Introduction to Scientific Python

Application to Oceanography

C. Troupin, E. Mason

SOCIB, IMEDEA

Cádiz, 27-29 January 2016

Table of contents

Table of contents

Introduction:

What? Why? How?

What is Python?

Programming language:

1. interpreted — instructions executed directly
2. dynamically typed — type checking at run-time
3. object-oriented — classes, objects, methods, ...
4. high-level — strong abstraction

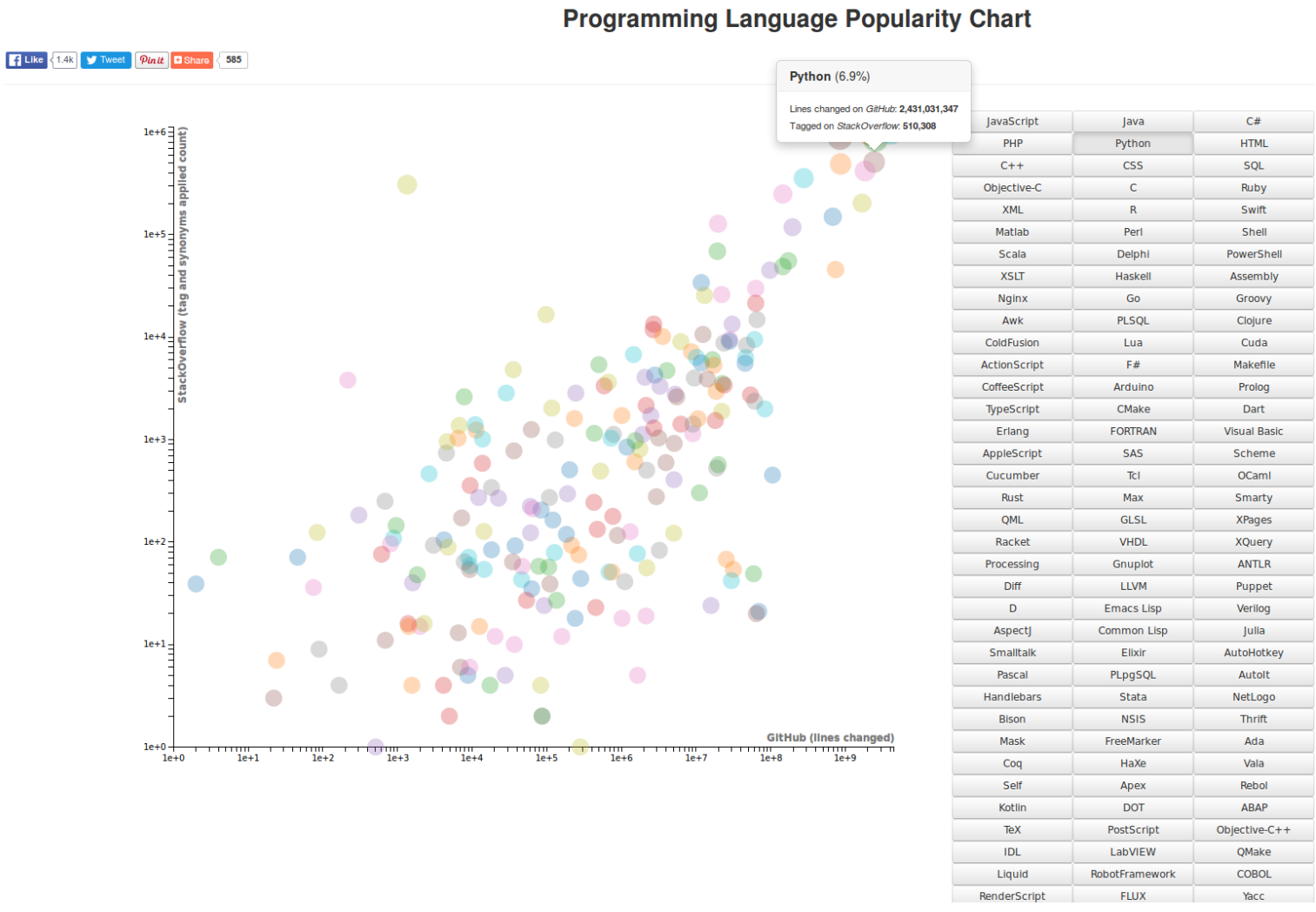
https://www.python.org

Why Python?

1. Simple, easy to learn syntax
2. Open
3. Large user community

doc, support, packages

Why Python?



Source: http://langpop.corger.nl/

Python vs. Matlab

Python	Matlab
General	
programming language	programming language + numerical computing environment
open	proprietary algorithms
general purpose	linear algebra
Indexing	
a[0]	a(1)
a[-1]	a(end)
a[::2]	a(1:2:end)
Functions	
a.max()	max(a)
a.shape()	size(a)

Numpy for Matlab users:
https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html

Quick example: hello.py

Quick example: hello.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
'''
This function prints "Hello world"
'''

def hello():
  print "Hello world"
  return

def main():
  hello()

if __name__ == '__main__':
  main()
```

Quick example: hello.py

1. Try to document you code

Quick example: hello.py

1. Try to document you code
2. Use

```
# -*- coding: utf-8 -*-
```

   if you're using non-ascii characters

Quick example: hello.py

1. Try to document you code
2. Use

```
# -*- coding: utf-8 -*-
```

   if you're using non-ascii characters

3. In Python 3:

```
print("Hello world!")
```

Quick example: hello.py

1. Try to document you code
2. Use

```
# -*- coding: utf-8 -*-
```

   if you're using non-ascii characters
3. In Python 3:

```
print("Hello world!")
```

4. Indentation matters!

A few definitions

Object: Python's abstraction for data
https://docs.python.org/2/reference/datamodel.html#index-0

Function: series of statements which returns some value to a caller
https://docs.python.org/2/glossary.html#term-function

Module: file containing Python definitions and statements
https://docs.python.org/2/tutorial/modules.html

Class: logical grouping of data and methods (functions)
https://docs.python.org/2/tutorial/classes.html

Data types

1. Numbers

Example:

```
g = 9.81
h = 4.135667662e-15
```

Data types

1. Numbers
2. String

Example:

```
name = "Rickman"
s = "this is a string"
```

Data types

1. Numbers
2. String
3. List

Example:

```
list = ['one', 2, 'three', 'four', 5]
```

Data types

1. Numbers
2. String
3. List
4. Tuple

Example:

```
tuple1 = ('one', 2, 'three', 'four', 5)
tuple2 = (1, '1', 'one', [1, 2], (1, 2, 3))
```

Tuples are *immutable* (fixed value)

Data types

1. Numbers
2. String
3. List
4. Tuple
5. Dictionary

Example:

```
email = {'Evan': 'emason@imedea.uib-csic.es',\
         'Irene': 'irene.laiz@uca.es',\
         'Charles': 'ctroupin@socib.es'}
email['Irene']
'irene.laiz@uca.es'
```

Data types

1. Numbers
2. String
3. List
4. Tuple
5. Dictionary

More details:
https://docs.python.org/2/tutorial/datastructures.html#

Resources

Web:

- https://docs.python.org/2.7/tutorial/index.html
- https://developers.google.com/edu/python/introduction?hl=en
  tutorial + exercises
- http://www.python-course.eu
- http://www.learnpython.org online code execution
  https://pythonprogramming.net
- https://www.gitbook.com/book/djangogirls/djangogirls-tutorial

Resources

Learning platforms:

- Programming Foundations with Python Learn Object-Oriented Programming (7 weeks)
- Code Academy (13 hours)

Resources

Youtube:

- Python Beginner Tutorial (For Absolute Beginners) (4 parts)
- Google Python Class (7×30 minutes)
- Zero to Hero with Python (11 hours)

Resources

Books:

- *Learn Python the hard way*, Z.A. Shaw, 2013
  http://learnpythonthehardway.org/book/
- *Learning Python, 5th Edition*, M. Lutz, 2013
- *Python Programming: An Introduction to Computer Science*, J.M. Zelle, 2002

Complete list: https://wiki.python.org/moin/PythonBooks

Python 2 vs Python 3

Some differences:

- Print function
- Integer division
- Unicode
- ...

Python 3.x = present and future of the language

More details:

Python 2 or Python 3

Will Scientists Ever Move to Python 3?

Objectives of the course

1. Use Python to solve oceanography-related problems

Objectives of the course

1. Use Python to solve oceanography-related problems
2. Read/write various types of files

Objectives of the course

1. Use Python to solve oceanography-related problems
2. Read/write various types of files
3. Generate high-quality figures

Objectives of the course

1. Use Python to solve oceanography-related problems
2. Read/write various types of files
3. Generate high-quality figures
4. Find resources on the internet for specific problems

Objectives of the course

1. Use Python to solve oceanography-related problems
2. Read/write various types of files
3. Generate high-quality figures
4. Find resources on the internet for specific problems

What we won't (can't) do:
teach you how to be a good programmer

About the trainers

Evan Mason
Oceanographer
Post-doctoral researcher at IMEDEA
10-year experience with Python

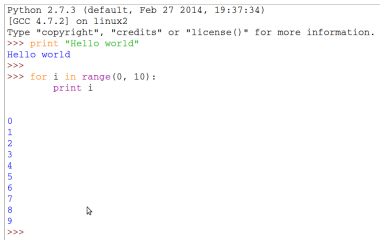
About the trainers

Evan Mason
Oceanographer
Post-doctoral researcher at IMEDEA
10-year experience with Python

Charles Troupin
Engineer, oceanographer
Head of SOCIB Data Center
2.5-year experience with Python

Structure of the course

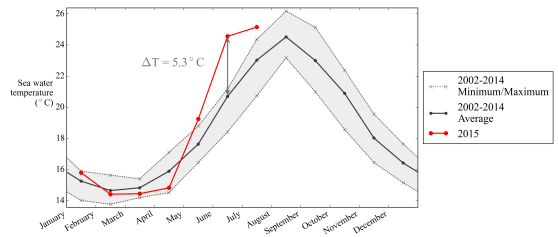
1. Reading/writing



Structure of the course

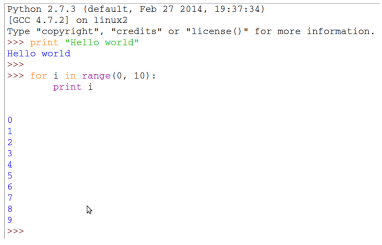
1. Reading/writing
2. Time series

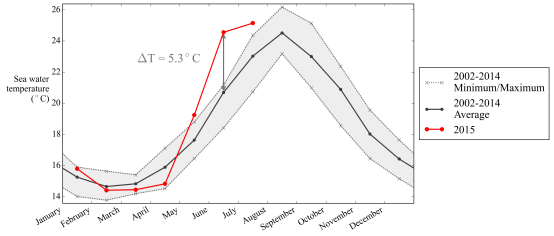
```
Python 2.7.3 (default, Feb 27 2014, 19:37:34)
[GCC 4.7.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello world"
Hello world
>>>
>>> for i in range(0, 10):
        print i


