STATE OF THE IOOS TOOLS

Filipe Fernandes

May 1, 2018

LONGER TITLE

The State of IOOS **(and some non-IOOS)** **(Python)** tools **(and its 105 repositories!?)**

whoami

ocefpaf

`whoami`

ocefpaf

- Physical Oceanographer

`whoami`

ocefpaf

- Physical Oceanographer
- Data Plumber

`whoami`

ocefpaf

- Physical Oceanographer
- Data Plumber
- Code Janitor

`whoami`

ocefpaf

- Physical Oceanographer
- Data Plumber
- Code Janitor
- CI babysitter

`whoami`

ocefpaf

- Physical Oceanographer
- Data Plumber
- Code Janitor
- CI babysitter
- Amazon-Dash-Button for conda-forge packaging

`whoami`

ocefpaf

- Physical Oceanographer
- Data Plumber
- Code Janitor
- CI babysitter
- Amazon-Dash-Button for conda-forge packaging

I'll probably have to add "GH Marie Kondo" after this presentation.

SOME BACKGROUND

SOME BACKGROUND

- year 01: skill score (SECOORA), system-test notebooks, packaging (ioos channel).

SOME BACKGROUND

- year 01: skill score (SECOORA), system-test notebooks, packaging (ioos channel).
- year 02: python 3 updates, CIs maintenance, packaging (conda-forge channel).

SOME BACKGROUND

- year 01: skill score (SECOORA), system-test notebooks, packaging (ioos channel).
- year 02: python 3 updates, CIs maintenance, packaging (conda-forge channel).
- year 03: CIs maintenance, Data Demo Center, some new tools, packaging.

SOME BACKGROUND

- year 01: skill score (SECOORA), system-test notebooks, packaging (ioos channel).
- year 02: python 3 updates, CIs maintenance, packaging (conda-forge channel).
- year 03: CIs maintenance, Data Demo Center, some new tools, packaging.
- year 04: CIs maintenance, some Data Demo Center, less new tools, packaging.

SOME BACKGROUND

- year 01: skill score (SECOORA), system-test notebooks, packaging (ioos channel).
- year 02: python 3 updates, CIs maintenance, packaging (conda-forge channel).
- year 03: CIs maintenance, Data Demo Center, some new tools, packaging.
- year 04: CIs maintenance, some Data Demo Center, less new tools, packaging.
- year 05: CIs maintenance, Data Demo Center(?), new tools(?), packaging.

GOALS

GOALS

- Streamline CI maintenance, e.g.: compliance-checker plugins.

GOALS

- Streamline CI maintenance, e.g.: compliance-checker plugins.
- Focus on the essential tools, deprecate what is not used.

GOALS

- Streamline CI maintenance, e.g.: compliance-checker plugins.
- Focus on the essential tools, deprecate what is not used.
- Reduce the overall maintenance burden.

GOALS

- Streamline CI maintenance, e.g.: compliance-checker plugins.
- Focus on the essential tools, deprecate what is not used.
- Reduce the overall maintenance burden.
- What tools are ready for Python 2.7 EOL?

GOALS

- Streamline CI maintenance, e.g.: compliance-checker plugins.
- Focus on the essential tools, deprecate what is not used.
- Reduce the overall maintenance burden.
- What tools are ready for Python 2.7 EOL?

Spoiler: our tools are 100% py3k compatible!

GOALS (CONTINUED)

GOALS (CONTINUED)

- Identify gaps in our tools and/or if we need new ones to deal with new challenges (bio-obis-taxa? EML checker?).

GOALS (CONTINUED)

- Identify gaps in our tools and/or if we need new ones to deal with new challenges (bio-obis-taxa? EML checker?).
- Increase the “bus” factor and lower the barrier for newcomers.

GOALS (CONTINUED)

- Identify gaps in our tools and/or if we need new ones to deal with new challenges (bio-obis-taxa? EML checker?).
- Increase the “bus” factor and lower the barrier for newcomers.
- Create a policy for releases, *sdist* publication, and packaging.

GOALS (CONTINUED)

- Identify gaps in our tools and/or if we need new ones to deal with new challenges (bio-obis-taxa? EML checker?).
- Increase the “bus” factor and lower the barrier for newcomers.
- Create a policy for releases, *sdist* publication, and packaging.

