

Scalable and Computationally Reproducible Approaches to Arctic Research

Matt Jones, Bryce Mecum, Jeanette Clark, Sam Csik

September 19, 2022

Table of contents

Preface	3
About	3
Schedule	3
Code of Conduct	3
Setting Up	5
Download VS Code and Python, if needed	5
Set up VS Code	6
Test your local setup	6
Set up Jupyter Notebooks Extension	7
About this book	7
1 Welcome and Introductions	8
2 Remote Computing	11
2.1 Learning Objectives	11
3 Python Syntax Refresher	12
3.1 Learning Objectives	12
4 Pleasingly Parallel Computing	13
4.1 Learning Objectives	13
5 Documenting and Publishing Data	14
5.1 Learning Objectives	14
References	15

Preface

About

This 5-day in-person workshop will provide researchers with an introduction to advanced topics in computationally reproducible research in python and R, including software and techniques for working with very large datasets. This includes working in cloud computing environments, docker containers, and parallel processing using tools like parsl and dask. The workshop will also cover concrete methods for documenting and uploading data to the Arctic Data Center, advanced approaches to tracking data provenance, responsible research and data management practices including data sovereignty and the CARE principles, and ethical concerns with data-intensive modeling and analysis.



Schedule

Code of Conduct

Please note that by participating in this activity you agree to abide by the [NCEAS Code of Conduct](#).

	Monday	Tuesday	Wednesday	Thursday	Friday	
08:00-08:30	Coffee (optional)	Coffee (optional)	Coffee (optional)	Coffee (optional)	Coffee (optional)	
08:30-09:00	1. Welcome and Course Overview (Jeanette)	6. Data structures and formats for large data (Bryce)	10. Spatial and Image Data using GeoPandas (Jeanette)	15. Google Earth Engine (Ingmar, Sam)	19. What is cloud computing anyways? (Matt)	
09:00-09:30			11. Data futures: Parquet and Arrow (Jeanette)			
09:30-10:00	2. Remote computing (Sam)					
10:00-10:30						
10:30-11:00	BREAK	BREAK	BREAK	BREAK	BREAK	
11:00-11:30	3. Python programming on clusters (Jeanette)	7. Parallelization with Dask (Bryce)	12. Software Design II (Bryce)	16. Billions of Ice Wedge Polygons (Chandi)	20. Reproducibility redux via containers (Bryce)	
11:30-12:00					Survey Feedback Q & A	
12:00-12:30	Lunch	Lunch	Lunch	Lunch	Adjourn	
12:30-13:00						
13:00-13:30	4. Pleasingly Parallel Programming (Matt)	8. Group project I Data staging and pre-processing (Jeanette)	13. Group project II Parallel data processing (Jeanette)	17. Group project III Visualizing big geospatial data (Jeanette)		
13:30-14:00						
14:00-14:30						
14:30-15:00						
15:00-15:30	Break	Break	Break	Break		
15:30-16:00	5. Documenting and Publishing Data (Daphne)	9. Software design I (Bryce)	14. Data Ethics (Matt)	18. Workflows for data staging and publishing (Jeanette)		
16:00-16:30			Breather Catch-up			
16:30-17:00	Q&A	Q&A	Q&A	Q&A		

Setting Up

In this course, we will be using Python (> 3.0) as our primary language, and VS Code as our IDE. Below are instructions on how to get VS Code set up to work for the course. If you are already a regular Python user, you may already have another IDE set up. We strongly encourage you to set up VS Code with us, because we will use your local VS Code instance to write and execute code on one of the NCEAS servers.

Download VS Code and Python, if needed

First, [download VS Code](#) if you do not already have it installed.

Next, if you aren't sure, check to make sure you have Python installed. To do this, from the terminal run:

```
python3 --version
```

If you get an error, it means you need to install Python. Here are instructions for getting installed, depending on your operating system.

- Windows: Download and run an installer from [Python.org](#).
- Mac: Install using [homebrew](#). If you don't have homebrew installed, follow the instructions from their webpage.

```
— brew install python3
```

After you run your install, make sure you check that the install is on your system PATH by running `python3 --version` again.

Set up VS Code

This section summarizes the official VS Code tutorial. For more detailed instructions and screenshots, see the [source material](#)

First, install the [Python extension for VS Code](#).

Open a terminal window in VS Code from the Terminal drop down in the main window. Run the following commands to initialize a project workspace in a directory called **training**. This example will show you how to do this locally. Later, we will show you how to set it up on the remote server with only one additional step.

```
mkdir training
cd training
code .
```

Next, we will select the Python interpreter for the project. Open the **Command Palette** using Command + Shift + P (Control + Shift + P for windows). The Command Palette is a handy tool in VS Code that allows you to quickly find commands to VS Code, like editor commands, file edit and open commands, settings, etc. In the Command Palette, type “Python: Select Interpreter.” Push return to select the command, and then select the interpreter you want to use (your Python 3.X installation).

