Introduction to Scientific Python

Application to Oceanography

C. Troupin (GHER), A. Capet (MAST)

University of Liège

Cádiz, 28-30 October 2019

ctroupin@uliege.be

Table of contents

Table of contents

Introduction:

What? Why? How?

What is Python?

Programming language (1st release: 1991):

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

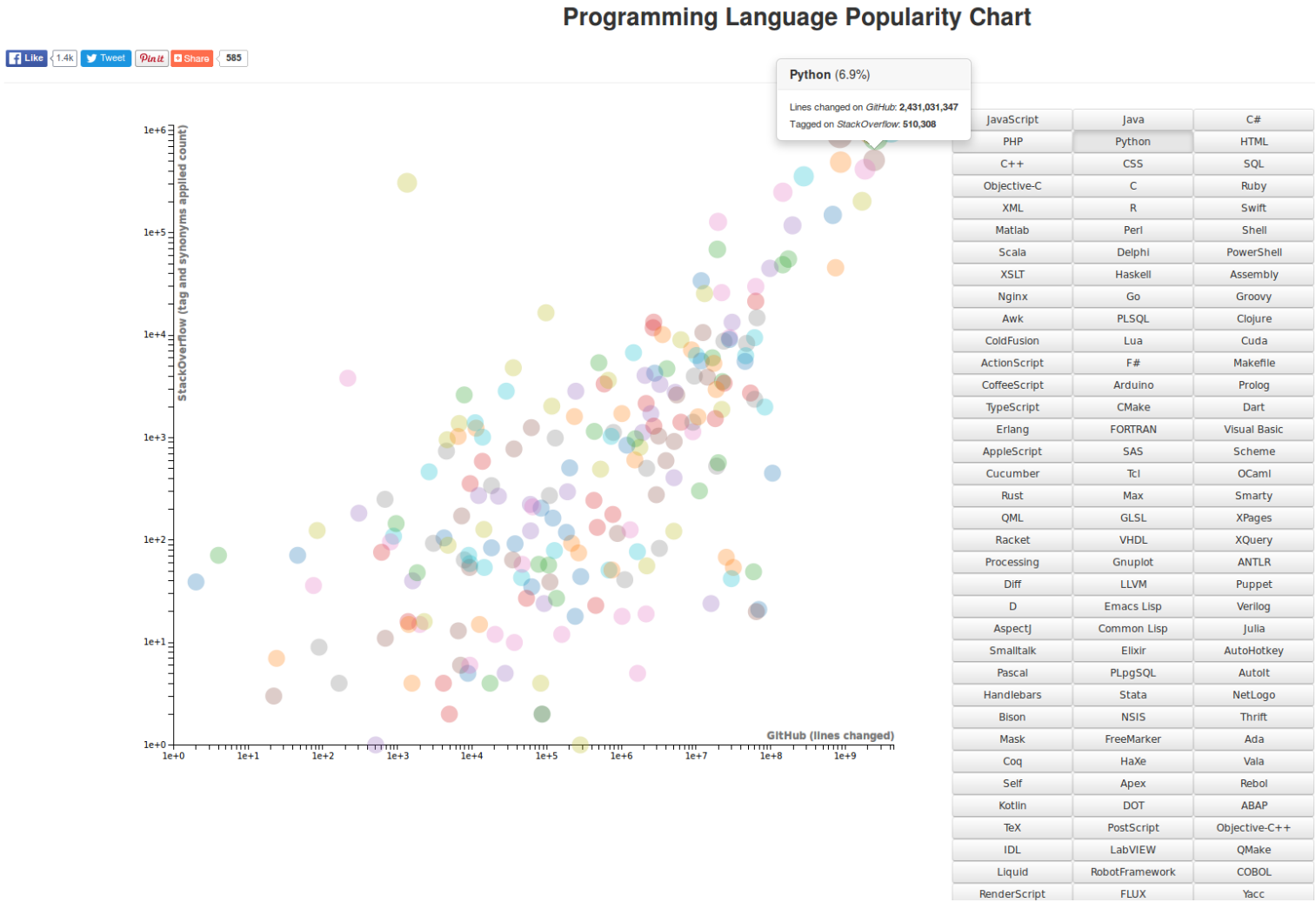
https://www.python.org

3 good reasons to use Python

1. Simple, easy to learn syntax
2. Multi-purpose language
3. Large user community,
   including Ocean Sciences

doc, support, packages

3 good reasons to use Python



Source: http://langpop.corger.nl/ (consulted in January 2016)

3 reasons not use Python?

1. Don’t have time to learn a new language
2. Slow? compared to C/C++, Fortran, Julia, …
3. Memory consumption

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://numpy.org/devdocs/user/numpy-for-matlab-users.html

Python 3 or Python 2

Python 3!!

Python 3 or Python 2

Some differences:

- Print function
- Integer division
- Unicode
- …

Python 3 or Python 2

”As of January 2020 Python 2 will be in EOL (End Of Life) status and receive no further official support.”

More details:

Python 2 or Python 3

Will Scientists Ever Move to Python 3? (January 2013)

Quick example: hello.py

Quick example: hello.py

```
#!/usr/bin/python
"""
This function prints "Hello world"
"""

def hello():
  print("Hello world")
  return

def main():
  hello()

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

Running your code: several solutions

Edit, then run in a shell:

```
$ python mycode.py
```

or

```
$ mycode.py
```

if *shebang* (#!) is present at the 1st line

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

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
```

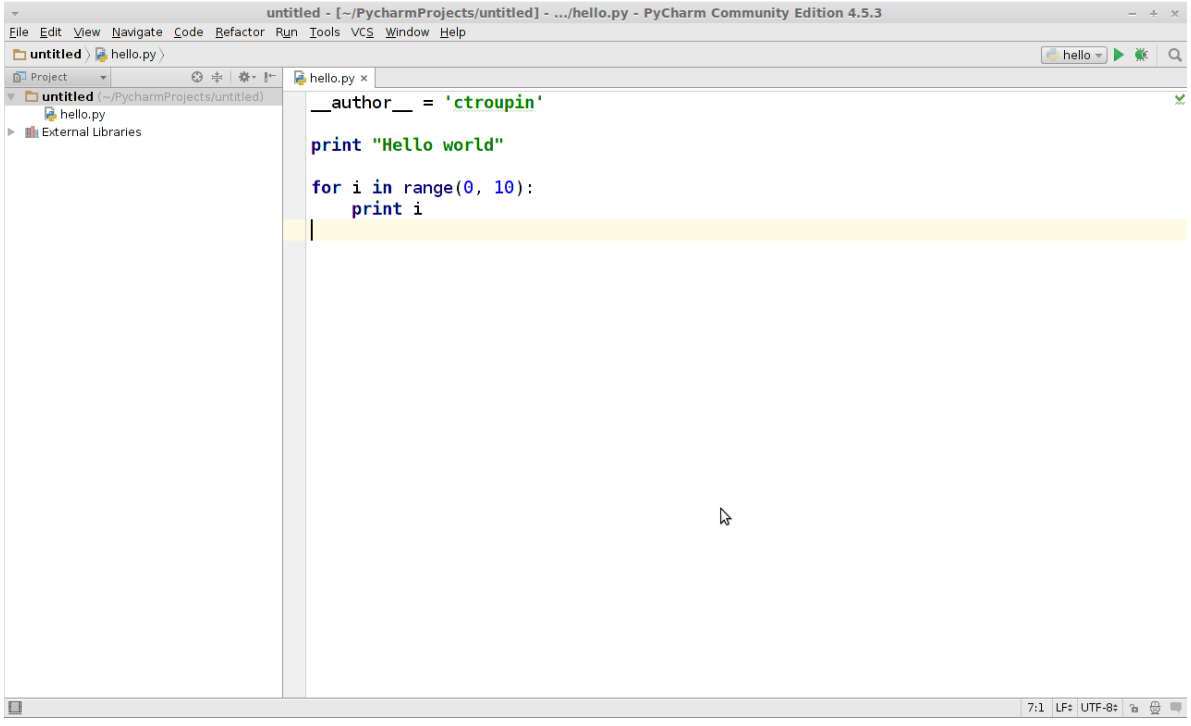
+ *magic* functions:

- `%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)

