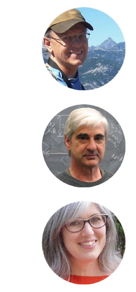
Lessons Learned While Implementing Opensource Computational Tools, Resources and Practices for Learning Quantitative Earth Sciences

Francis Jones, Phil Austin, & Tara Ivanochko. Dep't of Earth, Ocean and Atmospheric Sciences, U.B.C.



https://eoas-ubc.github.io

Earth Educators' Rendezvous
Pasadena, CA, July 10-14, 2023

Creative commons license:
Attribution-NonCommercial-ShareAlike 3.0 (CC BY-NC-SA 3.0 US)
https://creativecommons.org/licenses/by-nc-sa/3.0/us/

The OCESE Project: Opensource Computing for Earth Science Education

1. Project Goals

***Two main goals:** Develop open-source computing capacity…*

1. … to increase quantitative learning in *any* EOAS course, &
2. … enhance computing & math abilities of EOAS students.

***Five goals** in support of the main two:*

3. Develop & test sustainable cloud computing facilities;
4. Produce documentation, resources, guidelines, tutorials;
5. Support faculty to adopt consistent opensource practices;
6. Support UBC's BSc Minor in Data Science, especially DSCI 100;
7. Introduce open education materials & practices.

2. Project Deliverables

No. courses affected:

1. **Python & Jupyter Notebooks** (JNBs); new, or adapt MatLab & 'R'. **8**
2. **Dashboards**: Interactive apps for learning & demonstrations. **10**
3. **Data** gathered about students' and instructors' experiences. **11**
4. **Consulting** re. content, learning, pedagogy, or logistics. **14**
5. **Resources**: Guidelines for Python, JNBs, GitHub, dashboards, etc. **14**
6. **Faculty ProD**: COVID ➔ mainly 1-on-1 consulting.
7. **Dissemination**: 6 UBC events; 5 events beyond UBC.

3. Impacts: Courses and Resources

20 courses participated; ~2900 students affected, 2020 - 2023.

Table 1. Opensource computing to help expose more students to quantitative Earth Sciences

Course	1. jnb	2. dashb	3.data	4. consult	5. resource	6. FProD	7. dissem
ENVR 300		3	2				
EOSC 112		1	2	2			
EOSC 114				1			
EOSC 116				1	1		
EOSC 116, 326		2					
EOSC 310		1		1			
EOSC 323		1					
EOSC 325		3	4	3			
EOSC 340		1		1			
EOSC 372		1	2	1	1		
EOSC 373				1			
EOSC 429		1					
EOSC 442	Y*		1	2	1		

** All 4 computing labs were converted and students' projects required use of corresponding learned skills.*

Three types of changes to courses & Resources

.
.
.

Table 2. Enhance computing / math abilities of EOAS undergrads

Course	1. jnb	2. dashb	3.data	4. consult	5. resource
ATSC 301	y*		2		1
DSCI 100	y*		2	1	4
EOSC 211	y*		12	5	2
EOSC 354	y*	1	2	1	2
EOSC 410/510	y*		1	1	1
EOSC 471	y**				
EOSC 350	y^		1	1	

** Writing code is integral throughout these courses.*
*** All labs use extensive Python but students do little significant coding of their own.*
^ JNBs introduced ~6 yrs ago are used throught the course but students do not write code.

