OCESE: Open-source computing for Earth Sciences Education
Website: https://eoas-ubc.github.io

Focus of the breakout session:

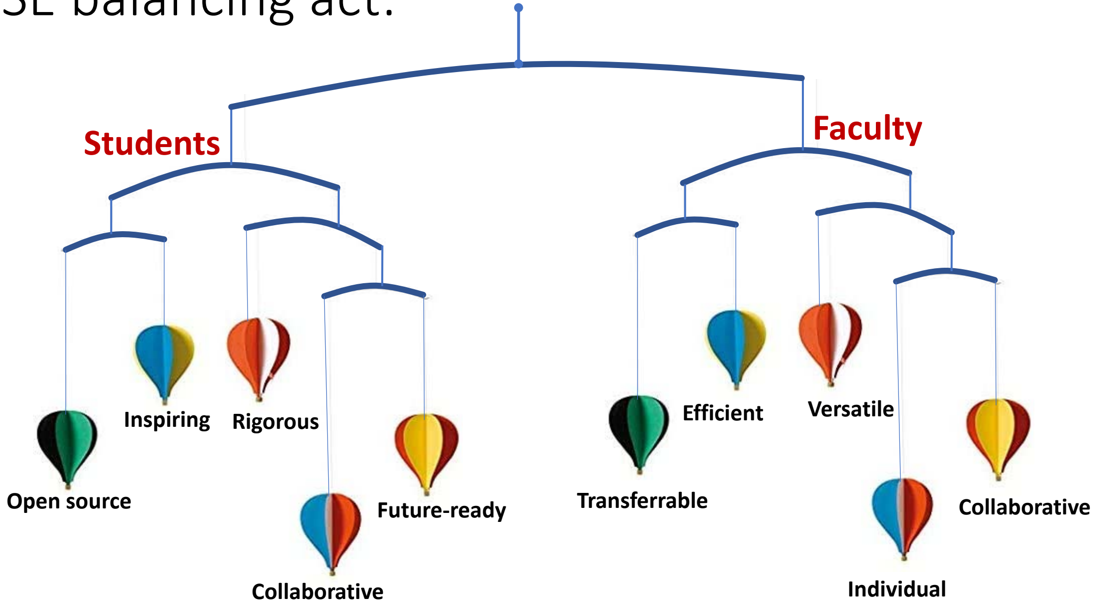
- Where you fit in the project (Francis presents specific case studies with concrete examples)
  - tag line: “low threshold, high ceiling”

Big picture: Decision making under uncertainty

- Why now? -- bridging EOAS, statistics and computer science
  - Wicked problems, exabytes of open access data, models with predictive skill
  - Open source collaboration, collective action and the climate emergency
  - EOAS faculty (new & existing) are keen -- Faculty Pro-D & support is a key project focus.

1

OCESE balancing act:



2

Faculty feedback from Nov 18:

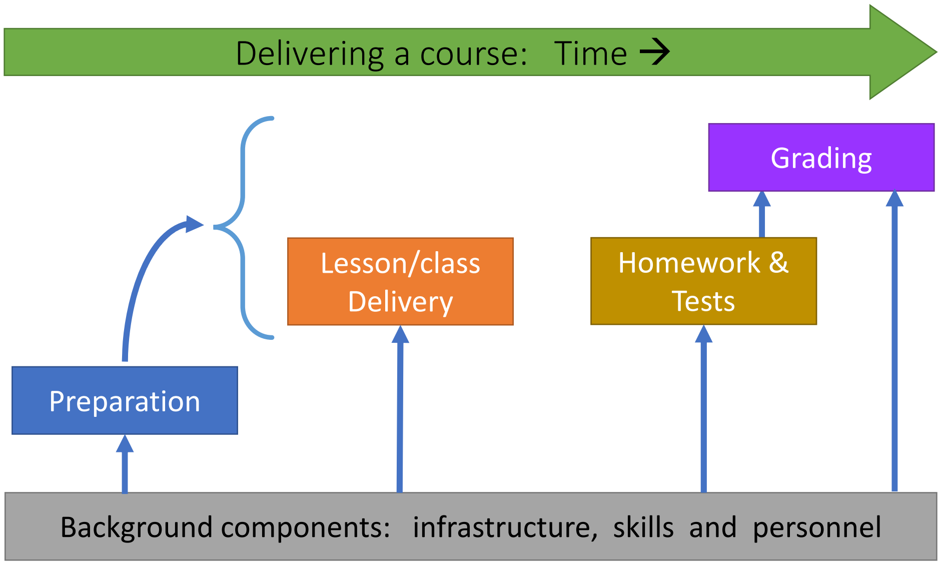
What's in it for me? **What benefits to students?**
What's it cost (in time, effort and maintenance)?

