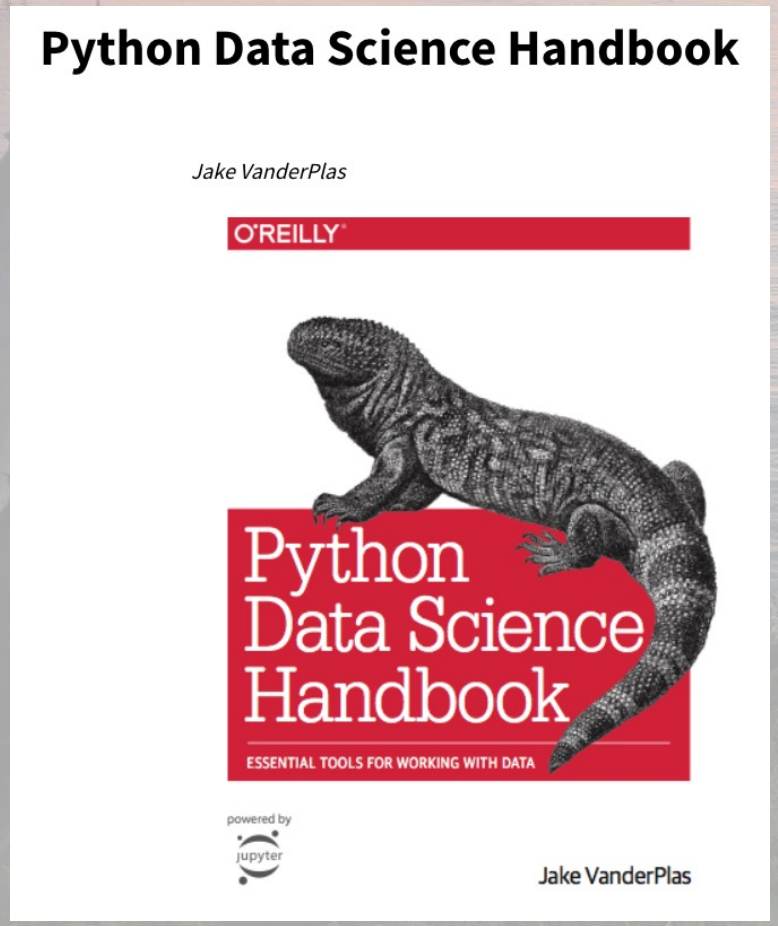
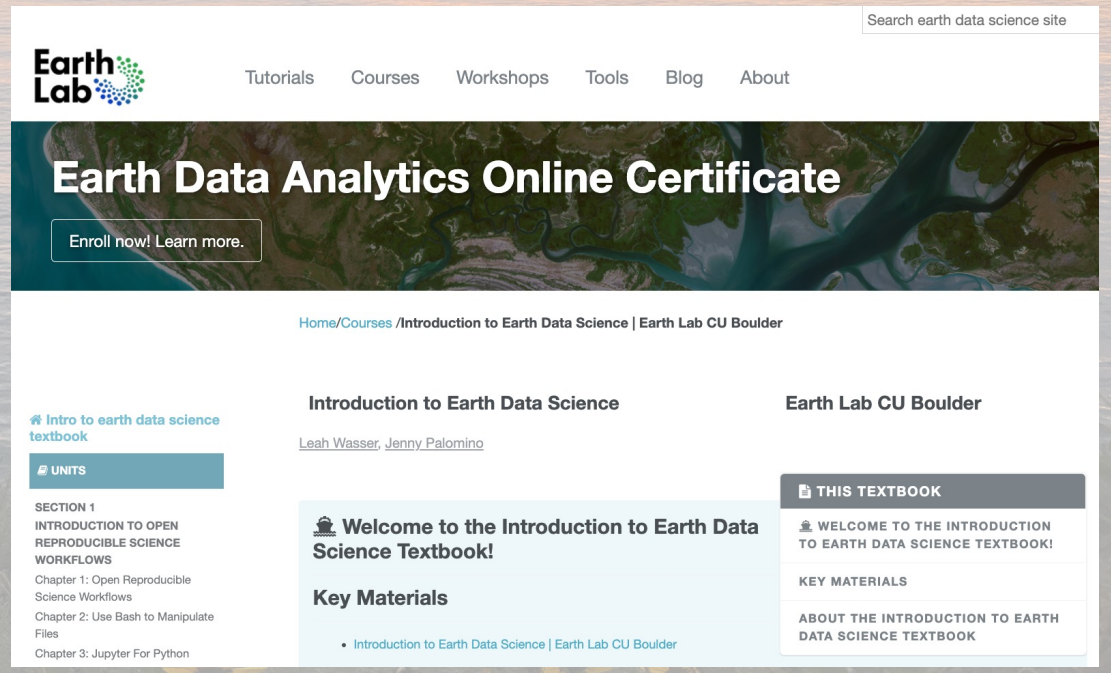
Python for data analysis and visualization (continued) and HW7

More Resources



https://jakevdp.github.io/PythonDataScienceHandbook/



https://www.earthdatascience.org/courses/intro-to-earth-data-science/

Why all of this data analysis and visualization?

- Unable to provide deep dive into scientific programming in the context of Advanced Oceanography, but we do have a few specific goals:
  - Provide high level survey of code capabilities
  - Introduce tools and how to use them
  - Work toward SLOs 2 and 3:

After completing OCN 350 students will be able to:

SLO 1: Use class information, primary literature, and other web-based sources to identify and describe important geological, chemical, and physical processes in the oceans (TTC 1, 3)

