, EOSC354, ATSC301, more soon.
- **Interactive dashboard** apps: 12 built. 10 piloted in 7 courses.
  Some examples :

Oceanography *Hydrogeology* *"daisyworld"* *Atmospheric CO2* *Climate factors*



For all 12 (&more), see
https://eoas-ubc.github.io/dashboards.html

OER Content

- **Jupyter Books** written for, or adapted from, open source:
  > Courses: e211, a301, a409 >startup & sftwr guidelines, >Programming.
  > Project documentation contents
  See https://eoas-ubc.github.io/index.html
- **Question sets** and question management tactics (see below).

Assessments

- Randomizing questions, especially with PrairieLearn.
- Autograding and management using nbgrader (in Jupyter notebooks),
- Interfacing Jupyter, nbgrader, Markdown, Canvas, PrairieLearn.

Python replacing MatLab or R

- In eosc211, 354, 410, 422, 429, envr420.
- Also, a new Python section for DSCI 100.

OCESE Documentation

Project & Outcomes
- Goals and Contributors
- Progress Reports >> (4)
- Course enhancements >> (21)
- Automating assessments
- Dashboards
- Faculty Pro-D
- OCESE Project evaluation
- Scholarly contributions

How-to Guides
- GitHub and Git Workflows
- Developing dashboards
- Dashboard deployment >> (3)
- Jupyter NB startup
- Servers and hubs
- Python startup
- Jupyter Books
- Other software tools
- Some challenges

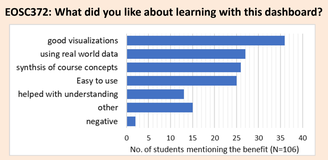
Tutorials
- Command line & Shells
- Conda & environments
- Programming environments
- Git Introduction
- Jupyterhub Tutorial
- Teaching with Jupyter NBs
- Teaching with dashboards

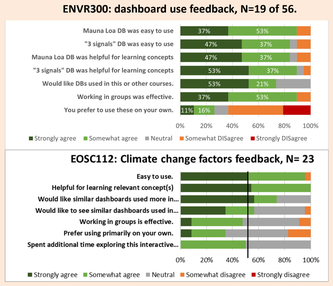
References
- Links & pointers
- Third-party dashboards
- Open Education Resources

Some reactions from students and instructors

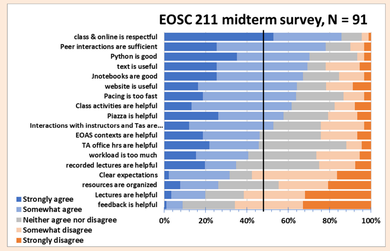
Student feedback survey results (selected)

- Prior knowledge & resources survey for EOSC 211.
- Dashboard feedback examples (well received) →





- Other survey data: EOSC 112, 211, 354, 325.
- Qualtrics feedback for each (1 line of code).
- NOTE: students always provide insightful recommendations after a pilot phase.



A few student quotes

- E112 dashboard feedback results:
  It explains the effects of the contributing factors in a visual way and was straight to the point when explaining with words.
  And *Very organized and easy to use, simple and concise explanations and good visuals.*
  Suggestion: *Maybe add more descriptions on what each contributing factor means and what does the sum selected factors actually do, especially for layperson*
- E372 student feedback: *I liked how the dashboard exercise applied various concepts using real data from the world's oceans. The online dashboard was quite easy to use and interesting as it allowed us to compare different nutrients/properties profiles in different oceans, synthesizing all the concepts in the course.*
- E211 student feedback (highly selected):
  > What was helpful? *"worksheets", "practice problems", "TAs are great", "peers",* etc.
  > What was challenging? "organization", coding and understand the question", "the labs", etc.
  > Recommendations: "more feedback", "explain code line-by-line", "focus in lectures", etc.