Participants asked questions like:

- **What does "open source" imply** for different courses, labs or activities?
- **Where do I start**? – "no spare time to innovate"
- What about my **teaching preferences**? (lecture, Socratic, active worksheets, clickers, etc.)
- Why Jupyterbooks & Jupyter notebooks?
- I like my current content formats – why change?
- Is there research on efficacy?
- Are there entry level guidelines?

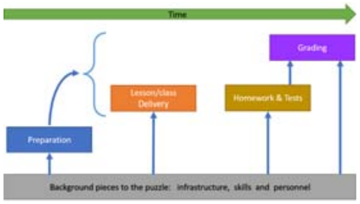
3

What course components will benefit?


Delivering a course: Time →
Grading
Lesson/class Delivery
Homework & Tests
Preparation
Background components: infrastructure, skills and personnel

4

OCESE project is working on …

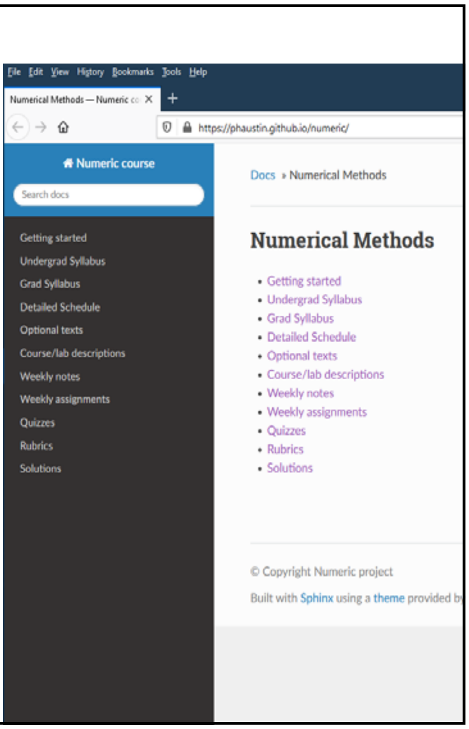

Time
Grading
Lesson/class Delivery
Homework & Tests
Preparation
Background pieces to the puzzle: infrastructure, skills and personnel

- Generating & maintaining custom *opensource* textbooks & documents
- Dashboards: *exploring & interacting with:* |data | concepts | models|
- Jupyter notebooks: mingling **text, math** and executable **code**.
- Assigning ‘authentic’ homework
- Managing test questions & automating grading
- Setting up EOAS, FoS and institutional infrastructure & personnel
- Collaborating with local and global colleagues

5

Textbook or docs. production

- Text-based (markdown)
- Rapid command-line construction with GitHub
- Easily convert LaTeX, MS-Word, PPTX, etc.
- Examples:
  - Project documentation: https://fhmjones.github.io/
  - Statistics used by STAT 201: https://moderndive.com/
  - Numerical methods: https://phaustin.github.io/numeric/
- **Easy to learn: ~half day max.**
- Open source (advisable!) ... or not.



Docs » Numerical Methods

Numerical Methods

- Getting started
- Undergrad Syllabus
- Grad Syllabus
- Detailed Schedule
- Optional texts
- Course/lab descriptions
- Weekly notes
- Weekly assignments
- Quizzes
- Rubrics
- Solutions

© Copyright Numeric project

Built with Sphinx using a theme provided by

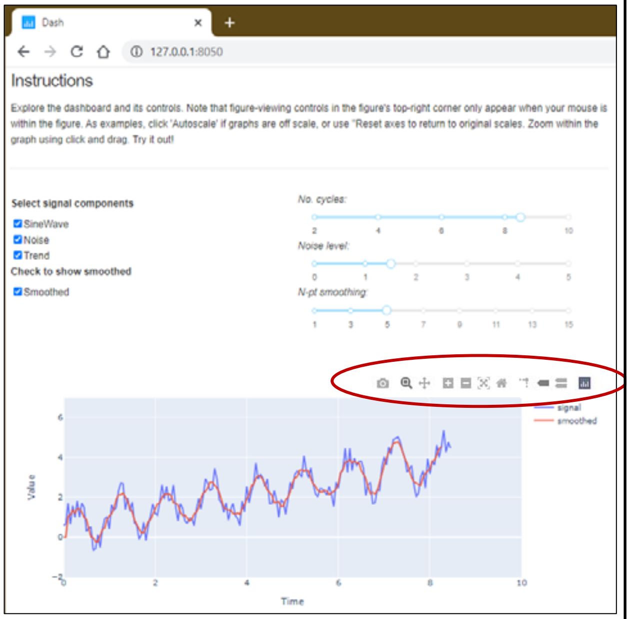
6

◆ Using Dashboards

Three types: explore *concepts*, *data*, *models*.