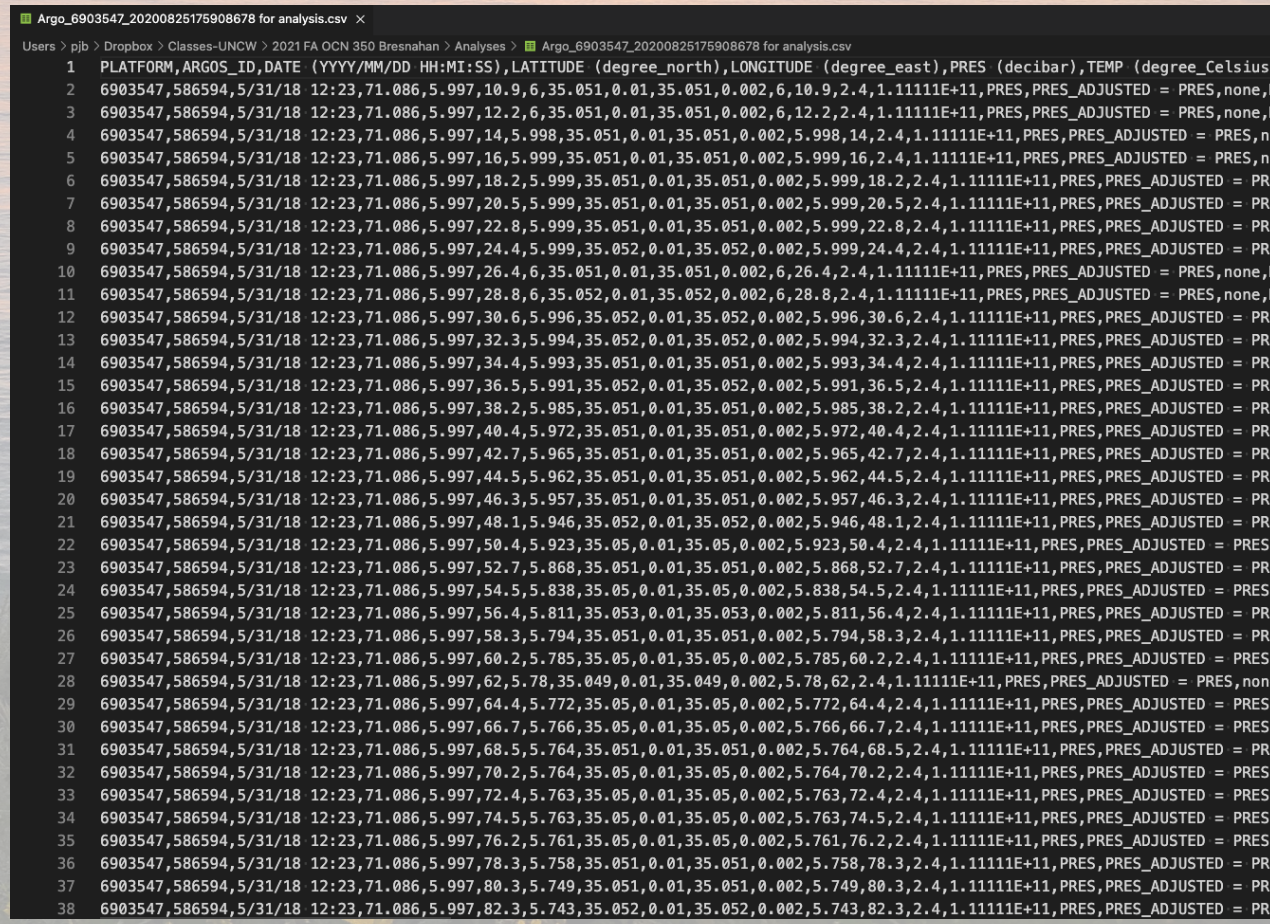
SLO 2: Use relevant data to identify and describe major features and/or processes operating in the ocean and linkages between them (TTC 4)

SLO 3: Use appropriate mathematical, statistical, or time series techniques to interpret oceanographic data (TTC 4)

SLO 4: Synthesize and communicate the way(s) that fundamental oceanographic processes and information can impact society through writing and oral presentations (TTC 1, 2, 3)

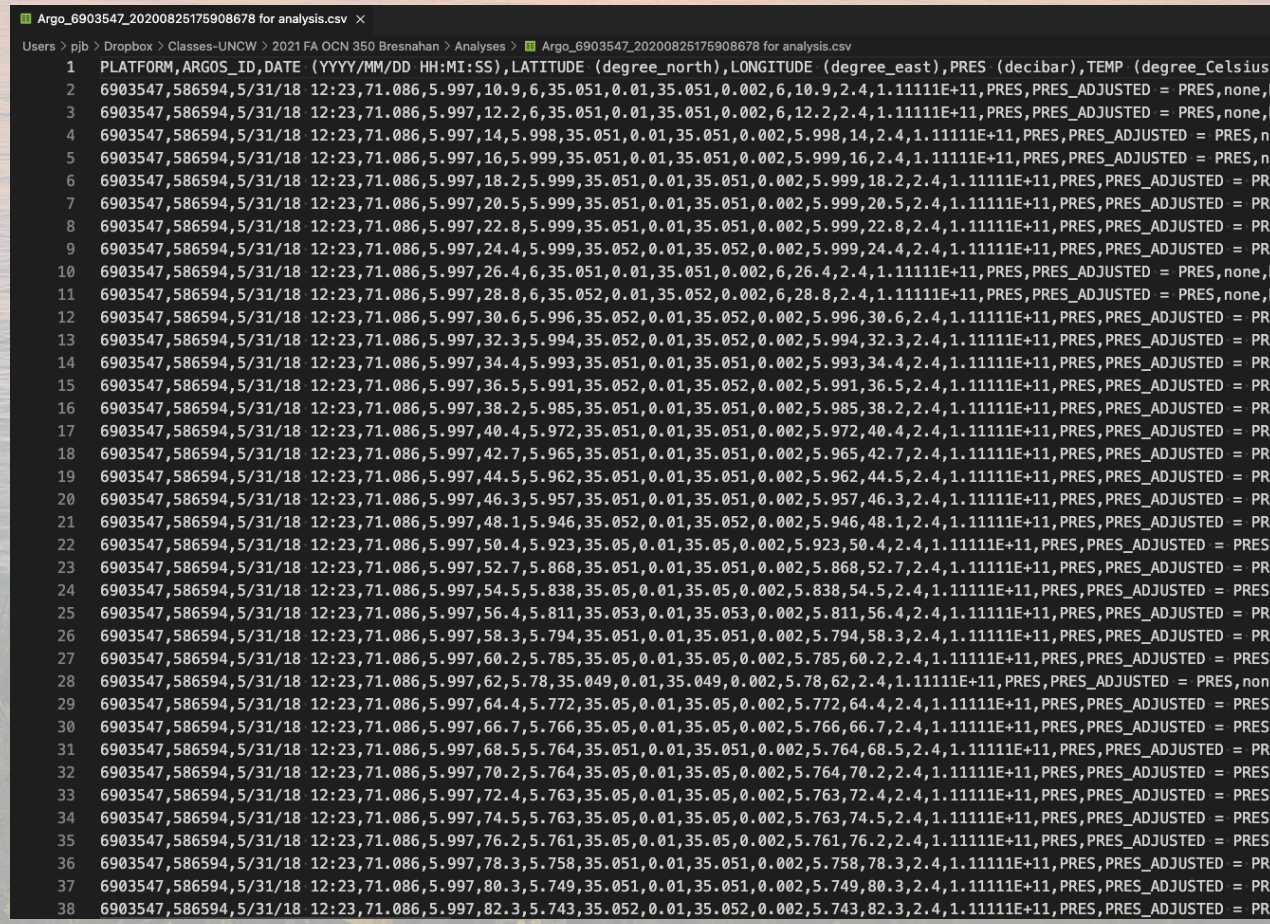
Common input data formats

- .csv or .txt
- comma-separated values or tab-delimited text
- Example: Argo data from HW3
- Advantages:
  - Can open with a simple text editor, Excel, or code
  - Simple, fairly easy to interpret without needing sophisticated programming skills
  - Should be persistent file type



Common input data formats

- .csv or .txt
- comma-separated values or tab-delimited text
- Example: Argo data from HW3
- Disadvantages:
  - Not much additional info (like when were the sensors calibrated? how? who deployed this? why?)



