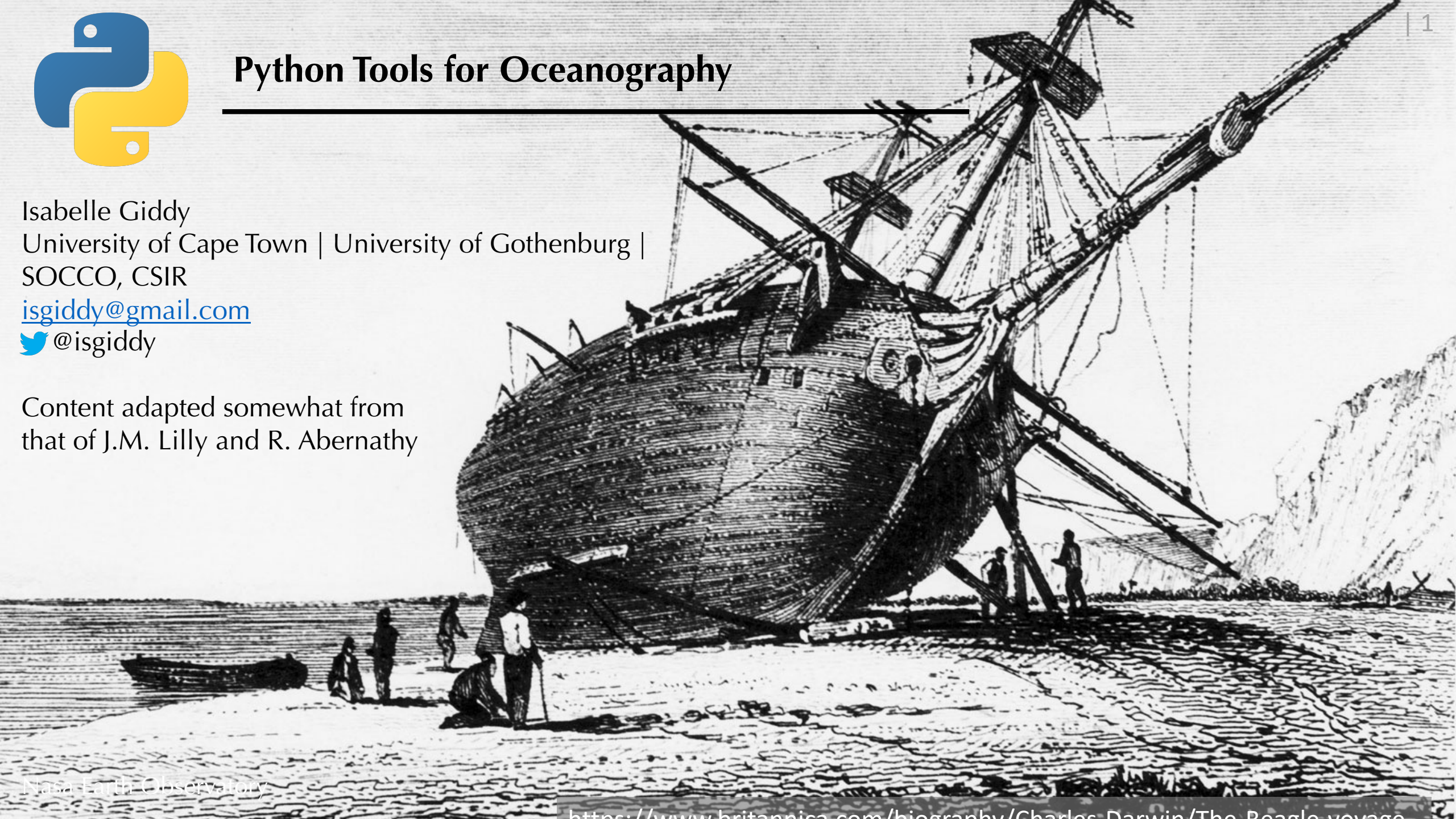
Python Tools for Oceanography

Isabelle Giddy
University of Cape Town | University of Gothenburg |
SOCCO, CSIR
isgiddy@gmail.com
@isgiddy

Content adapted somewhat from
that of J.M. Lilly and R. Abernathy



Nasa Earth Observatory

https://www.britannica.com/biography/Charles-Darwin/The-Beagle-voyage

Outline

16 August - Introduction, motivation, workflow/ using the terminal and git

23 August – Python fundamentals

30 August – Numpy

6 September – Matplotlib

13 September - Pandas/ working with tabular data/ low level statistics

20 September – Xarray for multidimensional data

27 September - Making maps/ cartopy

4 October – Summary / maybe brief intro to interactive plotting

Motivation

Data!

