An easy-to-use visualization framework for earth system models.

eViz

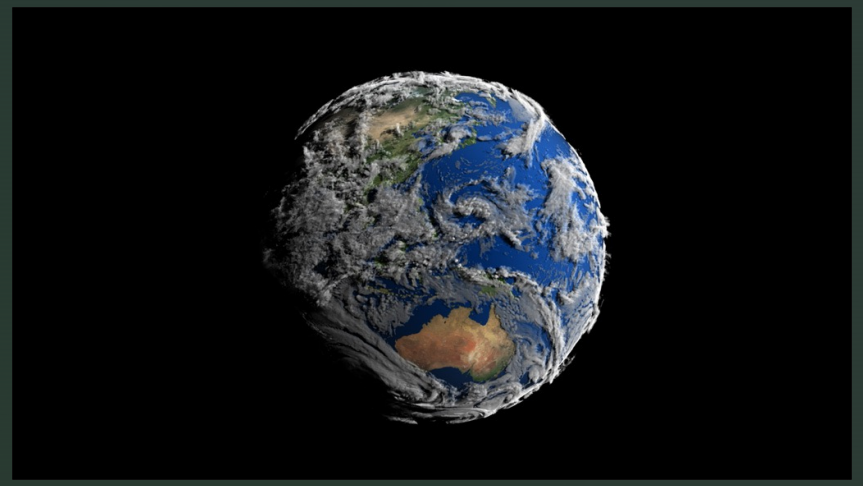
Carlos Cruz[1], Vanessa Valenti[1], Deepthi Raghunandan[3], Deon Kouatchou[3], Jules Kouatchou[1]

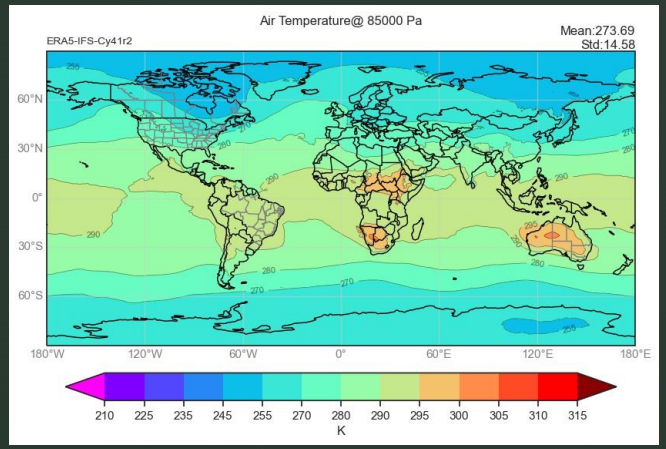
Megan Damon[1], Emma Knowland[2], Viral Shah[2]

[1]ASTG, [2]GMAO, [3]NASA/SSAI Interns

Visualization in Earth System Science

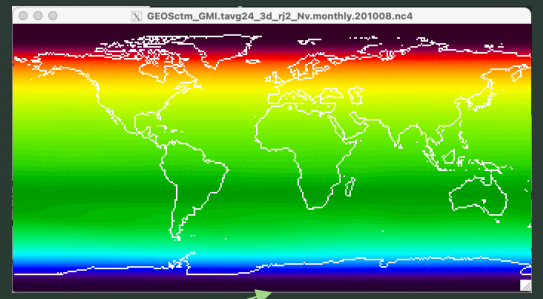
- Visualization is crucial step in the scientific process; plots can reveal issues and patterns in a model experiment, but often this occurs last due to the labor and time intensive current methods.