Common input data formats

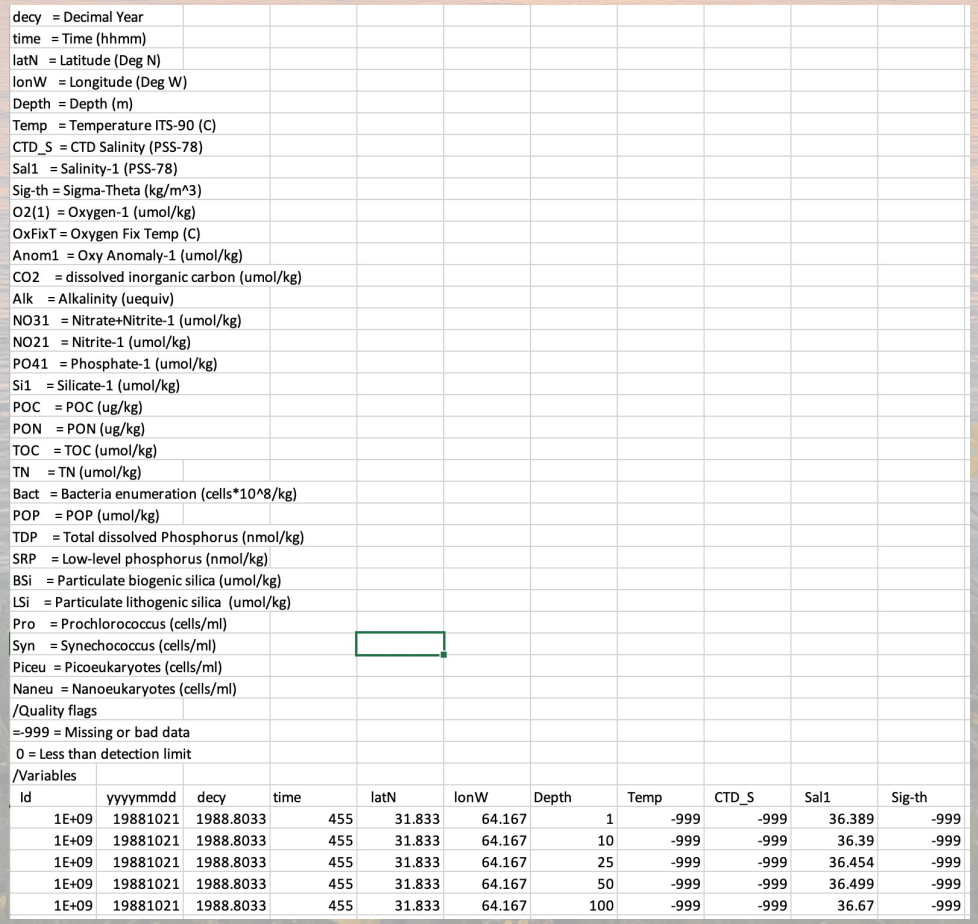
- Excel
- .xls or .xlsx
- Example: BATS data from HW5
- Advantages:
  - Can get a quick glance with more organization (columns)
  - Can do math/dataviz within the program that opens it

decy = Decimal Year
time = Time (hhmm)
latN = Latitude (Deg N)
lonW = Longitude (Deg W)
Depth = Depth (m)
Temp = Temperature ITS-90 (C)
CTD_S = CTD Salinity (PSS-78)
Sal1 = Salinity-1 (PSS-78)
Sig-th = Sigma-Theta (kg/m^3)
O2(1) = Oxygen-1 (umol/kg)
OxFixT = Oxygen Fix Temp (C)
Anom1 = Oxy Anomaly-1 (umol/kg)
CO2 = dissolved inorganic carbon (umol/kg)
Alk = Alkalinity (uequiv)
NO31 = Nitrate+Nitrite-1 (umol/kg)
NO21 = Nitrite-1 (umol/kg)
PO41 = Phosphate-1 (umol/kg)
Si1 = Silicate-1 (umol/kg)
POC = POC (ug/kg)
PON = PON (ug/kg)
TOC = TOC (umol/kg)
TN = TN (umol/kg)
Bact = Bacteria enumeration (cells*10^8/kg)
POP = POP (umol/kg)
TDP = Total dissolved Phosphorus (nmol/kg)
SRP = Low-level phosphorus (nmol/kg)
BSi = Particulate biogenic silica (umol/kg)
LSi = Particulate lithogenic silica (umol/kg)
Pro = Prochlorococcus (cells/ml)
Syn = Synechococcus (cells/ml)
Piceu = Picoeukaryotes (cells/ml)
Naneu = Nanoeukaryotes (cells/ml)
/Quality flags
=-999 = Missing or bad data
0 = Less than detection limit
/Variables

Id	yyyymmdd	decy	time	latN	lonW	Depth	Temp	CTD_S	Sal1	Sig-th
1E+09	19881021	1988.8033	455	31.833	64.167	1	-999	-999	36.389	-999
1E+09	19881021	1988.8033	455	31.833	64.167	10	-999	-999	36.39	-999
1E+09	19881021	1988.8033	455	31.833	64.167	25	-999	-999	36.454	-999
1E+09	19881021	1988.8033	455	31.833	64.167	50	-999	-999	36.499	-999
1E+09	19881021	1988.8033	455	31.833	64.167	100	-999	-999	36.67	-999

Common input data formats

- Excel
- .xls or .xlsx
- Example: BATS data from HW5
- Disadvantages:
  - Excel isn't free (though competitors are semi-free)
  - Gets extremely slow with large files
  - File types could go out of date
  - Again, not much additional info, though can add extra rows (could also do this with .txt or .csv)

