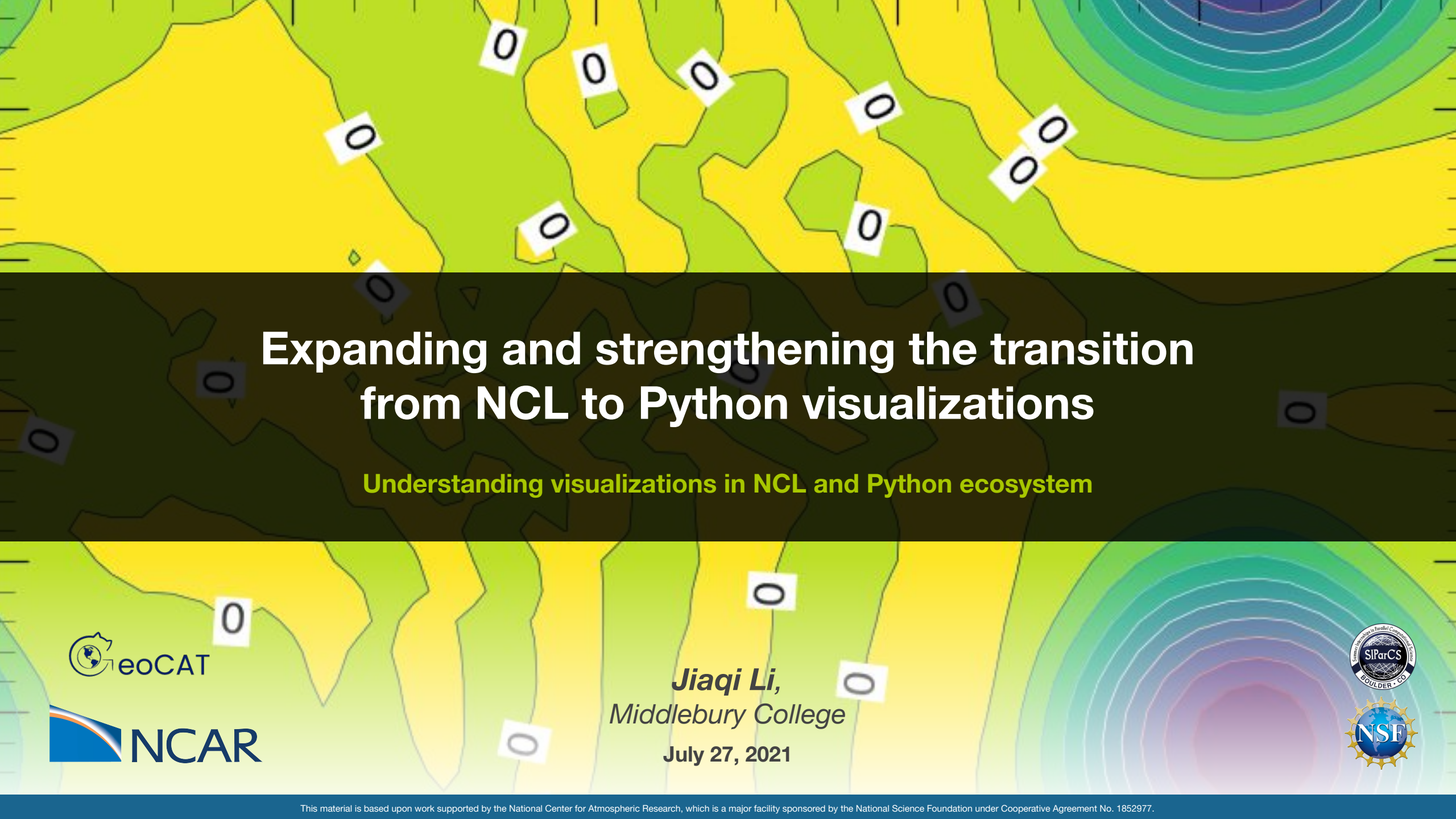


Expanding and strengthening the transition from NCL to Python visualizations

Understanding visualizations in NCL and Python ecosystem

Jiaqi Li,
Middlebury College

July 27, 2021

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

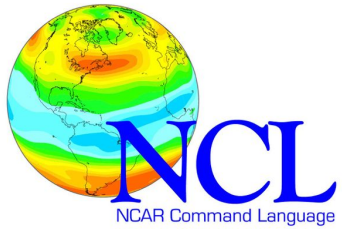
Pivot to Python:
GeoCAT team

In January of 2019, NCAR announced plans to transition away from NCL (NCAR Command Language) to Python Scientific Ecosystem

NCL is put into “maintenance mode”

