

Getting things done with Jupyter Notebooks

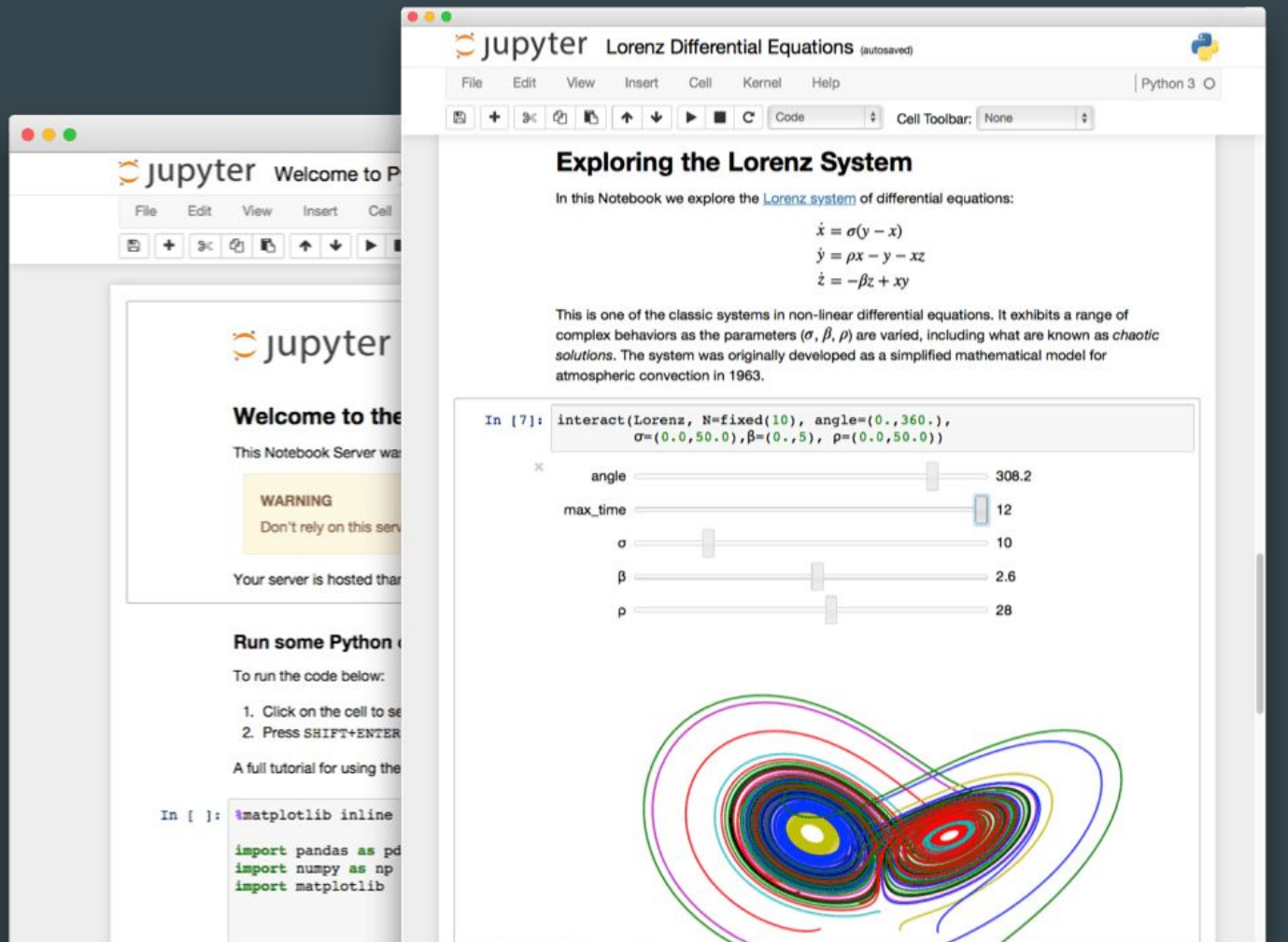
Chris Turner

Data Librarian at [Axiom Data Science](#)

Information Manager for [NGA LTER](#)

chris@axiomdatascience.com

Getting things done, with what?



A Jupyter Notebook is an IDE, kind of. But really it's just a file format that allows a mix of runnable code blocks, and documentation (markdown).

Getting things done, with what?

Project Jupyter

- Open source
- Non-profit
- Evolved from IPython Project
- Notebooks, and more!