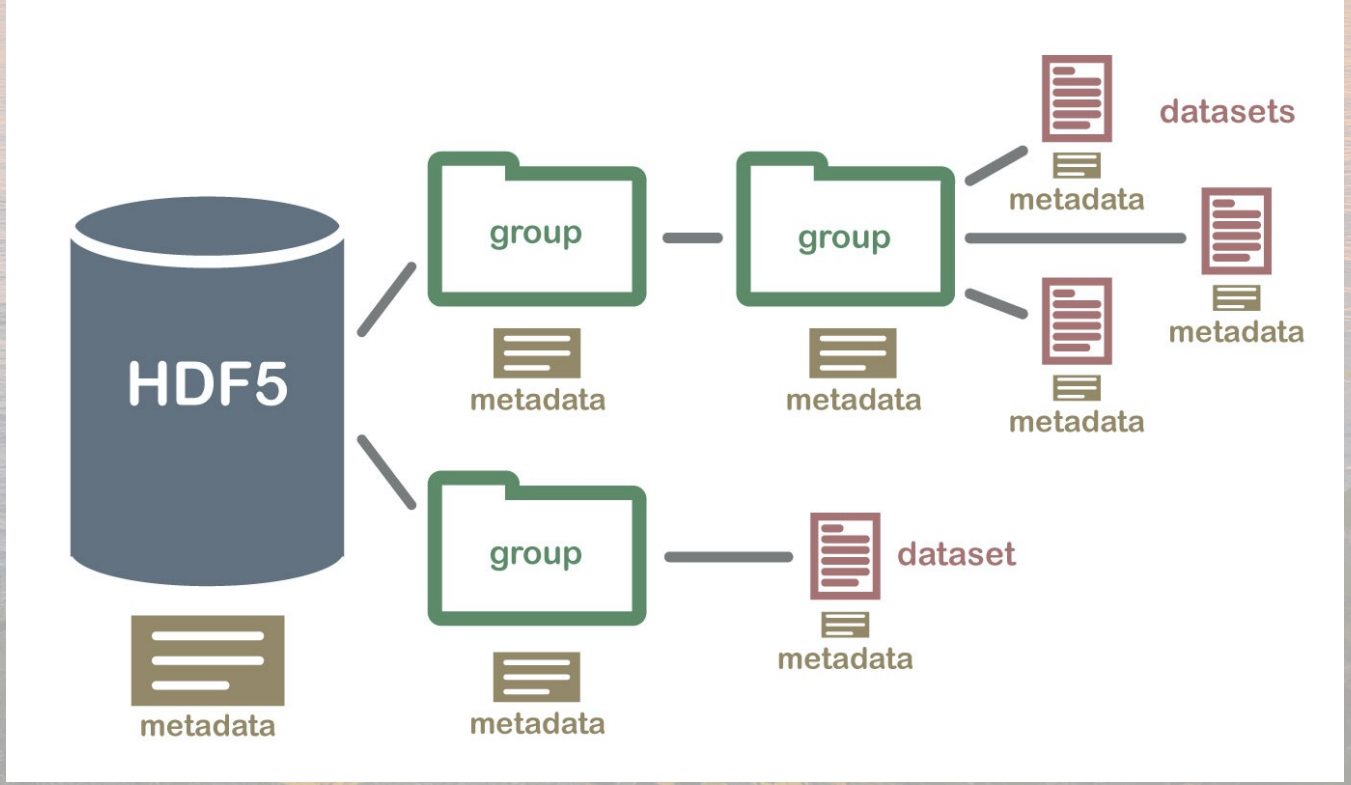


decy = Decimal Year
time = Time (hhmm)
latN = Latitude (Deg N)
lonW = Longitude (Deg W)
Depth = Depth (m)
Temp = Temperature ITS-90 (C)
CTD_S = CTD Salinity (PSS-78)
Sal1 = Salinity-1 (PSS-78)
Sig-th = Sigma-Theta (kg/m^3)
O2(1) = Oxygen-1 (umol/kg)
OxFixT = Oxygen Fix Temp (C)
Anom1 = Oxy Anomaly-1 (umol/kg)
CO2 = dissolved inorganic carbon (umol/kg)
Alk = Alkalinity (uequiv)
NO31 = Nitrate+Nitrite-1 (umol/kg)
NO21 = Nitrite-1 (umol/kg)
PO41 = Phosphate-1 (umol/kg)
Si1 = Silicate-1 (umol/kg)
POC = POC (ug/kg)
PON = PON (ug/kg)
TOC = TOC (umol/kg)
TN = TN (umol/kg)
Bact = Bacteria enumeration (cells*10^8/kg)
POP = POP (umol/kg)
TDP = Total dissolved Phosphorus (nmol/kg)
SRP = Low-level phosphorus (nmol/kg)
BSi = Particulate biogenic silica (umol/kg)
LSi = Particulate lithogenic silica (umol/kg)
Pro = Prochlorococcus (cells/ml)
Syn = Synechococcus (cells/ml)
Piceu = Picoeukaryotes (cells/ml)
Naneu = Nanoeukaryotes (cells/ml)
/Quality flags
=-999 = Missing or bad data
0 = Less than detection limit
/Variables

Id	yyyymmdd	decy	time	latN	lonW	Depth	Temp	CTD_S	Sal1	Sig-th
1E+09	19881021	1988.8033	455	31.833	64.167	1	-999	-999	36.389	-999
1E+09	19881021	1988.8033	455	31.833	64.167	10	-999	-999	36.39	-999
1E+09	19881021	1988.8033	455	31.833	64.167	25	-999	-999	36.454	-999
1E+09	19881021	1988.8033	455	31.833	64.167	50	-999	-999	36.499	-999
1E+09	19881021	1988.8033	455	31.833	64.167	100	-999	-999	36.67	-999

