python4oceanographers (https://ocefpaf.github.io/python4oceanographers/)

Turning ripples into waves

- Atom (http://ocefpaf.github.io/python4oceanographers//atom.xml)

Search

Navigate…

Navigate…

- Blog (http://ocefpaf.github.io/python4oceanographers)
- Archives (http://ocefpaf.github.io/python4oceanographers/archives.html)
- About (http://ocefpaf.github.io/homepage)

High resolution coastline

Jun 22, 2015

A friend asked for a nice figure for the Baía de Todos os Santos (https://en.wikipedia.org/wiki/Baía_de_Todos_os_Santos). Requests like that are always troublesome! It is hard to find high resolution datasets for the coastline for South America. They do exist, but are rarely online and, if you find someone that owns such data they usually won't share it.

The alternative are the global datasets. This post compare some of the available options. First let's define a plotting function:

In [3]:
```
import matplotlib.pyplot as plt

import cartopy.crs as ccrs
from cartopy.io import shapereader
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER


def make_map(projection=ccrs.PlateCarree()):
    fig, ax = plt.subplots(figsize=(9, 13),
                           subplot_kw=dict(projection=projection))
    gl = ax.gridlines(draw_labels=True)
    gl.xlabels_top = gl.ylabels_right = False
    gl.xformatter = LONGITUDE_FORMATTER
    gl.yformatter = LATITUDE_FORMATTER
    return fig, ax
```

Let's start with Google Maps tiles. They are not good for plotting overlays, but they are useful to convey a general idea of the area.

(I am uncertain if the data used to create the tiles are open and/or available online.)

In [4]:
```
import cartopy.io.img_tiles as cimgt

extent = [-39, -38.25, -13.25, -12.5]

request = cimgt.GoogleTiles()

fig, ax = make_map(projection=request.crs)
ax.set_extent(extent)

ax.add_image(request, 10)
```

We would like to plot a Bay contour that looks like that, with all the major rivers and islands, but without all the clutter.

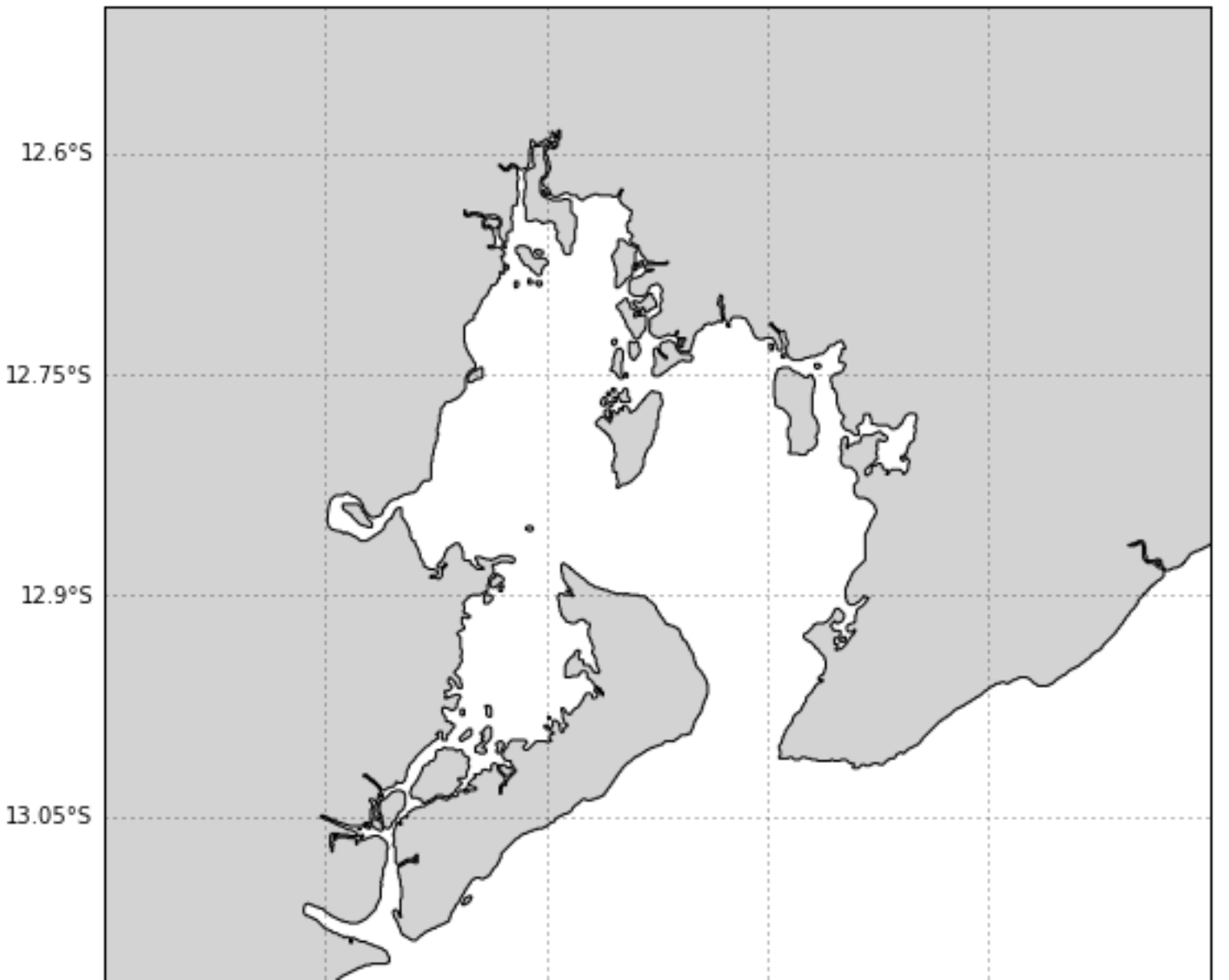
First let's try the Global Self-consistent, Hierarchical, High-resolution Geography Database (GSHHS) (http://www.ngdc.noaa.gov/mgg/shorelines/gshhs.html). The GSHHS is quite popular among oceanographers. It was used in the MatlabTM toolbox m_map (http://www.eos.ubc.ca/~rich/map.html) and it is the default choice in Python's Basemap (http://matplotlib.org/basemap/).