IP[y]: IPython
Interactive Computing

How to get started for free*

Jupyter

- jupyter.org
- [Binder](https://mybinder.org)

Big Names

- [Google Colab](https://colab.research.google.com)
- [Azure Notebooks](https://azure.microsoft.com/en-us/services/notebooks/)
- [Watson Studio Cloud](https://cloud.ibm.com/catalog/services/watson-studio)

Research / Publishing

- [CodeOcean](https://codeocean.com)
- [Kogence](https://kogence.com)
- [Research Workspace](https://research.workspace.com)

Data Science / ML

- [Kyso](https://kyso.io)
- [ModeAnalytics](https://modeanalytics.com)
- [Quantopian](https://quantopian.com)

Training / Education

- [CoCalc](https://coCalc.com)
- [Notebooks](https://notebooks.azure.com)
- [Gryd](https://gryd.io)
- [DataQuest](https://dataquest.io)
- [Kaggle](https://kaggle.com)

Reviews and Discussion

- [on blogs](#)
- [medium](https://medium.com)
- dataschool.io

... or, [host your own server](#)

How do we use Jupyter Notebooks?

Data management and processing

- ingest, processing scripts

Demonstration of capabilities and processes

- novel analyses and approaches

Compute near the data

- don't need to transfer large files or collections

How do we use Jupyter Notebooks?

Data Management and Processing Example: Ingest, style, and map Audubon data

Project:

























































- 1000+ spatial datasets
- Re-style layer to match print version of the [EABCBS](#)
- Create interactive data portal

Why Notebooks:

- Unique data
- Transparent to client
- Shareable
- One-off work

The Ecological Atlas of the Bering, Chukchi, and Beaufort Seas

- [Print \(pdf\)](#)
- [Web](#)

		Add Hover Tool.ipynb	57.7 kB		John Duna...	Jun 04
		add-layer-group-tags.ipynb	249.5 kB		Trevor Gol...	Jun 04
		add-modules-to-portals.ipynb	104.3 kB		Trevor Gol...	Jun 21
		add-raster-layers.ipynb	59.2 kB		Trevor Gol...	May 07
		add-updates-layers.ipynb	112.3 kB		Trevor Gol...	May 03
		add_mammal_layers.ipynb	192.3 kB		Trevor Gol...	May 03
		add_oikos_layers.ipynb	94.1 kB		Trevor Gol...	Apr 05
		apply_fish_styles.ipynb	2.7 kB		Trevor Gol...	Apr 05
		Associate_styles_with_layers_Geoser...	49.4 kB		John Duna...	May 30
		 bounds.xml	454.4 kB		John Duna...	Jun 03
		collapse-paau-regular-use-concent...	78.8 kB		Trevor Gol...	Jun 11
		collapse-updated-layers.ipynb	227.2 kB		Chris Turner	about an...
		Create Species Pivoted.sql	11.0 kB		John Duna...	Apr 17
		create-composite-mammals-table.i...	5.0 kB		Trevor Gol...	Apr 11
		create_composite_fishes_table.ipynb	4.8 kB		Trevor Gol...	Apr 03
		create_schemas.sql	234 B		Malcolm H...	Jan 31, 2...
		 create_schemas.txt	234 B		Melany Wil...	Jan 16
		crop-sealion-range-to-study-area.i...	102.0 kB		Trevor Gol...	Jun 11

How do we use Jupyter Notebooks?