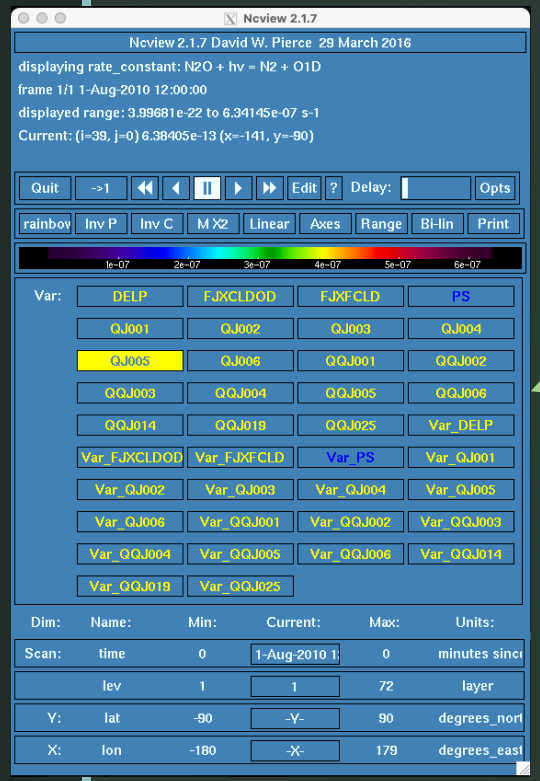




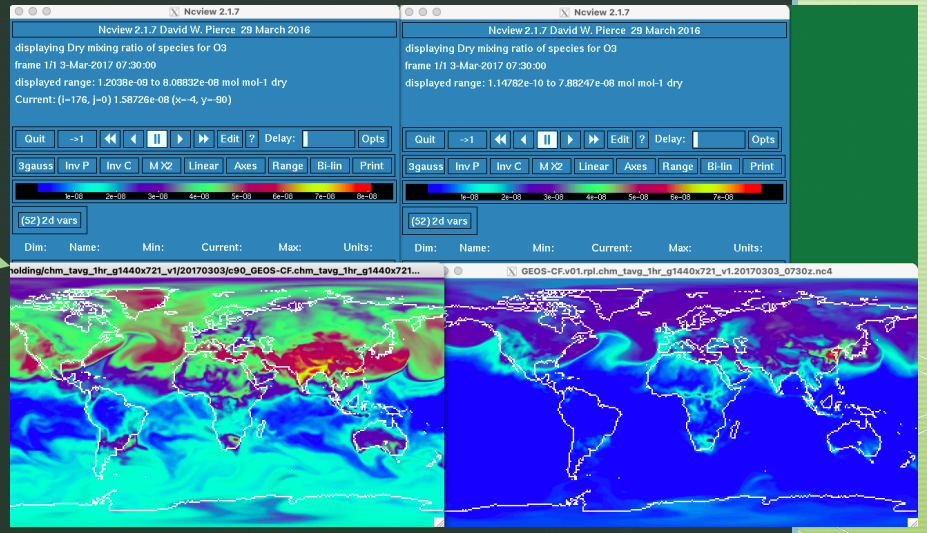
Current visualization methods

- **Current tools are limited by:**





- **Snapshots of data**
- **Non-interactive**
- **Not capable of comparing or analyzing**