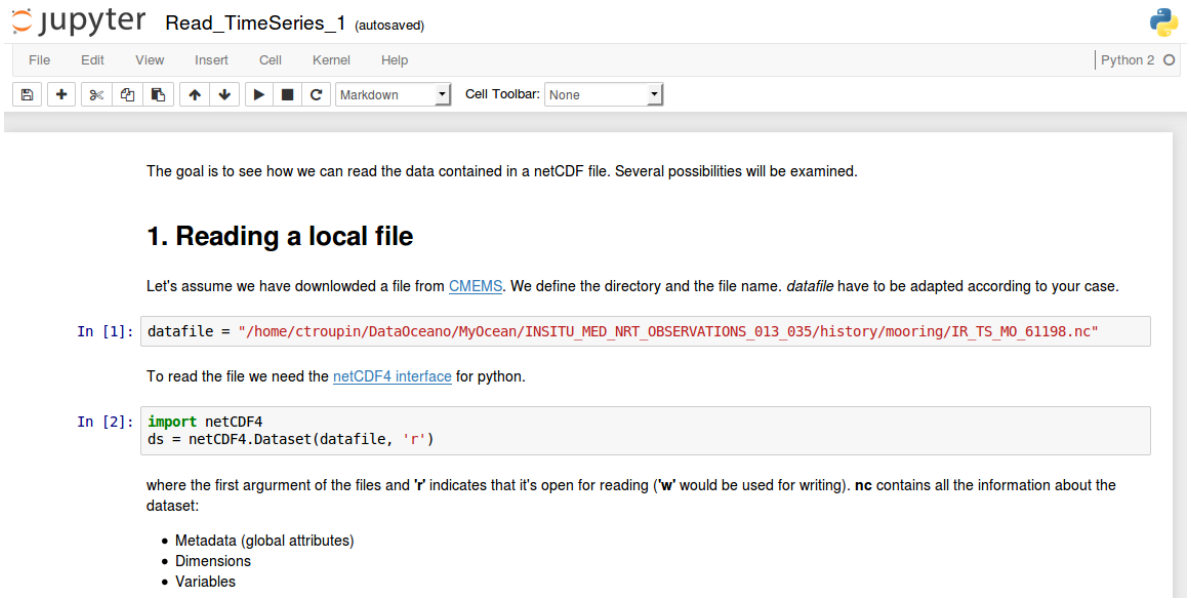


Examples: Atom, Eclipse, PyCharm, Idle, ...
Complete list: 🔗 https://wiki.python.org/moin/PythonEditors

Running your code: several solutions

Jupyter notebook

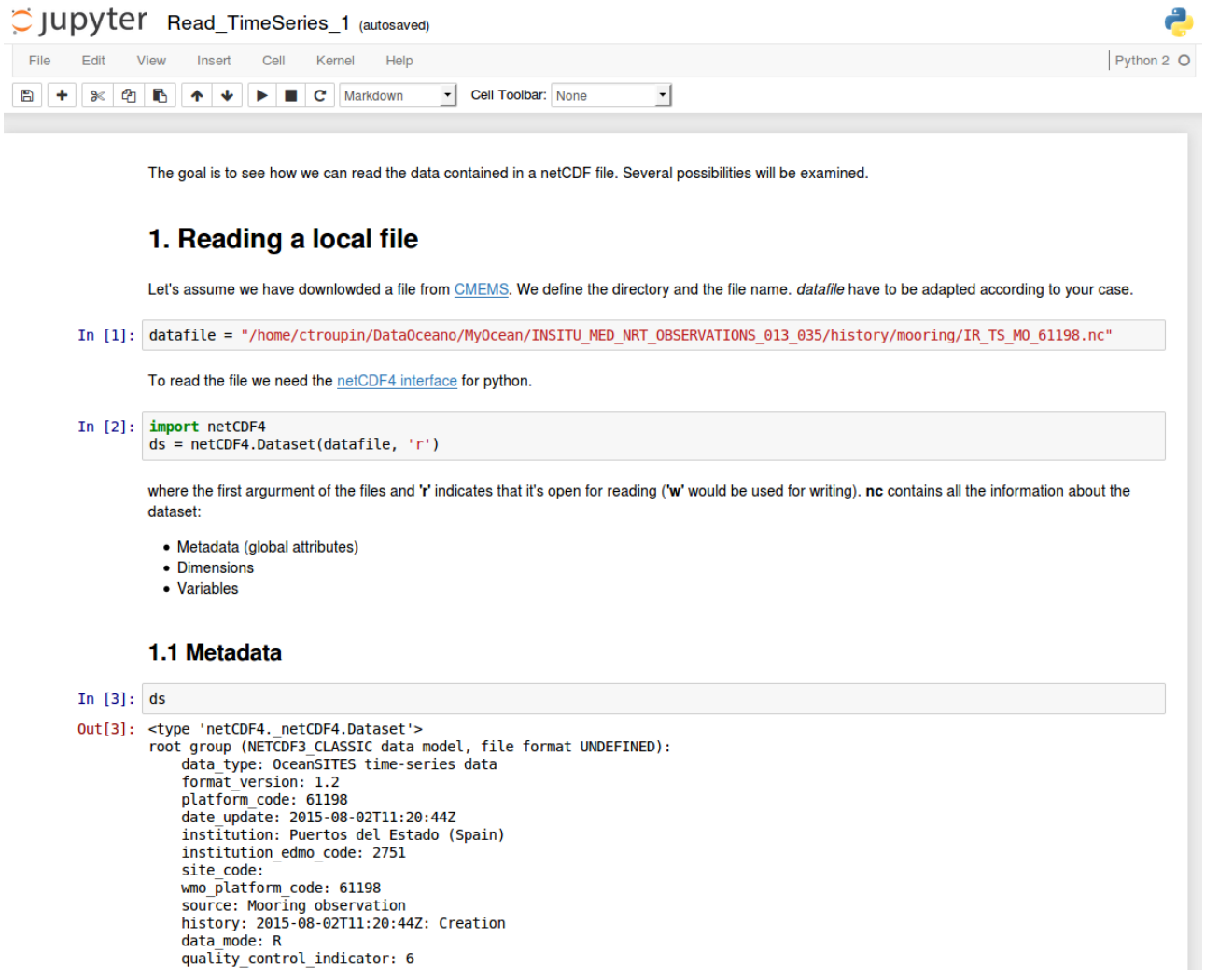
(interactive computational environment)



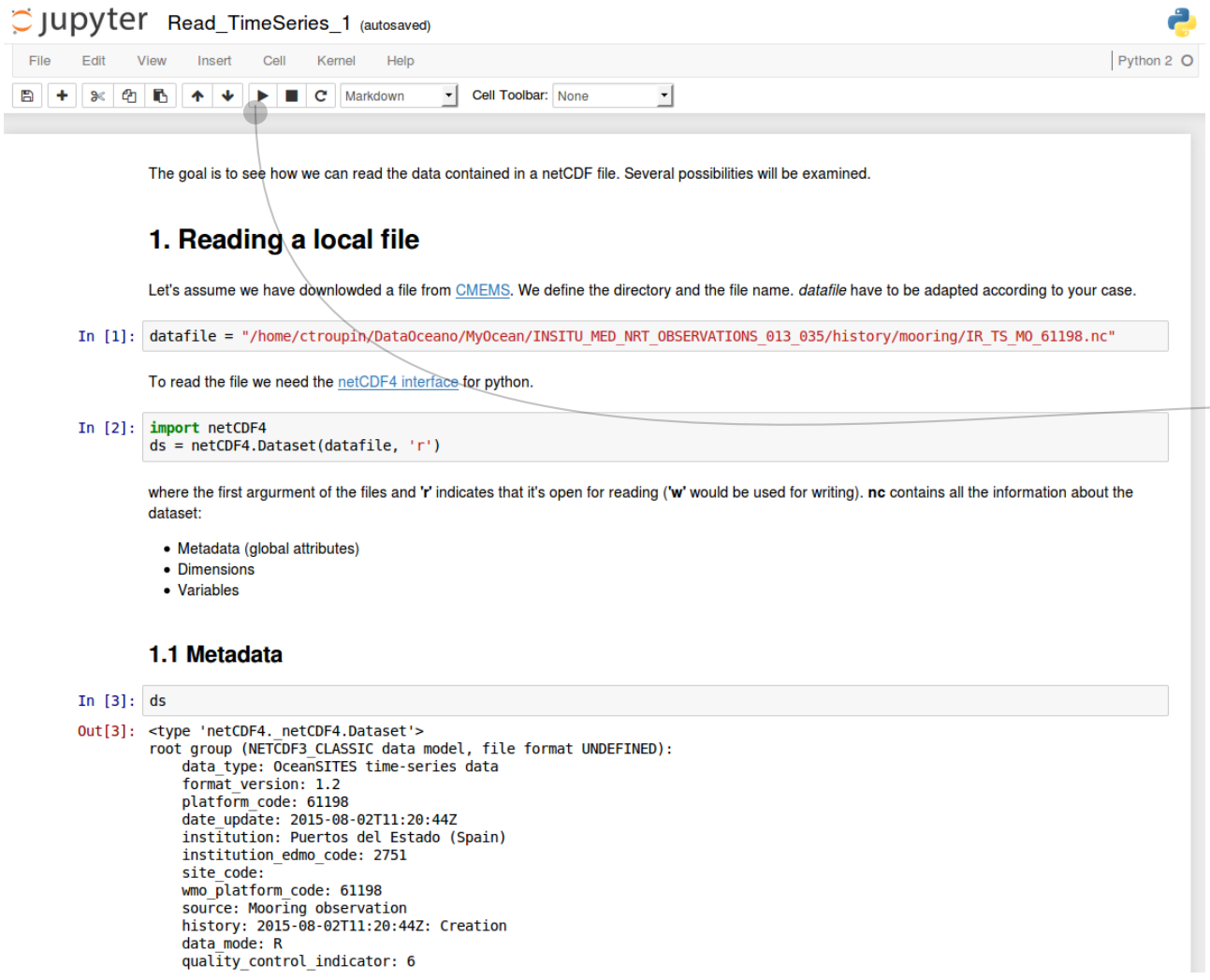
Rich text + command executions + figures + …