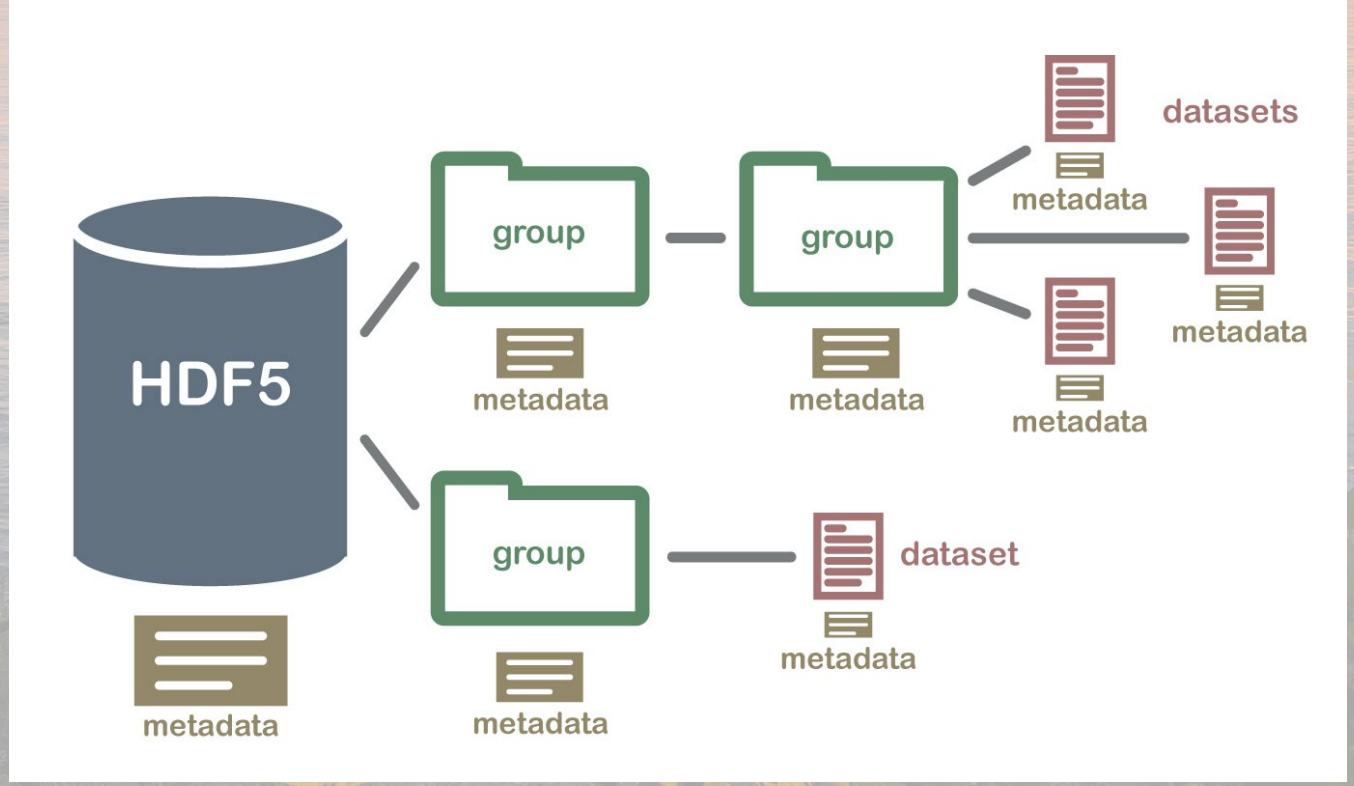
Common input data formats

- Hierarchical data formats
- .HDF, .nc (NetCDF)
- Analogous to a folder of files, but actually just a big file
- Advantages:
  - Can store different types of data and, critically, metadata in a single file
  - Handles multidimensional data very well (how many rows would you need in Excel to save data as a function of latitude, longitude, depth, and time?!)



Common input data formats

- Hierarchical data formats
- .HDF, .nc (NetCDF)
- Analogous to a folder of files, but actually just a big file
- Disadvantages:
  - Pretty complex!
  - Need to know a lot about file format to be able to use



```
Current time    5/22/2019 15:14
First sample    5/22/2019 15:14
Time zone   Local
File name   tanktest.txt
User initials   tw
Sampling period (s) 900
Standard interval   192
Standard multiple   2
pH Sample average   10
Pump on time (s)    10
Valve on time (s)   10
Sample Delay (s)    35
Standard Delay (s)  35
Low battery voltage (V) 10
TCOffset    0
Eo_int_25C  -0.4
Eo_ext_25C  -1.4
Default salinity (ppt)  33.5
Sensor name na
DuraFET SN  na
CAP adapter SN  na
ISE SN  na
MicroCAT SN na
Pump SN na
Pressure sensor full-scale (psi)    100
Vint gain   1
Vint sample rate (sps)  5
Deploy mode (pump on)

Sample #    Sample Time Main Batt   Vtherm  Vint    Vext Ref    Iso Batt    Controller Temp pH Temp Pressure (dBar) pH Int      pH Ext  Counter Le
#000000 5/22/2019 15:14 15.96   1.1449  0.049919    -1.003229   5.71    22.84   20.171  0.032   7.639134        6.124978    0   0.3379  5730    15
#000001 5/22/2019 15:30 15.96   1.15816 0.050541    -1.00323    5.73    21.98   19.784  0.063   7.652584    -0.01345    6.128183    0   0.3842  57
#000002 5/22/2019 15:45 15.96   1.16591 0.050774    -1.003203   5.73    21.57   19.56   0.057   7.658213    -0.005629   6.129792    0   0.4149  57
#000003 5/22/2019 16:00 15.96   1.17086 0.05083 -1.003249   5.71    21.33   19.417  0.141   7.660209    -0.001996   6.129409    0   0.4303  5730
#000004 5/22/2019 16:15 15.96   1.17413 0.050962    -1.00317    5.71    21.17   19.323  0.112   7.663165    -0.002956   6.130948    0   0.4469  57
#000005 5/22/2019 16:30 15.96   1.17641 0.051031    -1.003147   5.71    21.05   19.258  0.109   7.66483 -0.001665   6.131639    0   0.4468  5730
#000006 5/22/2019 16:45 15.96   1.17829 0.051071    -1.003129   5.73    20.96   19.204  0.136   7.665905    -0.001075   6.132409    0   0.4691  57
```

Defining some additional terms using a tab-delimited text file

```
1   Current time    5/22/2019 15:14
2   First sample    5/22/2019 15:14
3   Time zone   Local
4   File name   tanktest.txt
5   User initials   tw
6   Sampling period (s) 900
7   Standard interval   192
8   Standard multiple   2
9   pH Sample average   10
10  Pump on time (s)    10
11  Valve on time (s)   10
12  Sample Delay (s)    35
13  Standard Delay (s)  35
14  Low battery voltage (V) 10
15  TCOffset    0
16  Eo_int_25C  -0.4
17  Eo_ext_25C  -1.4
18  Default salinity (ppt)  33.5
19  Sensor name na
20  DuraFET SN  na
21  CAP adapter SN  na
22  ISE SN  na
23  MicroCAT SN na
24  Pump SN na
25  Pressure sensor full-scale (psi)    100
26  Vint gain   1
27  Vint sample rate (sps)  5
28  Deploy mode (pump on)
29
30  Sample #    Sample Time Main Batt   Vtherm  Vint        Temp pH Temp Pressure (dBar) pH Int      pH Ext  Counter Le
31  #000000 5/22/2019 15:14 15.96   1.1449  0.049919    -1.003          7.639134        6.124978    0   0.3379  5730    15
32  #000001 5/22/2019 15:30 15.96   1.15816 0.050541    -1.003          7.652584    -0.01345    6.128183    0   0.3842  57
33  #000002 5/22/2019 15:45 15.96   1.16591 0.050774    -1.003          7.658213    -0.005629   6.129792    0   0.4149  57
34  #000003 5/22/2019 16:00 15.96   1.17086 0.05083 -1.003249          60209   -0.001996   6.129409    0   0.4303  5730
35  #000004 5/22/2019 16:15 15.96   1.17413 0.050962    -1.003          7.663165    -0.002956   6.130948    0   0.4469  57
36  #000005 5/22/2019 16:30 15.96   1.17641 0.051031    -1.003          7.66483 -0.001665   6.131639    0   0.4468  5730
37  #000006 5/22/2019 16:45 15.96   1.17829 0.051071    -1.003          7.665905    -0.001075   6.132409    0   0.4691  57
```