0
1
2
3
4
5
6
7
8
9
>>>
```

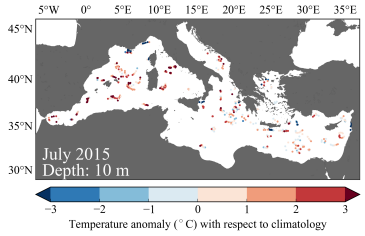


Structure of the course

1. Reading/writing
2. Time series
3. 2-D fields

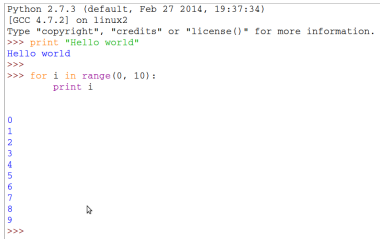


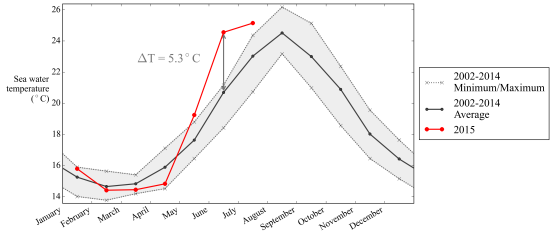


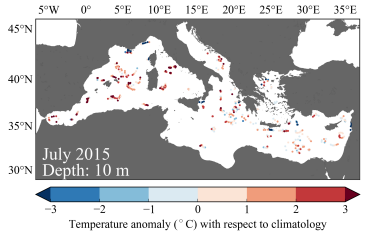


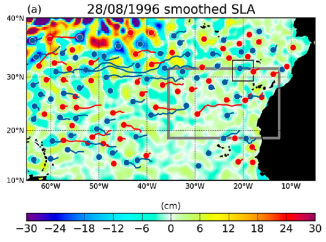
Structure of the course

1. Reading/writing
2. Time series
3. 2-D fields
4. Functions, classes, modules









Installation & use

Installing Python

Hard way: download source and compile:
https://www.python.org/downloads/

Installing Python

Hard way: download source and compile:
https://www.python.org/downloads/

Normal way: use installer or package:

- Windows:
  https://www.python.org/downloads/windows/
- Mac OS:
  https://www.python.org/downloads/mac-osx/
- Linux: package manager:
  python2.x and python2.x-dev packages

Installing Python

Hard way: download source and compile:
https://www.python.org/downloads/

Normal way: use installer or package:

- Windows:
  https://www.python.org/downloads/windows/
- Mac OS:
  https://www.python.org/downloads/mac-osx/
- Linux: package manager:
  python2.x and python2.x-dev packages

Easy way: Python distributions such as:

Anaconda	free
Enthought Canopy	free and commercial
Python(x,y)	free, Windows only

+ others

Installing modules

Example: SciPy (http://www.scipy.org/):
mathematics, science, and engineering

Installing modules

Example: SciPy (http://www.scipy.org/):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution

Installing modules

Example: SciPy (http://www.scipy.org/):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution

Intermediate: Linux, Mac: install package

- Linux: package manager
- Mac: MacPorts, Homebrew

Installing modules

Example: SciPy (http://www.scipy.org/):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution

Intermediate: Linux, Mac: install package

- Linux: package manager
- Mac: MacPorts, Homebrew

Harder: build from source

```
$ python setup.py install
```

Installing modules

Example: SciPy (http://www.scipy.org/):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution

Intermediate: Linux, Mac: install package

- Linux: package manager
- Mac: MacPorts, Homebrew

Harder: build from source

```
$ python setup.py install
```

Better: pip (https://pypi.python.org/pypi/pip)

Installing modules

Example: SciPy (http://www.scipy.org/):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution

Intermediate: Linux, Mac: install package

- Linux: package manager
- Mac: MacPorts, Homebrew

Harder: build from source

```
$ python setup.py install
```

Better: pip (https://pypi.python.org/pypi/pip)

Avoid: mixing installation methods

Using `pip` to manage modules

`pip` = recommended tool for installing Python packages

Using `pip` to manage modules

`pip` = recommended tool for installing Python packages

Installation:

- Included in recent Python version
- Otherwise: download and run `get-pip.py`
  https://pip.pypa.io/en/stable/installing/#install-pip

```
$ python get-pip.py
```

Using `pip` to manage modules

`pip` = recommended tool for installing Python packages

Installation:

- Included in recent Python version
- Otherwise: download and run `get-pip.py`
  https://pip.pypa.io/en/stable/installing/#install-pip

```
$ python get-pip.py
```

Usage:

- Install latest version + dependencies:

```
$ pip install Package
```

- Specify exact version:

```
$ pip install Package==x.y.z
```

- Specify minimum version:

```
$ pip install 'Package>=x.y.z'
```

More about `pip`

- Uninstall packages:

```
$ pip uninstall
```

More about `pip`

- Uninstall packages:

```
$ pip uninstall
```

- List installed packages:

```
$ pip list
```

```
aptoncd (0.1.98-bzr117-1.2)
backports.ssl-match-hostname (3.4.0.2)
basemap (1.0.7)
...
xhtml2pdf (0.0.6)
zope.interface (3.6.1)
```

More about `pip`

- Uninstall packages:

```
$ pip uninstall
```

- List installed packages:

```
$ pip list
```

- Output installed packages in requirements format:

```
$ pip freeze
```

```
aptoncd===0.1.98-bzr117-1.2
backports.ssl-match-hostname==3.4.0.2
basemap==1.0.7

...
xhtml2pdf==0.0.6
zope.interface==3.6.1
```

More about `pip`

- Uninstall packages:

```
$ pip uninstall
```

- List installed packages:

```
$ pip list
```

- Output installed packages in requirements format:

```
$ pip freeze
```

- Show information about installed packages:

```
$ pip show Package
```

```
Metadata-Version: 1.1
Name: numpy
Version: 1.9.2
Summary: NumPy: array processing for numbers, strings, ...
...
Requires:
```

Running your code: several solutions

Edit, then run in a shell:

```
$ python mycode.py
```

or

```
$ mycode.py
```

if *shebang*

```
#!/usr/bin/python
```

is present at the 1st line

Running your code: several solutions

Interactive python (ipython)
Auto-completion, exploring objects, . . .

```
In [2]: string = 'Hello all'

In [3]: string.
string.capitalize   string.encode       string.format       ...
string.rstrip       string.strip        string.upper
...
string.startswith   string.translate
```

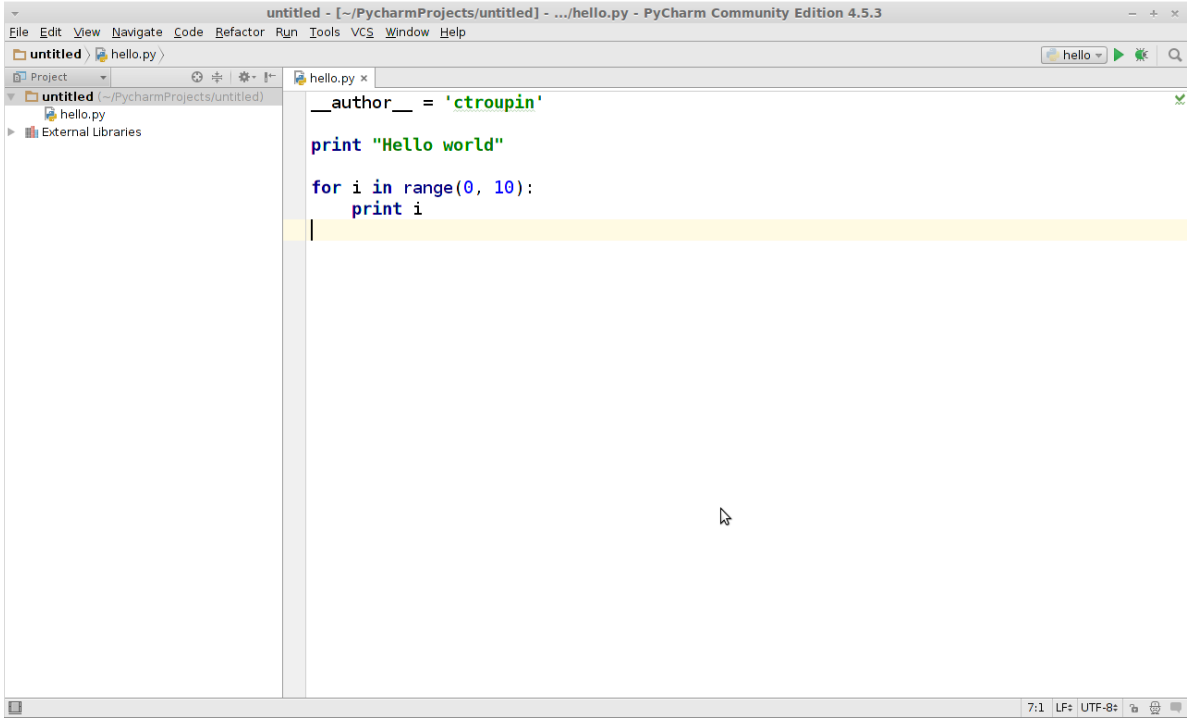
+ *magic* functions:

- `%run`: Run the named file inside IPython as a program
- `%timeit`: Time execution of a Python statement or expression
- `%who`: Print all interactive variables, with some minimal formatting

More: Built-in magic commands

Running your code: several solutions

Integrated Development Environment (IDE)
(editor + build automation tools + debugger)

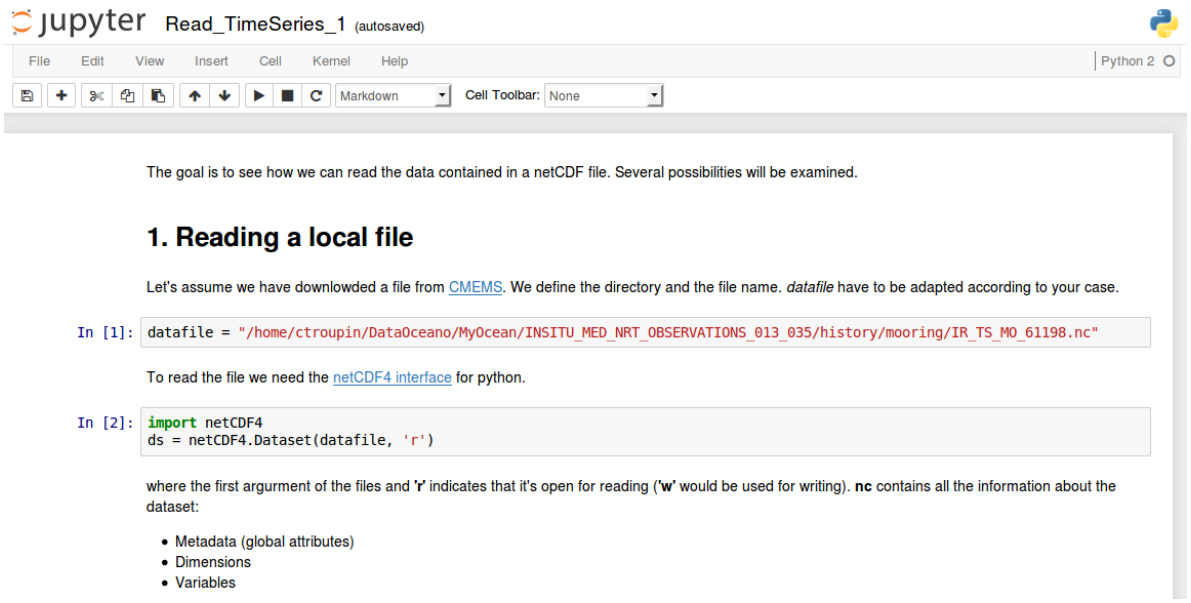


Examples: Atom, Eclipse, PyCharm, Idle, . . .
Complete list: https://wiki.python.org/moin/PythonEditors

Running your code: several solutions

ipython notebook

(interactive computational environment)



Rich text + command executions + figures + . . .

"*Data story telling*"

(see Programming in Python 2)

What do we work with?

Exercise 1: changecase.py

 :

Write a program that takes 2 arguments: the name and the surname, both written with a mix of upper and lowercase, and return the name with the first letter in uppercase and the surname with all the letters in uppercase.

Examples:

changecase allan rickman returns *Allan RICKMAN*

changecase aLlAn ricKmaN returns *Allan RICKMAN*

Tips: use the function sys.argv to be able to run the code as