1) Interactively explore a concept

- Signal components:
    - Signal | Noise | Trend
- Implications for measurement and interpretation
- Impact / limits of smoothing
- Questions to guide thinking
- "Easy" to wrap data or algorithms with interactive components.



Instructions

Explore the dashboard and its controls. Note that figure-viewing controls in the figure's top-right corner only appear when your mouse is within the figure. As examples, click 'Autoscale' if graphs are off scale, or use "Reset axes to return to original scales. Zoom within the graph using click and drag. Try it out!

Select signal components

☑ SineWave
☑ Noise
☑ Trend

Check to show smoothed

☑ Smoothed

No. cycles:

Noise level:

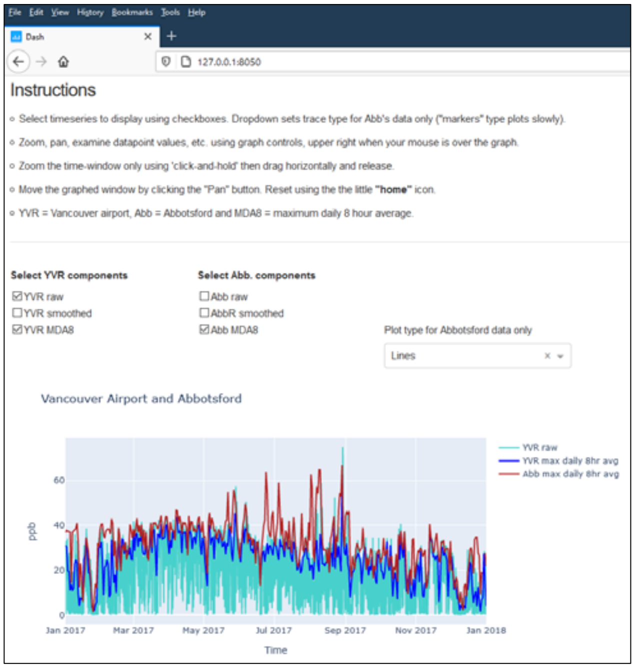
N-pt smoothing:

https://eoas05.eoas.ubc.ca/services/external/eoas-341/three-signals/

Demonstrate later if desired.

2) Explore datasets

- One year of hourly Ozone at 2 stns
- Compare two data sets
- Questions to guide thinking



Instructions

- Select timeseries to display using checkboxes. Dropdown sets trace type for Abb's data only ("markers" type plots slowly).
- Zoom, pan, examine datapoint values, etc. using graph controls, upper right when your mouse is over the graph.
- Zoom the time-window only using 'click-and-hold' then drag horizontally and release.
- Move the graphed window by clicking the "Pan" button. Reset using the the little **"home"** icon.
- YVR = Vancouver airport, Abb = Abbotsford and MDA8 = maximum daily 8 hour average.

Select YVR components

☑ YVR raw
☐ YVR smoothed
☑ YVR MDA8

Select Abb. components

☐ Abb raw
☐ AbbR smoothed
☑ Abb MDA8

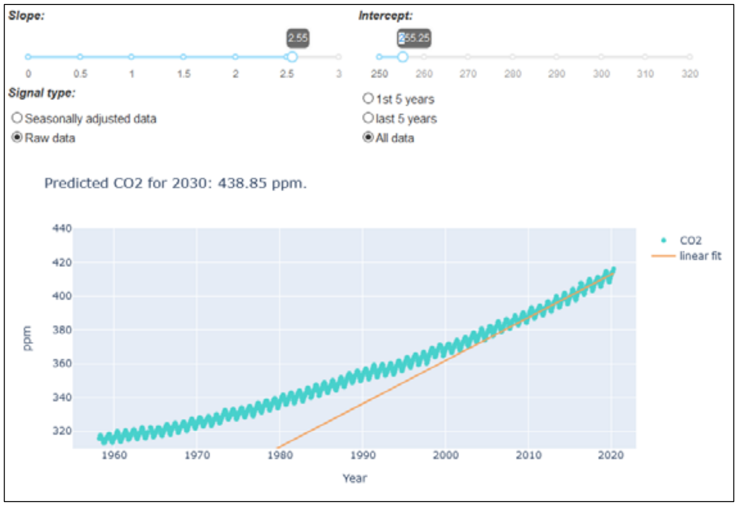
Plot type for Abbotsford data only: Lines

Demonstrate later if desired.