METADATA

COLUMN HEADERS

DATA

NetCDF

```
# Load the dataset using xarray
data = xr.open_dataset('nodc_7900673_prof.nc')
data
```

xarray.Dataset

- Dimensions: (n_calib: 1, n_history: 0, n_levels: 1017, n_param: 3, n_prof: 148)
- Coordinates: (0)
- Data variables: (65)
- Attributes: (49)

NetCDF attributes

```
# Load the dataset using xarray
data = xr.open_dataset('nodc_7900673_prof.nc')
data
```

xarray.Dataset

- ▶ Dimensions: (n_calib: 1, n_history: 0, n_levels: 1017, n_param: 3, n_prof: 148)
- ▶ Coordinates: (0)
- ▶ Data variables: (65)
- ▼ Attributes:

title :	Argo float vertical profile
institution :	FR GDAC
source :	Argo float
history :	2020-06-18T08:34:06Z boyer convAGDAC.f90 Version 1.2
references :	https://www.nodc.noaa.gov/argo/
user_manual_v…	3.1
Conventions :	GADR-3.0 Argo-3.0 CF-1.6
featureType :	trajectoryProfile
uuid :	be30d9b1-ed40-487b-bdc5-a2b73f6fb0ba
summary :	The U.S. National Centers for Environmental Information (NCEI) operates the Argo Global Data Repository (GADR). For information about organizations contributing data to GADR, see https://www.nodc.noaa.gov/argo/
file_source :	The Argo Global Data Assembly Center FTP server at ftp://ftp.ifremer.fr/ifremer/argo
keywords :	temperature, salinity, sea_water_temperature, sea_water_salinity
keywords_voca…	NCEI Data Types, CF Standard Names
creator_name :	Charles Sun
creator_url :	https://www.nodc.noaa.gov
creator_email :	Charles.Sun@noaa.gov
id :	0173560
naming_authorit…	gov.noaa.nodc
standard_name…	CF-1.6
Metadata_Conventions :	ta Dataset Discovery v1.0
publisher_name :	US DOC; NESDIS; NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION
publisher_url :	https://www.nodc.noaa.gov/
publisher_email :	NCEI.Info@noaa.gov

NetCDF variables

▼ Data variables:

data_type	()	object	...
format_version	()	object	...
handbook_version	()	object	...
reference_date…	()	object	...
date_creation	()	object	...
date_update	()	object	...
platform_number	(n_prof)	object	...
project_name	(n_prof)	object	...
pi_name	(n_prof)	object	...
station_paramet…	(n_prof, n_param)	object	...
cycle_number	(n_prof)	float64	...
direction	(n_prof)	object	...
data_centre	(n_prof)	object	...
dc_reference	(n_prof)	object	...
data_state_indi…	(n_prof)	object	...
data_mode	(n_prof)	object	...
platform_type	(n_prof)	object	...
float_serial_no	(n_prof)	object	...
firmware_version	(n_prof)	object	...
wmo_inst_type	(n_prof)	object	...
juld	(n_prof)	datetime64[ns]	...
juld_qc	(n_prof)	object	...
juld_location	(n_prof)	datetime64[ns]	...
latitude	(n_prof)	float64	...

```
array([-65.66364, -65.66287, -65.65826, -65.66265, -65.67238, -65.63203,
       -65.5    , -65.3    , -65.2    , -65.     , -64.8    , -64.7    ,
       -64.5    , -64.4    , -64.2    , -64.     , -63.9    , -63.7    ,
       -63.6    , -63.4    , -63.2    , -63.1    , -62.9    , -62.75633,
       -62.67307, -62.23222, -61.96943, -62.0714 , -61.72549, -61.39496,
       -61.03033, -60.58207, -60.12212, -59.87366, -59.28385, -58.16719,
       -58.4986 , -58.10733, -58.0471 , -57.24182, -57.13717, -56.71731,
```

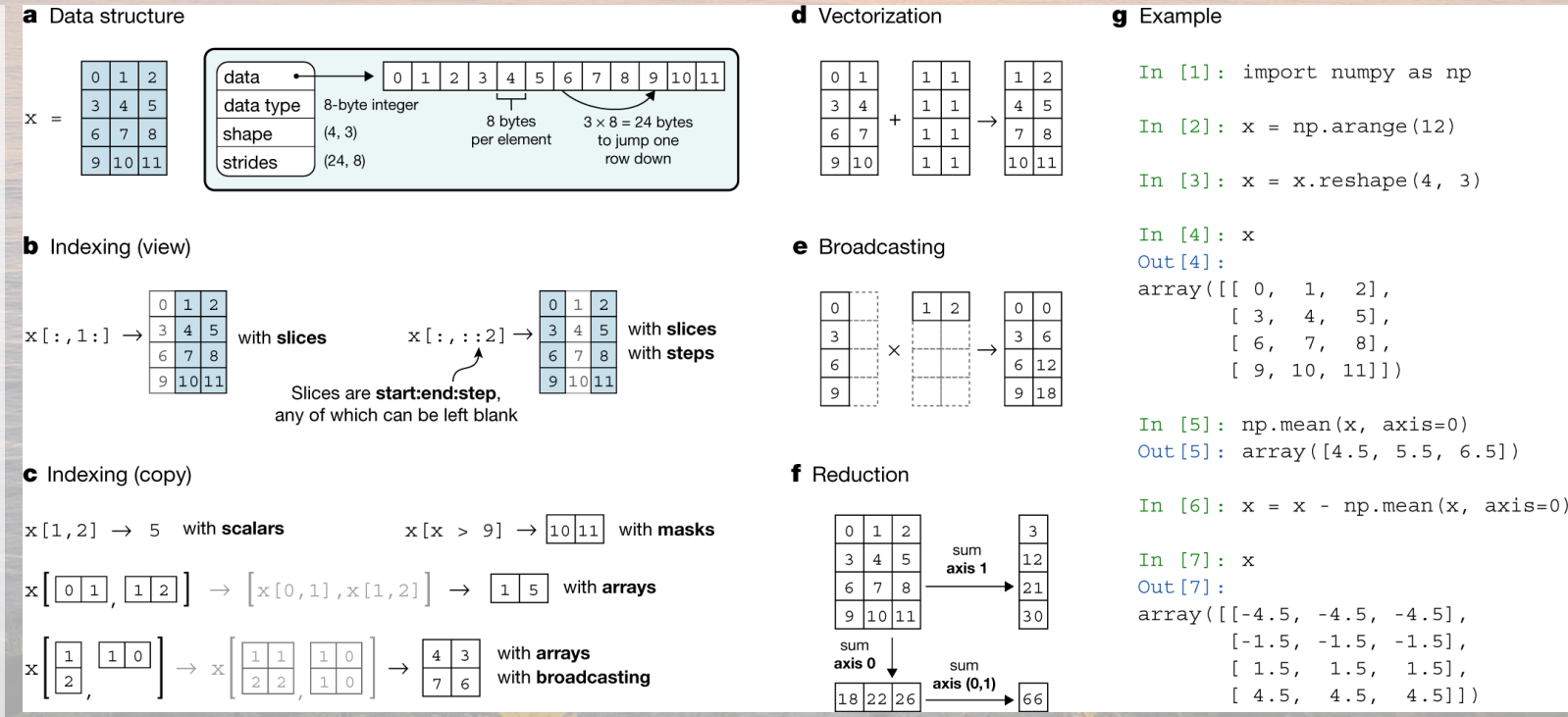
Python's main modules for dealing with numerical data