GeoCAT team is in charge of making the transition as smooth as possible

- GeoCAT-Comp
- GeoCAT-Viz
- GeoCAT-Examples



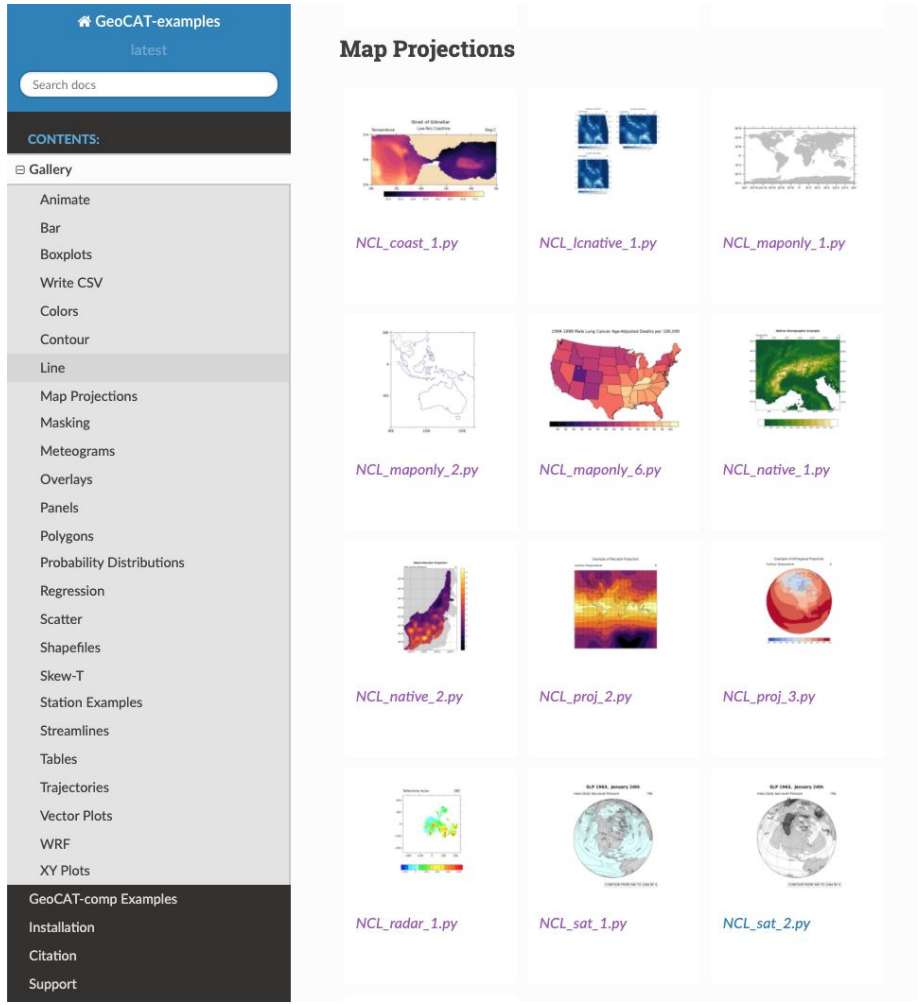
GeoCAT-Examples

A diverse a set of NCL plots recreated in Python scientific ecosystem

Each template includes

- complete Python script and documentations
- applications of key computational routines
- differences between the NCL and recreated plot

GeoCAT-examples serves as plotting templates that recreate NCL counterparts **almost exactly**