”*Data story telling*”

Structure of a notebook

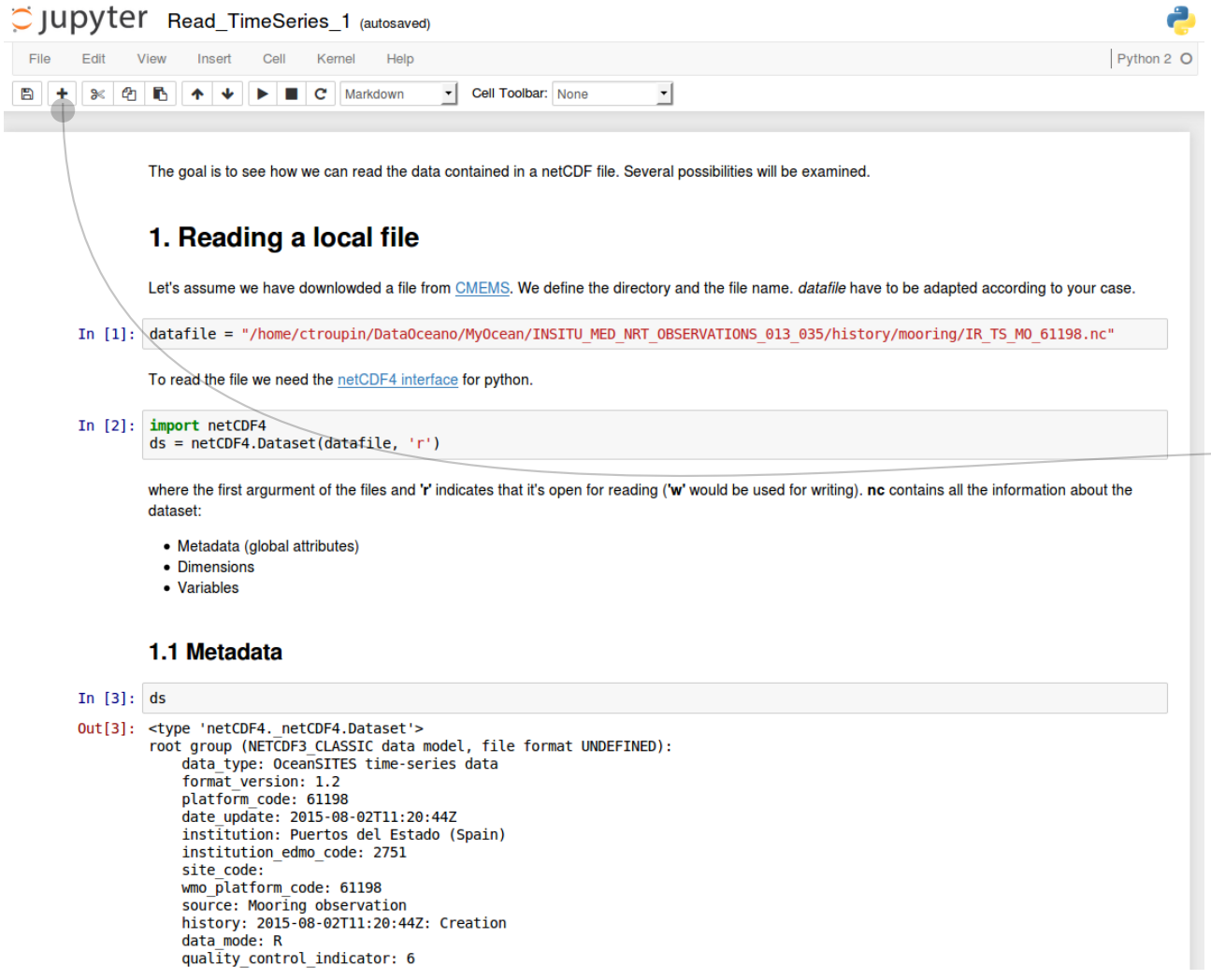


Structure of a notebook



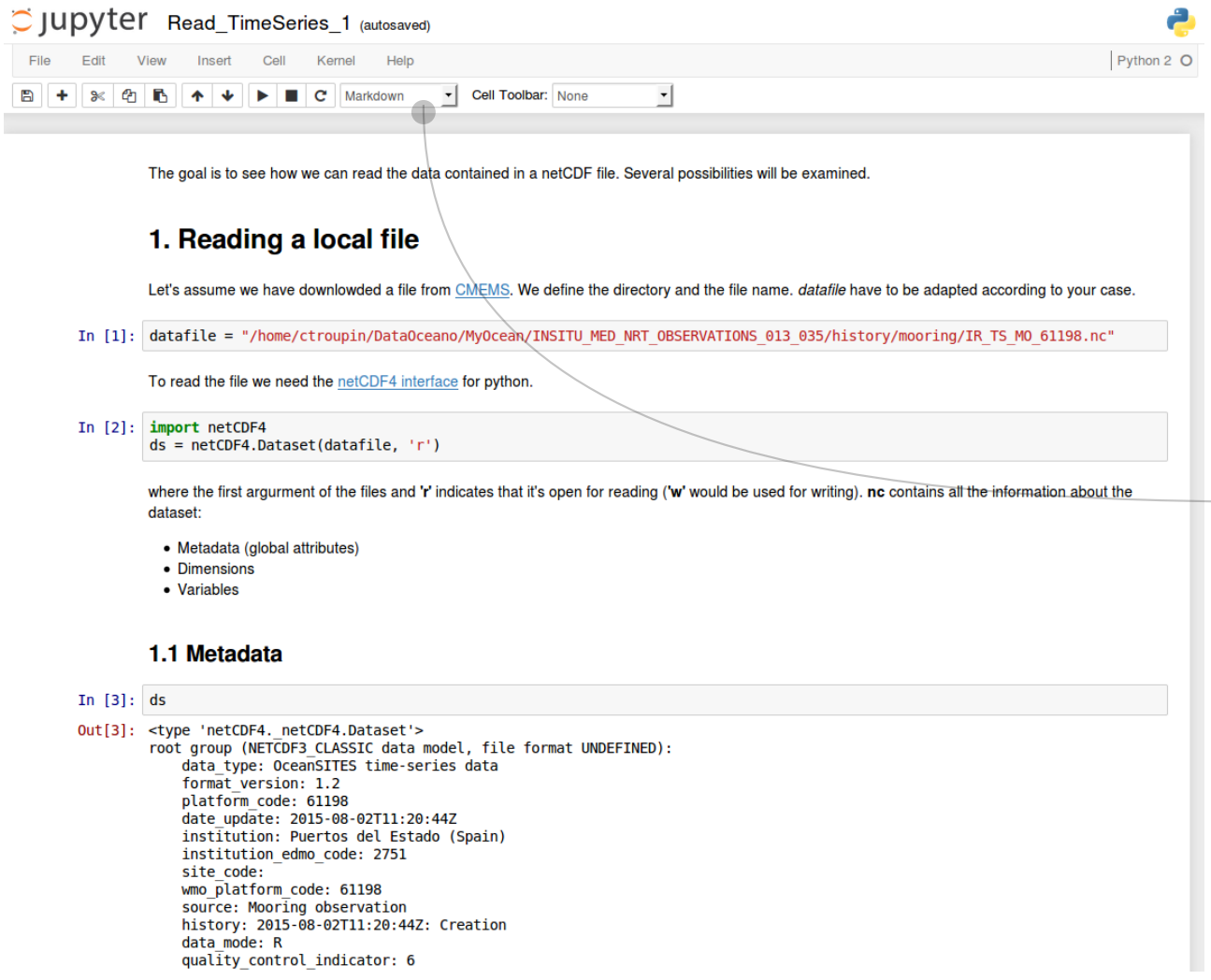
Run current cell

Structure of a notebook



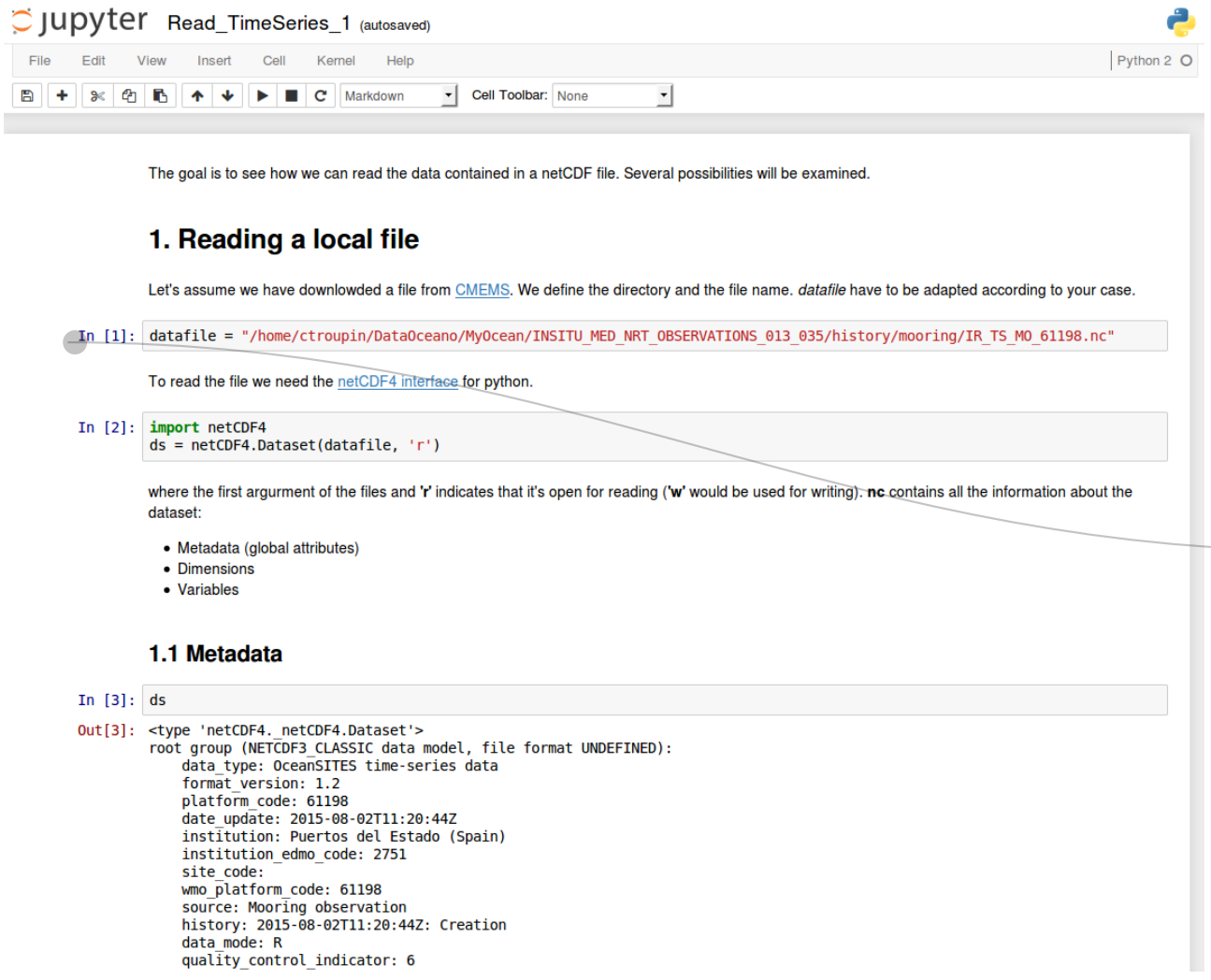
Run current cell
Add a new cell

Structure of a notebook



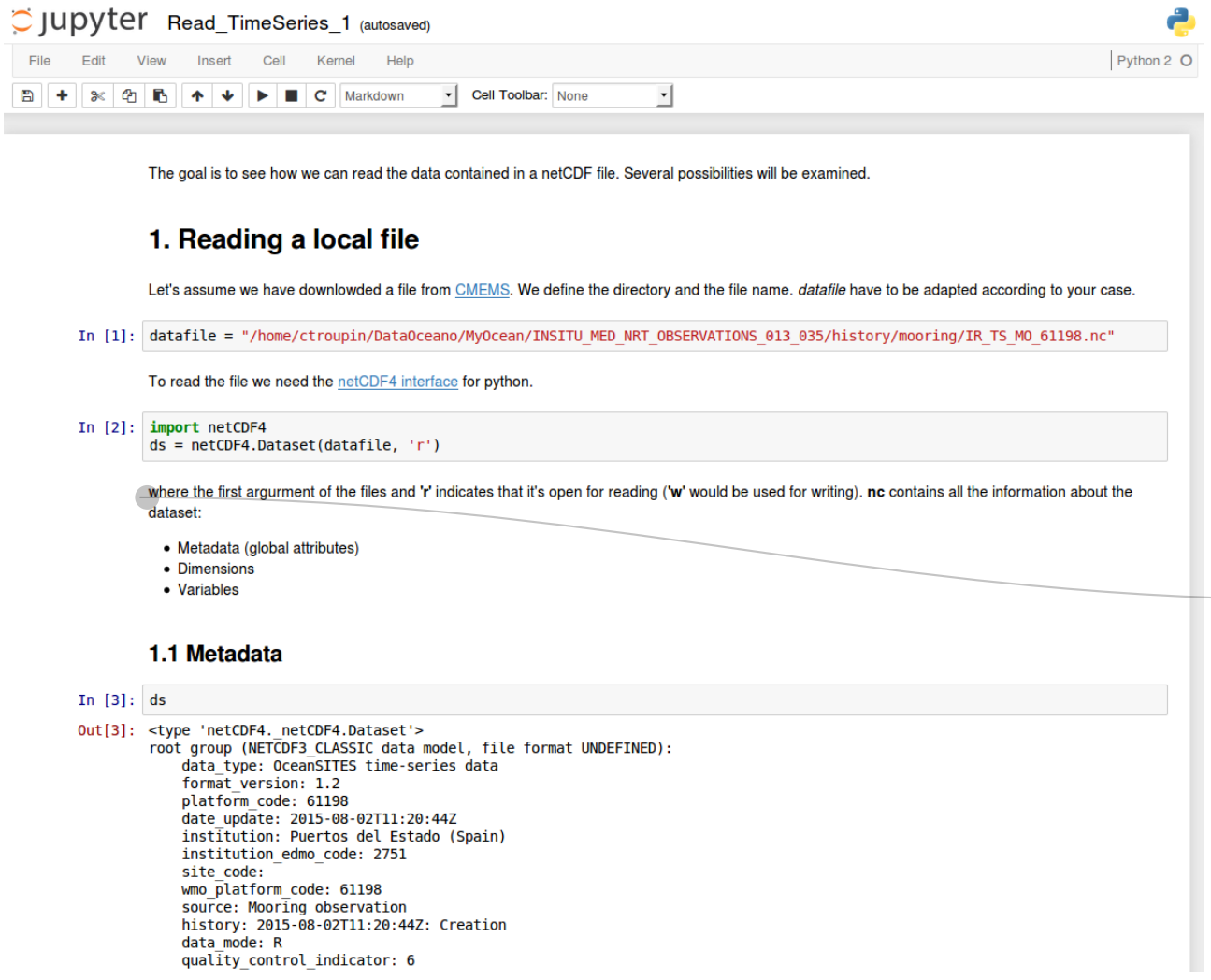
Run current cell
Add a new cell
Select type of cell

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

Some data types

Let's start to work!

🔗 Intro/data-types.ipynb

🔗 Intro/control_flow.ipynb

Resources

Web

- https://docs.python.org/3/tutorial/
- https://developers.google.com/edu/python/introduction?hl=en

  tutorial + exercises
- http://www.learnpython.org — online code execution
- https://pythonprogramming.net

Resources

Learning platforms

- Udacity: Introduction to Python Programming (5 weeks)
- Code Academy (25 hours)
- DataCamp

Resources

videos

- Python Beginner Tutorial (For Absolute Beginners) (4 parts)
- 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

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

Objectives of the course

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

What won't be done:
make you a good programmer

About the trainers

Intro/teacher-map.ipynb

Arthur Capet
Oceanographer
Post-doctoral researcher at MAST
4-year experience with Python

About the trainers

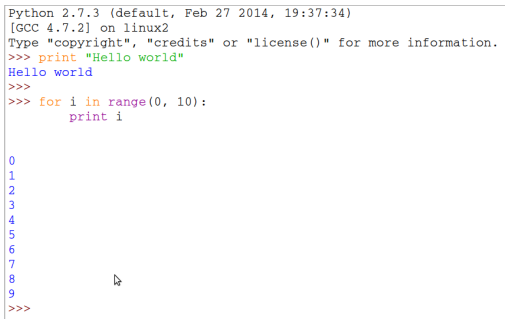
Intro/teacher-map.ipynb

Arthur Capet
Oceanographer
Post-doctoral researcher at MAST
4-year experience with Python

Charles Troupin
Engineer, oceanographer
Post-doctoral researcher at GHER
7-year experience with Python