3) Explore modelling

- CO2 at MonaLoa
  - Linear fit to segments.
  - Compare predictions

- Guiding questions via …
  - On screen
  - In worksheets
  - Via Canvas questions
  - For built-in delivery (coming soon …)



Slope: 2.55

Intercept: 255.25

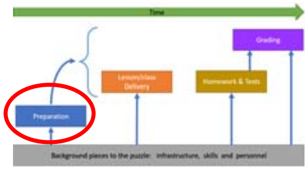
Signal type:

○ Seasonally adjusted data
● Raw data

○ 1st 5 years
○ last 5 years
● All data

Predicted CO2 for 2030: 438.85 ppm.

◆ Dashboards summary



Time

Grading

Lesson/class Delivery

Homework & Tests

Preparation

Background pieces to the puzzle: infrastructure, skills and personnel

- “Low threshold” ways to engage & explore data, concepts, models
  - **Instructors**: Easily add to lessons or assignments.
  - **Students**: Easy to use – “near zero” learning curve.
- Benefits:
  - Active, inquiry-based learning.
  - Guided discussions using on-the-fly trial & error and cause & effect.
- Evidence:
  - Instructors appreciate engaging in lively, demonstration-oriented discussions.
  - See also https://phet.colorado.edu/en/research

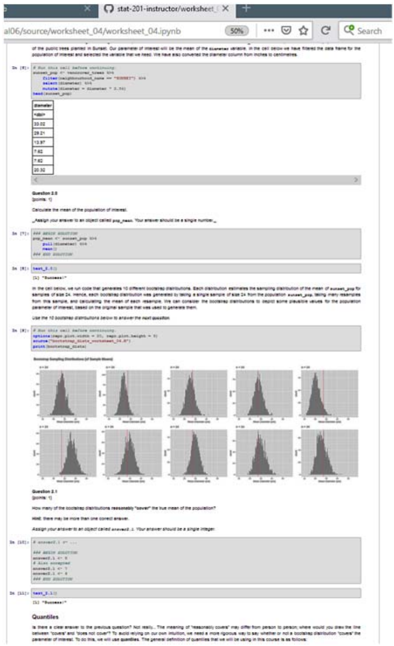
◆ Jupyter notebooks

- Why go there? https://jupyter.org/
  - ***Literate coding*** = text/math/figs mingled with executable code snippets.
  - Open, reproducible, articulate **science**
  - Open, consistent, evolving **education**
- What does teaching / learning with them look like?

11

Jupyter Notebook: STAT 201 tutorial

- Worksheet as a Jupyter notebook online here.
  - Cells with **instructions** (text, math, tables, diagrams …)
  - Cells with **code** and/or **graphics**
  - In-line, in-**context questions** to guide students' thinking.
  - Question **solutions** in the *instructor's* version.
- Students (or instructors) work through the Jupyter notebook one cell at a time.
  - Alone – in groups – labs – tutorials – in class – etc.
- Open-source textbook: https://moderndive.com/


stat-201-instructor/worksheet_
al06/source/worksheet_04/worksheet_04.ipynb
50%
Search

of the public trees planted in Sunset. Our parameter of interest will be the mean of the diameter variable. In the cell below we have filtered the data frame for the population of interest and selected the variable that we need. We have also converted the diameter column from inches to centimetres.

```
In [5]: # Run this cell before continuing.
        sunset_pop <- vancouver_trees %>%
            filter(neighbourhood_name == "SUNSET") %>%
            select(diameter) %>%
            mutate(diameter = diameter * 2.54)
        head(sunset_pop)
```

diameter
<dbl>
33.02
29.21
13.97
7.62
7.62
20.32

Question 2.0
{points: 1}

Calculate the mean of the population of interest.