```
$ changecase name surname
```

Programming in Python #1

importing/exporting data

El més important, que quedi clar el zen de python

Objective

Be able to read/write data in various formats used in oceanography

Objective

Be able to read/write data in various formats used in oceanography

Text files (ascii)
CSV
NetCDF
HDF
Images (geotiff)
matlab files
...

Objective

Be able to read/write data in various formats used in oceanography

Text files (ascii)
CSV
NetCDF
HDF
Images (geotiff)
matlab files
...

Be able to find the resources to read another format

What is an ipython notebook?

What is an ipython notebook?

Python: high-level programming language
https://www.python.org/

What is an ipython notebook?

Python: high-level programming language
https://www.python.org/

IPython: command shell for interactive computing
http://ipython.org/

What is an ipython notebook?

Python: high-level programming language
https://www.python.org/

IPython: command shell for interactive computing
http://ipython.org/

IPython notebook: web-based interactive computational environment combining code, text, figures, ...
http://ipython.org/notebook.html

Why ipython notebooks?

IP[y]: IPython
Interactive Computing

- User-friendly
- Free, easy to write, easy to read
- Code and results visible online via http://nbviewer.ipython.org

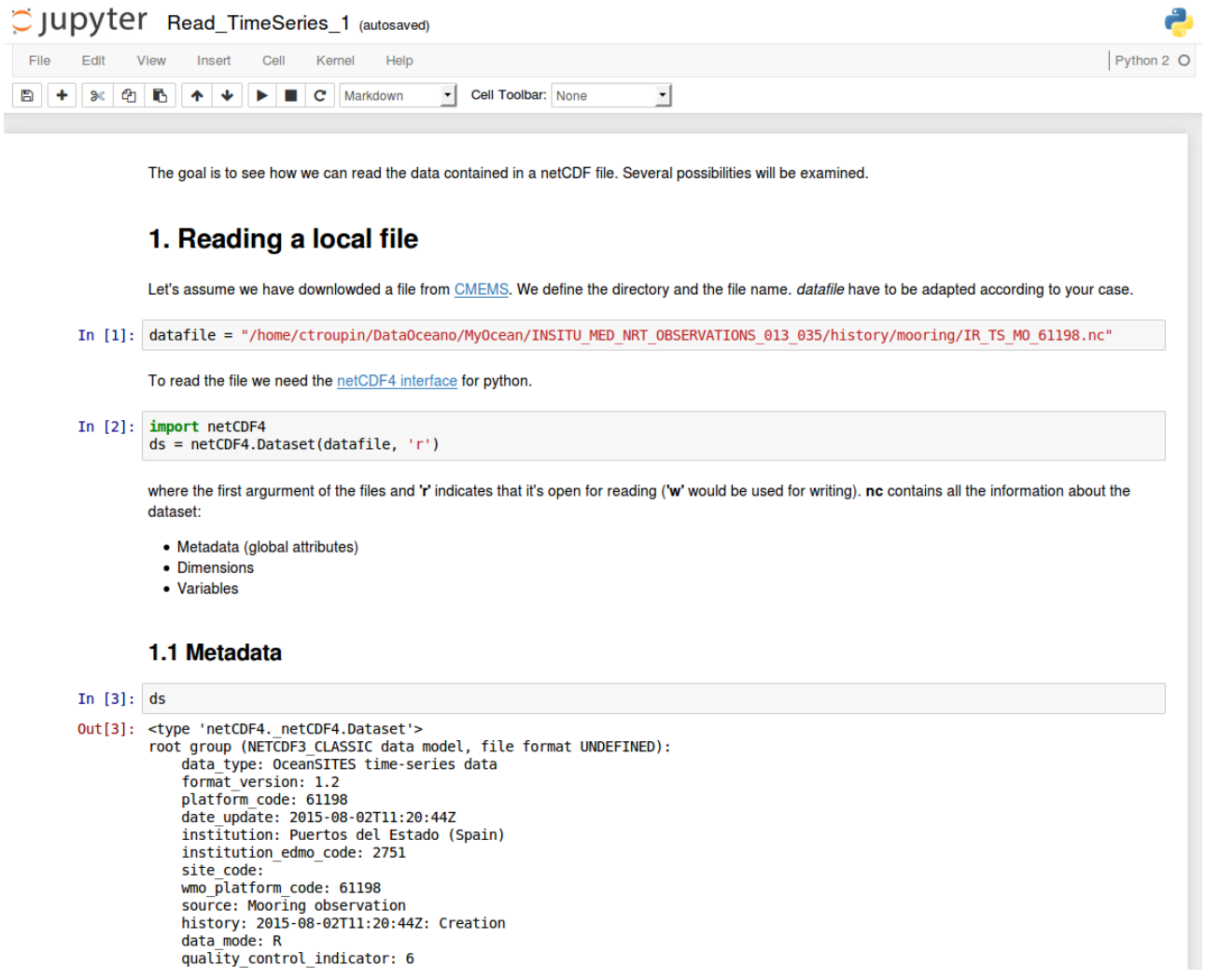
Why ipython notebooks?

IP[y]: IPython
Interactive Computing

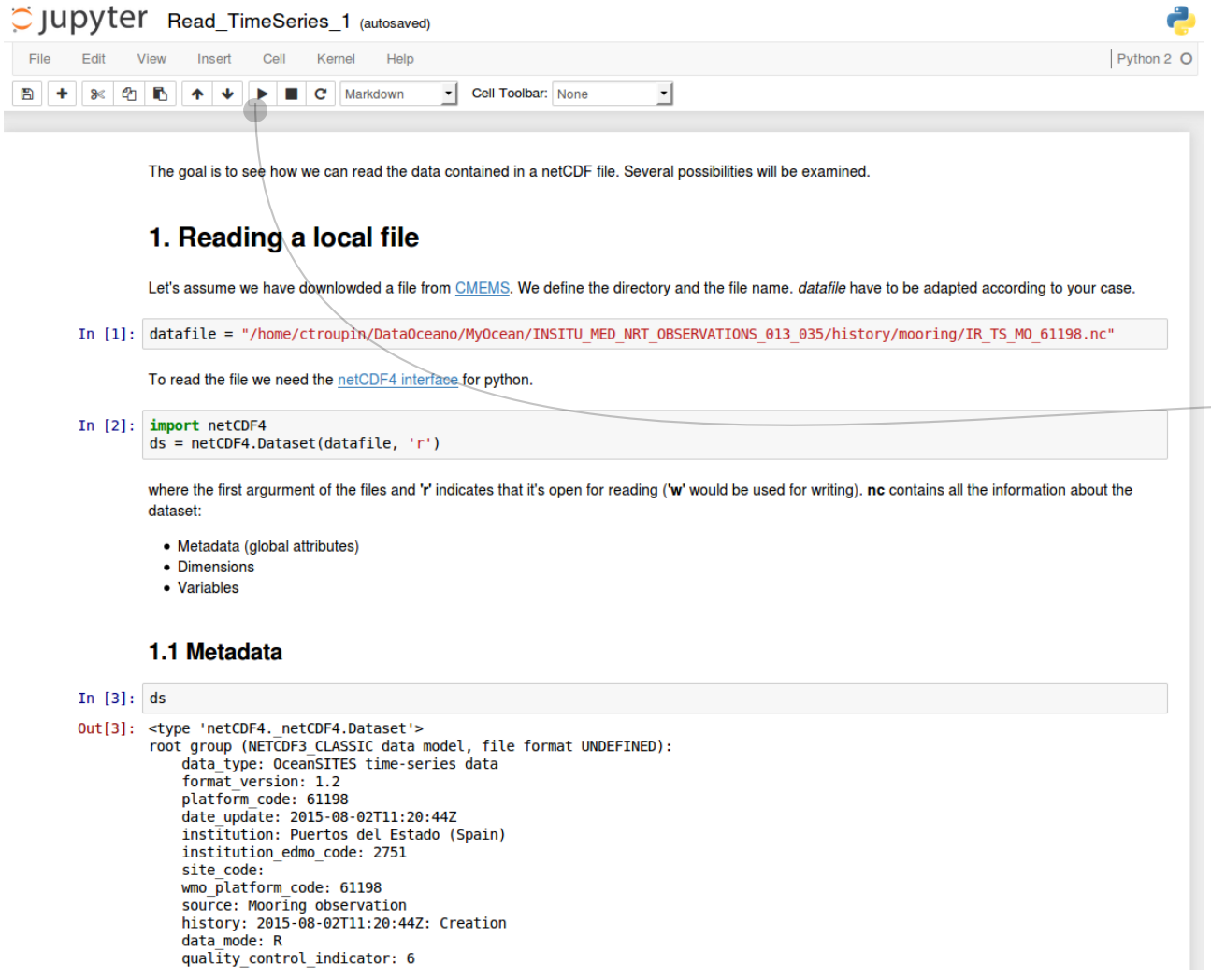
- User-friendly
- Free, easy to write, easy to read
- Code and results visible online via http://nbviewer.ipython.org

- *Data story-telling*

Structure of a notebook

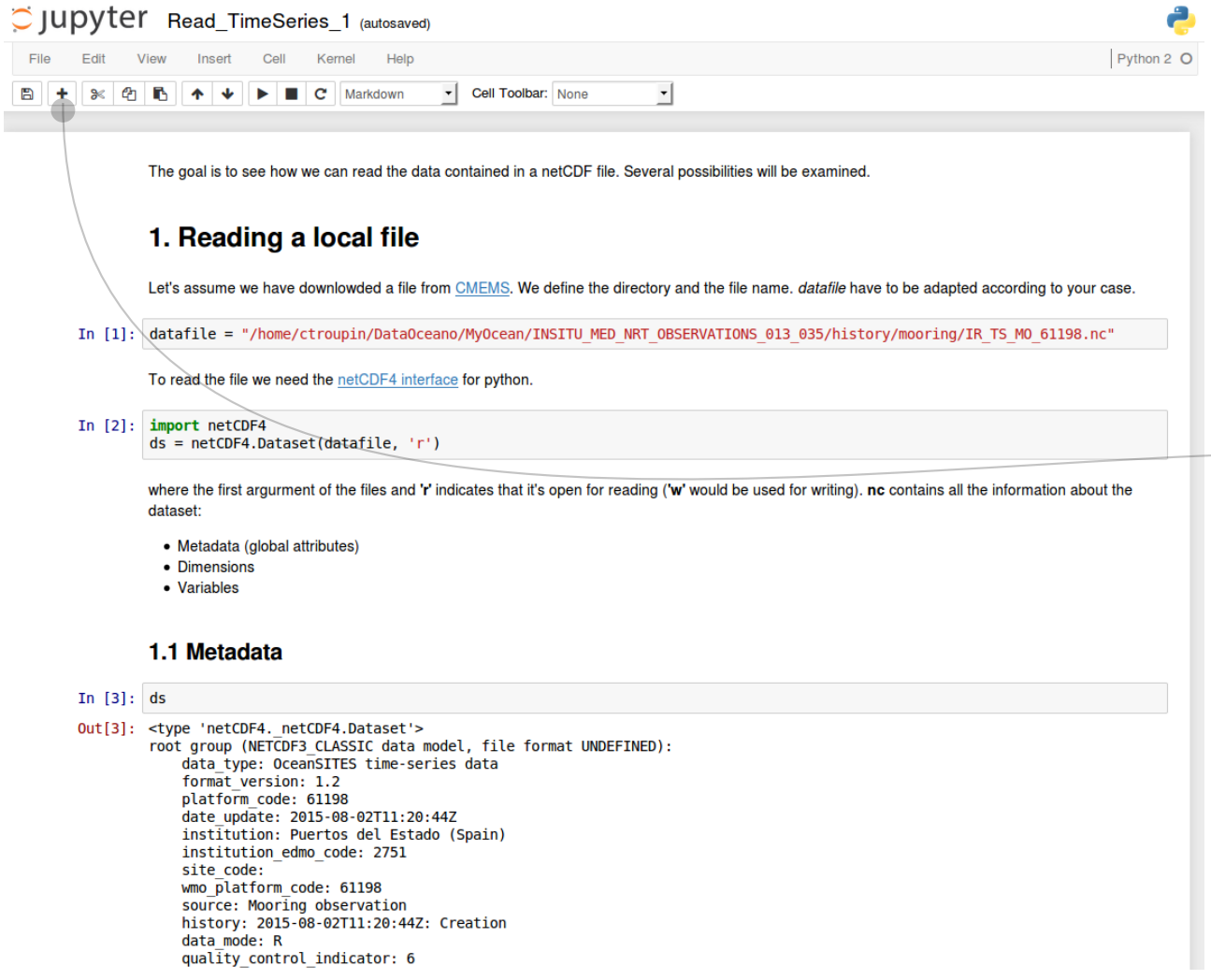


Structure of a notebook



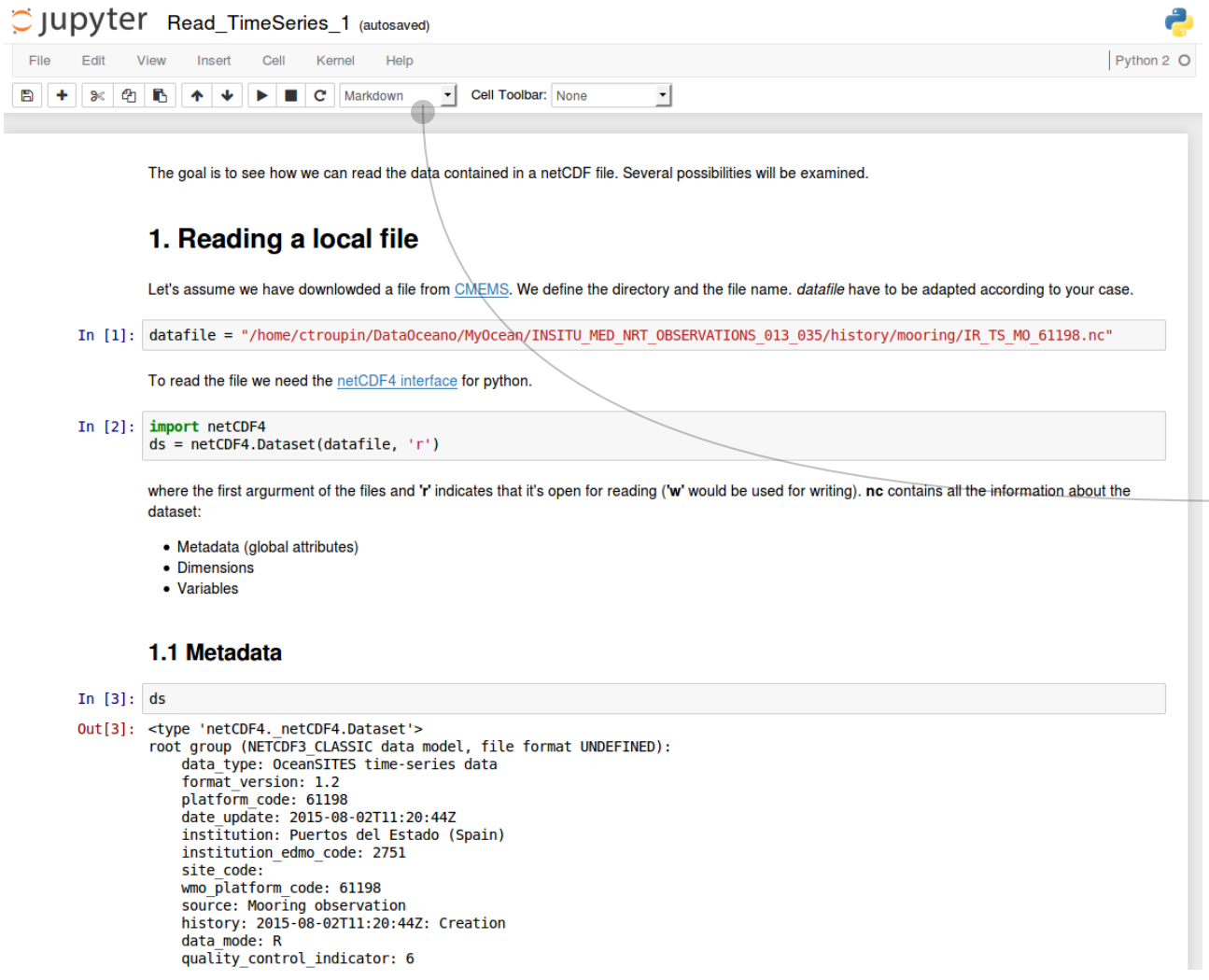
Run current cell

Structure of a notebook



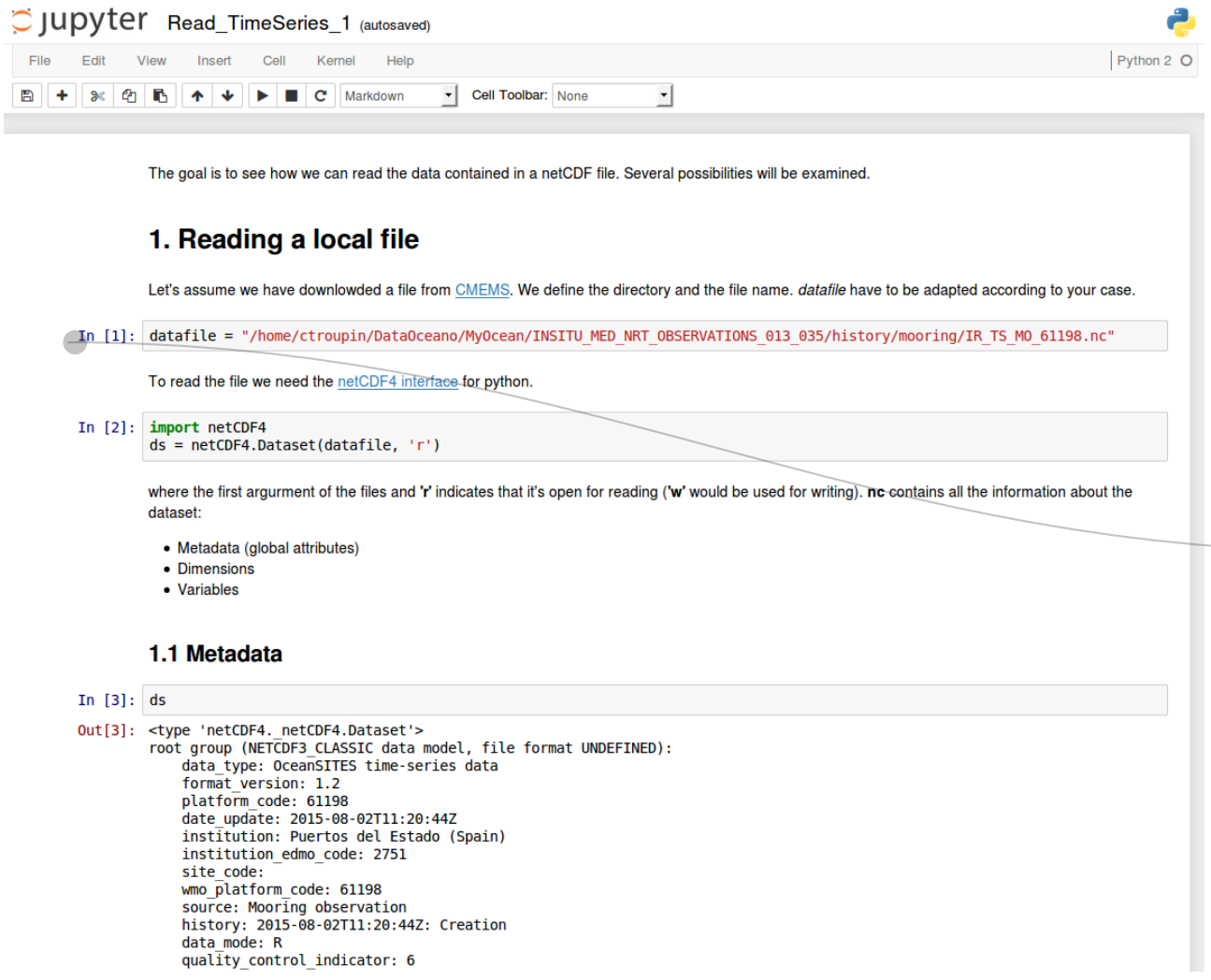
Run current cell
Add a new cell

Structure of a notebook



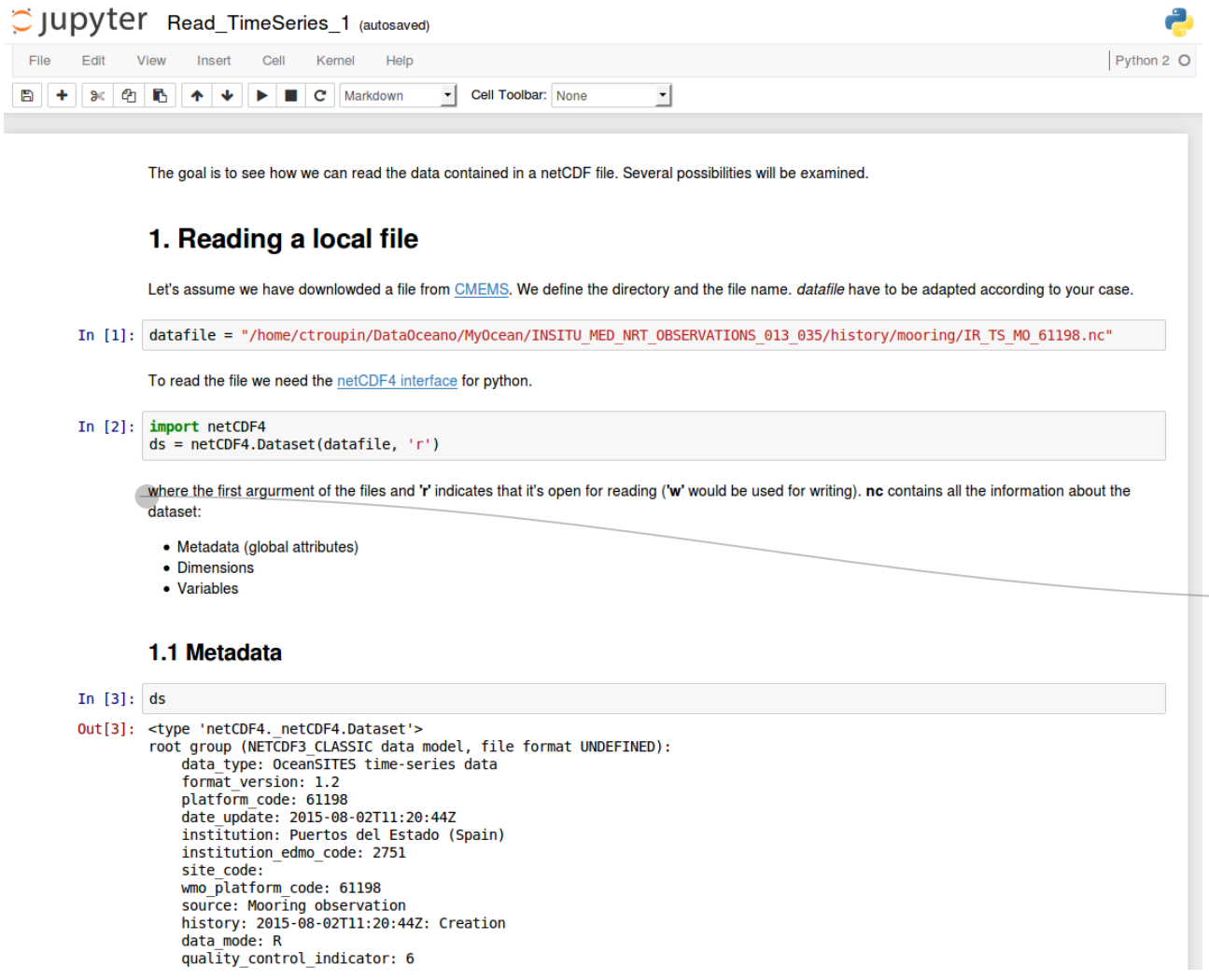
Run current cell
Add a new cell
Select type of cell

Structure of a notebook



Run current cell
Add a new cell
Select type of cell
Code cell

Structure of a notebook



Run current cell
Add a new cell
Select type of cell
Code cell
Text cell

Comma-Separated Values file

Example: file `buoy-canaldeibiza_SALT_SBE37.csv`

Example of CSV file