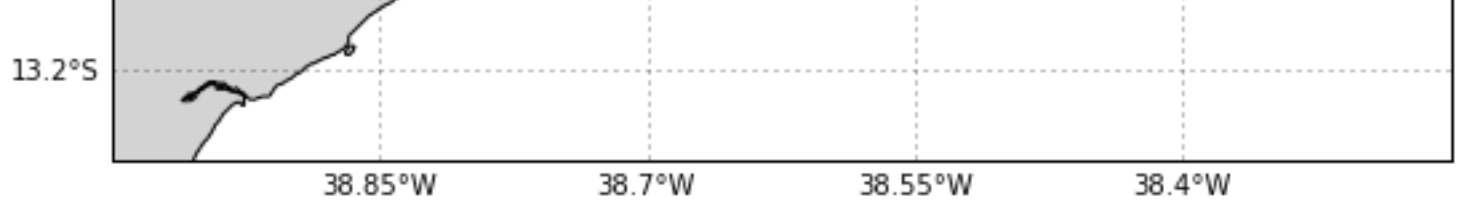
Here we use Cartopy instead of Basemap because we are loading a custom cut version of the fine resolution database.

I always cut my Shapefiles around Brazil with GDAL for faster plots and to save some disk space. Here is the commando to do so:

```
ogr2ogr -f "ESRI Shapefile" <output>.shp <input>.shp -clipsrc -82 -45 -32 10
```

In [5]:
```
fig, ax = make_map(projection=ccrs.PlateCarree())
ax.set_extent(extent)

shp = shapereader.Reader('./data/GSHHS/bts_GSHHS_f_L1')
for record, geometry in zip(shp.records(), shp.geometries()):
    ax.add_geometries([geometry], ccrs.PlateCarree(), facecolor='lightgray',
                      edgecolor='black')
```