_Assign your answer to an object called pop_mean. Your answer should be a single number._

```
In [7]: ### BEGIN SOLUTION
        pop_mean <- sunset_pop %>%
            pull(diameter) %>%
            mean()
        ### END SOLUTION

In [8]: test_2.0()

        [1] "Success!"
```

In the cell below, we run code that generates 10 different bootstrap distributions. Each distribution estimates the sampling distribution of the mean of sunset_pop for samples of size 24. Hence, each bootstrap distribution was generated by taking a single sample of size 24 from the population sunset_pop, taking many resamples from this sample, and calculating the mean of each resample. We can consider the bootstrap distributions to depict some plausible values for the population parameter of interest, based on the original sample that was used to generate them.

Use the 10 bootstrap distributions below to answer the next question.

```
In [9]: # Run this cell before continuing.
        options(repr.plot.width = 20, repr.plot.height = 8)
        source("bootstrap_dists_worksheet_04.R")
        print(bootstrap_dists)
```

Bootstrap Sampling Distributions (of Sample Means)

Question 2.1
{points: 1}

How many of the bootstrap distributions reasonably "cover" the true mean of the population?

Hint: there may be more than one correct answer.

Assign your answer to an object called answer2.1. Your answer should be a single integer.

```
In [10]: # answer2.1 <- ...

         ### BEGIN SOLUTION
         answer2.1 <- 6
         # Also accepted
         answer2.1 <- 7
         answer2.1 <- 8
         ### END SOLUTION

In [11]: test_2.1()

         [1] "Success!"
```

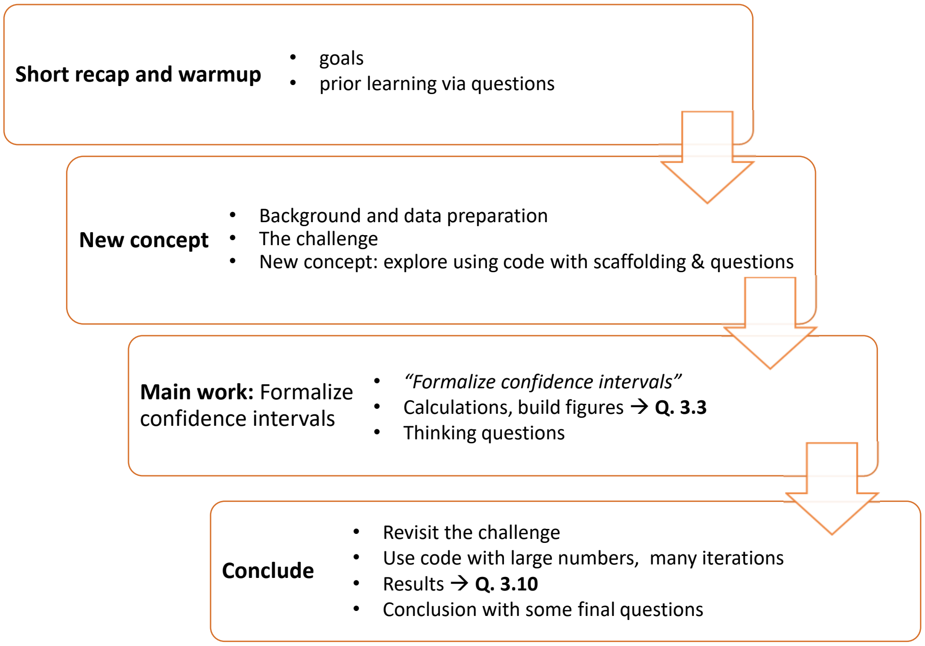
Quantiles

Is there a clear answer to the previous question? Not really... The meaning of "reasonably covers" may differ from person to person; where would you draw the line between "covers" and "does not cover"? To avoid relying on our own intuition, we need a more rigorous way to say whether or not a bootstrap distribution "covers" the parameter of interest. To do this, we will use quantiles. The general definition of quantiles that we will be using in this course is as follows:

⋮

12

Workflow for the STAT 201 tutorial:



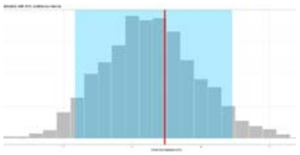
Short recap and warmup
- goals
- prior learning via questions

New concept
- Background and data preparation
- The challenge
- New concept: explore using code with scaffolding & questions