Observations

Science

Reproducibility

Trust

Impact

Motivation Data – Observations - Science

In oceanography we learn a lot of theory and methods, we spend time deploying instruments, calibrating them, cleaning the data

We say we are observing the ocean – but most of the observations are made by sensors beneath the ocean/ in the lab which we do not see with our eyes

It can be difficult to gain intuition for this hidden world

One possibility is to train yourself to ***observe the data***

To facilitate this, having a good hold of a coding language is key

Motivation Reproducibility - Trust - Impact

FAIR principles

“findable, accessible, interoperable, and reusable.”

Workflow

Set up your project

Workflow

Set up your project

Use *conda* environments for projects

ANACONDA®

Workflow

Set up your project

Use *conda* environments for projects

ANACONDA®

Code interactively in jupyter lab

jupyter

Workflow

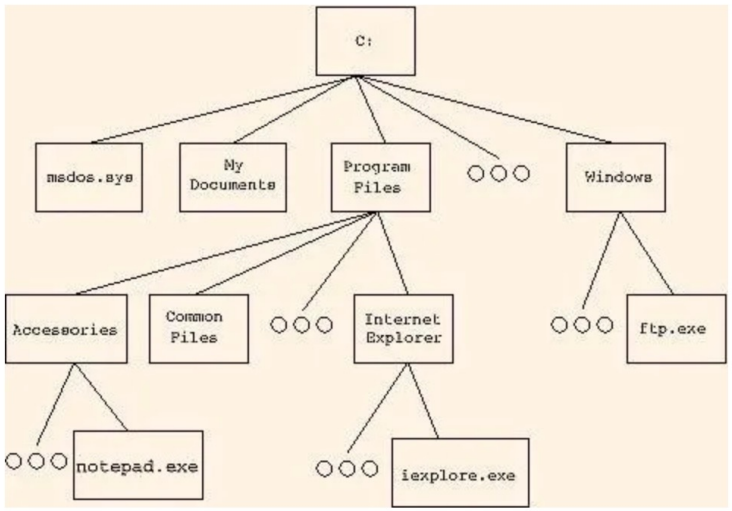
Set up your project/ understand a computers file system

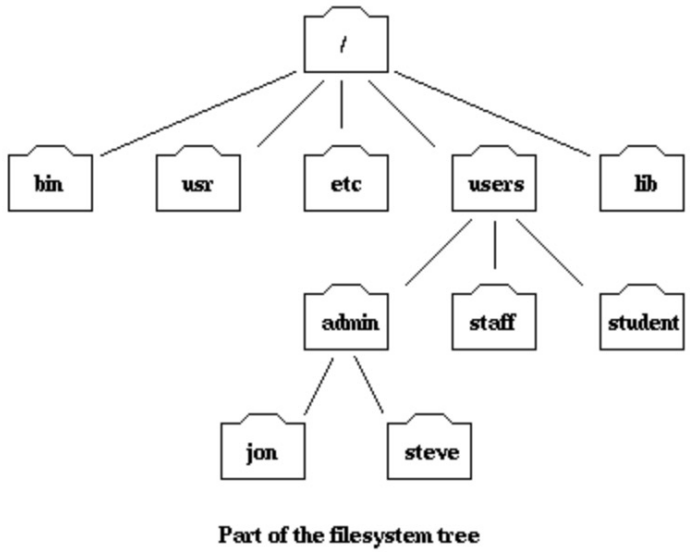
Use *conda* environments for projects

Code interactively in jupyter lab

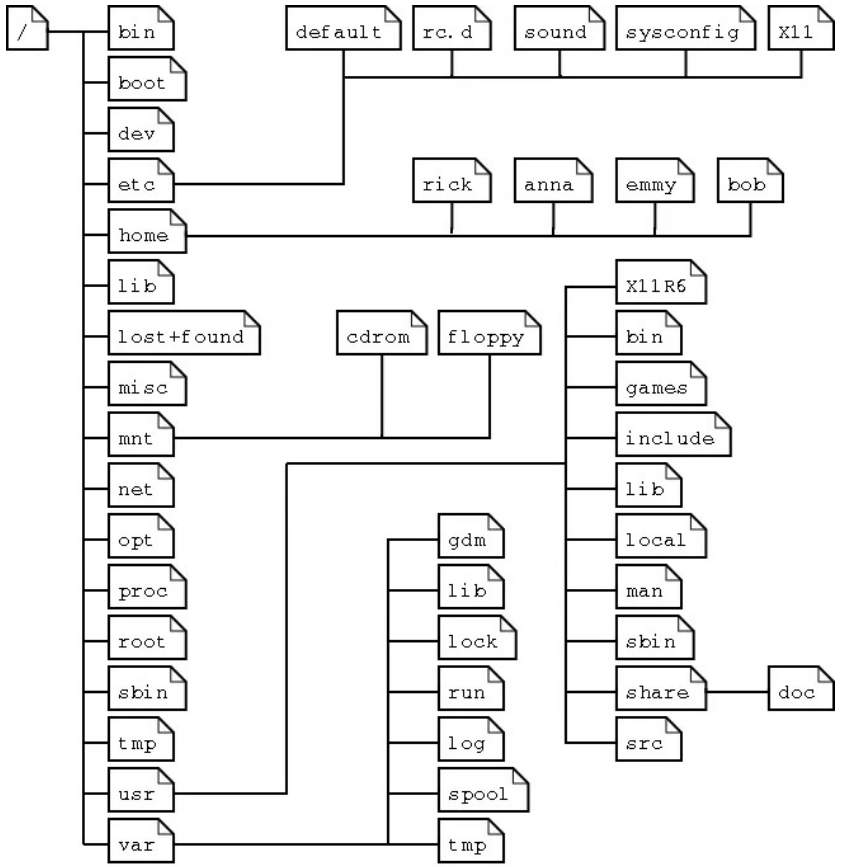
Use version control, collaborate and share with github