Content of the course

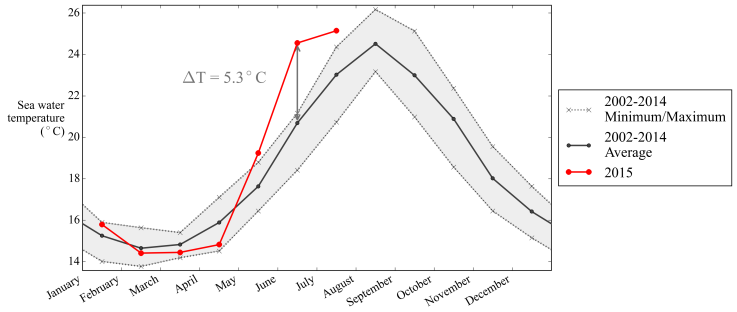
1. Reading/writing data files



Content of the course

1. Reading/writing data files
2. Working with time series

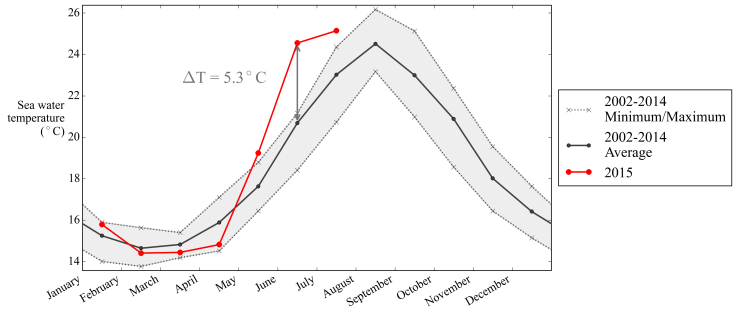
```
Python 2.7.3 (default, Feb 27 2014, 19:37:34)
[GCC 4.7.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello world"
Hello world
>>>
>>> for i in range(0, 10):
        print i


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

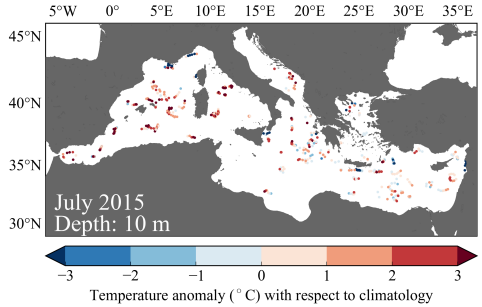


Content of the course

1. Reading/writing data files
2. Working with time series
3. Plotting 2-D fields

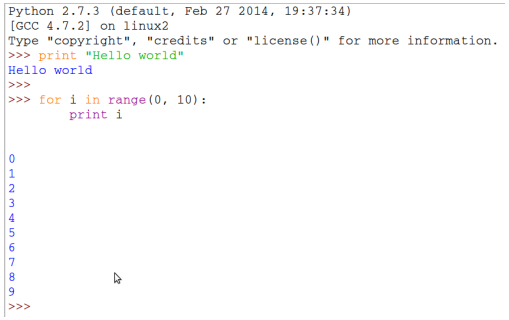
```
Python 2.7.3 (default, Feb 27 2014, 19:37:34)
[GCC 4.7.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello world"
Hello world
>>>
>>> for i in range(0, 10):
        print i


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

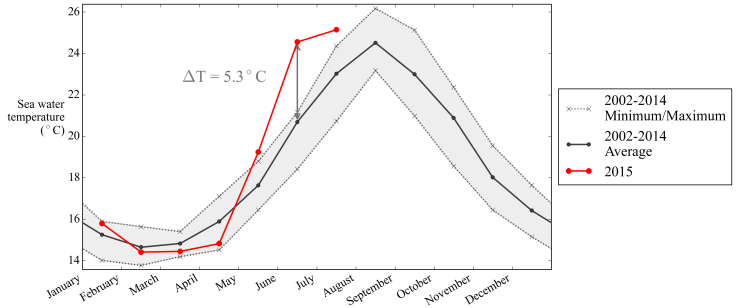


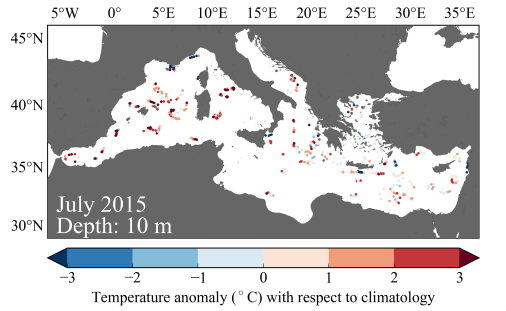


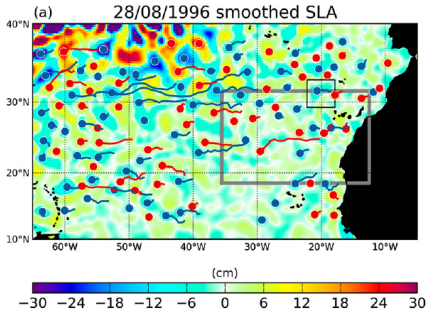
Content of the course

1. Reading/writing data files
2. Working with time series
3. Plotting 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:
  python3.x and python3.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:
  python3.x and python3.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 (https://www.scipy.org/install.html):
mathematics, science, and engineering

Installing modules

Example: SciPy (https://www.scipy.org/install.html):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution (ex: Anaconda)

Installing modules

Example: SciPy (https://www.scipy.org/install.html):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution (ex: Anaconda)

Intermediate: Linux, Mac: install package

- Linux: package manager
- Mac: MacPorts, Homebrew

Installing modules

Example: SciPy (https://www.scipy.org/install.html):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution (ex: Anaconda)

Intermediate: Linux, Mac: install package

- Linux: package manager
- Mac: MacPorts, Homebrew

Harder: build from source

```
python setup.py install
```

Installing modules

Example: SciPy (https://www.scipy.org/install.html):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution (ex: Anaconda)

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 (https://www.scipy.org/install.html):
mathematics, science, and engineering

Easy way: Windows, Linux, Mac:
use Scientific Python distribution (ex: Anaconda)

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 PackageName
```

- Specify exact version:

```
pip install PackageName==x.y.z
```

- Specify minimum version:

```
pip install 'PackageName>=x.y.z'
```

More about `pip`

- Uninstall packages:

```
pip uninstall PackageName
```

More about `pip`

- Uninstall packages:

```
pip uninstall PackageName
```

- List installed packages:

```
pip list
```

Output:

```
Package          Version
---------------- ---------
asn1crypto       0.24.0
Cartopy          0.17.0
certifi          2019.6.16
cffi             1.12.3
...
```

More about `pip`

- Uninstall packages:

```
pip uninstall PackageName
```

- List installed packages:

```
pip list
```

- List installed packages in requirements format:

```
pip freeze
```

Output:

```
asn1crypto==0.24.0
Cartopy==0.17.0
...
tornado==6.0.3
urllib3==1.25.3
```

More about `pip`

- Show information about installed packages:

```
pip show PackageName
```

```
ctroupin@GHER-ULg-Laptop ~ $ pip show numpy
Name: numpy
Version: 1.17.1
Summary: NumPy is the fundamental package for array computing with
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email: None
License: BSD
Location: /home/ctroupin/miniconda3/lib/python3.7/site-packages
Requires:
Required-by: scipy, pykdtree, matplotlib, Cartopy
```

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

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 ^^I# It will return the header, as a dictionary.
profile.keys()^^I^^I# It will list the available variables.
profile['TEMP']   ^^I# 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
```

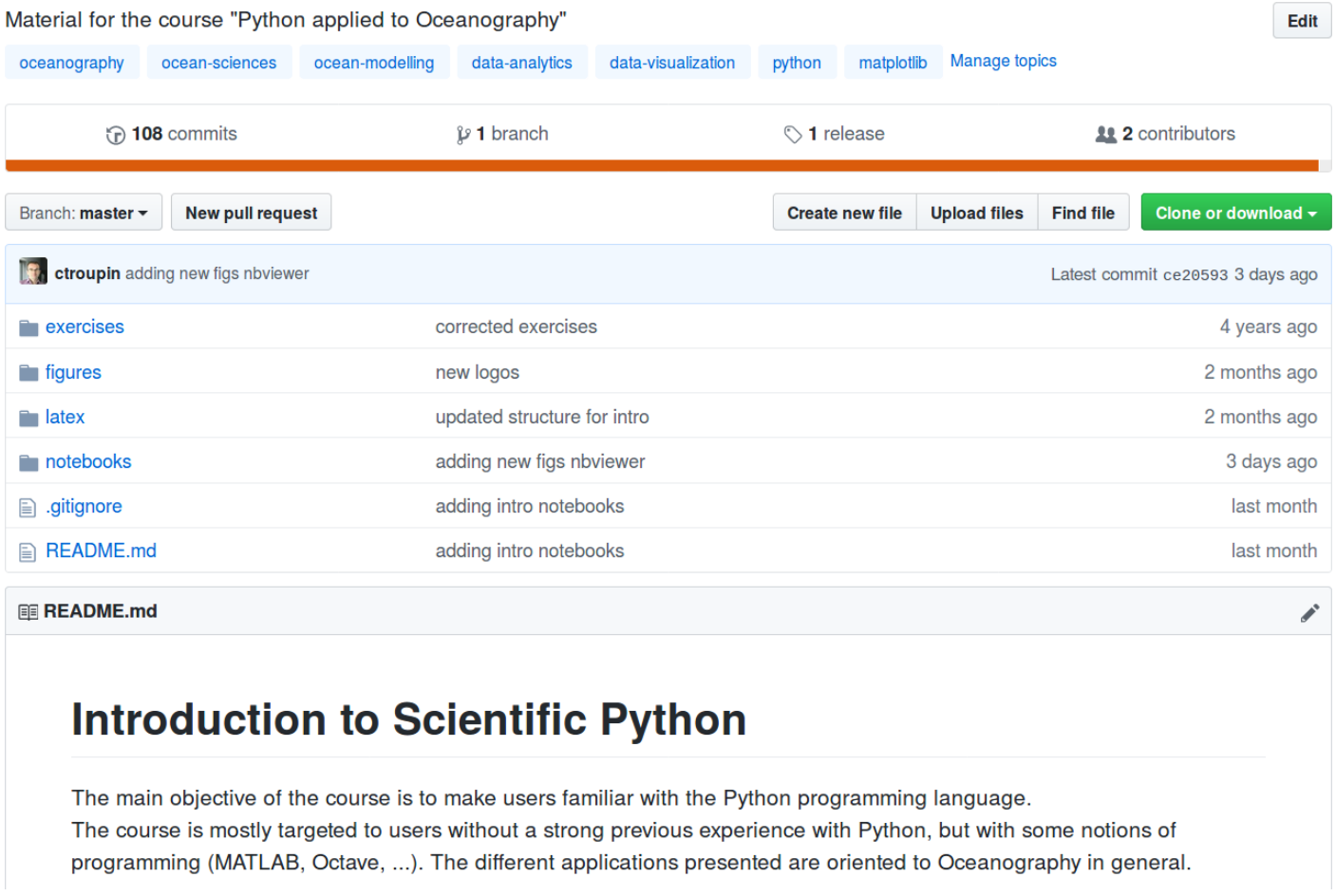
- From the doc:

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

: `read_cnv_file.ipynb`

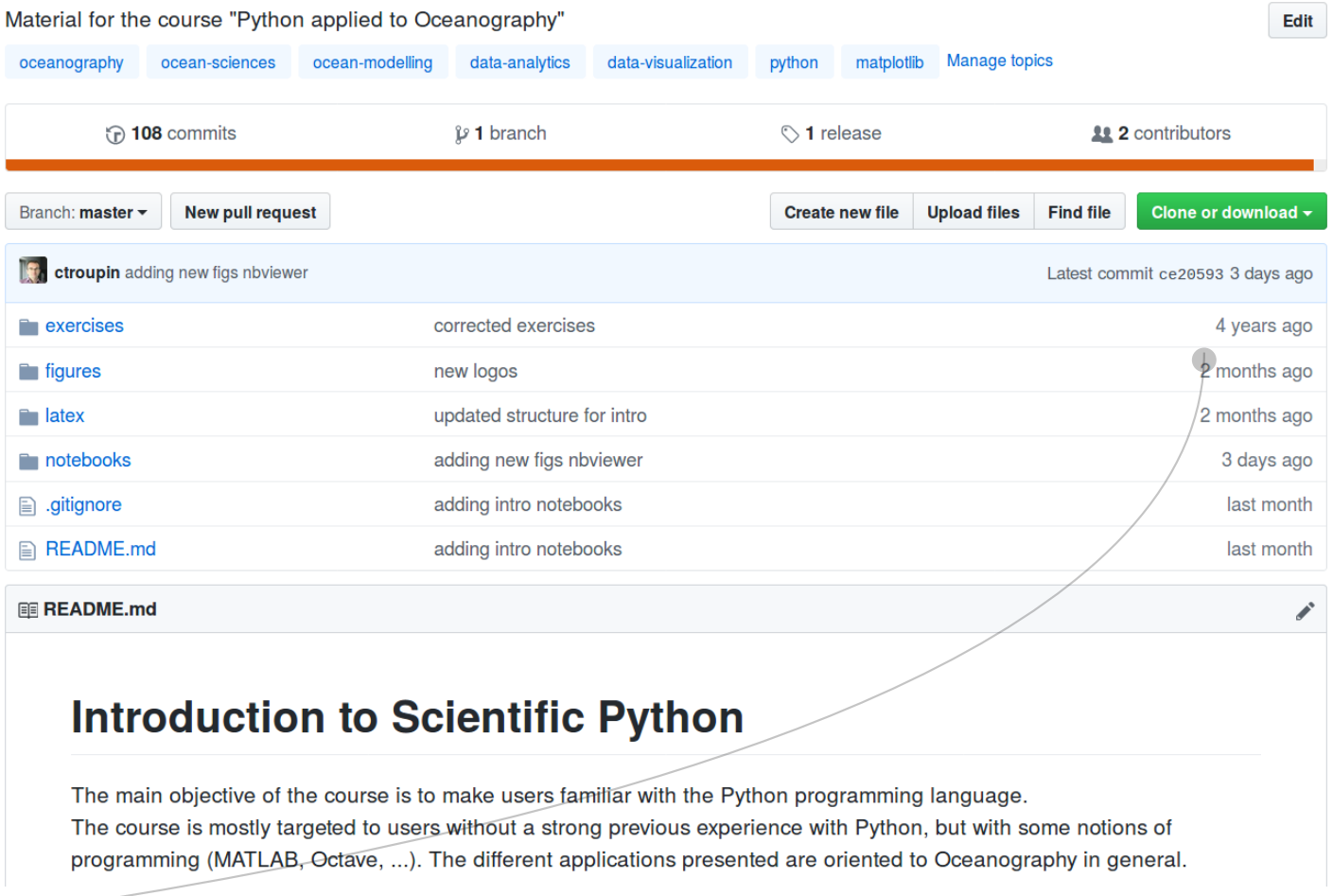
How to get code from github

https://github.com/ctroupin/PythonCourseCadiz2016



How to get code from github

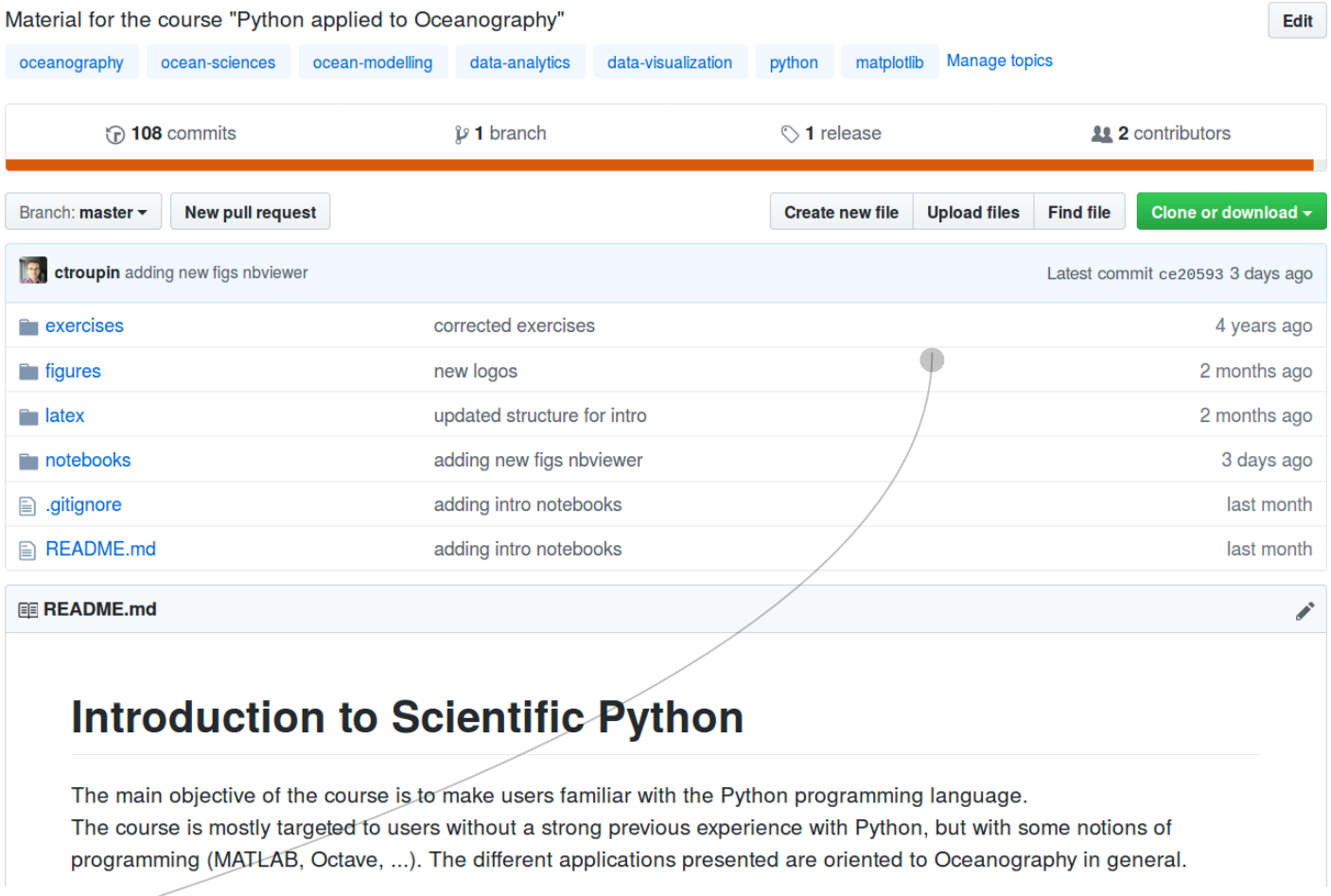
https://github.com/ctroupin/PythonCourseCadiz2016



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. ⏳3 (if there were modifications in the repository)