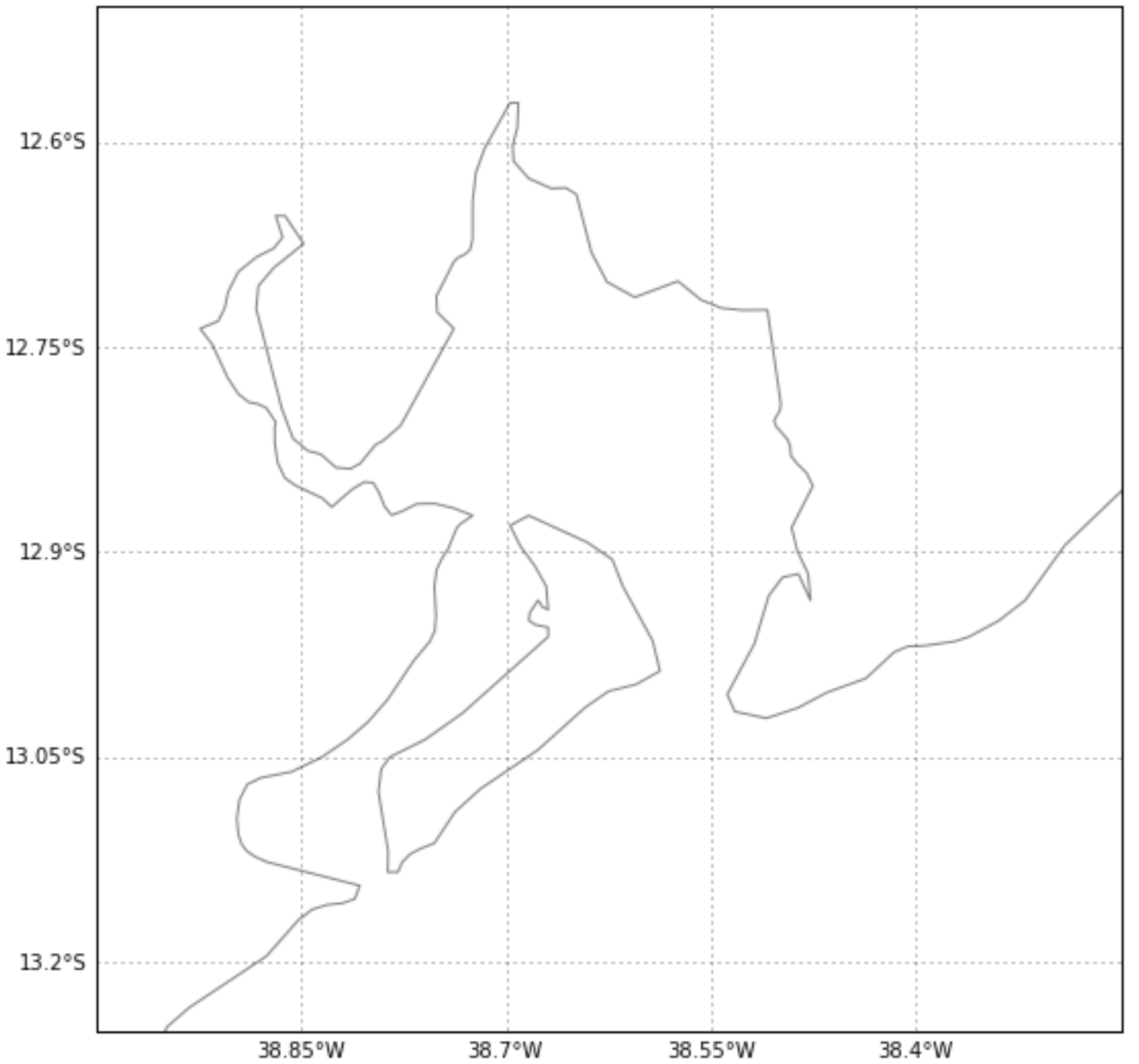




I can see all the major islands, but not the rivers. Let's try Natural Earth instead (NE).

NE is the default choice in Cartopy, so we can use Cartopy's `NaturalEarthFeature` do download the data.

In [6]:
```
from cartopy.feature import NaturalEarthFeature

coast = NaturalEarthFeature(category='physical', scale='10m',
                            facecolor='none', name='coastline')

fig, ax = make_map(projection=ccrs.PlateCarree())

ax.set_extent(extent)

feature = ax.add_feature(coast, edgecolor='gray')
```