A few instructor quotes

- E372 Instructor feedback: *I am so impressed ... I love how (a) sliders constrain and adjust axes, (b) data at various real stations can be chosen on a map and compared, (c) results can be saved to submit for assessment. I agree that now is the time to finalize an assignment, so thank you to the team!*
- Envr300 paraphrased: *The in-class group activity that used the atmospheric CO_2 dashboard worked very well. I will likely use the same procedure when we are in-person again.*
- E325: email end of term: *Thanks so much for detailed assessment. Students suggestions are very helpful and I will reflect on them in the rest of this term and certainly incorporate them in the course design next fall.*

Class or lab observations (in person and online)

- 211: first week (copus) + stwr install lab; notes & TA feedback on effectiveness.
- 354: support and feedback for an instructor (PhD student) new to teaching.
- 112: observations and feedback for a TA delivering a new climate modeling lesson with an activity using IPCC's online Climate Atlas. https://interactive-atlas.ipcc.ch/
- **Iterative design cycle**:
  Prototype → instructor explores→ plan activity→ adjust, revisit, iterate.

Lessons (being) Learned

***NOTE:** Priorities & abilities to participate evolved during 2 years of COVID restrictions.*

Backend infrastructure challenges & effort were greater than anticipated. *Examples:*
- Local vs Cloud; • Hubs vs laptops; • 'small' vs scalable; • containers, libraries & software.
- System debugging during a "live" course was stressful for instructors, TAs and students.

Question mgmt & auto-grading became new, challenging priorities.
- Assessing Jupyter notebooks is straightforward with < ~20 students (e.g., EOSC354)
- Autograding & randomized isomorphic qns are critical for ~90+ students (e.g., EOSC211)
- We are learning from expert colleagues, especially those teaching DSCI 100.

Personal capacity of instructors to adapt lessons collapsed (COVID)
- Evolving online / in-person / hybrid settings are not conducive to innovation.

"**Pythonization**"
- Easier on students than instructors. • Python "setup" (etc.) adds overhead.
- Students in **all** courses are **pleased** that Python is being taught (feedback surveys).
- Adapting "R" → Python for 1 section of DSCI100 (identical learning goals) has been surprisingly complex and labor intensive.

Professional development for instructors and TAs
- Faculty support became one-on-one during COVID.
- Paired teaching → key for EOSC211; 112, 340, 372 less so, other courses not.
- slack → key development, support and evaluation tool for EOSC211.
  - Save channel traffic to Google Sheet; analyze for topics, priorities & timing using zapier.com
  - Eg: "*Should we use a new 'better' library or a simpler, older library to avoid cognitive overload?*"
  - Eg: TAs discussing student difficulties prior to teaching next lab section.
  - Eg: Teaching team discussion that kept scope-creep out of a newly designed lab exercise.
- New course, EOSC325, received extra "SES support" as well as new dashboards.

Open source and OERs
- Critical for software development (colleagues in other UBC dep'ts and institutions)
- Critical that UBC works within the opensource ecosystem (solo = unsustainable)
- Institutional community is active and growing (finally ☺) – needs fostering!
- TAs and instructors learn development techniques (GitHub, Jupyter, etc.)
- Yet - challenging when critical components go un-supported for a while (nbgrader).

Looking Ahead: final year expectations

- **Python &/or Jupyter** for 6 – 12 additional courses (some more ambitious than others)
- **Dashboards**: 4 – 10 more for topics in climate, geophysics, oceanography, & others
- **Documentation:** See → https://eoas-ubc.github.io/index.html
  - project outcomes & evaluation,
  - how-to guides & tutorials,
  - references & resources.

  } jupyter {book}
- **Recommendations** re. UBC computing infrastructure for undergraduate learning.
- **ProD events & activities**
  - Continued support / development for specific courses.
  - Two showcase events in EOAS.
  - One workshop for incorporating dashboards into your course.
  - One workshop on teaching with Jupyter Notebooks.
  - Engage with UBC Jupyter-for-teaching community.
- **Reflections to be written on education development** projects during COVID
  - E.g., assessing impacts on • instructors • students • project team
  - E.g., how & when stakeholders buy in or bow out …

More details:
- GitHub: https://github.com/eoas-ubc/eoas_tlef
- Website: https://eoas-ubc.github.io/
- Summary: https://www.eoas.ubc.ca/education/current-major-initiatives/ocese