Secret motivation: make year 06 all about new tools and the Data Demo Center!

HOW?

HOW?

- Finding the active and inactive projects and deprecating the latter.

HOW?

- Finding the active and inactive projects and deprecating the latter.
- Consolidate similar tools.

HOW?

- Finding the active and inactive projects and deprecating the latter.
- Consolidate similar tools.
- Adding auto-PyPI *sdist* publication.

HOW?

- Finding the active and inactive projects and deprecating the latter.
- Consolidate similar tools.
- Adding auto-PyPI *sdist* publication.
- Adopt a Release Early Release Often (RERO) policy.

HOW?

- Finding the active and inactive projects and deprecating the latter.
- Consolidate similar tools.
- Adding auto-PyPI *sdist* publication.
- Adopt a Release Early Release Often (RERO) policy.
- Write documentation of GH good practices.

IOOS TOOLS “HEALTH METRIC”

https://bit.ly/2019-DMAC

IOOS TOOLS “HEALTH METRIC”

https://bit.ly/2019-DMAC

- The data was collected on April 27th 2019 (4 days ago).

IOOS TOOLS “HEALTH METRIC”

https://bit.ly/2019-DMAC

- The data was collected on April 27th 2019 (4 days ago).
- The metric is based on: last commit, last release, number contributors, and py3k testing

IOOS TOOLS

software	last commit	last release	contributors	py3k testing
compliance-checker	2019-04-24	2019-02-27	28	py37
erddapy	2019-04-21	2019-03-06	3	py37
cc-plugin-glider	2019-02-20	2019-02-20	8	py36
cc-checker-ugrid	2019-01-09	2019-01-09	5	py36
pyoos	2019-02-24	2017-03-30	11	py35
ciso	2019-02-07	2019-02-07	2	py37
cc-plugin-ncei	2019-01-16	2017-10-17	4	py36
sensorml2iso	2018-09-12	2018-08-22	6	py36
odvc	2019-04-27	2018-03-02	3	py37
thredds_crawler	2018-03-16	2018-03-16	5	py36
petulant-bear	2016-02-03	2016-02-03	6	py35
wicken	2016-02-03	2016-02-03	5	py35
qartod	2016-14-14	NA	4	py35
cc-plugin-sgrid	2016-02-04	NA	1	py35

OTHER TOOLS (PYOCEANS)

- gridgeo
- ioos_tools
- pocean-core
- ~~erddapy~~

OTHER TOOLS (PYOCEANS)

- `gridgeo`
- `ioos_tools`
- `pocean-core`
- ~~`erddapy`~~

There are more tools in the pyoceans org. I only listed those that I know are used by IOOS in some places.

OTHER TOOLS (ASA-RPS)

WARNING: This list is not comprehensive! Also, we are not expecting any action from ASA-RPS! The goal is to identify what tools here are useful to the IOOS community!

OTHER TOOLS (ASA-RPS) LIST

- `qartod` (not a rare pókemon)
- `paegan`
- `paegan-viz`
- `paegan-transport`
- `sci-wms`
- `thredds_crawler_matlab`
- `udunitspy` (`compliace-checker` adopted `cf-units` instead)

OTHER TOOLS (AXIOM)

WARNING: This list is not comprehensive! Also, we are not expecting any action from Axiom! The goal is to identify what tools here are useful to the IOOS community!

OTHER TOOLS (AXIOM) LIST

- `pyncml`
- `epic2cf`
- `wera2netcdf`
- `codar2netcdf`
- `modflow2netcdf`
- `gutils` (lives in the SECOORA organization)

OTHER TOOLS (AXIOM) CONTINUATION

- `pygc`
- `pngpack`
- `pyaxiom` (predecessor of `pocean-core`)
- `sci-wms` (déjà vu)
- `ioos_qc` (`qartod` pókemon evolved form)

OTHER-OTHER TOOLS

- MetOffice stack: `iris`, `cartopy`, `nc-time-axis`, and `cf-units`
- PyViz: `bokeh`, `panel`, `hvplot`
- `matplotlib`
- `windrose`
- `folium`
- `geopandas`
- `gsw`
- `utide`

OTHER-OTHER TOOLS (CONTINUATION)

- nco
- netcdf4
- pysgrid
- pyugrid
- gridded
- xarray
- bagit

OTHER-OTHER TOOLS (CONTINUATION)

- `nco`
- `netcdf4`
- `pysgrid`
- `pyugrid`
- `gridded`
- `xarray`
- `bagit`

Feel free to add more in the hackpad.