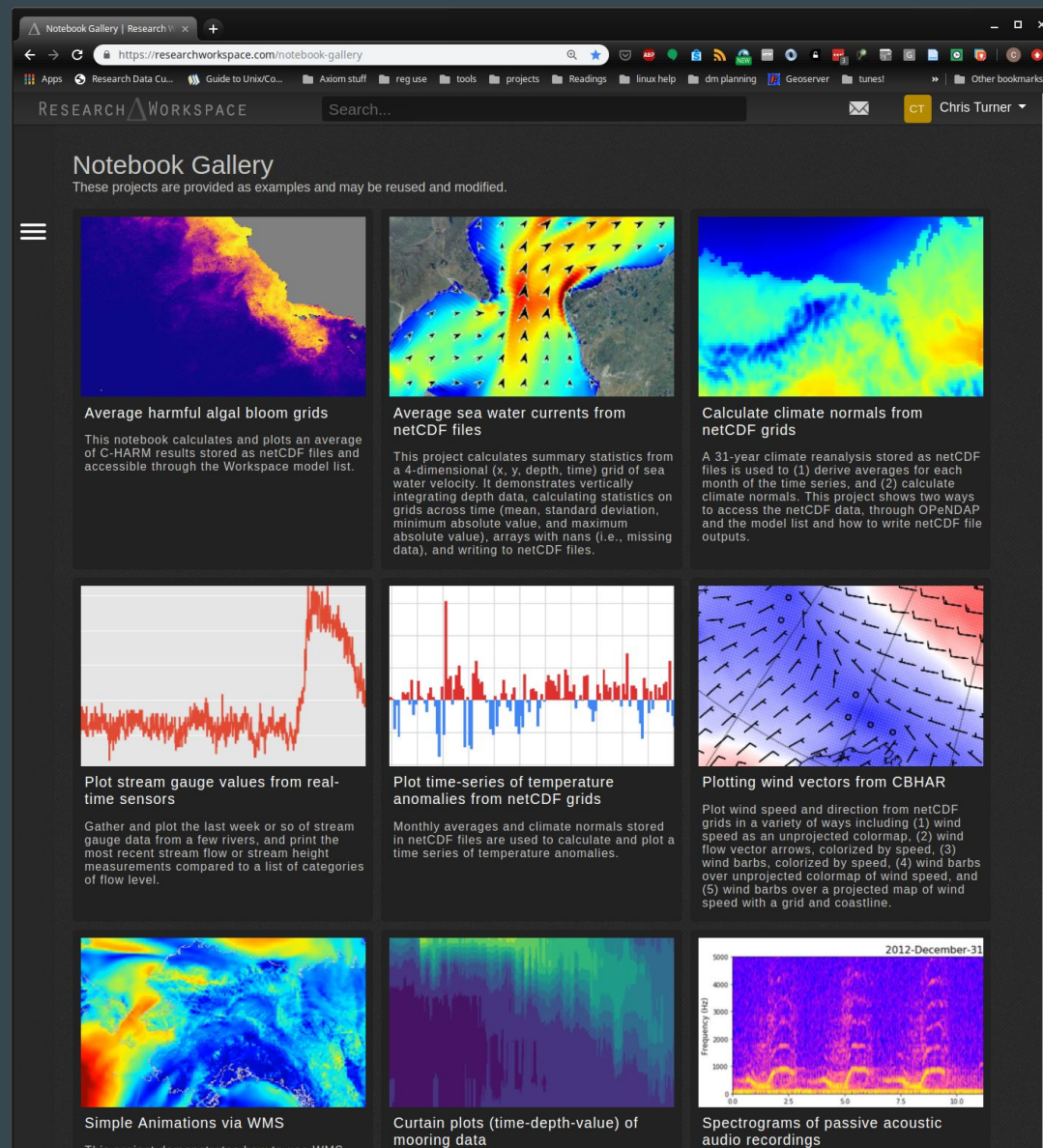
Example Demonstrations: Research Workspace Notebook Gallery

Project(s):