```
Platform, Instrument, Paramenter, Units, Date, Value, QC value
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 12:00:00, 36.916, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 01:00:00, 36.936, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 02:00:00, 36.929, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 03:00:00, 36.927, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 04:00:00, 36.925, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 05:00:00, 36.948, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 06:00:00, 36.95, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 07:00:00, 36.954, 1.0
Buoy_CanalDeIbiza, SCB-SBE37006, sea_water_salinity, psu, 2015-12-01 08:00:00, 36.933, 1.0
```

Comma-Separated Values file

- Type ”python read csv file” in search engine

Comma-Separated Values file

- Type ”python read csv file” in search engine
- Result: https://docs.python.org/2/library/csv.html

Comma-Separated Values file

- Type ”python read csv file” in search engine
- Result: https://docs.python.org/2/library/csv.html
- From the doc:

```
import csv
csvfile = open('buoy-canaldeibiza\_SALT_SBE37.csv', 'rb')
reader = csv.reader(csvfile)
for row in reader:
  print row
csvfile.close()
```

Comma-Separated Values file

- Type ”python read csv file” in search engine
- Result: https://docs.python.org/2/library/csv.html
- From the doc:

```
import csv
csvfile = open('buoy-canaldeibiza\_SALT_SBE37.csv', 'rb')
reader = csv.reader(csvfile)
for row in reader:
  print row
csvfile.close()
```

: `read_csv_file.ipynb`

Seabird CTD file

Example: file `dsb01.cnv`

Example of CNV file

```
* Sea-Bird SBE 9 Data File:
* FileName = C:\OCEANO\CTD\DATOS\IMEDEA-SHEBEX\SB01.hex
* Software Version Seasave V 7.22
* Temperature SN = 5427
* Conductivity SN = 3872
...
38.86539   2.78989   ...  3.4535e-02     7.459
38.86542   2.78986   ...  223  0.0000e+00
```

Seabird CTD file

- Type ”python read cnv file” in search engine

Seabird CTD file

- Type ”python read cnv file” in search engine
- Result: https://pypi.python.org/pypi/cnv

”Sorry for the inconvenience, but I moved all the functionalities into the package seabird”

Seabird CTD file

- Type ”python read cnv file” in search engine
- Result: https://pypi.python.org/pypi/cnv

  ”Sorry for the inconvenience, but I moved all the functionalities into the package seabird”

- Use pip to search for the package

```
pip search seabird
seabird     — Non official parser for Sea—Bird's sensors.
```

Seabird CTD file

- Type ”python read cnv file” in search engine
- Result: https://pypi.python.org/pypi/cnv

  ”Sorry for the inconvenience, but I moved all the functionalities into the package seabird”

- Use pip to search for the package

```
pip search seabird
seabird     — Non official parser for Sea—Bird's sensors.
```

- Install package:

```
pip install —user seabird==0.6.3
```

Seabird CTD file

- Type ”python read cnv file” in search engine
- Result: https://pypi.python.org/pypi/cnv

”Sorry for the inconvenience, but I moved all the functionalities into the package seabird”

- Use pip to search for the package

```
pip search seabird
seabird      - Non official parser for Sea-Bird's sensors.
```

- Install package:

```
pip install --user seabird==0.6.3
```

- From the doc:

```
from seabird.cnv import fCNV
profile = fCNV('')
profile.attributes        # It will return the header, as a dictionary.
profile.keys()            # It will list the available variables.
profile['TEMP']           # Get the data
```

Seabird CTD file

- Type "python read cnv file" in search engine
- Result: https://pypi.python.org/pypi/cnv

  "Sorry for the inconvenience, but I moved all the functionalities into the package seabird"

- Use pip to search for the package

```
pip search seabird
seabird      - Non official parser for Sea-Bird's sensors.
```

- Install package:

```
pip install --user seabird==0.6.3
```

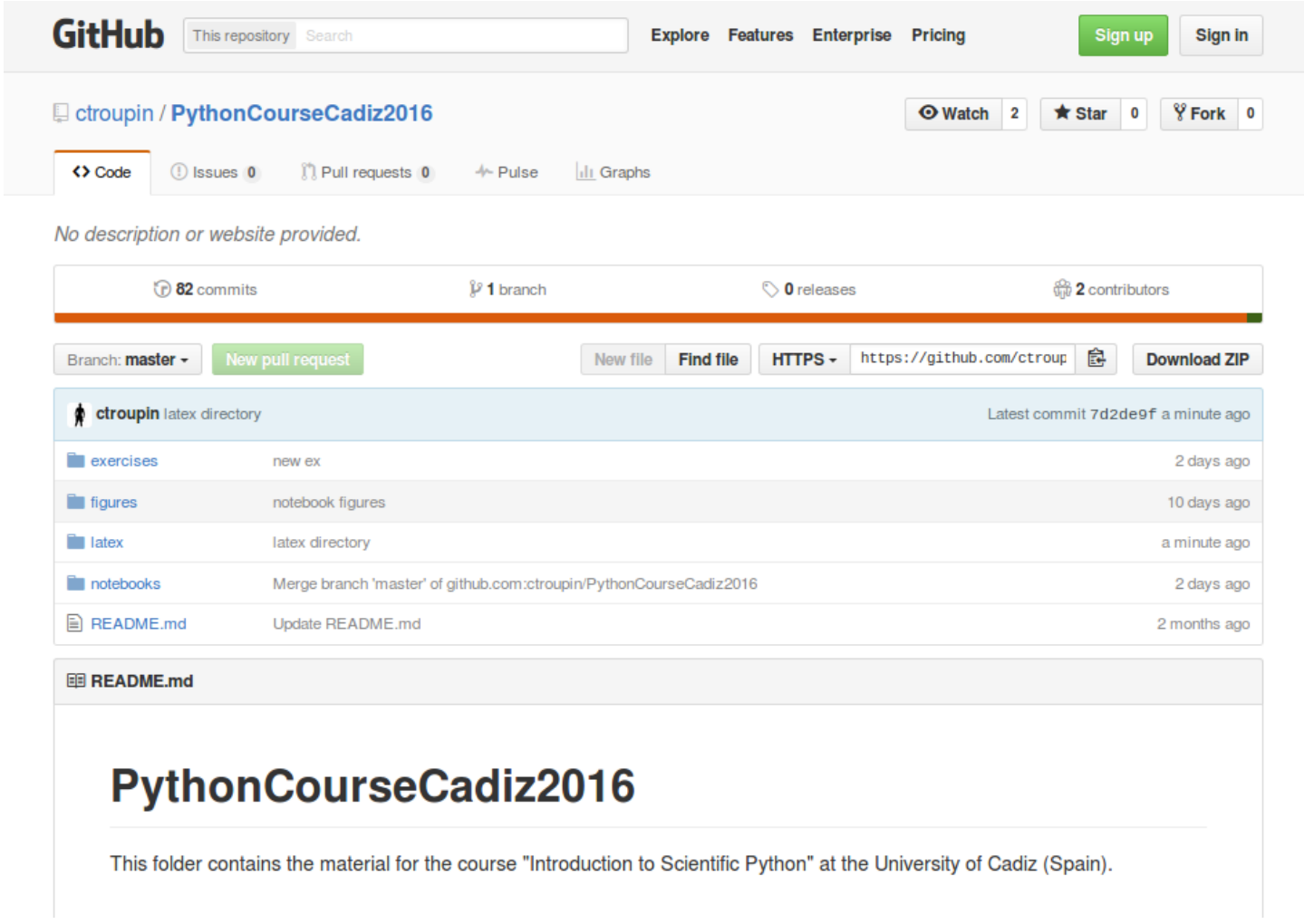
- From the doc:

```
from seabird.cnv import fCNV
profile = fCNV('')
profile.attributes       # It will return the header, as a dictionary.
profile.keys()           # It will list the available variables.
profile['TEMP']          # Get the data
```

: `read_cnv_file.ipynb`

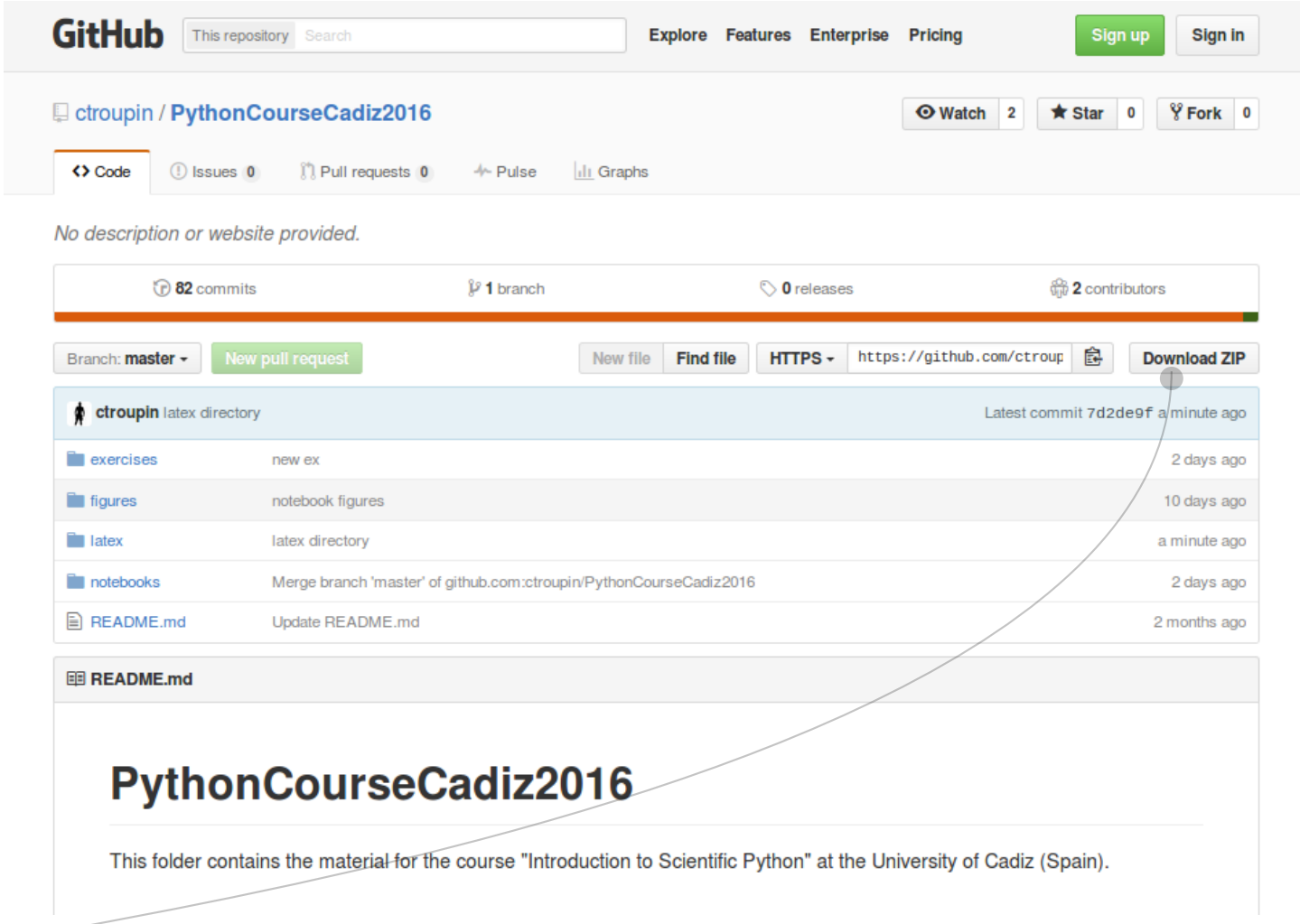
How to get code from github

https://github.com/ctroupin/PythonCourseCadiz2016



How to get code from github

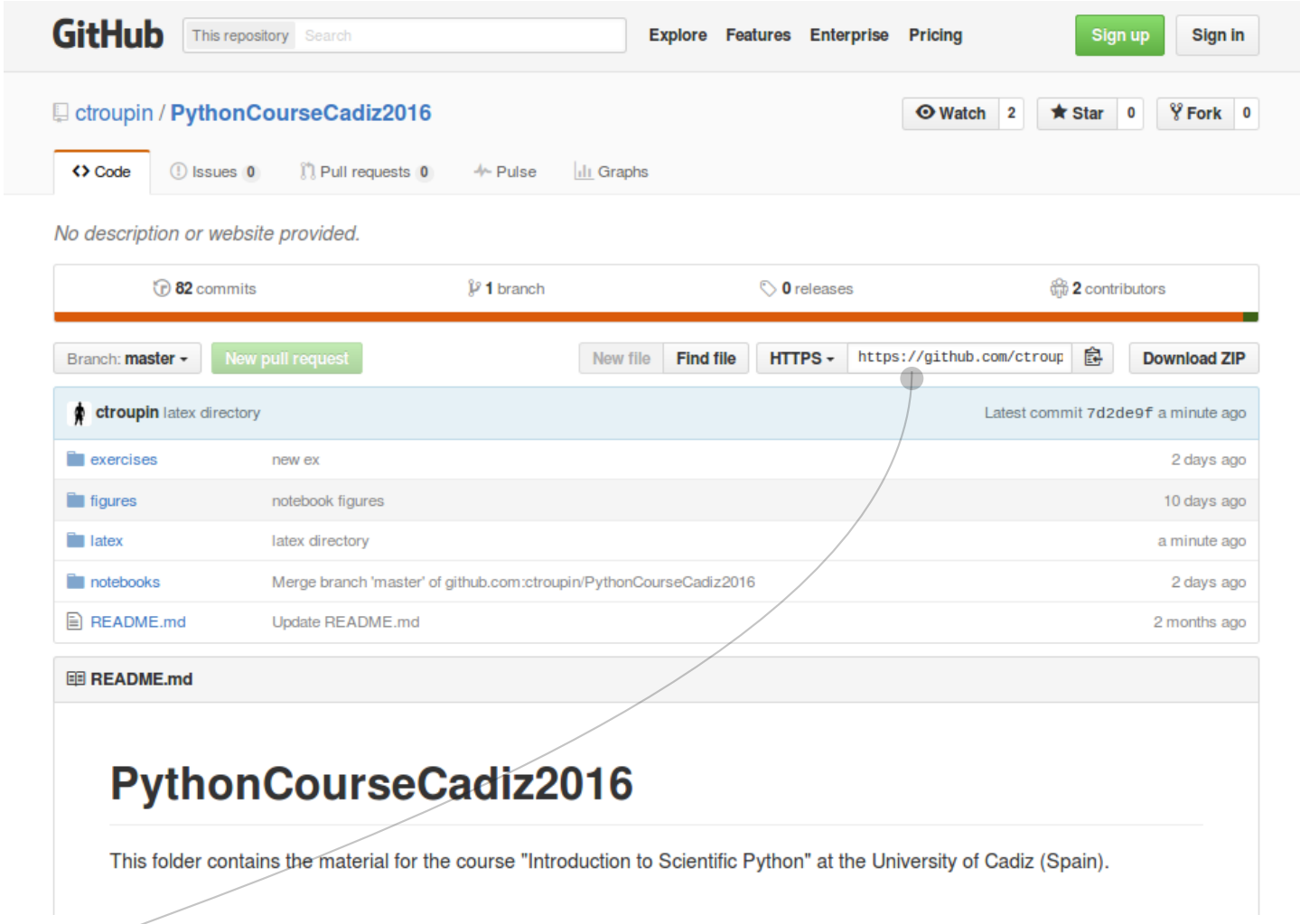
https://github.com/ctroupin/PythonCourseCadiz2016



Click and download the `.zip` file or ...

How to get code from github

https://github.com/ctroupin/PythonCourseCadiz2016



Click and download the `.zip` file or ...

Copy the URL and type `git clone url` in a terminal

Update the code

1. Clone the repository

```
git clone https://github.com/... .git
```

Update the code

1. Clone the repository

```
git clone https://github.com/... .git
```

2. ⌛ (if there were modifications in the repository)

Update the code

1. Clone the repository

```
git clone https://github.com/... .git
```

2. ⌛ (if there were modifications in the repository)
3. Update your version

```
git pull
```

Update the code

1. Clone the repository

```
git clone https://github.com/... .git
```

2. ⌛ (if there were modifications in the repository)
3. Update your version

```
git pull
```

More about github:

- https://github.com/
- http://rogerdudler.github.io/git-guide/

NetCDF

Definition: software libraries and self-describing, machine-independent data formats (source: http://www.unidata.ucar.edu/software/netcdf/).

NetCDF

Definition: software libraries and self-describing, machine-independent data formats (source: http://www.unidata.ucar.edu/software/netcdf/).

Users: atmospheric/ocean observing/modelling communities: NOAA, EUMETSAT, NASA/JPL, USGS, Copernicus Marine Service, ...

NetCDF

Definition: software libraries and self-describing, machine-independent data formats (source: http://www.unidata.ucar.edu/software/netcdf/).

Users: atmospheric/ocean observing/modelling communities: NOAA, EUMETSAT, NASA/JPL, USGS, Copernicus Marine Service, ...

Tools: many tools and libraries to inspect, visualise and explore data sets.

NetCDF

Definition: software libraries and self-describing, machine-independent data formats (source: http://www.unidata.ucar.edu/software/netcdf/).

Users: atmospheric/ocean observing/modelling communities: NOAA, EUMETSAT, NASA/JPL, USGS, Copernicus Marine Service, ...

Tools: many tools and libraries to inspect, visualise and explore data sets.

Structure:

- header: dimensions, attributes, variables
- data: arrays

Quick inspection: ncdump

https://www.unidata.ucar.edu/software/netcdf/docs/netcdf/ncdump.html

text representation of a netCDF dataset (header information, variables, . . .)

ncdump applied on a file