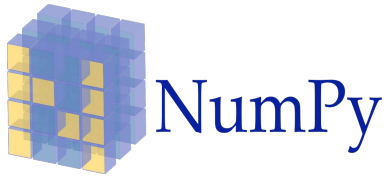
NCL and Python

NCL

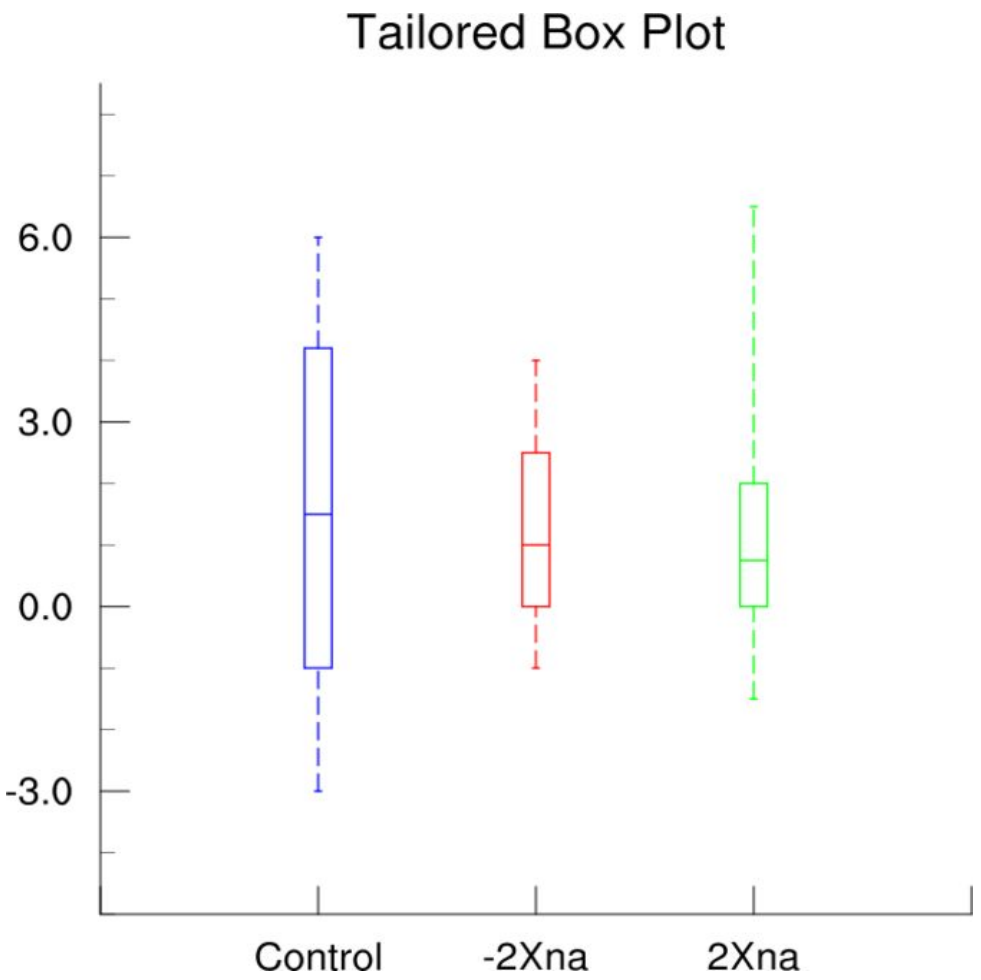
- Interpreted language
- Open-source
- Built specifically for data analysis and visualization in atmospheric and oceanic sciences
- Wrapper of niche computational routines and plotting procedures

Python

- Interpreted language
- Open-source
- Available for multi-purpose developments and/or computations
- Incorporation (imports) and manipulation of multiple packages is needed to achieve the same level of functionality