Workflow





Part of the filesystem tree



Windows

UNIX

LINUX

Workflow

- The file system is responsible for managing information on the disk
- Information is stored in files, which are stored in directories (folders)
- `cd path` changes the current working directory
- `ls path` prints a listing of a specific file or directory ; `ls` on its own lists the current working directory
- `pwd` prints the user's current working directory
- `whoami` shows the user's current directory
- / on its own is the root directory of the whole file system
- A relative path specifies a location starting from the current location
- An absolute path specifies a location from the root of the file system
- Directory names in a path are separated with '/' on Unix, but '\\' on Windows
- '..' means " the directory above this one "
- "." means " the current directory"
- Most files' names are something.extension. The extension isn't requires, and don't guarantee anything, but is normally used to indicate the type of data in the file
- Most commages take comtions (flags) which begin with a '-'

Workflow

Useful commands

- `cd path` change directory
- `ls path` / `ls` list files
- `pwd` prints the user's current working directory
- `whoami` shows the user's current directory
- `mkdir` create a folder
- `rm / rm -r` remove file/ folder
- `mv old new` moves (renames) file/folder
- `cp old new` copies a file
- `touch <filename> / nano <filename> / notepad <filename>` creates a file

Workflow

Set up a project

Example folder organisation:

/PythonTools

- Readme.md *general introduction to your project, how to use it, dependencies, acknowledgements,*
- LogBook.md *keep a daily logbook of what you do*
- /data *store your data here*
- /notebooks *all the notebooks you work on will go here*
- /src *any source code you use/develop goes here (more later)*
- /results *all your figures are saved here*
- /manuscript *accompanying manuscript if applicable*

Try: Use the command line to create yourself a project folder with a similar structure as above

Workflow

ANACONDA

Step 1: Download Anaconda https://www.anaconda.com/products/individual#Downloads

Step 2: Set up an environment in the terminal or in Anaconda Prompt

```
conda create -n myenv python=3.6
```

`conda env create -f environment.yml` for this course we will use a pre-constructed environment

Step 3: Activate new environment

```
conda activate myenv
source activate myenv
```

```
conda info --envs lists environments
conda list lists packages
conda remove --name myenv --all remove environment
conda env export --from-history export environment
```

Step 4: Install more packages

```
conda install <packagename>
conda install -c conda-forge <packagename>
pip install <packagename>
```

More: https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#

Workflow

Step 1: Install Jupyter in Anaconda / check already there

```
conda install -c conda-forge jupyterlab
```

Step 2: Navigate to your project folder

Step 3: Launch in the terminal/ Anaconda Prompt/ Command Prompt

```
jupyter-lab
```

Read the JupyterLab documentation
https://jupyterlab.readthedocs.io/en/stable/user/interface.html

https://zenodo.org/record/4910038#.YQpTTO2xVQI

Why Version Control

Version control is a powerful way to organize, back up, and share with collaborators your research computing code. A Version control system keeps track of a set of files and saves snapshots (i.e. *versions*, *commits*) of the files at any point in time.

Using version control allows you to confidently make changes to your code (any any other files), with the ability to roll back to any previous state.

Version control also allows you to share code with collaborators, make simultaneous edits, and merge your changes in a systematic, controlled way.

More: http://swcarpentry.github.io/git-novice/

https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2021EA001797

Workflow

GitHub

Step 1: Create a github account

https://github.com/

Step 2: Create your first repository

follow on jupyter notebook

Step 3: Make your first commit - prepare for the rest of the course!

Step 4: Clone the course repo to keep up to date with the latest content.

More info:
https://docs.github.com/en/github/

https://docs.github.com/en/github/importing-your-projects-to-github/importing-source-code-to-github/adding-an-existing-project-to-github-using-the-command-line

https://cyberhelp.sesync.org/faq/git-and-jupyterlab.html