Update the code

1. Clone the repository

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

2. ⏳3 (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. ⏳3 (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:
^^Itime = 1 ;
^^Ilat = 1580 ;
^^Ilon = 3308 ;
variables:
^^Iint time(time) ;
^^I^^Itime:long_name = "reference time" ;
^^I^^Itime:standard_name = "time" ;
^^I^^Itime:axis = "T" ;
^^I^^Itime:calendar = "Gregorian" ;
^^I^^Itime:units = "seconds since 1981-01-01 00:00:00" ;
...
^^I^^I^^I"SUBSAMP=1\n",
^^I^^I^^I"OUTMODE=0\n",
^^I^^I^^I"" ;
}
```

Ferret

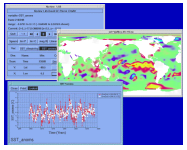
http://www.ferret.noaa.gov/Ferret/

visualization and analysis environment

Ferret to get basic info on file

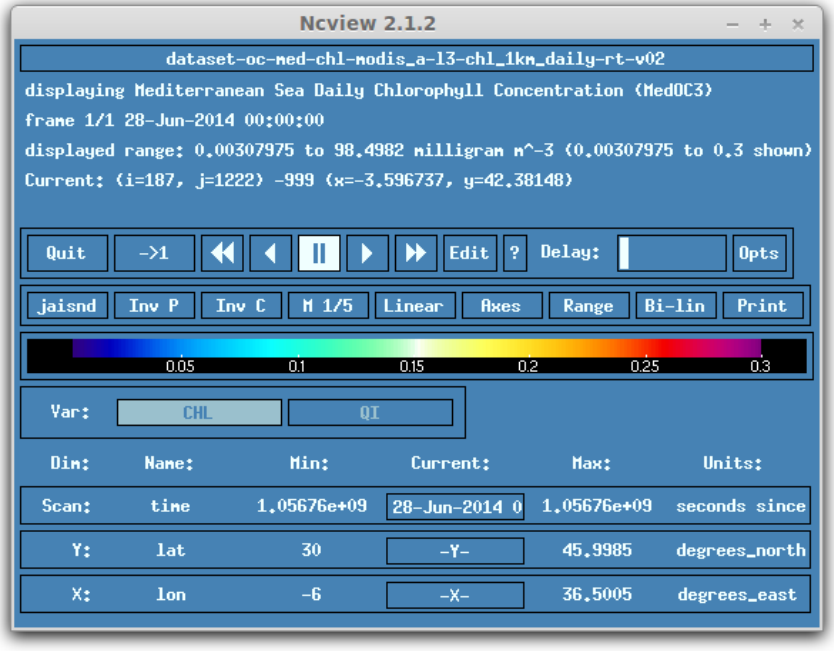
```
ctroupin@SCBD046 ~/Desktop $ ferret_c
 ^^INOAA/PMEL TMAP
 ^^IFERRET v6.62
 ^^ILinux(gfortran) 2.6.9—89.0.20.ELsmp — 07/06/13
 ^^I25—Nov—15 12:23

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?
```

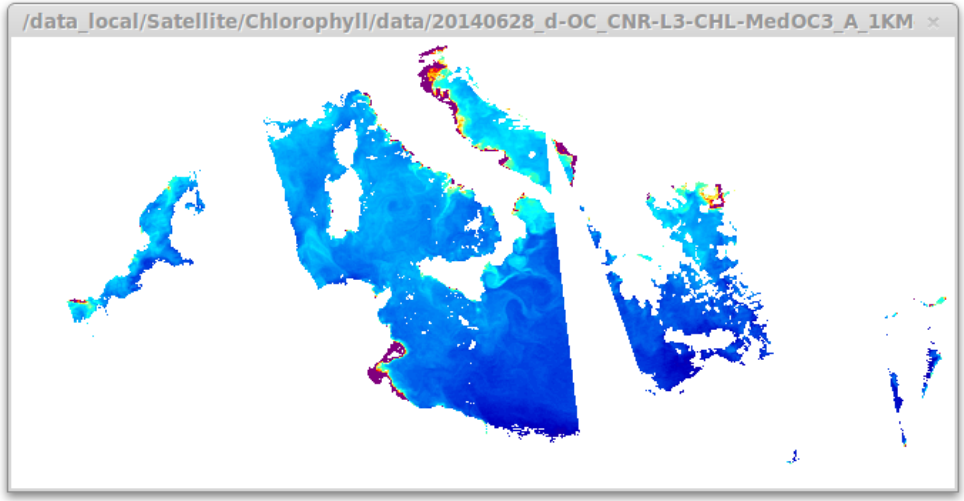
ncview



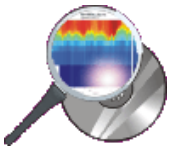
http://meteora.ucsd.edu/~pierce/ncview_home_page.html

quick visualisation of 3-4D fields



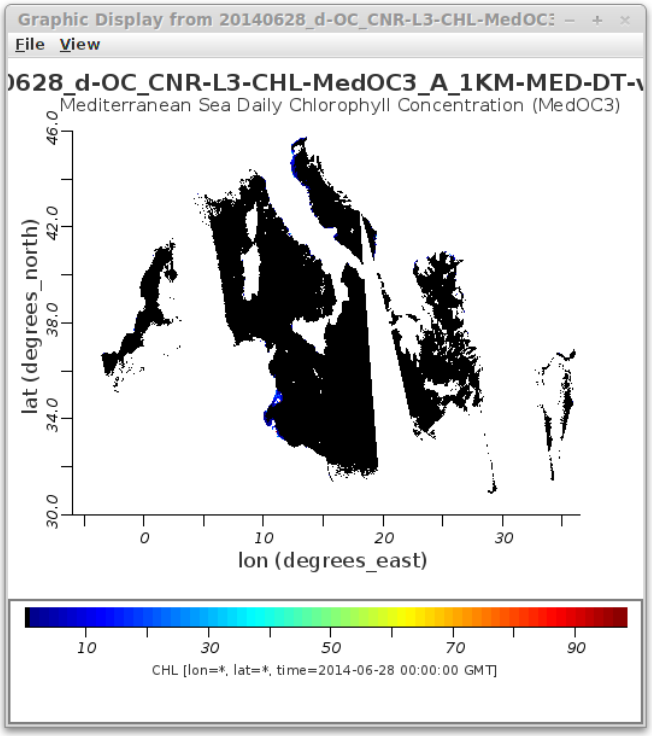


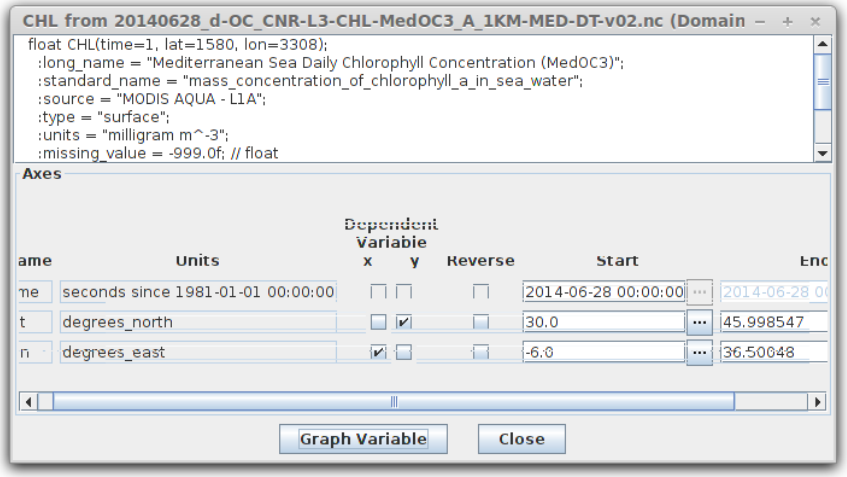
ncbrowse



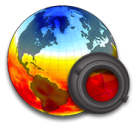
http://www.epic.noaa.gov/java/ncBrowse/

interactive graphical display



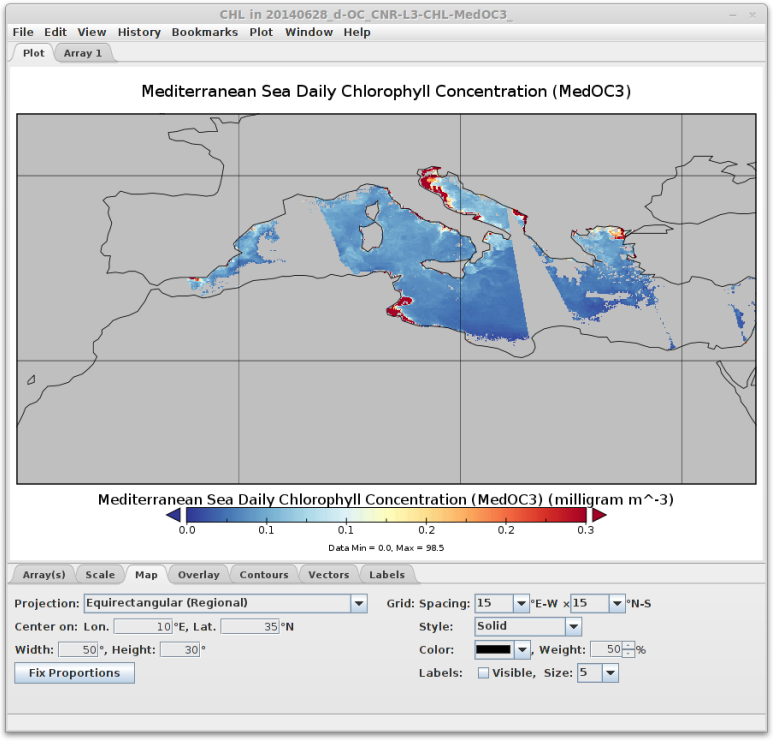


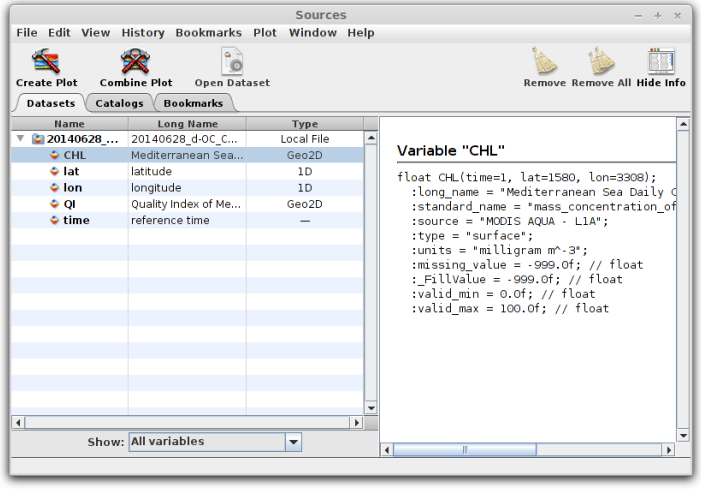
Panoply



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
```