SMALL ASIDE: BEST PRACTICES

SMALL ASIDE: BEST PRACTICES

- Always have a README file.

SMALL ASIDE: BEST PRACTICES

- Always have a README file.
- Always publish on PyPI (conda-forge will be updated automatically).

SMALL ASIDE: BEST PRACTICES

- Always have a README file.
- Always publish on PyPI (conda-forge will be updated automatically).
- Auto-publish docs and `sdist` is a plus.

SMALL ASIDE: BEST PRACTICES

- Always have a README file.
- Always publish on PyPI (conda-forge will be updated automatically).
- Auto-publish docs and `sdist` is a plus.
- Always add test with new code.

SMALL ASIDE: BEST PRACTICES

- Always have a README file.
- Always publish on PyPI (conda-forge will be updated automatically).
- Auto-publish docs and `sdist` is a plus.
- Always add test with new code.
- Adopting `flake8`, `black` and `isort` can be daunting at first but pays off in the end.

SMALL ASIDE: BEST PRACTICES

- Always have a README file.
- Always publish on PyPI (conda-forge will be updated automatically).
- Auto-publish docs and `sdist` is a plus.
- Always add test with new code.
- Adopting `flake8`, `black` and `isort` can be daunting at first but pays off in the end.
- Should we have an IOOS boilerplate repo?

SMALL ASIDE: BEST PRACTICES

- Always have a README file.
- Always publish on PyPI (conda-forge will be updated automatically).
- Auto-publish docs and `sdist` is a plus.
- Always add test with new code.
- Adopting `flake8`, `black` and `isort` can be daunting at first but pays off in the end.
- Should we have an IOOS boilerplate repo?

Some of these are part of the PyOpenSci packaging guide.

REPOSITORIES

- catalog-docker-base
- catalog-docker-ckan
- catalog-docker-ckan-harvest
- catalog-docker-pycsw
- comt
- comt_1_archive
- comt_2
- configuration-management
- configuration-management-hugo

REPOSITORY CLEAN-UP RECOMMENDATIONS

REPOSITORY CLEAN-UP RECOMMENDATIONS

- Aggressive archiving of repositories to avoid user confusion.

REPOSITORY CLEAN-UP RECOMMENDATIONS

- Aggressive archiving of repositories to avoid user confusion.
- Do not delete! Deletions are not reversible!

REPOSITORY CLEAN-UP RECOMMENDATIONS

- Aggressive archiving of repositories to avoid user confusion.
- Do not delete! Deletions are not reversible!
- Add a README.{md,txt,rst} file to all active repositories!

REPOSITORY CLEAN-UP RECOMMENDATIONS

- Aggressive archiving of repositories to avoid user confusion.
- Do not delete! Deletions are not reversible!
- Add a README.{md,txt,rst} file to all active repositories!
- Aggregate pages and docs into a sub-org/prefix.