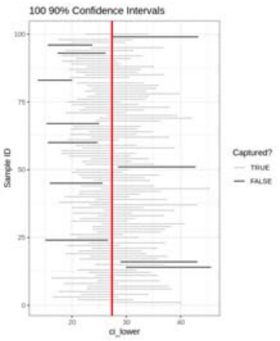
Main work: Formalize confidence intervals
- *"Formalize confidence intervals"*
- Calculations, build figures → **Q. 3.3**
- Thinking questions

Conclude
- Revisit the challenge
- Use code with large numbers, many iterations
- Results → **Q. 3.10**
- Conclusion with some final questions

<u>Example plots students make</u>

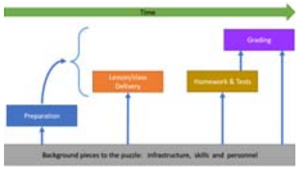


Q. 3.3: Plot confidence interval and true mean of the population over the bootstrap distribution.



Q. 3.10: Visualize 100 confidence intervals

Jupyter notebooks summary


Time
Grading
Lesson/class Delivery
Homework & Tests
Preparation
Background pieces to the puzzle: infrastructure, skills and personnel

Currently used in: STAT 201, ATSC 301, EOSC 350, EOSC 213, & others.

- Complete lessons or tutorials deployed as **Jupyter Notebooks**.
- Learning sequence includes ***guidance*** and active ***practicing***:
  - ***Guiding*** cells include math, figures, tables, diagrams, plots
  - ***Executable*** cells: > Write code snippets & run. > Supply answers & submit.
- **Scaffolding** helps students focus on concepts not code syntax.
- Embedded **low-stakes questions** to guide thinking (like clickers).
- **Auto grading** of questions.
- Careful choice of working **data**: real, relevant, both "*ideal*" & "*real*".

Benefits to students

Low threshold

- Rapid, reliable engagement with computer-based active learning in class.
- Opportunities to explore concepts & datasets without overhead of finding data & processing it.
- Engagement with real and meaningful data sets, concepts and problems.
- Reduced barriers due to diverse computing platforms.
- More active exposure to challenging concepts rather than abstract, “static” presentation of learning content.

Higher ceiling

- Exposure to, and practice with, **notebook-based thinking** about quantitative problems.
- Practice applying fundamentals (math, programming) in higher level contexts
- Authentic practice using emerging collaborative procedures
- Earlier exposure to higher level analysis or interpretation tasks
- Easily modified tests & assignments means reliable, up-to-date, authentic practice & assessment.
- Eg: students may be able to participate in https://cic.ubc.ca/ with sufficient background.

15

Learning curve for instructors

“Low threshold” options for engaging with OCESE goals and initiatives:

- Familiarity with basic GitHub functions for storing and collaborating
- Convert local formats (LaTex, MSWord, Powerpoint, etc.) -to- Markdown text.
- A dashboard app for a class or assignment.
  - Plans for: ENVR 300, EOSC 329, EOSC 372, 373, and others.

16

Learning curve for instructors

"Low threshold" options for engaging with OCESE goals and initiatives:

- Familiarity with basic GitHub functions for storing and collaborating
- Local software → Markdown conversion
- A dashboard app for a class or assignment.

Move towards more sophisticated opportunities via one or more of …

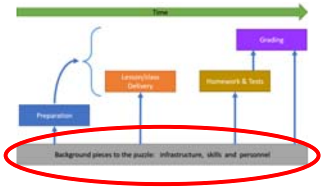
- *Jupyter notebooks* for coding, demonstrating, practicing.
- Making *Jupyter Books* for textbooks, or readings or lectures.
- *Python* as a "future ready" framework for developing coding skills.
- "*Containers*" for packaging dashboards or software for universal deployment.
- Python / Plotly / Dash for making or maintaining *interactive dashboards*.

17

OCESE infrastructure development:

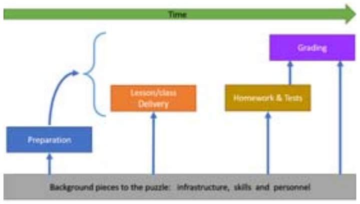
OCESE is collaborating with UBC and global colleagues to enable …

1) Local and/or cloud-based servers; "Jupyter hubs"
2) Deployments that guarantee success on all platforms; "containers".
3) Effective use of GitHub repositories *(open or private)*
4) Software & practices for the long term:
   >agility, >maintainability, >"shareability"
5) Professional development and training
6) Systems personnel


Time
Grading
Lesson/class Delivery
Homework & Tests
Preparation
Background pieces to the puzzle: infrastructure, skills and personnel