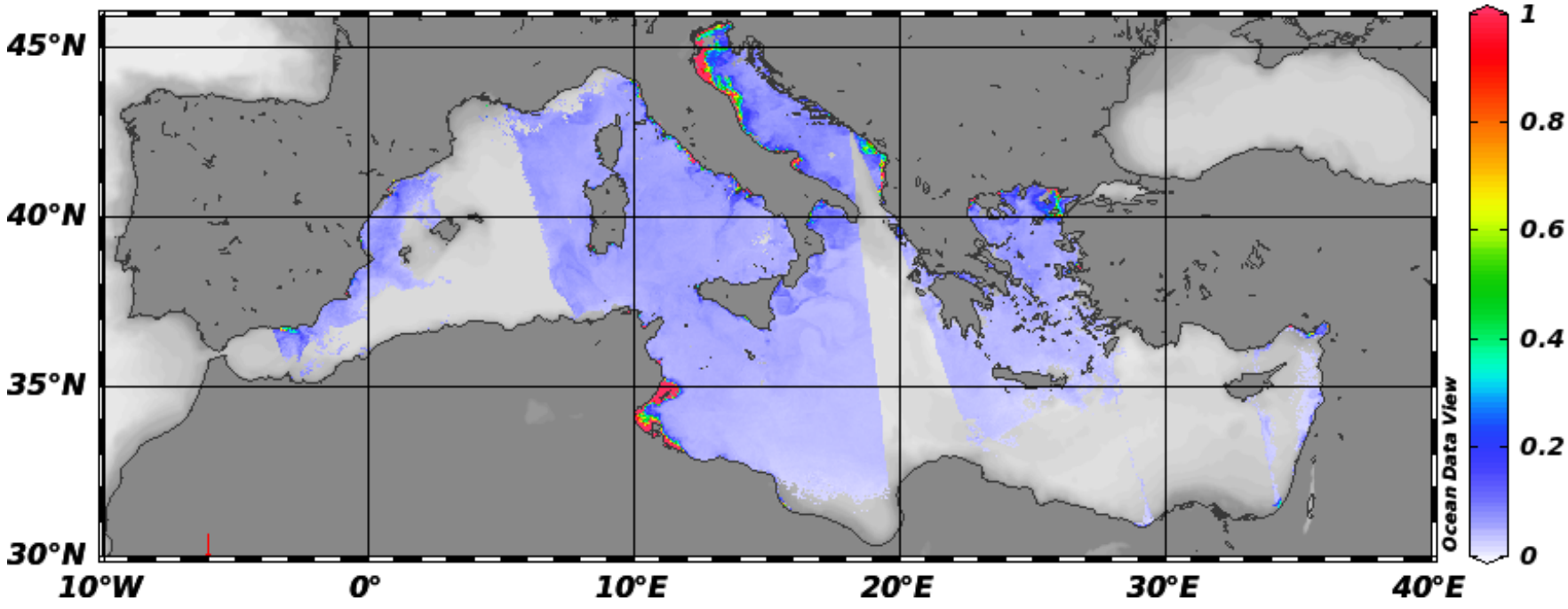
- Extract subregion:

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

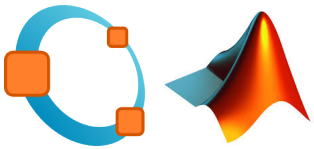
ODV – Ocean Data View

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

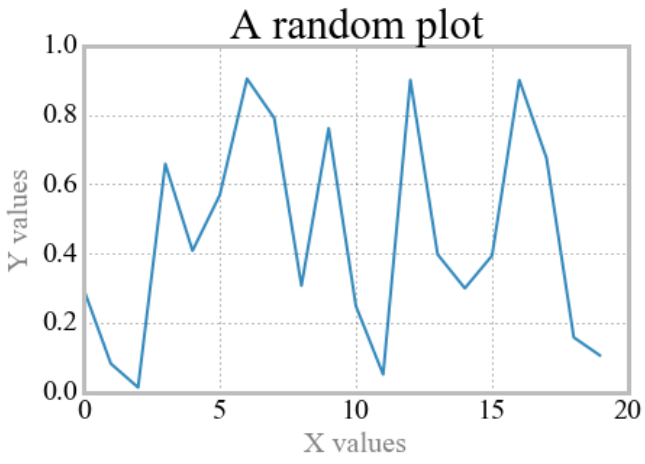
Located in `Plotting`

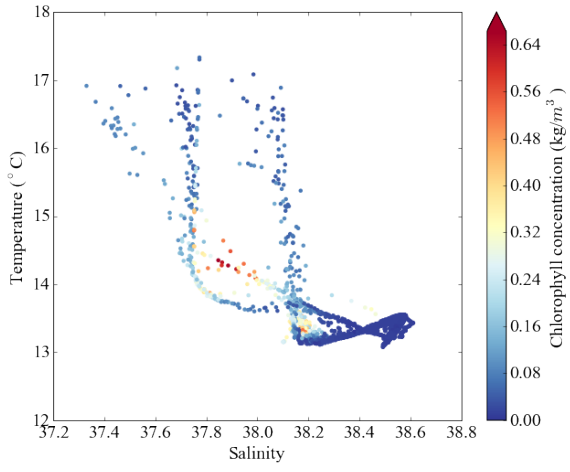
plot_line.ipynb: simple line plots and configuration

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

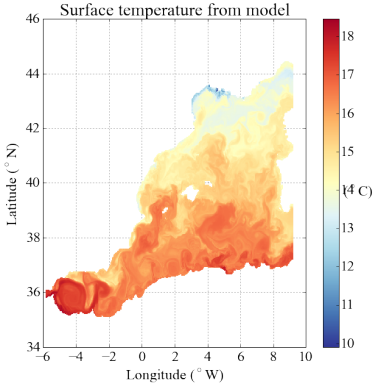
Located in `Plotting`

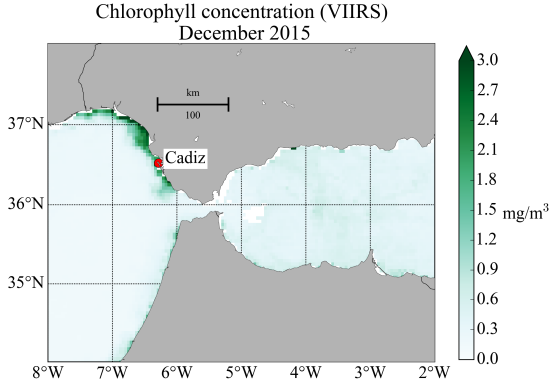
plot2D_contours_pcolor.ipynb: plot 2D fields with contour or pseudo-color techniques

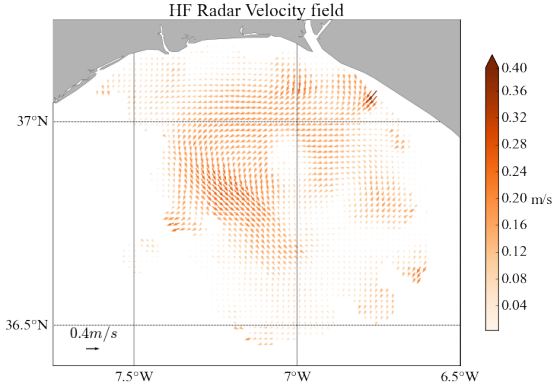
plot2D_map_field.ipynb: plot 2D field on a map

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`