```
ncdump -h 20140628_d-OC_CNR-L3-CHL-MedOC3_A_1KM-MED-DT-v02.nc

netcdf \20140628_d-OC_CNR-L3-CHL-MedOC3_A_1KM-MED-DT-v02 {
dimensions:
        time = 1 ;
        lat = 1580 ;
        lon = 3308 ;
variables:
        int time(time) ;
                time:long_name = "reference time" ;
                time:standard_name = "time" ;
                time:axis = "T" ;
                time:calendar = "Gregorian" ;
                time:units = "seconds since 1981-01-01 00:00:00" ;
...
                        "SUBSAMP=1\n",
                        "OUTMODE=0\n",
                        "" ;
}
```

Ferret

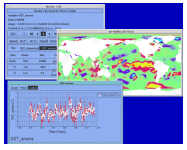
http://www.ferret.noaa.gov/Ferret/

visualization and analysis environment

Ferret to get basic info on file

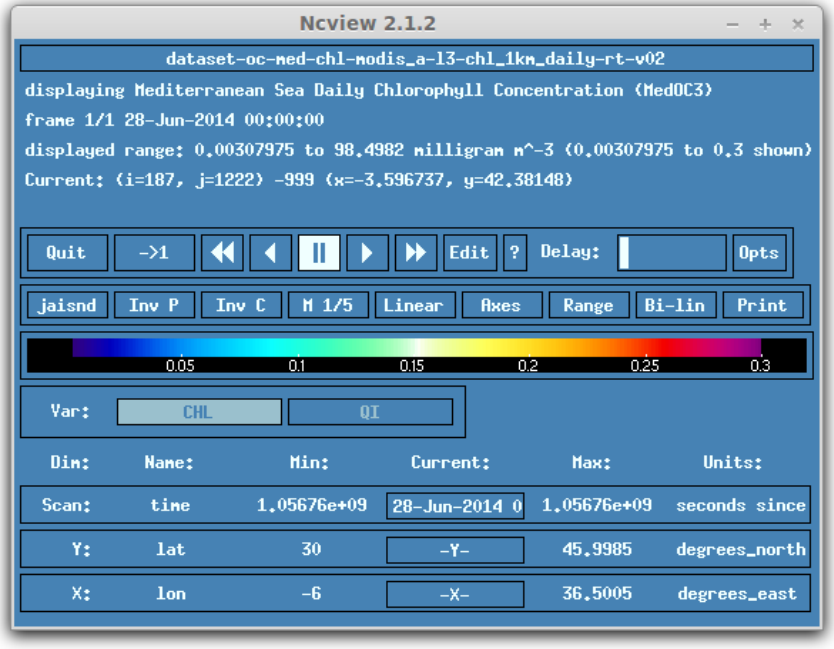
```
ctroupin@SCBD046 ~/Desktop $ ferret_c
         NOAA/PMEL TMAP
         FERRET v6.62
         Linux(gfortran) 2.6.9-89.0.20.ELsmp - 07/06/13
         25-Nov-15 12:23

yes? SET DATA 20140628_d-OC_CNR-L3-CHL-MedOC3_A_1KM-MED-DT-v02.nc
yes? SHOW DATA
      currently SET data sets:
    1> 20140628_d-OC_CNR-L3-CHL-MedOC3_A_1KM-MED-DT-v02.nc  (default)
 name      title                                I          J          K          L
 CHL       Mediterranean Sea Daily Chlorop  1:3308     1:1580     ...        1:1
 QI        Quality Index of Mediterranean   1:3308     1:1580     ...        1:1

yes?
```

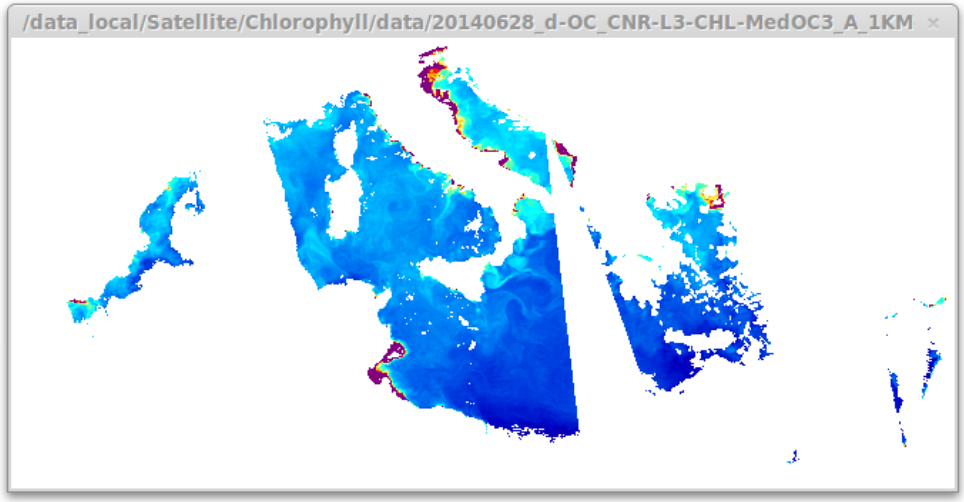
ncview



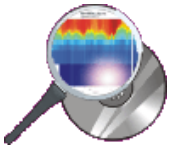
http://meteora.ucsd.edu/~pierce/ncview_home_page.html

quick visualisation of 3-4D fields



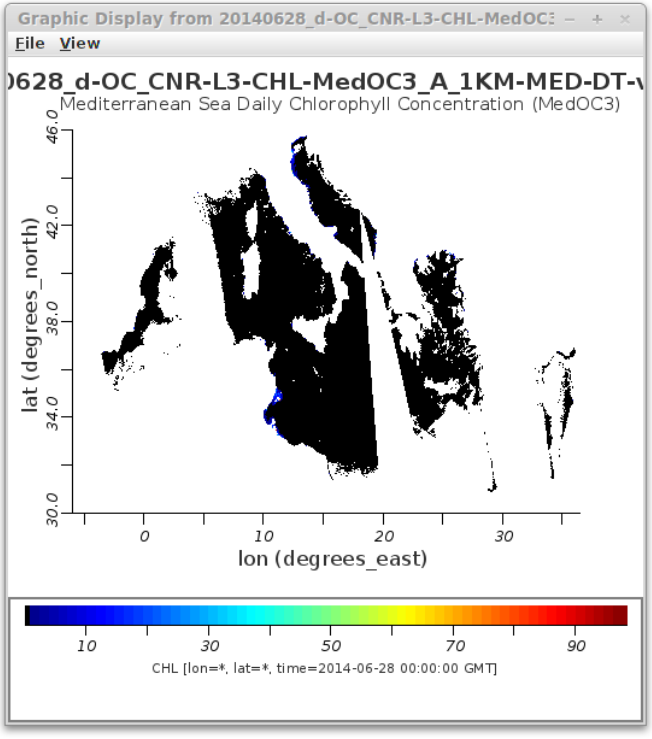


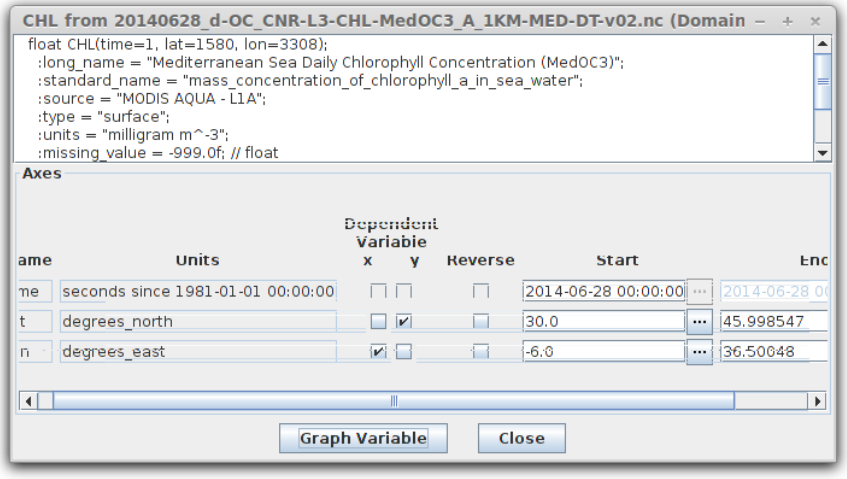
ncbrowse



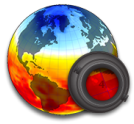
http://www.epic.noaa.gov/java/ncBrowse/

interactive graphical display



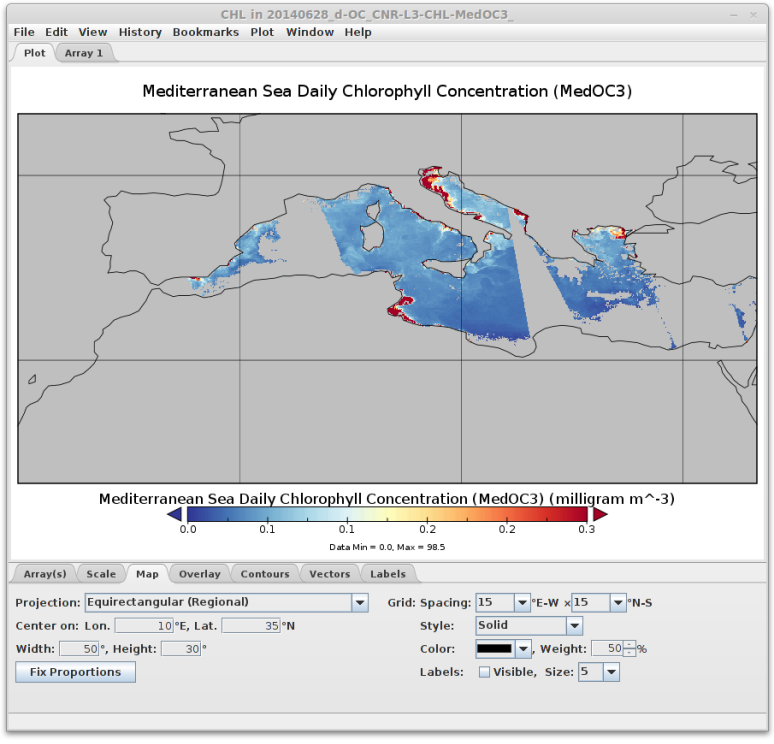


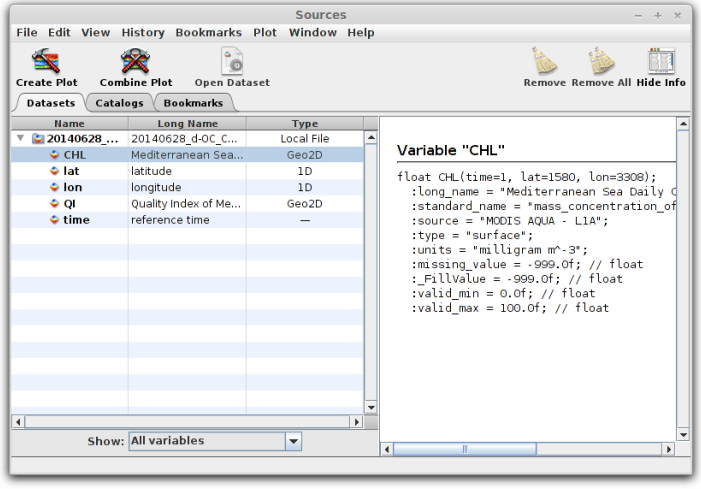
Panoply



http://www.giss.nasa.gov/tools/panoply/

plot, slice, combine, overlay, . . .





cdo – Climate Data Operators

- https://code.zmaw.de/projects/cdo
- manipulate (merging, averaging) netCDF files (+other formats)

Examples:

- Basic info (min, max, avg, size, . . .):

```
cdo info input.nc
```

- Compute standard deviation:

```
cdo fldstd input.nc output.nc
```

NCO – netCDF Operators

- http://nco.sourceforge.net/
- command line operations on netCDF files

Examples:

- Average variable over domain:

```
ncwa -O -a lon,lat input.nc output.nc
```

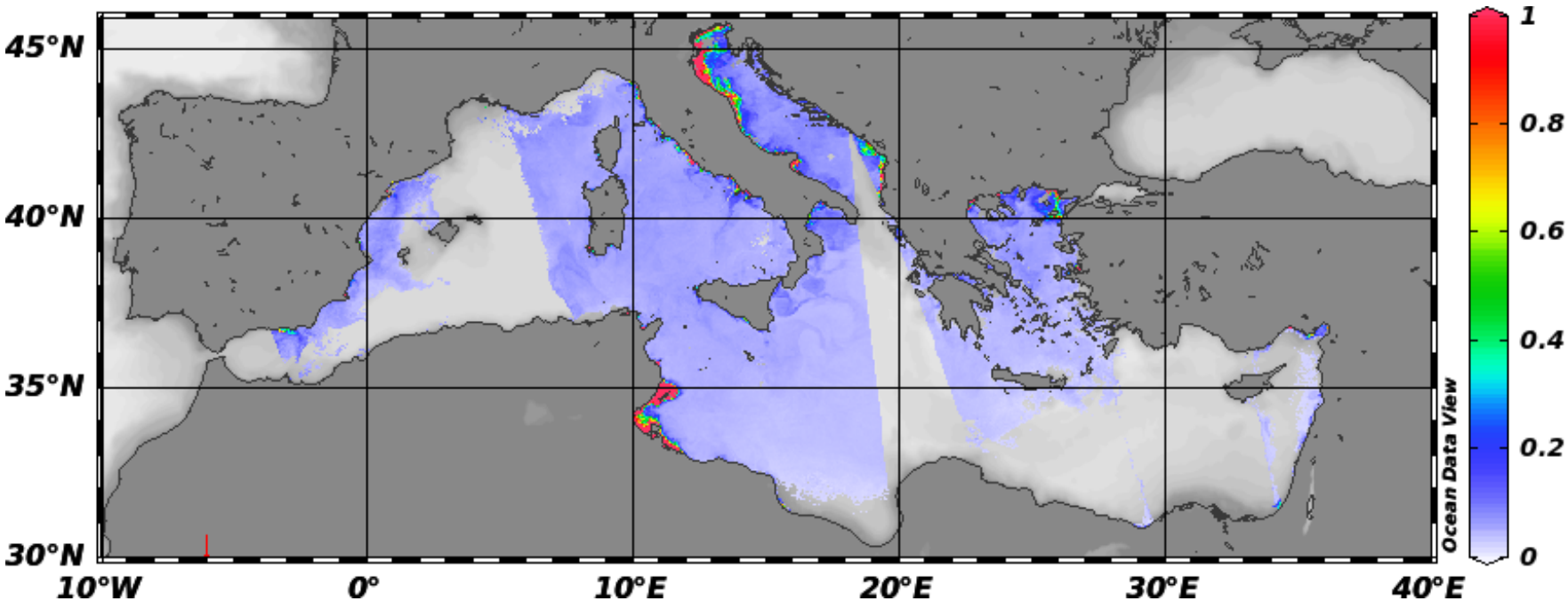
- Extract subregion:

```
ncks -d lon,13.,18.0 -d lat,33.0,36.0
input.nc output.nc
```

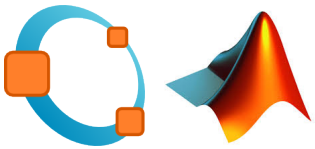
ODV – Ocean Data View

Ocean Data View

- http://odv.awi.de/en/home/
- interactive exploration, analysis and visualization of oceanographic data



Octave / Matlab



High-level functions to read/write data from/to a netCDF file:

- http://octave.sourceforge.net/netcdf/overview.html
- http://es.mathworks.com/help/matlab/network-common-data-form.html

Example with Octave

```
nc = netcdf('input.nc','r');                % open netcdf file in read-only
CHL = nc{'CHL'}(:);                         % retrieve variable
CHL_units = nc{'CHL'}.units;                % retrieve the attribute units
CHL_valid_range = nc{'CHL'}.valid_range;    % retrieve the attribute valid_range
global_history = nc.history;                % retrieve the global attribute history
```

Python

Python interface to the netCDF C library:
http://unidata.github.io/netcdf4-python/
Example: file
`dep0001_station-santantoni_scb-wlog001_L1_2016-01.nc`

Example with ipython

```
import netCDF4
nc = netCDF4.Dataset('dep0001_station-santantoni_scb-wlog001_L1_2016-01.nc')
print nc
<type 'netCDF4._netCDF4.Dataset'>
root group (NETCDF3_CLASSIC data model, file format UNDEFINED):
    title: Data from instrument SCB-WLOG001 on platform Station SantAntoni
    institution: SOCIB (Sistema de Observacion y prediccion Costero de las Islas Baleares)
    netcdf_version: 3.0
    Conventions: CF-1.6
    abstract: Deployment of instrument SCB-WLOG001 at Sant Antoni station in endurance line

    ...
nc.close()
```

List of notebooks

Located in `Data_ReadWrite`

read_csv_file.ipynb: Comma separated values

read_cnv_file.ipynb: SeaBird CTD file

read_netcdf_local.ipynb: local netCDF file

read_netcdf_opendap.ipynb: netCDF on thredds data server

read_netcdf_cf.ipynb: netCDF using CF conventions

read_shapefile.ipynb: geospatial vector data

read_geotiff.ipynb: geotiff image

read_image.ipynb: jpg image

read_HDF_file.ipynb: Hierarchical Data Format

read_mat_file.ipynb: .mat file

Exercise 1: data reading

Objective: read *unknown* file format

⚙: `1-read_CMEMS_indexfile.ipynb`

Programming in Python #2

time series, scatter plots, . . .

Useful packages

NumPy: package for scientific computing
http://www.numpy.org/

SciPy: open-source software for mathematics, science, and engineering http://www.scipy.org/

Useful packages

NumPy:	package for scientific computing http://www.numpy.org/
SciPy:	open-source software for mathematics, science, and engineering http://www.scipy.org/
matplotlib:	2D plotting library which produces publication quality figures http://matplotlib.org/

List of notebooks

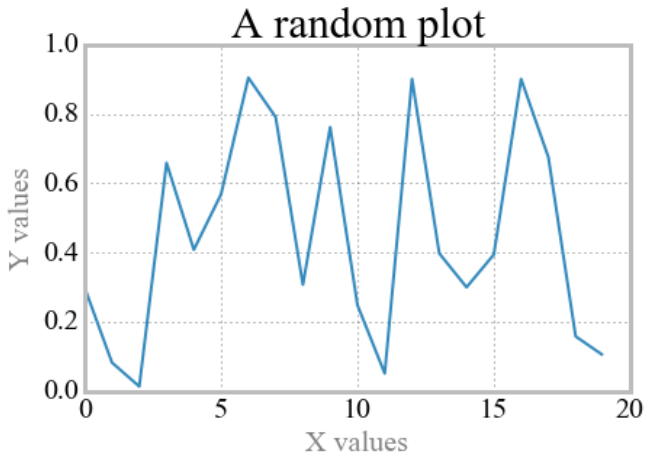
Located in `Plotting`

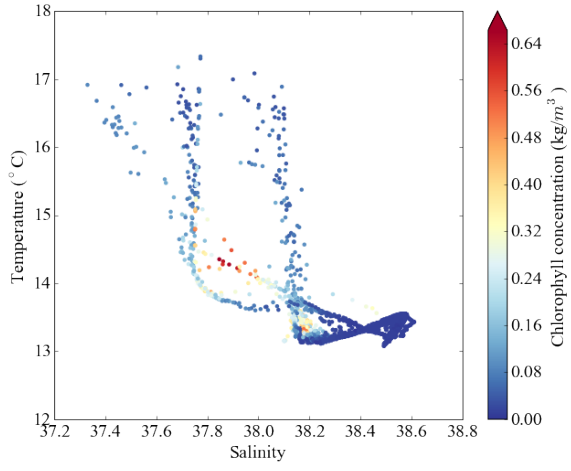
plot_line.ipynb: simple line plots and configuration

plot_subfigure.ipynb: organizing plots in sub-figures

plot_TS_diagram.ipynb: scatter plot using temperature and salinity from CTD

oceanography.mplstyle: file to define figure style





Exercise 2: mooring time series

Objective: plot time series of temperature and salinity

⚙: `2-plot_ibiza_temperature_salinity.ipynb`

Programming in Python #3

2D plots and plots on map

Useful packages

NumPy: package for scientific computing
http://www.numpy.org/

SciPy: open-source software for mathematics, science, and engineering http://www.scipy.org/

Useful packages

NumPy:	package for scientific computing http://www.numpy.org/
SciPy:	open-source software for mathematics, science, and engineering http://www.scipy.org/
matplotlib:	2D plotting library which produces publication quality figures http://matplotlib.org/

Useful packages

NumPy: package for scientific computing
http://www.numpy.org/

SciPy: open-source software for mathematics, science, and engineering http://www.scipy.org/

matplotlib: 2D plotting library which produces publication quality figures http://matplotlib.org/

Basemap toolkit : library for plotting 2D data on maps in Python
http://matplotlib.org/basemap/

Useful packages

NumPy: package for scientific computing
http://www.numpy.org/

SciPy: open-source software for mathematics, science, and engineering http://www.scipy.org/

matplotlib: 2D plotting library which produces publication quality figures http://matplotlib.org/

Basemap toolkit : library for plotting 2D data on maps in Python
http://matplotlib.org/basemap/

Bokeh: interactive visualization library
(http://bokeh.pydata.org/en/latest/)

List of notebooks

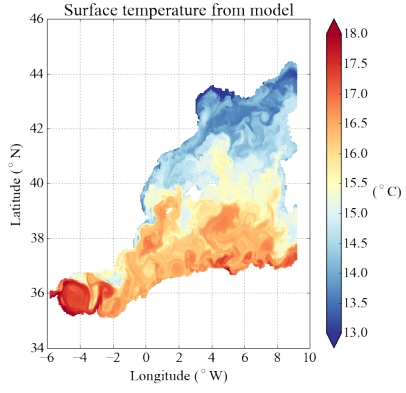
Located in `Plotting`

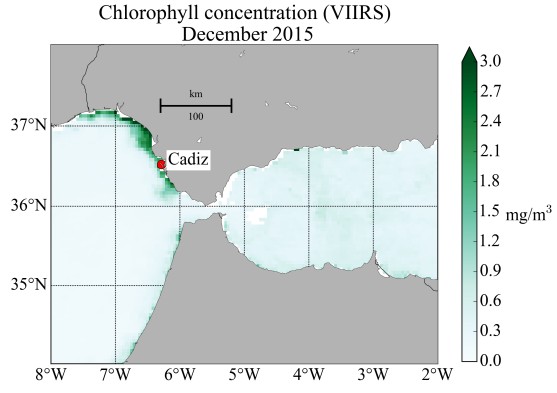
plot2D_contours_pcolor.ipynb: plot 2D fields with contour or pseudo-color techniques

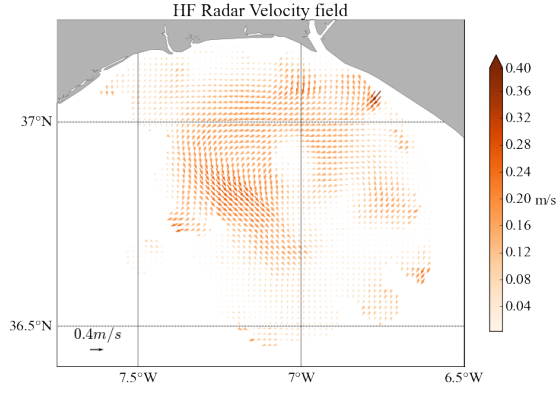
plot2D_map_field.ipynb: plot 2D field on a map

plot2D_map_vector.ipynb: plot vector field on a map

plot2D_map_scatter.ipynb: scatter plot on a map







Exercise 3: 2D plot

Objective: create a plot with the salinity from a numerical model and the velocities taken from a HF radar system.

⚙: `3-plot_radar_salinity.ipynb`