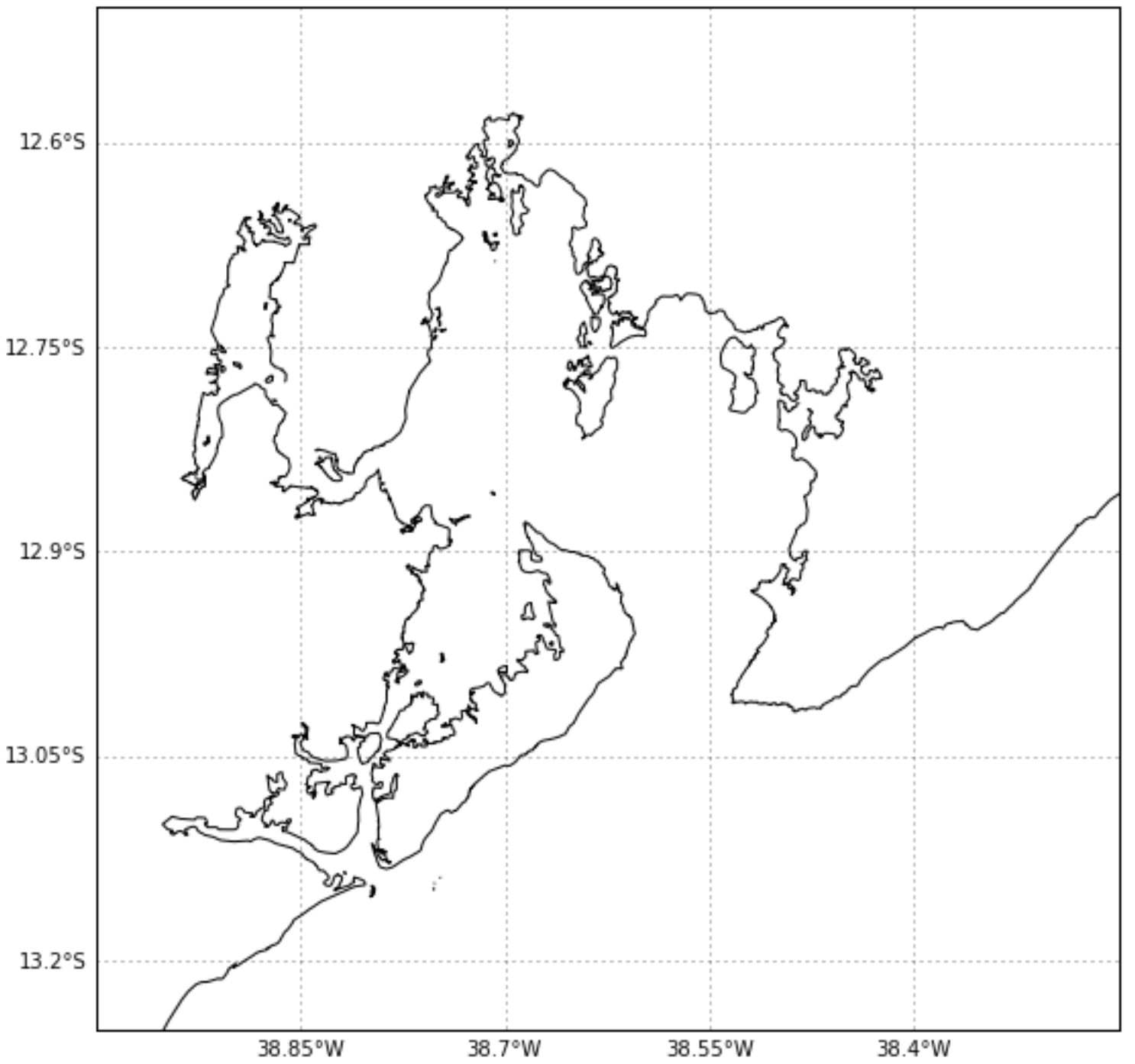
Ouch! NE is great for North America and Europe, but it sucks big time for South America! We would be better using this (https://upload.wikimedia.org/wikipedia/commons/9/9b/Mapa_da_Baia_de_Todos_os_Santos_numa_parede.jpg) image.

While trying to find the data behind Google tiles we found out that Open Street Maps data are easily available for download (http://openstreetmapdata.com/data).

They have the coastlines (http://data.openstreetmapdata.com/coastlines-split-4326.zip), water, land (http://data.openstreetmapdata.com/land-polygons-complete-4326.zip), and even Antarctic ice sheet polygons.

