How we are using Jupyter Notebooks* in the Northeast U.S. Shelf (NES) LTER

Stace Beaulieu (stace@whoi.edu),
NES-LTER Information Manager at Woods Hole Oceanographic Institution (WHOI)
Coordinator, WHOI Ocean Informatics initiative

Joe Futrelle (jfutrelle@whoi.edu),
Applications Development in WHOI Information Services

* And R Markdown, but our focus today is on Jupyter Notebooks with Python
We are using Jupyter Notebooks in NES-LTER for:

- Data management:
  https://github.com/WHOIGit/nes-lter-notebooks

- Engaging scientists and students with NES-LTER data (e.g., data analysis and visualization):
  https://github.com/WHOIGit/nes-lter-examples
Example for data management

Challenge: How can we clean ship-provided data when something out-of-the-ordinary occurred during a cruise?

e.g., underway data, CTD data

Why a notebook?

● We do not want to include these specific fixes in our code library.
● We want to record the provenance for these specific fixes.
Example for data management

Need to add data from a particular instrument to the underway data

```python
In [3]: import pandas as pd

lines = lines_that_match(notes_file, r'.*recording')
datetimes = []

Out[3]:
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>2019-02-02 21:45:00+00:00 en627_20190202_214500 start</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>2019-02-02 23:16:00+00:00 en627_20190202_214500 stop</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2019-02-02 23:17:30+00:00 en627_20190202_213730 start</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>2019-02-03 03:38:00+00:00 en627_20190202_213730 stop</td>
</tr>
</tbody>
</table>

Create ~1Hz timestamps for each FLRTD log starting from the start time

```python
In [5]: def ticks(start_time, frequency, n):
    return [start_time + pd.Timedelta(frequency) * i for i in range(n)]

# 1.18Hz = 847ms
```

Example for engaging scientists and students with NES-LTER data*

Challenge: How can we help compare their post-cruise analyses of samples with ship-provided data?

e.g., underway data, CTD data

Why a notebook?

- Nice way to provide code, visualization, and some documentation in a single interface.
- Jupyter Notebooks in particular render nicely in GitHub.

* With R Markdown, we are addressing additional quick visualizations per cruise for PI-provided data. Our EDI Fellow and one of our REUs this summer are developing reproducible workflows.
Example for engaging scientists and students with NES-LTER data

NES-LTER: Comparison between CTD and sampled chlorophyll concentration estimates

This notebook combines chlorophyll concentration estimates derived from a CTD-mounted fluorometer with corresponding estimates derived from lab processing of samples. This enables confirming that the estimates match, which aids in the decision of when to take samples.

In [1]:

```python
import pandas as pd

BASE_URL = 'https://nes-lter-data.whoi.edu/api/
chl = pd.read_csv(BASE_URL + 'chl/en608.csv')
btl = pd.read_csv(BASE_URL + 'ctd/en608/bottles.csv')
```

Note: NES-LTER REST API

In [4]:

```python
# merge sample and CTD data per-niskin
merged = btl.merge(chl_avg, on=['cruise','cast','niskin'])

# display a few rows to make sure we're doing it right
merged[['cruise','cast','niskin','chl','fleco_afl','par']].head()
```

Out[4]:

<table>
<thead>
<tr>
<th>cruise</th>
<th>cast</th>
<th>niskin</th>
<th>chl</th>
<th>fleco_afl</th>
<th>par</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN608</td>
<td>1</td>
<td>2</td>
<td>5.737033</td>
<td>3.9609</td>
<td>4.365900e+00</td>
</tr>
<tr>
<td>EN608</td>
<td>1</td>
<td>5</td>
<td>5.174035</td>
<td>2.9635</td>
<td>2.574600e+01</td>
</tr>
<tr>
<td>EN608</td>
<td>1</td>
<td>9</td>
<td>5.397047</td>
<td>2.5430</td>
<td>1.283600e+02</td>
</tr>
<tr>
<td>EN608</td>
<td>1</td>
<td>19</td>
<td>5.344230</td>
<td>1.7145</td>
<td>6.748100e+02</td>
</tr>
<tr>
<td>EN608</td>
<td>19</td>
<td>1</td>
<td>0.014856</td>
<td>0.0020</td>
<td>1.0000000e-12</td>
</tr>
</tbody>
</table>

https://github.com/WHOIGit/nes-lter-examples/blob/master/notebooks/compare_ctd_chl_api.ipynb
Example for engaging scientists and students with NES-LTER data

In [5]: %matplotlib inline

# now plot CTD against sampled data
ax = merged.plot.scatter(
    x='chl',
    y='fleco_afl',
    c='par',
    s=50,
    cmap='viridis',
    edgecolor='black',
    title='CHL concentration: CTD vs. sampled',
    grid=True,
    figsize=(8,8),
    sharex=False
)
ax.set_xlabel('sampled')
ax.set_ylabel('CTD');

https://github.com/WHOIGit/nes-lter-examples/blob/master/notebooks/compare_ctd_chl_api.ipynb
Strengths... and weaknesses for Jupyter Notebooks with Python

Strengths

● Data management: Keeps special cases out of our code library
● Engaging scientists and students: Proven success with this software with teams of ocean scientists.*

Weaknesses

● Data management: Now we have to manage notebooks!
● Engaging scientists and students: Learning curve not only for Jupyter Notebooks but also Python

Strengths... and weaknesses for Jupyter Notebooks in any language

I Don’t Like Notebooks
Joel Grus (@joelgrus) #JupyterCon 2018

Mixed Markdown and Code

I love well-documented code!

Accordingly, I love the idea of mixing markdown and code.

For example, one underappreciated Python library is `itertools`:

```
In [1]:
    import itertools

    binary_numbers = [''.join(digits)
                        for digits in itertools.product(*['0',
                                                     '1'])
                                                     binary_numbers[:8]]

Out[1]:
```
We are offering training to NES-LTER students this summer

XSEDE Jetstream Education Allocation:

Navigating an Ocean of Data: Curriculum Development and Implementation | TG-OCE190011

Thanks!

Stace Beaulieu (stace@whoi.edu)
Extra: Next notebook we are building for ship-provided NES-LTER data

Underway fluorometer matchups for EN608

Compare underway fluorometer data with CTD cast data

Step 1: parse underway data

```
In [1]: import os
   DIR = r'D:\nes_lter_ims-test-data\en608_underway'
   assert os.path.exists(DIR)
```

```
In [2]: import re
   from io import StringIO
   import pandas as pd
```

Step 5: plot

```
In [9]: # index downcast data by datetime
   surf = sdf.copy()
   surf.index = surf.pop('date')
   # index merged uw/bti data by date
   ...
```

https://nbviewer.jupyter.org/gist/joefutrelle/09f2efba6b7629fbe19e13553a78f7cc
“Getting Stuff Done with R, Python and Jupyter Notebooks”

ESIP Summer Meeting, 17 July 2019
Moderator: M. Gastil-Buhl
Speakers: John Porter • Colin Smith • Chris Turner and Stace Beaulieu