REPOSITORY CLEAN-UP RECOMMENDATIONS

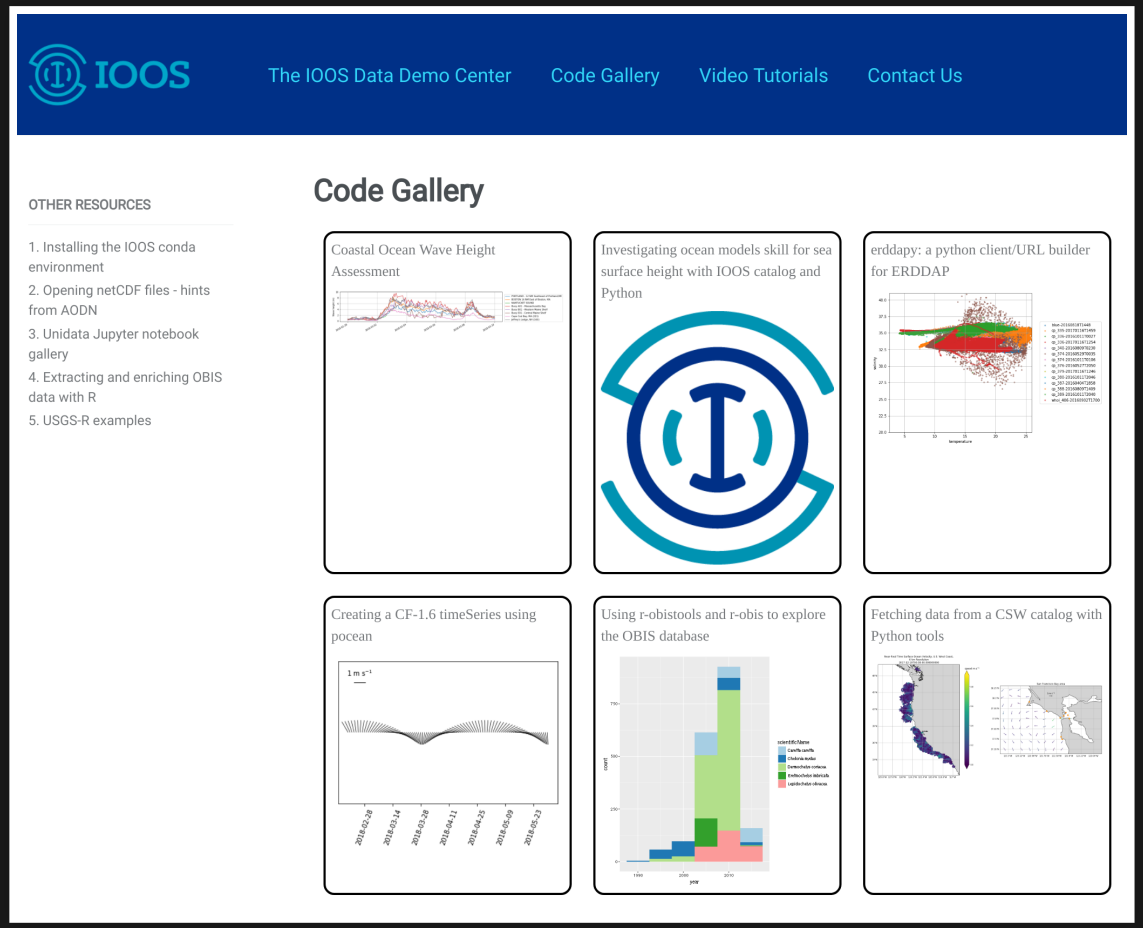
- Aggressive archiving of repositories to avoid user confusion.
- Do not delete! Deletions are not reversible!
- Add a README.{md,txt,rst} file to all active repositories!
- Aggregate pages and docs into a sub-org/prefix.
- Add Repo Health app: https://repohealth.info.

REPOSITORY CLEAN-UP RECOMMENDATIONS

- Aggressive archiving of repositories to avoid user confusion.
- Do not delete! Deletions are not reversible!
- Add a README.{md,txt,rst} file to all active repositories!
- Aggregate pages and docs into a sub-org/prefix.
- Add Repo Health app: https://repohealth.info.

https://repohealth.info/report/pyoceans/python-ctd

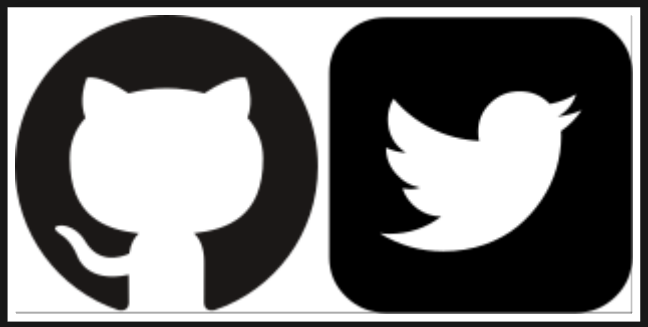
CODE GALLERY



http://ioos.github.io/notebooks_demos/code_gallery

PYVIZ DEMO

END



(ocefpaf)

https://ocefpaf.github.io/2019-DMAC