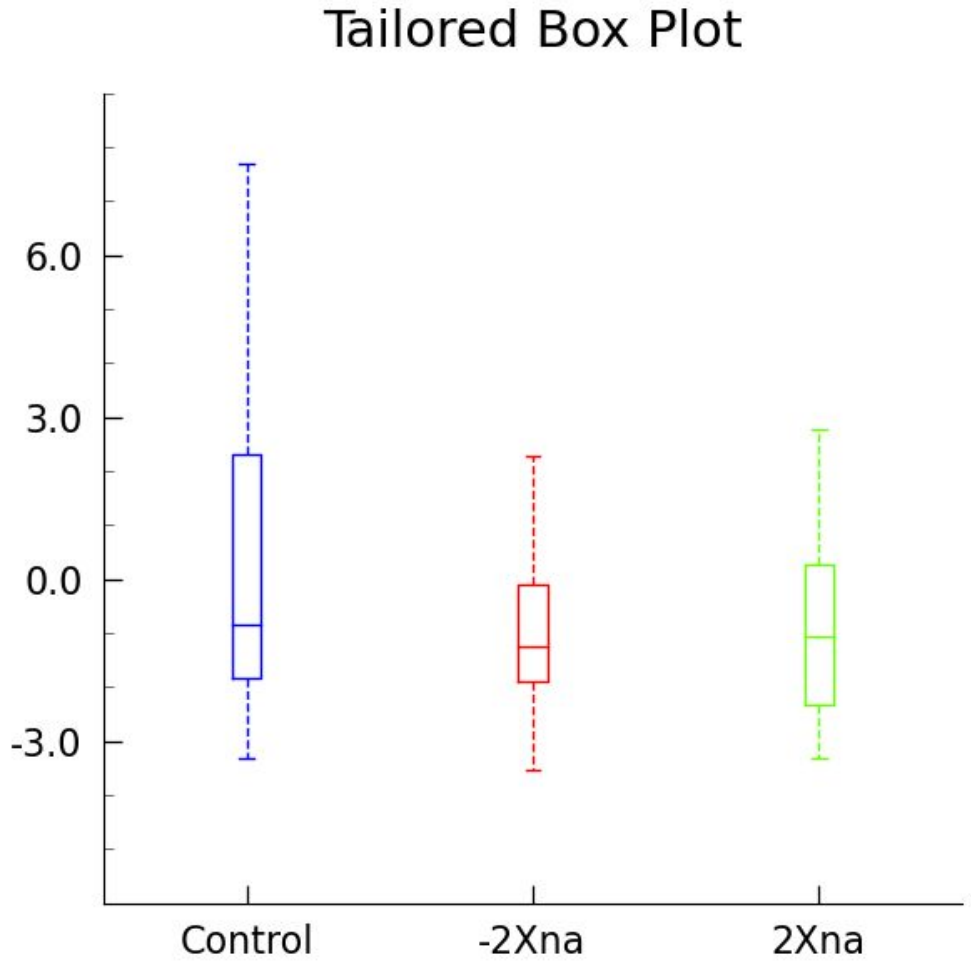


Contrast 1: Visual Differences without Functional Loss



Original NCL plot

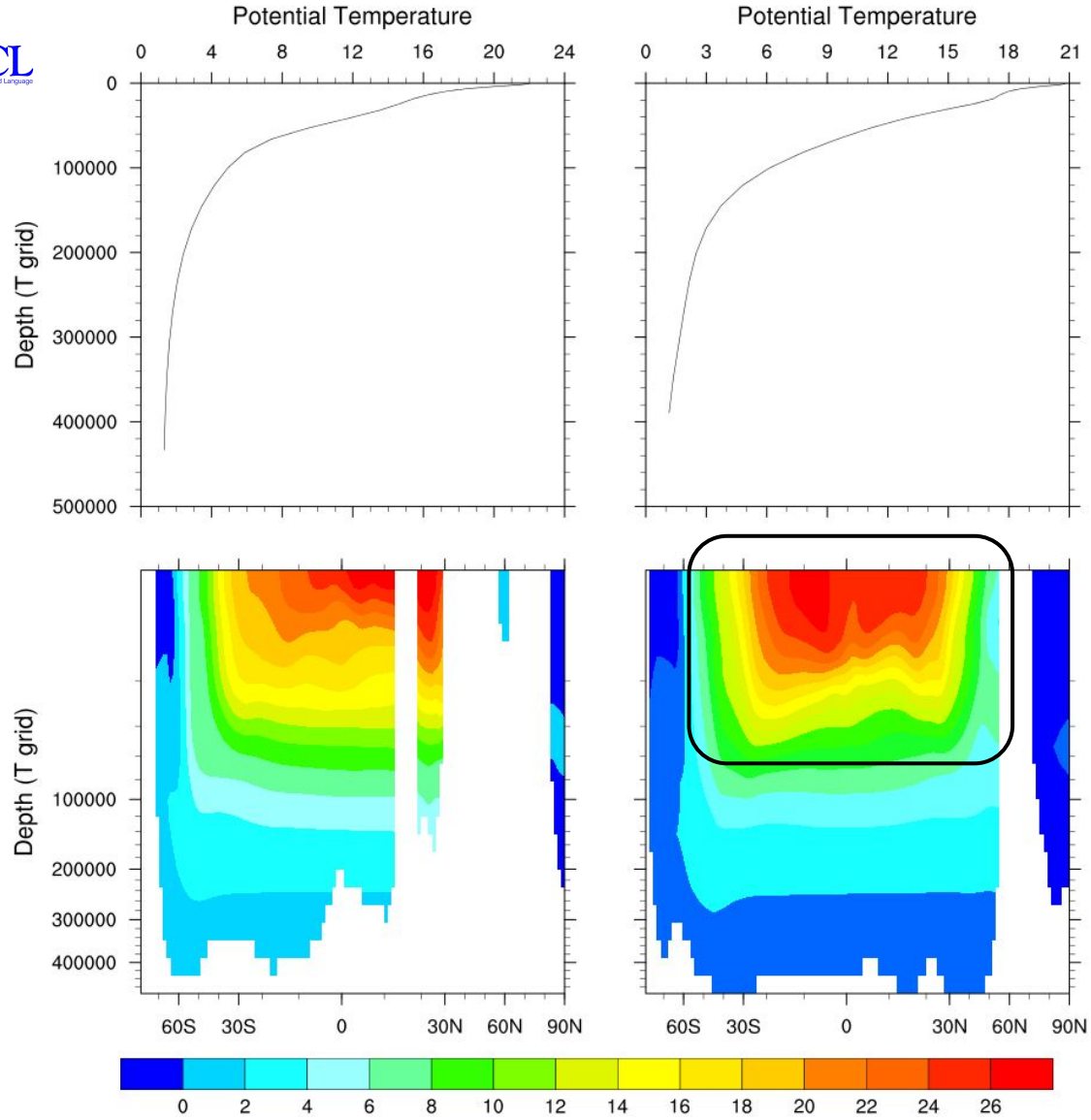
box_2.ncl



Plot Recreated

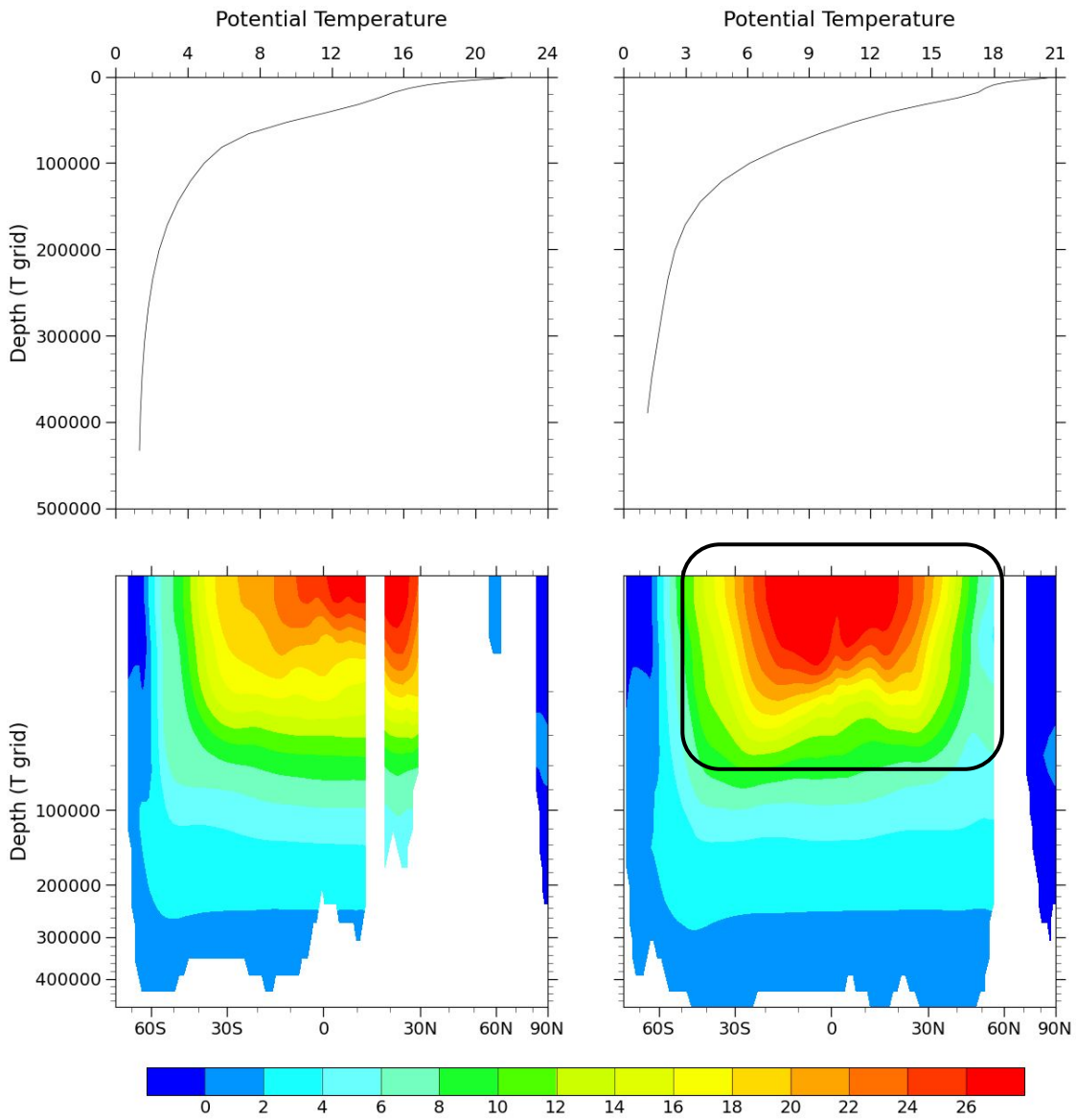
NCL_box_2.py

Contrast 2: NCL Implicit Axes Manipulation



Original NCL plot