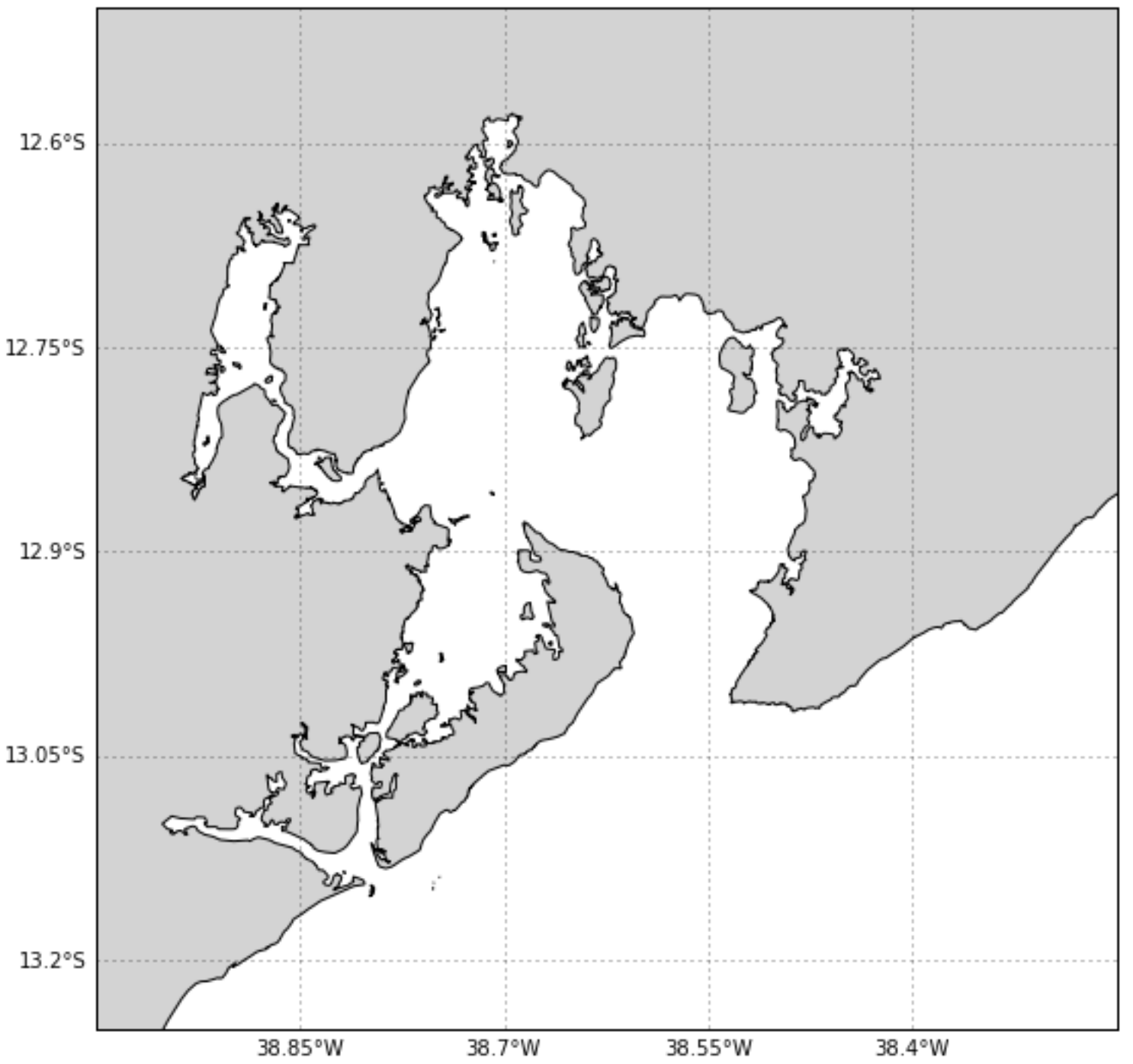
Let's test the coastline data,

In [7]:
```
fig, ax = make_map(projection=ccrs.PlateCarree())
ax.set_extent(extent)

shp = shapereader.Reader('./data/OSM_coastline/BTS')
for record, geometry in zip(shp.records(), shp.geometries()):
    ax.add_geometries([geometry], ccrs.PlateCarree(), facecolor='w',
                      edgecolor='black')
```



and land data:

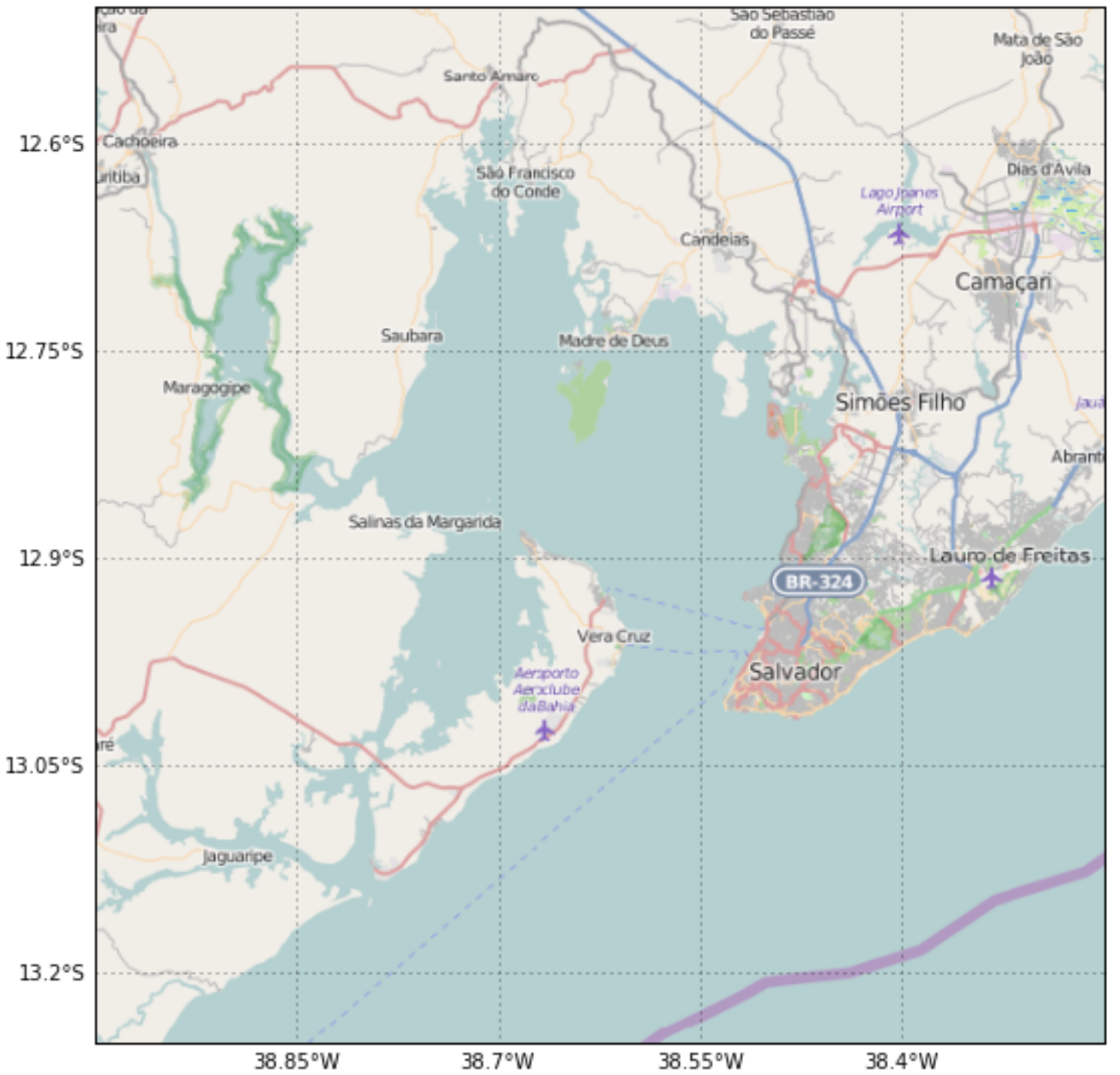
In [8]:
```
fig, ax = make_map(projection=ccrs.PlateCarree())
ax.set_extent(extent)

shp = shapereader.Reader('./data/OSM_land/BTS')
for record, geometry in zip(shp.records(), shp.geometries()):
    ax.add_geometries([geometry], ccrs.PlateCarree(), facecolor='lightgray',
                      edgecolor='black')
```



Note that we purposely used white as the `facecolor` in the coastline example. The OSM coastline data have some open lines, and coloring the polygons would produce a wacky image. The land dataset looks OK though, and it is the most detailed we found so far!