Table 3. Opensource computing capacity in EOAS; not course-specific

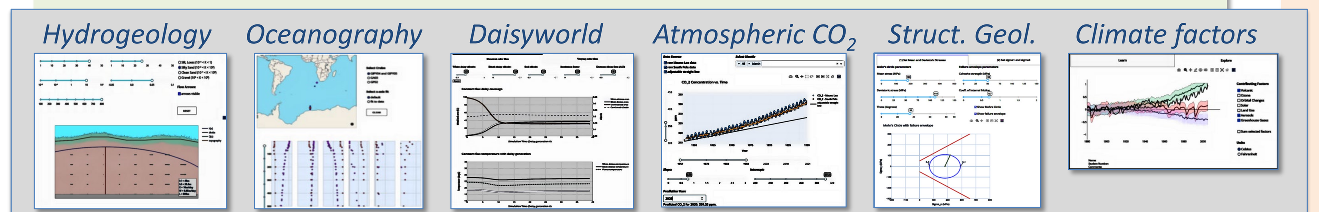
Item	5. resource	6. FProD	7. dissem
Testdrive* UBC Open JNB hub	3		
Testdrive* 3rd party cloud Hub	1		
Dep't server for dashboards	1		
Assess dep't computing needs	1		
Docs: accomplishments	1		
OERs: project repository	1		
Docs: Project website**		… 1 …	
Event: eoas		3	1
Event: outside			4
Event: UBC			4

**Cloud-based Notebook hubs must be stable & scalable*
***Website includes reporting, tutorials & guidelines docs.*
See https://eoas-ubc.github.io/

4. Lessons learned: Engaging with data & concepts

Dashboards: *interactive learning resources to engage students with quantitative concepts and data.*

- **Low-stakes**, easy to adopt, BUT instructors need inspiration.
- **Early vs late adopters**: We had 3 early, 6 late & now >20 are keen.



See all 17 at https://eoas-ubc.github.io/dashboards.html

Build, deploy, sustain

- **Coding skills** needed are "strong undergraduate" level.
- **Opensource** code libraries enable licensing as OERs.
- **Start with** interactive & explorative learning goals.
- **Geoscience education coordinator** minimizes instructor time and supports pedagogic best practices.
- **Iterate**: design/build, pilot V1 with students, feedback, V2.
- **Temporary host** facilitates the design cycle (e.g. https://render.com/).
- **In-house server** needs corresponding skills to host.
  - Docker containers, GitHub, a dedicated server.
  - ~5hrs/mth of time with syst. mgr. skills.
- **Jupyter Notebooks** also work well if a hub is available.

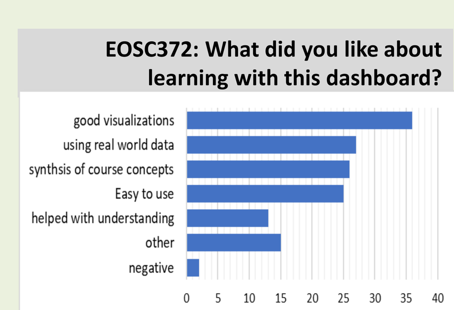
Cost to build
- *~ 20 hrs for e.g. **Mohr's circles** for structural geology.* **$**
- *~ 2 mths for e.g. **cmip6-dash** compares CMIP6 models and scenarios for different climate variables.* **$$**

See https://eoas-ubc.github.io/dashboards.html for details.

Instructors' costs **$**
- *3-4 short meetings during design & prototyping;*
- *Prepare / manage 1st live use (like any new learning activity);*
- *Gather feedback data (GeoSci Ed support can build & analyze)*
- *1-2 short meetings to followup and fine tune*

Learning

- **Follow PhET guidelines** for "*teaching with simulations*". Students start by exploring, then tackle meaningful tasks.
- **Groups** work better than solo learning.
- **Focus** on concepts & real data, not details or "toy" examples.
- **Keep apps versatile**; give assignment instructions separately.
- **Students are inspired** by "hands on" learning opportunities.
  - Analyzed feedback from 106 3rd year students: oceanography elective, EOSC 372. ➔
  - Similarly, in a 1st yr course: 75% respondents agree or strongly agree that they *"would like more use of dashboards …".*
  - From instructor: *"I am so impressed … I love how*
    - sliders constrain / adjust axes,
    - data at real stations are chosen on a map and compared,
    - graphic results can be saved to submit for assessment."



5. Lessons learned: Computing for EOAS students

Convert courses to Python, Jupyter NBs, and opensource practices.