18

Recap: OCESE Project components


Time
Preparation
Lesson/class Delivery
Homework & Tests
Grading
Background pieces to the puzzle: infrastructure, skills and personnel

- ◆ ✓ Generating & maintaining custom *opensource* textbooks & documents
- ◆ ✓ Dashboards: *exploring* & *interacting with:* |data | concepts | models|
- ◆ ✓ Jupyter notebooks: mingling **text, math** and executable **code**.
- ◆ ✓ Assigning 'authentic' homework
- ◆ Managing test questions & automating grading
- ◆ Establishing EOAS, FoS and institutional infrastructure & personnel
- ◆ Collaborating with local and global colleagues

19

Questions to drive today's discussion

- Any questions? Are any project initiatives unclear?
- Which of these project components are you most excited about?
- Which do you think will improve your life as a teacher?
- Which will help students gain better, more transferrable skills?
- Which will make learning more inspiring?
- Is there anything that seems to be missing?
- What topics or skills will be most important to you for training sessions?

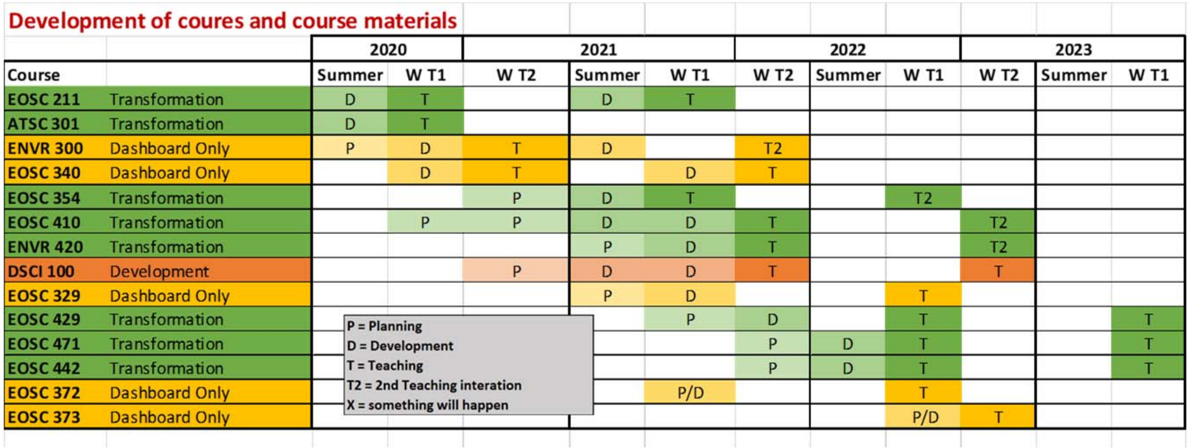
20

ADDITIONAL SLIDES

Included if needed for discussion

21

Time line – and targeted courses



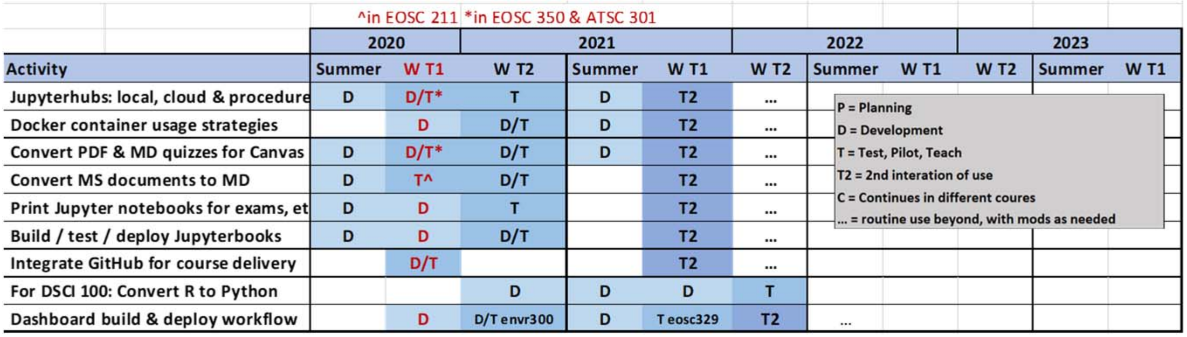
Development of coures and course materials