I could not find any OSM shapefile for the rivers, but the OSM tile service is very similar do Google Maps and they do display the rivers properly. So that data exists somewhere...

In [9]:
```
request = cimgt.OSM()

fig, ax = make_map(projection=request.crs)
ax.set_extent(extent)

ax.add_image(request, 10)
```

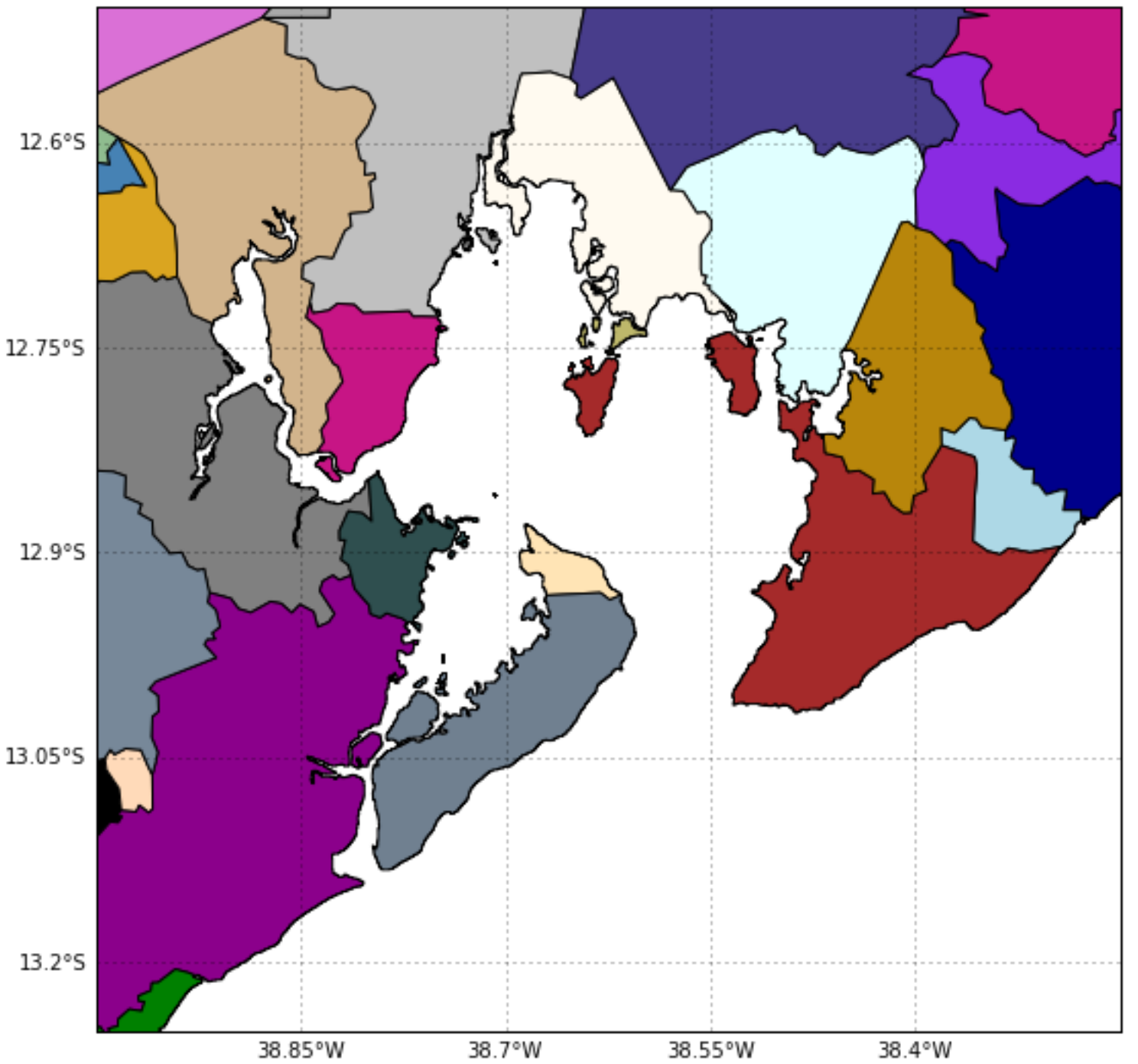


The sad part in this story is that we see high resolution coatline datasets used in several papers published, but unfortunately the data openness movement is still in its infancy in Brazil...