- Use Notebooks to process or analyze assets in the Axiom data system

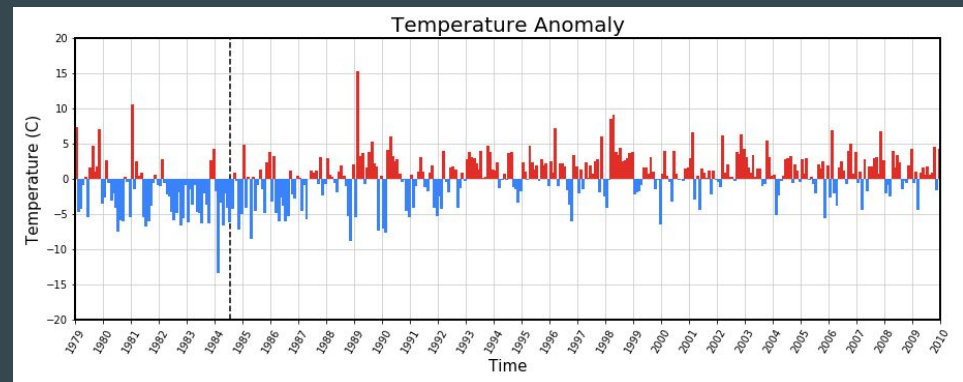
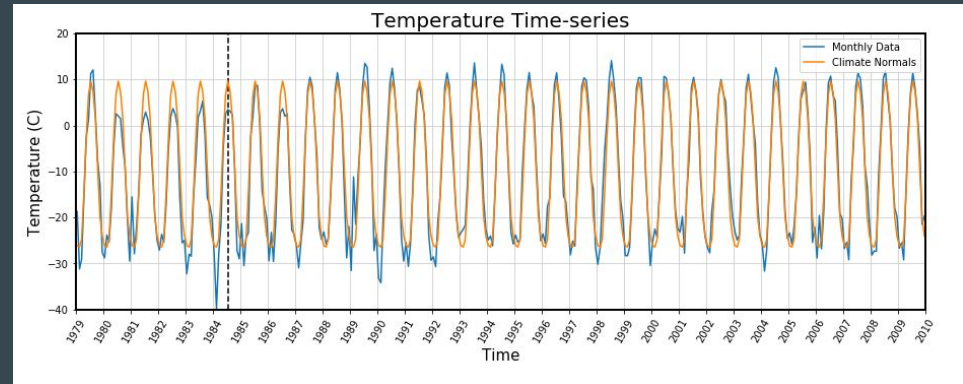
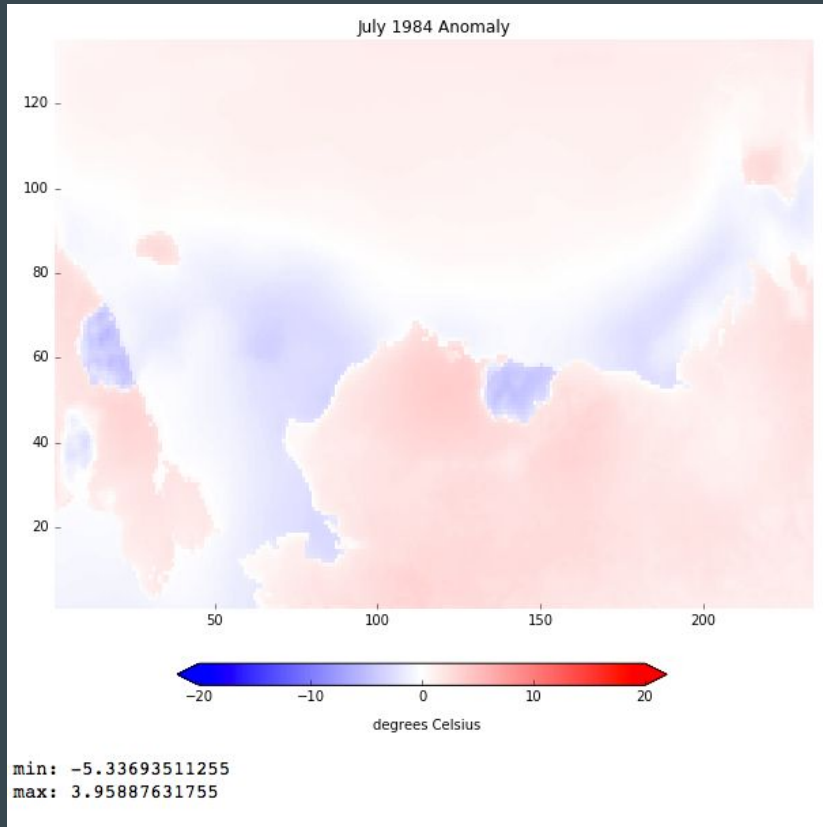
Why Notebooks:

- Shareable
- Documentable
- Interactive



<https://researchworkspace.com/notebook-gallery>

Time-series Anomalies: CBHAR model



- Calculate climate normals on a 31-year long, multi-terabyte dataset
- Then plot temperature anomalies over a region

Averaging Many Gridded Files: C-HARM model

```
# We have many files in the directory, but we want to subset those by time. In this example,
# we'll just subset by a start and end date.
start_day = datetime.datetime(2015,6,1,0,0)
end_day = datetime.datetime(2015,8,31,0,0)

# These lines build a datacube of time slices between our start and end dates,
# expanding the dimensions as it goes.

first = True

for i in trange(nfiles):

    netcdf = netCDF4.Dataset(filenamees[i])
    # extract the time, turn it into a date
    t = np.array(netcdf.variables['time'])[0]
    t = netCDF4.num2date(t, time_var.units, time_var.calendar)

    # compare the date of the time slice to our set start and end dates
    if start_day <= t <= end_day:
        # get the data from the netcdf file, remove the first axis (time)
        thisdata = np.array(netcdf.variables[variable_name])[0,:,:]

        if first:
            # If this is the first filename, create an array
            datacube = np.expand_dims(thisdata, axis=0)
            first = False
        else:
            thisdata=np.expand_dims(thisdata, axis=0)
            # If this is not the first filename, add to the existing array
            datacube= np.append(datacube, thisdata, axis=0)

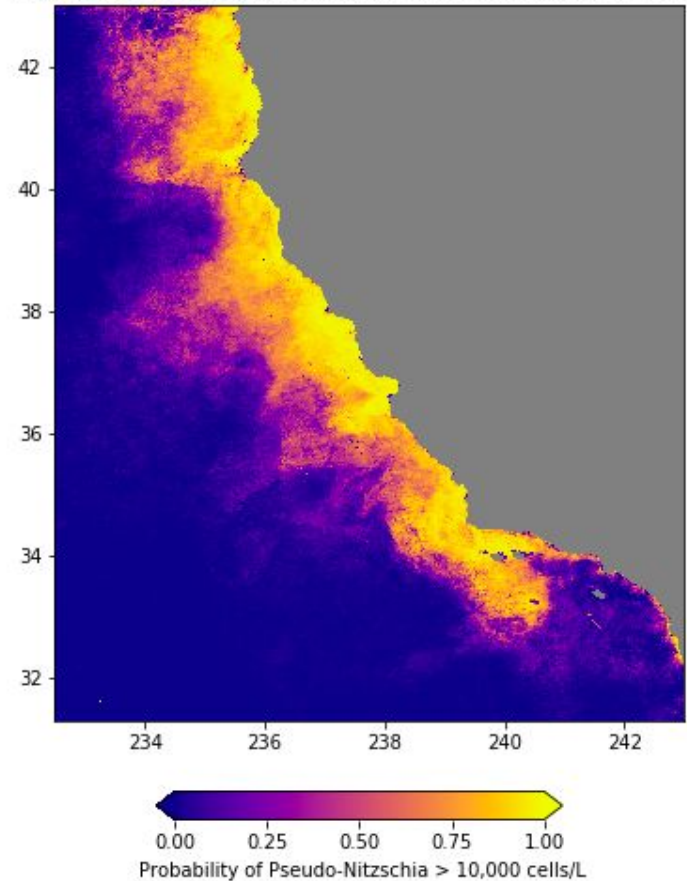
    # close each file that we open
    netcdf.close()

100%|██████████| 1222/1222 [00:24<00:00, 50.09it/s]

# Apply the mask to the datacube
datacube = ma.masked_values(datacube, -999.)
```