		2020		2021			2022			2023		
Course		Summer	W T1	W T2	Summer	W T1	W T2	Summer	W T1	W T2	Summer	W T1
EOSC 211	Transformation	D	T		D	T						
ATSC 301	Transformation	D	T									
ENVR 300	Dashboard Only	P	D	T	D		T2					
EOSC 340	Dashboard Only		D	T		D	T					
EOSC 354	Transformation			P	D	T			T2			
EOSC 410	Transformation		P	P	D	D	T			T2		
ENVR 420	Transformation				P	D	T			T2		
DSCI 100	Development			P	D	D	T			T		
EOSC 329	Dashboard Only				P	D			T			
EOSC 429	Transformation					P	D		T			T
EOSC 471	Transformation						P	D	T			T
EOSC 442	Transformation						P	D	T			T
EOSC 372	Dashboard Only					P/D			T			
EOSC 373	Dashboard Only								P/D	T		

P = Planning
D = Development
T = Teaching
T2 = 2nd Teaching interation
X = something will happen

22

Timeline for infrastructure development



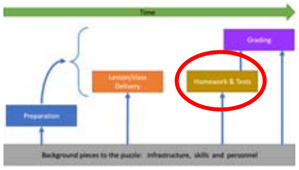
^in EOSC 211 *in EOSC 350 & ATSC 301

	2020		2021			2022			2023		
Activity	**Summer**	**W T1**	**W T2**	**Summer**	**W T1**	**W T2**	**Summer**	**W T1**	**W T2**	**Summer**	**W T1**
Jupyterhubs: local, cloud & procedure	D	D/T*	T	D	T2	...					
Docker container usage strategies		D	D/T	D	T2	...					
Convert PDF & MD quizzes for Canvas	D	D/T*	D/T	D	T2	...					
Convert MS documents to MD	D	T^	D/T		T2	...					
Print Jupyter notebooks for exams, et	D	D	T		T2	...					
Build / test / deploy Jupyterbooks	D	D	D/T		T2	...					
Integrate GitHub for course delivery		D/T			T2	...					
For DSCI 100: Convert R to Python			D	D	D	T					
Dashboard build & deploy workflow		D	D/T envr300	D	T eosc329	T2	...				

P = Planning
D = Development
T = Test, Pilot, Teach
T2 = 2nd interation of use
C = Continues in different coures
... = routine use beyond, with mods as needed

23

◆ Managing test questions

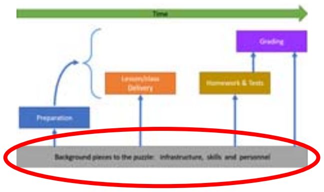

Time
Grading
Lessons/class Delivery
Homework & Tests
Preparation
Background pieces to the puzzle: infrastructure, skills and personnel

- Why:
  - Manage your question banks **outside** of Canvas.
  - Build **unique test questions** for each student.
  - **Autograding** in Canvas **AND** other tools.
- Currently piloting several techniques
  - EOSC 350 (clicker questions)
  - EOSC 211 (quizzes)
- Autograding of code or math work:
  - Not “new”, but consistency across courses is desirable.
  - **Stats** (DataScience) has adopted *nbgrader*. **Comp. Sci.** is piloting other techniques.
  - EOAS will likely pilot *nbgrader*.

24

Infrastructure currently being developed

1) Local and/or cloud-based servers – Jupyter hubs
   a) Secure access
   b) Large vs small classes
   c) Heavy or light computing loads
   d) Storage for students, or not
   e) Containers (Docker) for ease of delivery and maintenance of running codes.
   f) Transparent or visible back-end (eg, do students need GitHub accounts?)
2) GitHub repositories
   a) The OCESE standard for development & project management.
   b) Instructors – *eases collaboration with students, UBC colleagues and beyond*
   c) Students – *learn transferrable skills and encounter the opensource community*
3) Commitment to software and practices that will ensure long term agility and reliability
   a) Grading – nbgrader? (currently used by DSCI 100)
   b) Jupyter notebooks
   c) Jupyter books with Jupytext
   d) Communication between repositories and Canvas
   e) Python, and chosen key libraries
4) Professional development and training
   a) Docs, tutorials, workshops for tools and workflows (TAs & instructors, plus tutorials for students)
   b) Instructors prepared to adopt consistent and transferable skills and knowledge.
5) Systems personnel
   a) Individuals, Dep't and/or institution.


Time
Grading
Lesson/class Delivery
Homework & Tests
Preparation
Background pieces to the puzzle: infrastructure, skills and personnel