Table 4. Course conversions from original code environment to Python using Jupyter Notebooks.

Course	Original language	text: rewrite	text: adopt OER	redo class materials	labs &/or assigs	auto-grading	hubs	local installs
DSCI 100 *	R	y		y	y	y	y	
EOSC 211 *	Matlab		y	y	y	y	y	y
EOSC 442 ^	MatLab				y		y	
EOSC 354 ^	MatLab				y			y
EOSC 410/510 ^	MatLab				y			y
EOSC 471 ^	MatLab				y		y	
ATSC 301 **			y		y	y	y	y

** Introductory courses - complex, labor intensive, time consuming, multifaceted.*
^ Senior courses: some programming assumed, conversion can be straightforward.
*** ATSC 301, already Python-based, served as precedent for several OCESE tasks.*

When course conversion is straightforward

*When students are **not beginners**, e.g. 3rd, 4th year courses.*

- **Convert** labs, lessons, etc. to Python & Jupyter Notebooks with no fundamental change to course content.
- **Check starting skills** & provide catch-up resources.
- **Opensource** resources: e.g. PROJECT PYTHIA ➔
- **Develop workflows** to assign, submit, grade & give feedback.
- **Jupyter hubs?** Only if code/datasets are huge or change often. A common goal at 3rd & 4th year level is for students to become self-sufficient. Therefore, most students use their own laptops.
- **Assessing** Jupyter notebooks is easy(*ish*) with < ~25 students

Cost for straightforward course transformations **$$**
- *Convert ~10 assignments / labs: ~2mths student programmer;*
- *Pilot first term: a "strong" TA, but otherwise little else changed.*
- *Minor adjustments to workflow and lessons after pilot.*

When course conversion is complex & costly

***First exposure to computing**; i.e. larger 1st or 2nd year courses.*

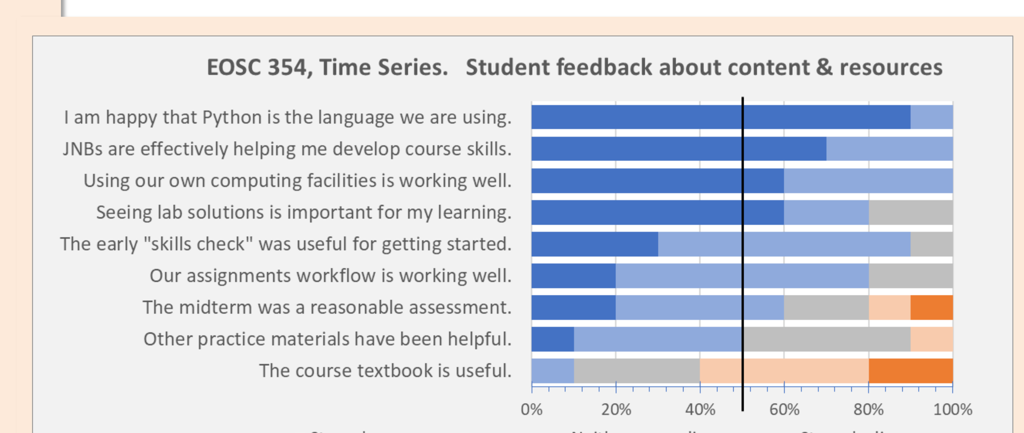
- **Critical support**: Geoscie. Ed. + excellent TAs.
- Jupyter hubs must be reliable, scalable & "well managed". Refer to open source community experience (eg. https://2i2c.org/)
- For students on laptops (~33%): install using conda lockfiles.
- Assessment management (*a new, emerging priority*):
  - **Auto-grading**: non-trivial but essential for 100+ students. E.g. *PrairieLearn, nbgrader, ottergrader, gradescope, LMS,* etc.
  - **Improve LMS efficiency**: Manage questions via its API.