- Averaging gridded files between a start and end date

Mean Pseudo-Nitzschia Probability: Jun-01-2015 to Aug-31-2015



Test Implementation: CRAWL State Space Model

The screenshot shows a Jupyter Notebook interface. The title bar reads "Step 1. bearded-seals-example-crawl-notebook.ipynb". The notebook content includes a title "CRAWL-ing with bearded seals", a subtitle "Applying the `crawl` state space model (SSM) to adult bearded seal telemetry data from the North Slope", and introductory text about the CRAWL package and the author Josh London. It also includes a section "1. Import the libraries" with a code cell containing R code for installing and loading various packages.

CRAWL-ing with bearded seals

Applying the `crawl` state space model (SSM) to adult bearded seal telemetry data from the North Slope

The CRAWL package (continuous-time correlated random walk model for animal telemetry data) can be found here:
<https://github.com/NMML/crawl>

The code for this notebook was written by Josh London (@jmlondon) in his ebook "A Guide to Crawl-ing with R" which can be found here:
<https://jmlondon.github.io/crawl-workshop/index.html#preface>

Code was adapted by Axiom Data Science for this example notebook, and any errors are our fault.

1. Import the libraries

```
In [1]: #OPTIONS
options(unzip = 'internal')

#DATA
library(tidyverse)
library(purrr)
library(lubridate)
if(!require(devtools)) install.packages('devtools')
library(sp)
library(xts)

#PLOTS
install.packages('ggthemes')
library(sf)
library(leaflet)
devtools::install_github('bhaskarvk/leaflet.extras')
library('leaflet.extras')
install.packages('htmlwidgets')
library(htmlwidgets)

#MODEL
install.packages('pander')
library(pander)
```

<https://jmlondon.github.io/crawl-workshop/crawl-practical.html>

Test Implementation: CRAWL State Space Model

```
In [33]: n <- length(unique(sf_pred_lines$deployid))
pal <- colorFactor(topo.colors(5),
                  domain = sf_pred_lines$deployid)

pal2 <- colorFactor(ggthemes::hc_pal(palette = "default")(n),
                  domain = sf_pred_lines$deployid)

#sf::st_transform(4326) %>%
m <- leaflet() %>%
  addProviderTiles("Esri.OceanBasemap") %>%
  # addCircleMarkers(data = sf_locs, radius = 2,
  #                 weight = 2, opacity = 1,
  #                 color = ~pal(deployid)) %>%
  addPolylines(data = sf_lines, weight = 2, color = ~pal(deployid)) %>%
  addPolylines(weight = 2, data = sf::st_transform(sf_pred_lines,4326), color = ~pal2(deployid)) %>%
  # addLegend(pal = pal, values = ~deployid, labels = ~deployid) %>%
  suspendScroll()

m
```



Animation: CRAWL + models

plot-size-tests.ipynb

PYTHON 3

EDIT

PREVIEW

Make a movie of covariates along an SSM track

Purpose

This notebook uses an animal track through time to extract data from other related datasets. It resamples the track to consistent, hourly increments (taking the mean location). Then it extracts bathymetry data along the track from the ARDEM 2.0 dataset, and then extracts sea surface temperature from the GHR SST MUR data product and sea ice concentration from NSIDC. We save extracted values to CSV and create an animation of the track, sea ice, and graphs.