- numpy: `import numpy as np`
- pandas: `import pandas as pd`
- xarray: `import xarray as xr`

- Plus plotting: `import matplotlib.pyplot as plt`

numpy

- Create and manipulate arrays
  - Add, subtract, multiply, take median, mean, etc.
- Arrays are essentially rows, tables, or cubes of data comprising many elements



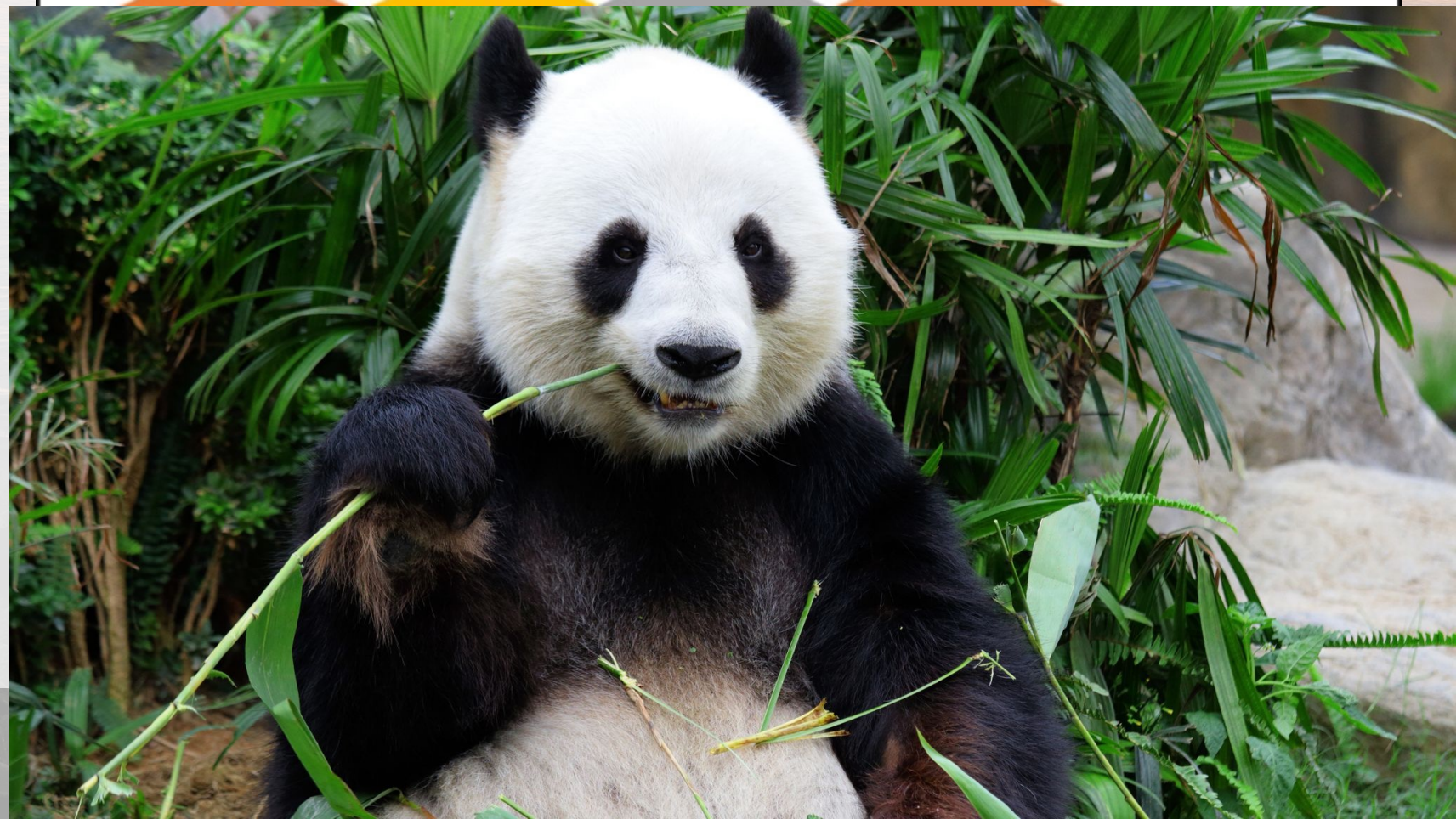
https://www.nature.com/articles/s41586-020-2649-2

pandas

- Named for “panel data,” common terminology in the finance biz but not really in the natural sciences
- Adds some functionality to numpy
- One of the biggest: dealing with time!

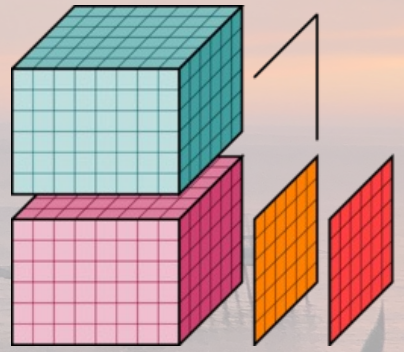
pandas: sorry it's not this..