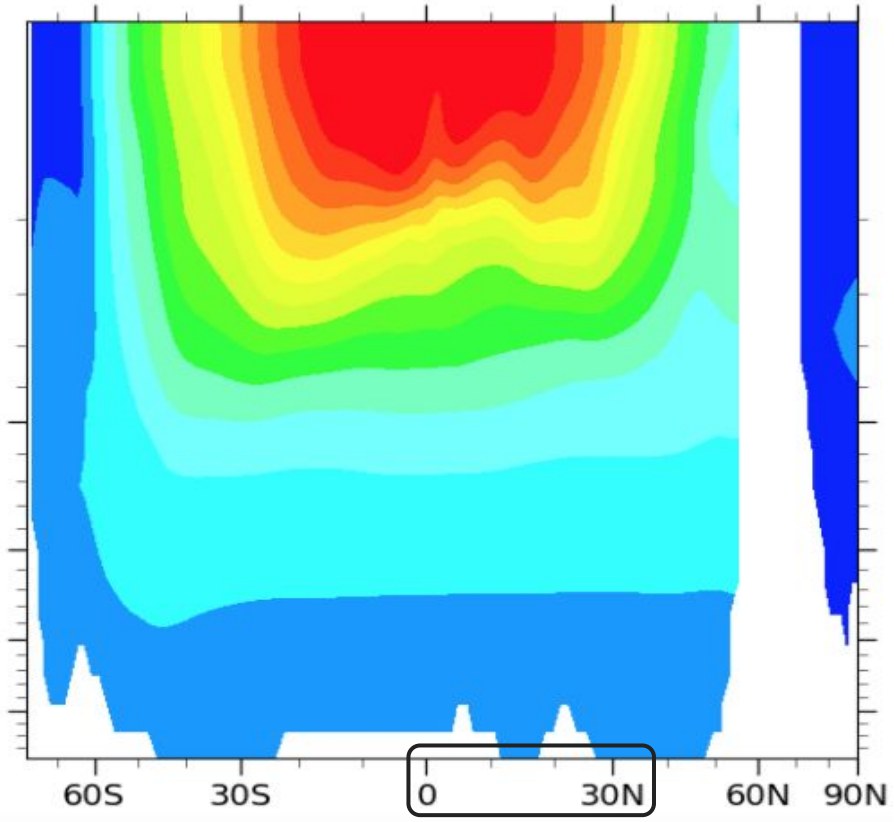
panel_14.ncl



Plot recreated

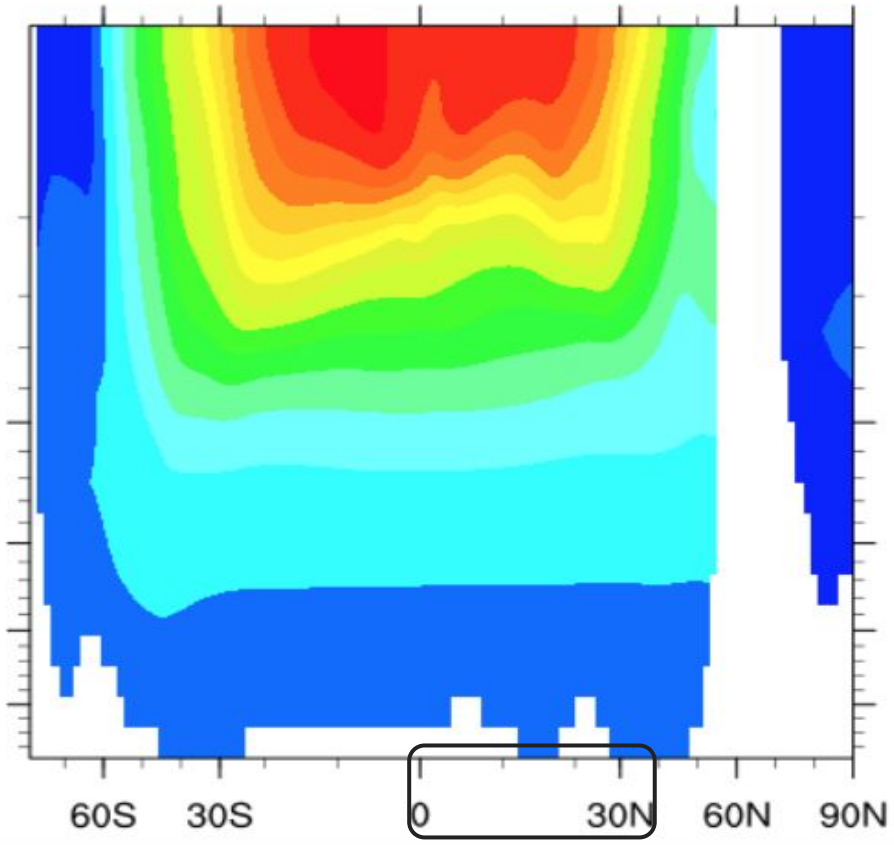
NCL_panel_14.py

Contrast 2: NCL Implicit Axes Manipulation



Original NCL plot

panel_14.ncl



Plot recreated

NCL_panel_14.py

Solutions for Implicit Axis Manipulation