Inputs

- **Bathymetry data from ARDEM 2.0:** [Alaska Regional Digital Elevation Model 2.0](#) from Seth Danielson. It shares values with the International Bathymetric Chart of the Arctic Ocean, but extends further to the south. It's has 1-km resolution spacing.
- **SST and ice fraction values from MUR:** Sea surface temperatures and sea ice fractions are extracted from the [GHR SST Level 4 MUR Global Foundation Sea Surface Temperature Analysis \(.01deg/1km\)](#).
- **NSIDC Sea Ice Concentration:** Sea ice concentration values are extracted from the [NSIDC Sea Ice Concentration \(Nimbus-7 and Near-Real-Time DMSP\)](#).

Outputs

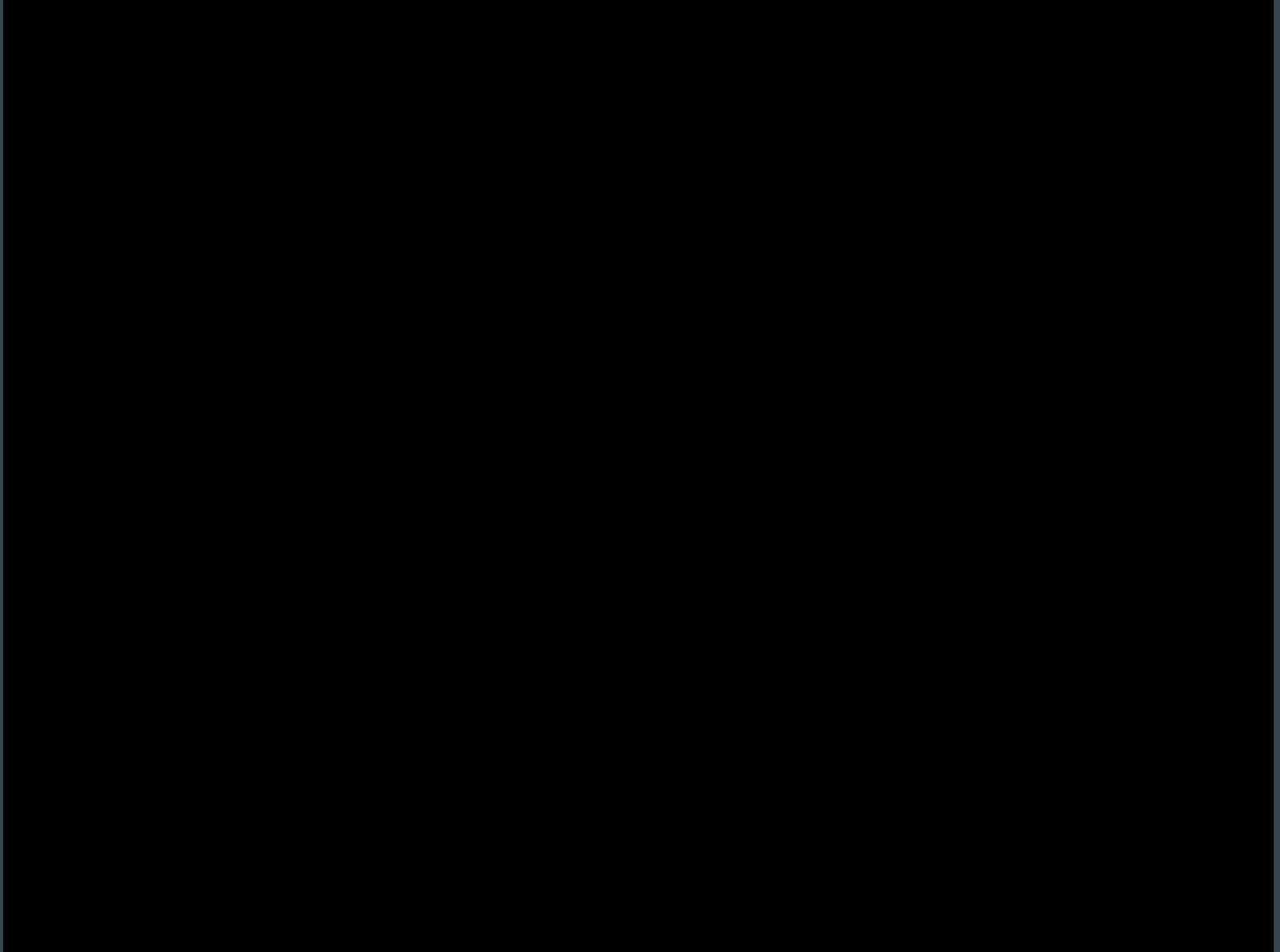
This notebook has two outputs.