- Named for "panel data," common terminology in the finance biz but not really in the natural sciences
- Adds some functionality to numpy



pandas in 10’
(only 4’ now, watch the rest for homework)

https://pandas.pydata.org/getting_started.html



xarray

- Advantageous for complex scientific data across many dimensions
- Designed to work with NetCDF files
- Attributes—>

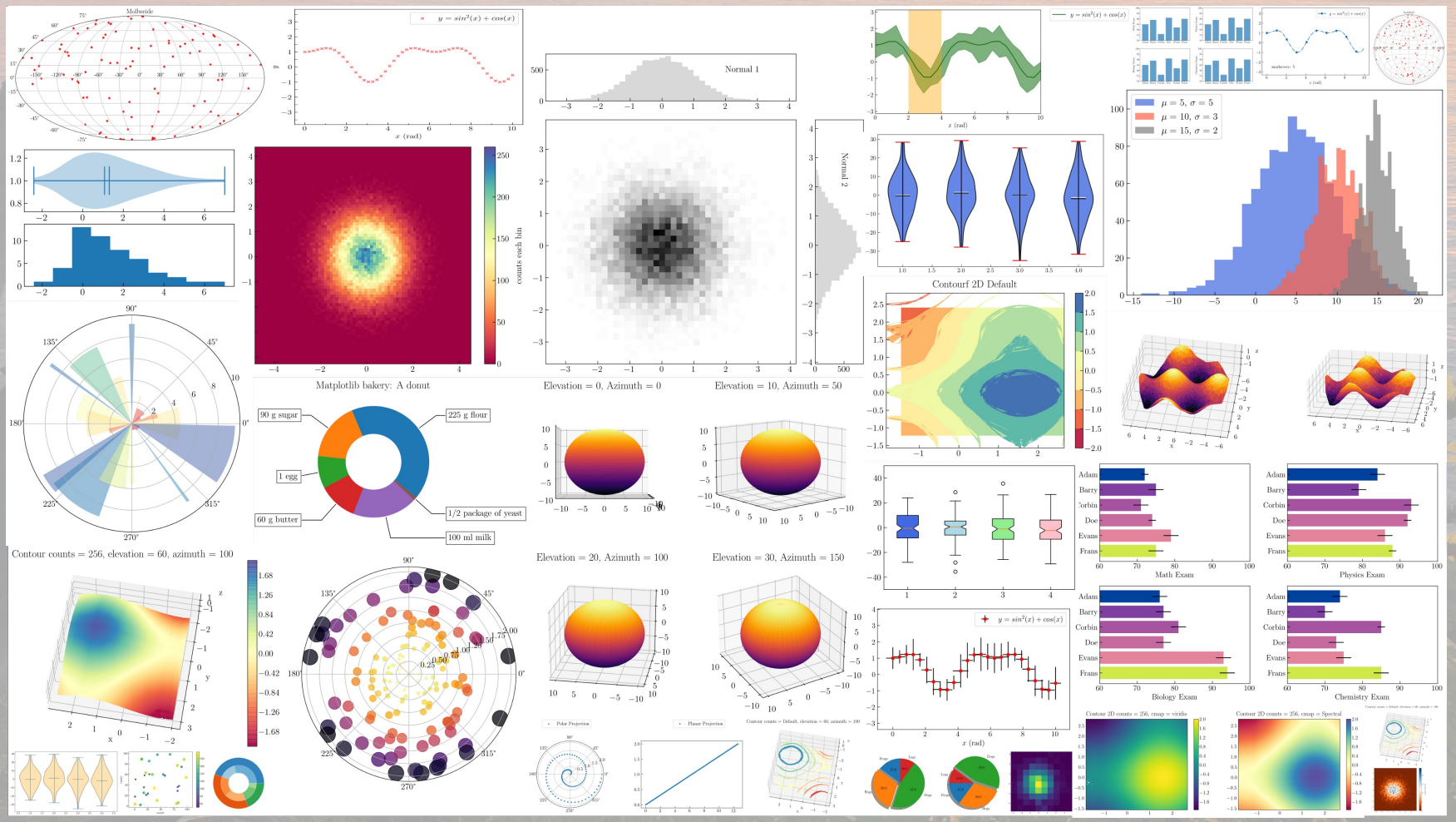
Attributes

While you're setting up your DataArray, it's often a good idea to set metadata attributes. A useful choice is to set `data.attrs['long_name']` and `data.attrs['units']` since xarray will use these, if present, to automatically label your plots. These special names were chosen following the NetCDF Climate and Forecast (CF) Metadata Conventions. `attrs` is just a Python dictionary, so you can assign anything you wish.

```
In [15]: data.attrs["long_name"] = "random velocity"

In [16]: data.attrs["units"] = "metres/sec"

In [17]: data.attrs["description"] = "A random variable created as an example."

In [18]: data.attrs["random_attribute"] = 123

In [19]: data.attrs
Out[19]:
{'long_name': 'random velocity',
 'units': 'metres/sec',
 'description': 'A random variable created as an example.',
 'random_attribute': 123}

# you can add metadata to coordinates too
In [20]: data.x.attrs["units"] = "x units"
```

Plotting:

```
import matplotlib.pyplot as plt
```



https://towardsdatascience.com/visualizations-with-matplotlib-part-1-c9651008b6b8?gi=e2a4e69a6943

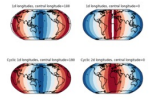
Mapping

- If you're really into mapping, I highly recommend our GIS (geographic information systems) and remote sensing classes and GIS minor
- Check out
  - Intro to GIS
  - Spatial Programming
  - Much more here: https://uncw.edu/earsci/geography-minorgeospatialtech.html
- Today: putting dots on a map using Cartopy

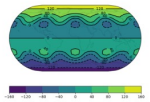
https://github.com/SUPScientist/Advanced-Quantitative-Oceanography/blob/main/notebooks/Intro_07_EXO2_MapAnalysis_cartopy.ipynb

Cartopy: powerful Python mapping

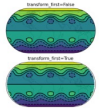
Scalar data



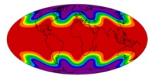
Adding a cyclic point to help with wrapping of global data



Contour labels

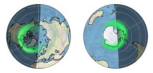


Contour transform options

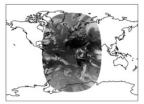


Filled contours

Map tile acquisition

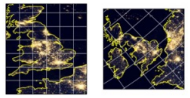


Plotting the Aurora Forecast from NOAA on Orthographic Polar Projection



Reprojecting images from a Geostationary projection

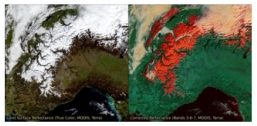
Web services



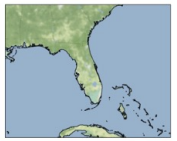
Displaying WMTS tiled map data on an arbitrary projection

Interactive WMS (Web Map Service)

Interactive WMTS (Web Map Tile Service)



Web Map Tile Service time dimension demonstration



Web tile imagery

Homework 7: Map data in Cartopy

- https://github.com/SUPScientist/Advanced-Quantitative-Oceanography/blob/main/notebooks/Intro_07_EX02_MapAnalysis_cartopy.ipynb