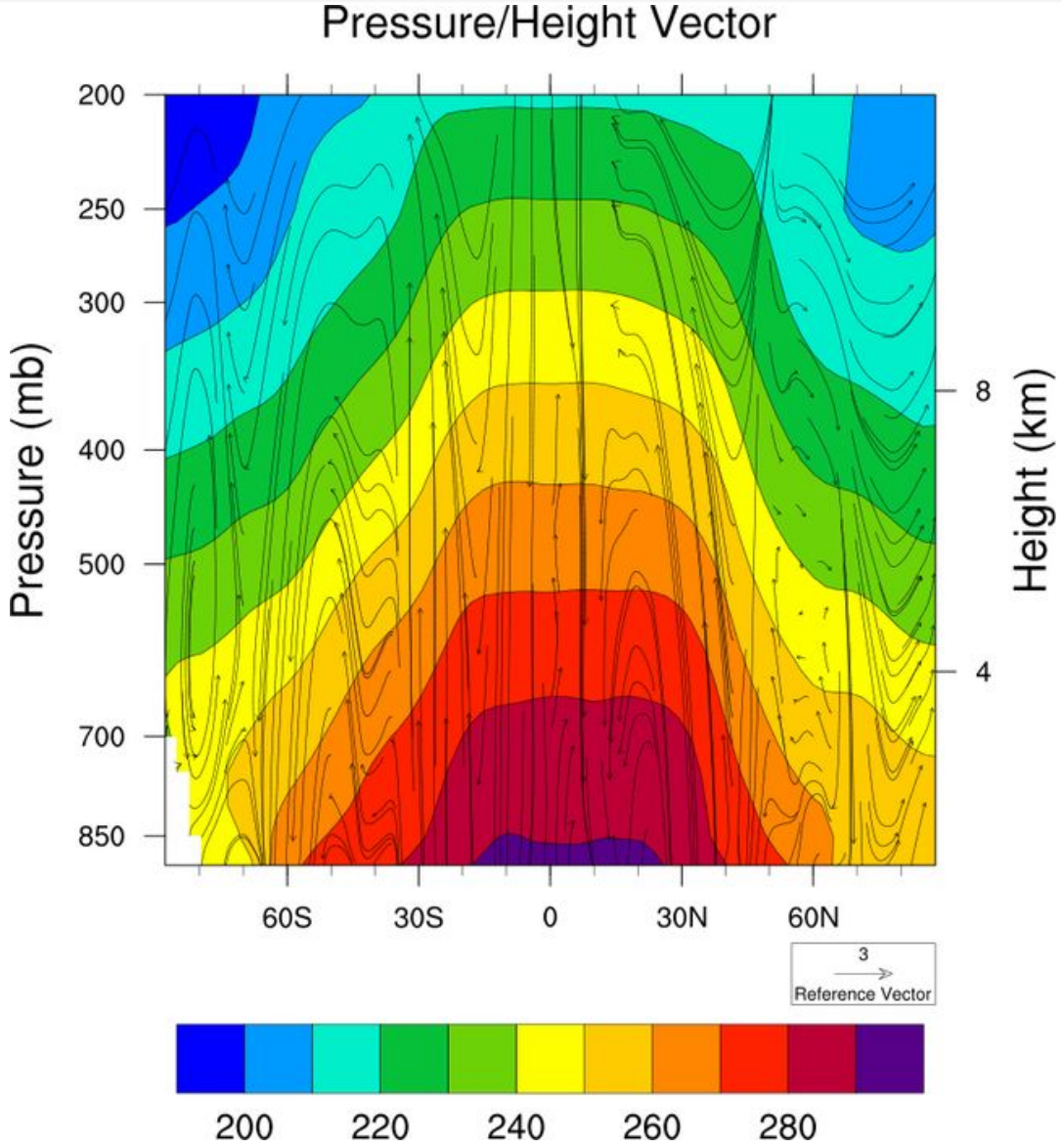
Current solution

```
# Define functions for axis scales
# Function x**(1/3) and its inverse
def yforward(x):
    return np.power(x, 1 / 3)

def yinverse(x):
    return np.power(x, 3 / 1)

# Function Mercator transform and its inverse
def xforward(a):
    a = np.deg2rad(a)
    return np.rad2deg(np.arctan(np.sinh(a)))

def xinverse(a):
    a = np.deg2rad(a)
    return np.rad2deg(np.log(np.abs(np.tan(a) + 1.0 /
np.cos(a))))


# Set scales of X axis and Y axis
axes.set_yscale('function', functions=(yforward, yinverse))
axes.set_xscale('function', functions=(xforward, xinverse))
```