1. It saves the hourly, extracted track data to a CSV file.
 - **date_time:** Date and time of the format YYYY-MM-DD HH-MM-SS (e.g., 2011-06-18 04:00:00)
 - **Longitude:** decimal degrees, positive east (0-360), epsg:4326
 - **Latitude:** decimal degrees, epsg:4326
 - **geometry:** lat/lon values of points as WKT, for convenience
 - **km_traveled_per_period:** the distance calculated between each point, which is a measure of if the tag was hanging around in one location or moving between locations. In this example we've resampled the track to be hourly, but that's somewhat arbitrary, and this distance traveled should not be confused with speed in the water (e.g., if an animal does very fast laps or dives, that would not be reflected in this value).
 - **bathymetry:** Elevation of land surface in meters (negative = depth) extracted from ARDEM
 - **sst_c:** Sea surface temperature extracted from GHR SST MUR.
 - **sst_ice_fraction:** Sea ice fraction extracted from GHR SST MUR model.
 - **nsidc_ice_percent:** Sea ice concentration extracted from the NSIDC satellite data.
2. It writes an mp4 to the workspace.

Modification History

2018-10-17: Prototype complete (W. Koeppen, Axiom)
2018-11-05: Cleaned up docs (W. Koeppen, Axiom)
2018-11-20: reworked to include sea ice in movie (W. Koeppen, Axiom)
2019-01-30: adding metadata information (W. Koeppen, Axiom)

Animation: CRAWL + models



Strengths of Jupyter Notebooks

- They're easy, interactive, and (can be) very readable
- Shareable
 - <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>
 - <https://www.researchworkspace.com/notebook-gallery>
 - <https://plot.ly/ipython-notebooks/>
 - <https://unidata.github.io/python-gallery/examples/index.html>
 - <https://proba-v-mep.esa.int/documentation/manuals/notebook-sample-gallery>
- Language and kernel options:
Python, R, Julia, and [130+ other kernels](#)

Downsides of Jupyter Notebooks?

- Easy to make mistakes
 - All cells must be run in order
 - No linting, syntax highlighting, etc.
- Encourage bad habits
 - Difficult to version, merge, and test code
 - Environment can be mysterious

Downsides of Jupyter Notebooks?

- Easy to make mistakes
 - All cells must be run in order
 - No linting, syntax highlighting, etc.
- Encourage bad habits
 - Difficult to version, merge, and test code
 - Environment can be mysterious

Criticisms:

- [Why Jupyter Is Not My Ideal Notebook](#)
- [I Don't Like Notebooks](#)
- [5 Reasons Why Jupyter Notebooks Suck](#)

Best Practices:

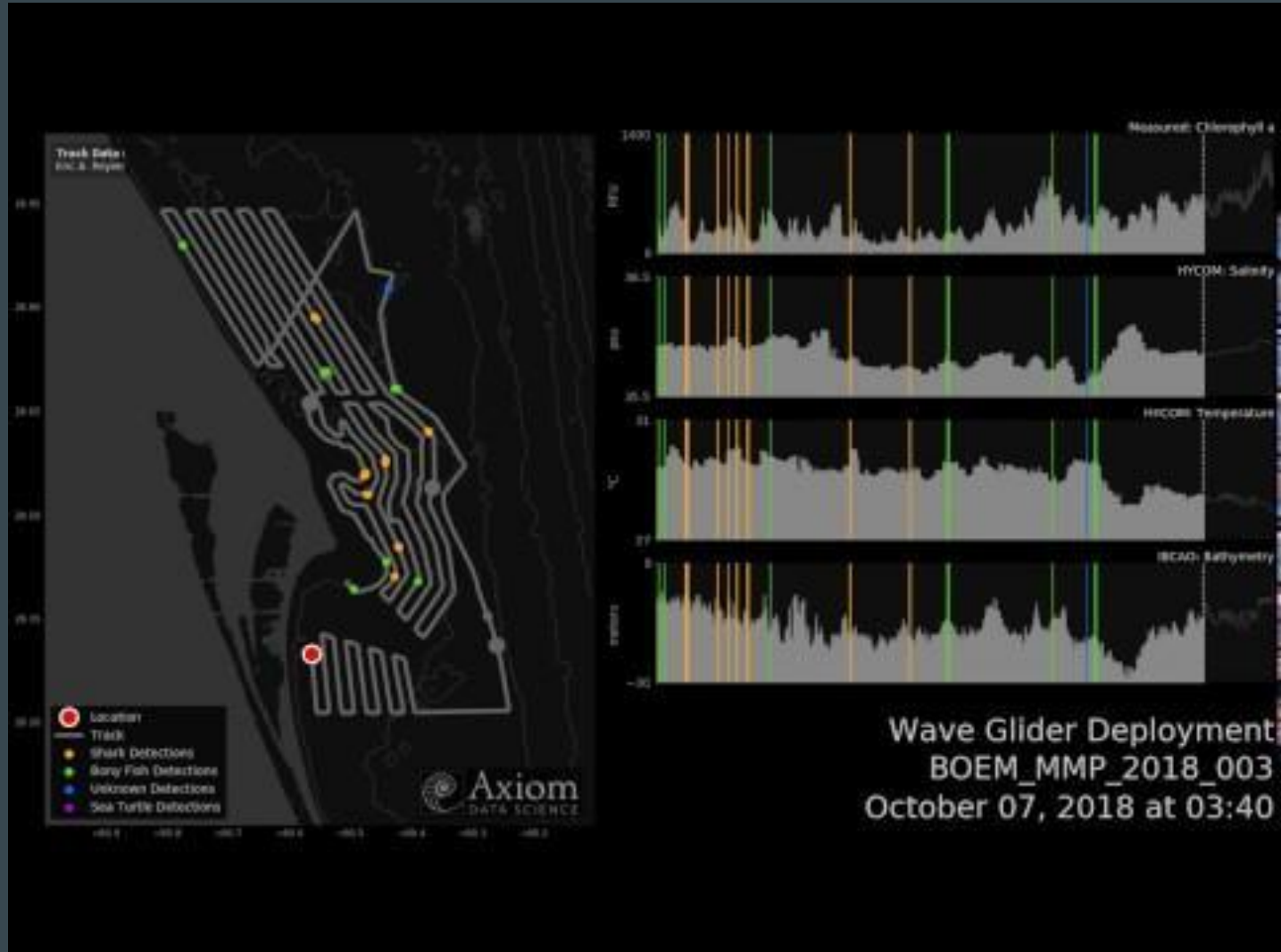
- [Jupyter Notebook Manifesto](#)
- [How to Version Control Jupyter Notebooks](#)
- [Making Publication Ready Jupyter Notebooks](#)