Test your local setup

To make sure you can write and execute code in your project, [create a Hello World test file](#).

- From the File Explorer toolbar, or using the terminal, create a file called **hello.py**
- Add some test code to the file, and save

```
msg = "Hello World"
print(msg)
```

- Execute the script using either the Play button in the upper-right hand side of your window, or by running `python3 hello.py` in the terminal.
 - For more ways to run code in VS Code, see the [tutorial](#)

Set up Jupyter Notebooks Extension

Finally, download the [Jupyter extension](#). You can create a test Jupyter Notebook document from the command palette by typing “Create: New Jupyter Notebook” and selecting the command. This will open up a code editor pane with a notebook that you can test.

About this book

These written materials reflect the continuous development of learning materials at the Arctic Data Center and NCEAS to support individuals to understand, adopt, and apply ethical open science practices. In bringing these materials together we recognize that many individuals have contributed to their development. The primary authors are listed alphabetically in the citation below, with additional contributors recognized for their role in developing previous iterations of these or similar materials.

This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

Citation: Matthew B. Jones, Bryce Mecum, S. Jeanette Clark, Samantha Csik. 2022. Scalable and Computationally Reproducible Approaches to Arctic Research.

Additional contributors: Amber E. Budden, Natasha Haycock-Chavez, Noor Johnson, Stephanie Hampton, Jim Regetz, Bryce Mecum, Julien Brun, Julie Lowndes, Erin McLean, Andrew Barrett, David LeBauer, Jessica Guo.

This is a Quarto book. To learn more about Quarto books visit <https://quarto.org/docs/books>.

1 Welcome and Introductions



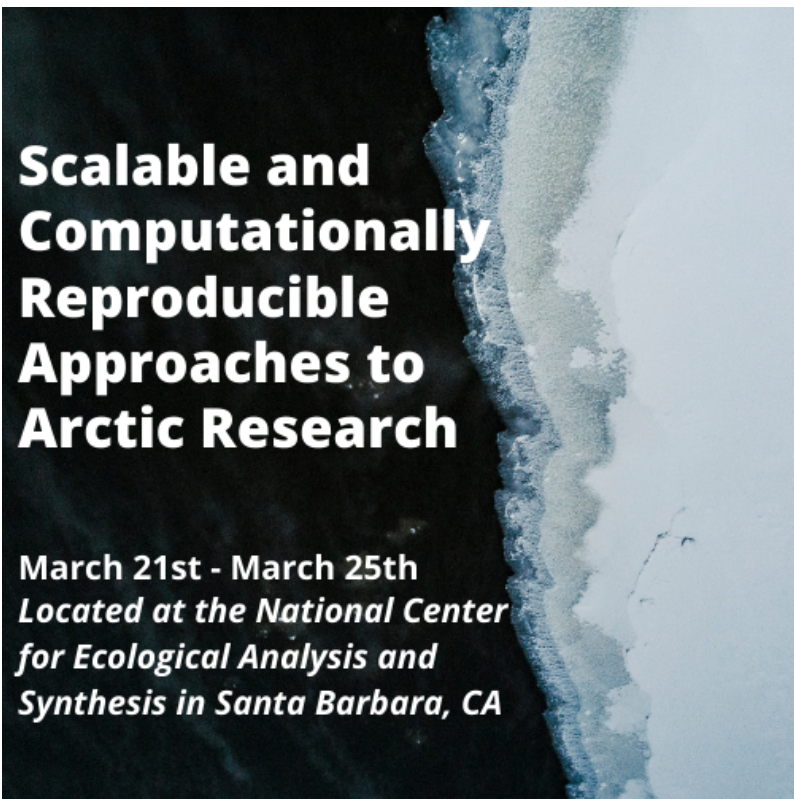
This course is one of three that we are currently offering, covering fundamentals of open data sharing, reproducible research, ethical data use and reuse, and scalable computing for reusing large data sets.





Reproducible Practices for Arctic Research Using R

**February 14th - February
18th, 2022**
*This course will be taught
virtually*



Scalable and Computationally Reproducible Approaches to Arctic Research

March 21st - March 25th
*Located at the National Center
for Ecological Analysis and
Synthesis in Santa Barbara, CA*

2 Remote Computing

2.1 Learning Objectives

3 Python Syntax Refresher

3.1 Learning Objectives

4 Pleasingly Parallel Computing

4.1 Learning Objectives

5 Documenting and Publishing Data

5.1 Learning Objectives

References