Updated: The OSM data seems to be mostly from the Global Administrative Areas (http://www.gadm.org/country). I already wrote about it here (https://ocefpaf.github.io/python4oceanographers/blog/2013/08/12/brazil-states-map/). If you go straight to the Global Administrative Areas data you can do things like this:

In [10]:
```
from matplotlib.colors import cnames

fig, ax = make_map(projection=ccrs.PlateCarree())
ax.set_extent(extent)

shp = shapereader.Reader('./data/BRA/BRA_adm2')

k = 0
colors = list(cnames.keys())
for record, geometry in zip(shp.records(), shp.geometries()):
    if record.attributes['NAME_1'].decode('latin-1') == u'Bahia':
        if k+1 == len(colors):
            k = 0
        else:
            k += 1
```

```
        color = colors[k]
        ax.add_geometries([geometry], ccrs.PlateCarree(),
                          facecolor=color, edgecolor='black')
```



In [11]:
```
HTML(html)
```

Out[11]: This post was written as an IPython notebook. It is available for download (https://ocefpaf.github.com/python4oceanographers/downloads/notebooks/2015-06-22-osm.ipynb) or as a static html (https://nbviewer.ipython.org/url/ocefpaf.github.com/python4oceanographers/downloads/notebooks/2015-06-22-osm.ipynb).

Posted by Filipe Fernandes Jun 22, 2015

Comments

Veronica Ruiz Xomchuk • 3 years ago

Hey, thanks a lot for this! I'm migrating from Basemap to Cartopy, and this helped a lot. I have a question tho... Is there a way to get rid of the box? I can't find a way to separate the box from the labels in the formater :S

ocefpaf Mod → Veronica Ruiz Xomchuk • 3 years ago

Hi Veronica! I am away from my laptop for a while to test it but I believe you can do that if you omit the lines:

gl.xlabels_top = gl.ylabels_right = False
gl.xformatter = LONGITUDE_FORMATTER
gl.yformatter = LATITUDE_FORMATTER

Take a quick look at cartopy's gallery in the docs too. There should be an example on how to do that.

Reply • Share ›

Madeleine Sánchez Gácita • 3 years ago

Muito bom o seu blog! Trabalho com meteorologia e estou achando as suas entradas muito úteis... Obrigada por compartilhar.

Reply • Share ›

ALSO ON **LEARN PYTHON WITH EXAMPLES APPLIED TO MARINE SCIENCES.**

Displaying ...

2 comments • 4 years ago

ocefpaf — If you are on Firefox there is an "amazing" safety feature that prevents HTTP pages to be displayed from inside a HTTPS page. Maybe you have an icon near the URL

Arctic ...

2 comments • 5 years ago

ocefpaf — You can find it here: https://raw.githubuserconte...

Plotting ...

3 comments • 4 years ago

Blake Anderson — Great post! Interactive Map Software

netCDF4 ...

2 comments • 4 years ago

ocefpaf — I realized I was installing pyaxiom everywhere just to get that method. So it made sense to send it upstream.

Recent Posts

- Plotting a GeoDataFrame with folium (https://ocefpaf.github.io/python4oceanographers/blog/2015/12/14/geopandas_folium/)
- Python tools for sgrid (https://ocefpaf.github.io/python4oceanographers/blog/2015/12/07/pysgrid/)
- Active Fire Data WMS Modis layer (https://ocefpaf.github.io/python4oceanographers/blog/2015/11/30/chapada/)
- Desabafo de um Pesquisador (https://ocefpaf.github.io/python4oceanographers/blog/2015/11/23/FAP/)
- Displaying a field of arrows with folium (https://ocefpaf.github.io/python4oceanographers/blog/2015/11/16/folium_quiver/)

Blogroll

- PyAOS (http://pyaos.johnny-lin.com/)
- EarthPy (http://earthpy.org/)
- PyHOGs (http://pyhogs.github.io/)
- drclimate (http://drclimate.wordpress.com/)
- Software Carpentry (http://software-carpentry.org/blog/index.html)