Costs: larger intro. courses need complete rebuilds. E.g. …
First year stats course, "R" ➔ Python: **$$$**
- *9mths, 4 students, 3 profs. • Rewrite original opensource text.*
- *Adapt & test all lessons, learning activities & resources.*
- *Stay compatible with original "R" version of the course.*

Second year Earth science computing, MatLab ➔ Python: **$$$$**
- *12mths, 2 students, 2 profs. • Adopt an opensource text.*
- *Adapt all lessons & learning activities. • Pilot use of Jupyter hubs (twice).*
- *Re-work autograding workflow.*

General observations re. course transformation

- "**Pythonization**" was easier on students than instructors.
- **Opensource textbooks** are efficient & sustainable.
- **Students** want to learn Python; feedback surveys, e.g. ➔
- **2-3 years** to shift from MatLab to Python across curriculum.
- **TA & student-worker** support was critical!
- **Geosci. Ed. coordinator**: critical for efficiency & pedagogy.



6. Lessons learned: Dep't / Institution

Documentation & tutorials - https://eoas-ubc.github.io

Supporting instructors & TAs (pedagogy & logistics)

- **COVID** reduced Faculty's capacity to participate. Faculty support became 1-on-1 during COVID.
- **Geoscience ed'n specialist** can coordinate multi-course projects (i.e. large teams) and keep the emphasis on learning. Also can support development of new courses or learning activities.
- **Teaching assistants**' energy & talent was critical for development, implementation, and supporting instructors.
- **Workshops** are NOT agile enough, AND inappropriate before successful implementation.
- Building **opensource** sftwr & docs (GitHub, Jupyter Books, etc.) is a "*hard sell*" for those new to such practices.
- Paired teaching a key to success in 5 of 20 courses.
- Slack channel data highlights some challenges & concerns. E.g.:
  >"Should we use a new 'better' library or a simpler, older library to avoid cognitive overload?"
  >TAs discuss student difficulties prior to teaching a lab section.
  >Teaching team discusses scope-creep in a new lab exercise.

Infrastructure & servers / hubs

- **Challenges & effort** were greater than anticipated.
- System debugging during a "live" course was stressful for instructors, TAs and *not good* for students.
- **JNB issues (e.g.):** • Hubs vs laptops; • 'small' vs scalable; • containers & environments; • libraries & software.
- **Dashboard servers** need admin-level computing skills or staff.
- **Deployment** must be "invisible" to instructors.
- **Don't "reinvent the wheel"**; J-hubs and server techniques are known & opensource (e.g. https://2i2c.org/).

Open source and Open Education Resources (OERs)

- Critical for **software** development.
- Critical to work within the **opensource ecosystem**. *"Going it alone"* is not sustainable.
- **Yet** - challenging when critical components go un-supported.
- An important & useful **learning goal for students**.
- **Local Jupyter community** is growing; needs fostering! e.g.
- We will be delivering **project products as OERs.**

Parallel project: ***QuEST**, Quantitative Earth Sciences Transformation*
Rejuvenating our quantitative Earth science curriculum.
https://blogs.ubc.ca/eoasquest/

Contributors 2020 - 2023

20 Faculty: T. Ivanochko, P. Austin, F. Jones, C. Johnson, V. Radic, A. Ameli, S. Waterhouse, M. Maldonado, K. Orians, S. Sutherland, R. Beckie, S. Allen, A. Orsi, M. Bostock, L. Heagey, L. Porritt, K. Hodge, M. Lipsen, T. Timbers (Stats), T. Campbell (Stats)
11 Graduate students: R. Merrill, N. Dahiya, W. Ye, Y. Egorovo, Y. Kuzmenko, A. Loeppky, J. McFarlane, C. Rodell, F. Rossmann, Y. Su, H. Umashankar. || **6 Undergraduates:** J. Byer, B. Chang, M. Colclough, D. Platonov, C. Zhang, I. Sadeh,

Acknowledgements :

*Many thanks for financial support provided by UBC Vancouver students via the **Teaching and Learning Enhancement Fund**, and the UBC **Work Learn Program** for helping employ student contributors.*