Future effort

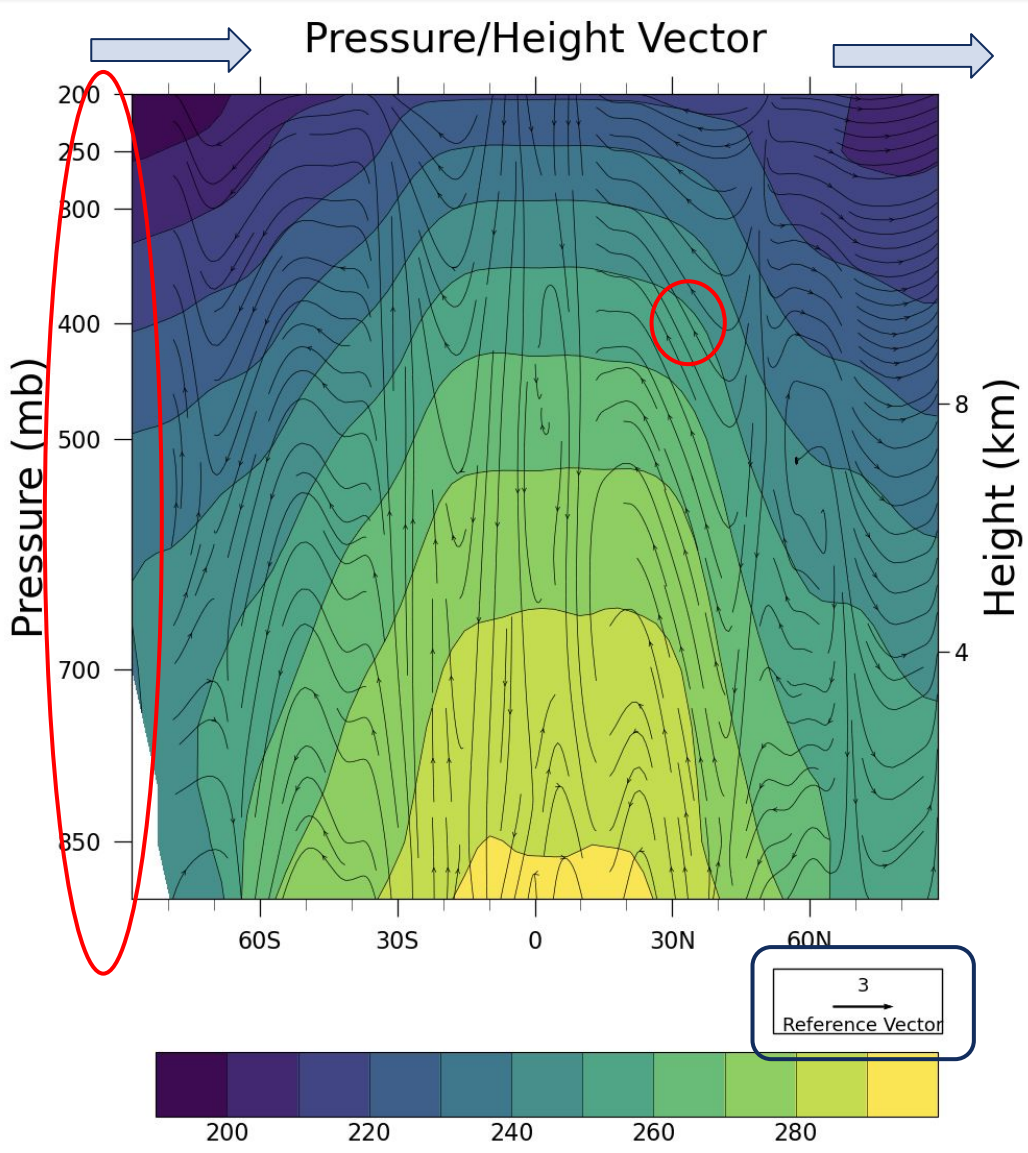
- Investigate the axis scaling routine in NCL
- Understand the reasoning for the particular axes scaling
- Choose to replicate similar routines or
  - discard axis scaling when not specified
- Wrap the functionality into a GeoCAT-Viz function

Contrast 3: NCL Curly Vector



Original NCL plot

vector_5.ncl



Plot recreated

NCL_vector_5.py

Solutions for Curly Vector

Current solution

- Use **plt.streamplot** function in replacement of **plt.quiver** function designated for straight vectors
- Use Scipy's interpolation routine (**interp2d**) to interpolate data sets onto an evenly spaced 2-dimensional grid
- Open up an issue with Matplotlib
- Inverse the y-velocities of the vectors dataset
- Change the density of streams

Future effort

- Further research into implementation of curly vector in Matplotlib
- Further research into best (efficient, suitable for data set) interpolation routine if Streamplot continues to be utilized
- Wrap the functionality into a GeoCAT-Viz function

Summary: Project Components

Motivation: Mismatches between NCL functionalities and Python ecosystem exist; a large amount of boilerplate code exists

GeoCAT-Examples

Attempt to create exact matches of corresponding NCL plots

Explore the strengths and weaknesses between plotting in Python and NCL

Serve as resources and teaching tools for scientific visualizations

GeoCAT-Viz

Wrappers around repeated codes for common plotting components

Wrappers around computational/plotting routines

GeoCAT Team!

Anissa Zacharias, Michaela Sizemore, Julia Kent, Orhan Eroglu, Alea Kootz

Erin Lincoln, Heather Craker

SIParCS Program!

Virginia Do, AJ Lauer, Jerry Cyccone

Max Cordes Galbraith