Done.

Questions?

chris@axiomdatascience.com

More Animations



Another example: <https://twitter.com/secoora/status/1040379622486147078>

Data Types in the Axiom Data System

Biodiversity

count, richness, diversity indices



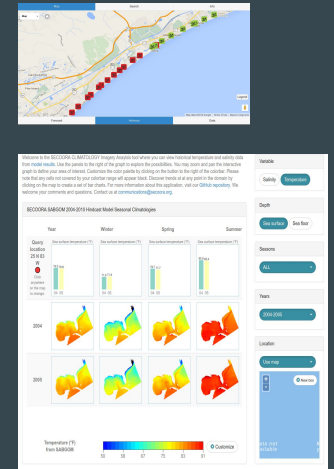
Platforms

moorings, shore stations



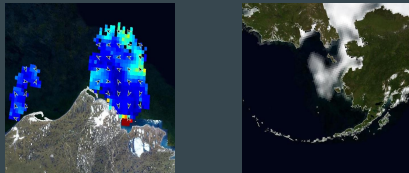
Products

skill assessment, shoreline change, etc.



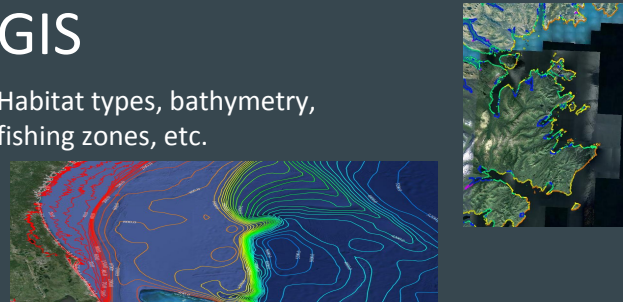
Grids

models, satellite, radar



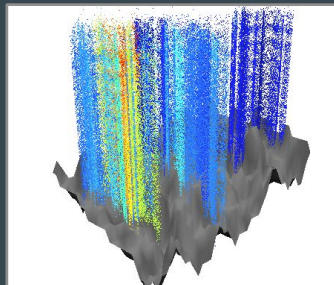
GIS

Habitat types, bathymetry, fishing zones, etc.

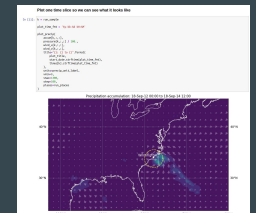


Moving Platforms

Gliders, Cruises



Unstructured Data



The Research Workspace

- Organize into projects, research campaigns and organizations
- Coordinate data exchange across networks, groups, programs
- ISO 19115-2/19110 metadata editor
- Execute server side Jupyter Notebooks on uploaded data AND any data in Axiom Data System
- Mint DOIs
- Archive pathway to DataONE (NCEI coming soon)