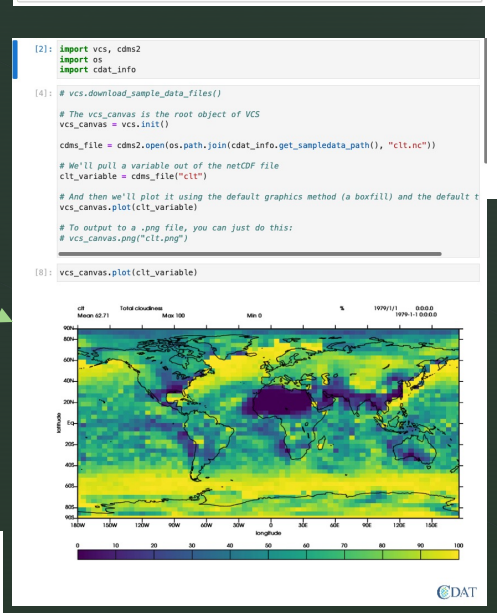
Current visualization methods

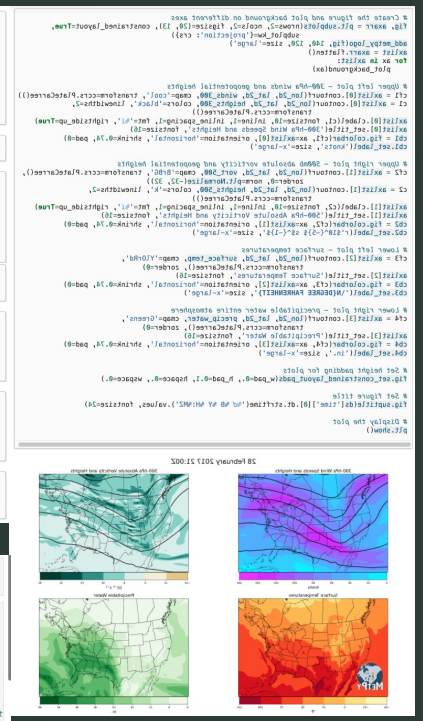
- **Current tools are limited by:**

Requires writing code and knowledge of scripts

Complex interfaces

```
python PlotCommonFields_GEOS-GMI.py  -c GEOSctm_GMI.tavg24_3d_dac_Nv.monthly.200408.nc4 -g gmic_HindcastMR2V2-GMIctm_2004_aug.amonthly.nc -r 0 -d 202008 -n PBS_NODEFILE -p 12 -s const_labels -v const -t Q
```





Current visualization methods

- **Current tools are limited by:**

Requires writing code and knowledge of scripts

Complex interfaces

```
python PlotCommonFields_GEOS-GMI.py  -c GEOSctm_GMI.tavg24_3d_dac_Nv.monthly.200408.nc4 -g gmic_HindcastMR2V2-GMIctm_2004_aug.amonthly.nc -r 0 -d 202008 -n PBS_NODEFILE -p 12 -s const_labels -v const -t Q
```

```
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.pyplot as plt
import numpy as np
import scipy.ndimage as ndimage
import xarray as xr

from metpy.cbook import get_test_data
from metpy.plots import add_metpy_logo
```

```
crs = ccrs.LambertConformal(central_longitude=-100.0, central_latitude=45.0)
```

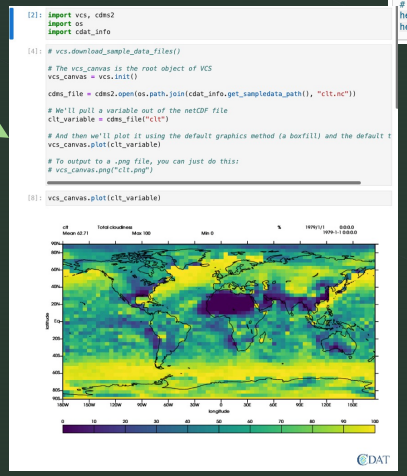
```
# Function used to create the map subplots
def plot_background(ax):
    ax.set_extent([235., 290., 20., 55.])
    ax.add_feature(cfeature.COASTLINE.with_scale('50m'), linewidth=0.5)
    ax.add_feature(cfeature.STATES, linewidth=0.5)
    ax.add_feature(cfeature.BORDERS, linewidth=0.5)
    return ax
```

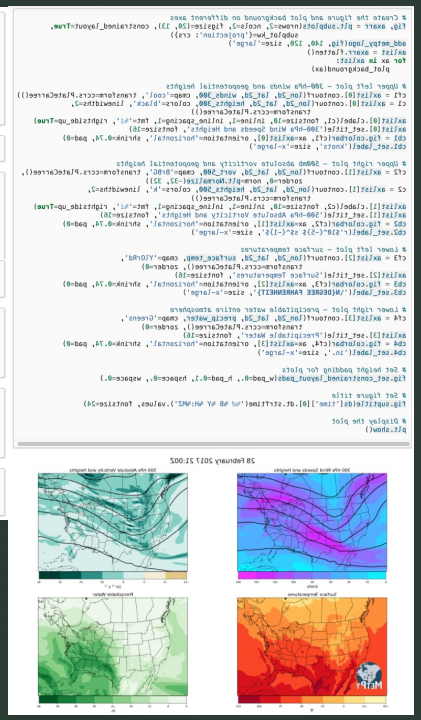
```
# Combine 1D latitude and longitudes into a 2D grid of locations
lon_2d, lat_2d = np.meshgrid(ds['lon'], ds['lat'])
```

```
# Pull out the data
vort_500 = ds['vort_500'][0]
surface_temp = ds['temp'][0]
precip_water = ds['precip_water'][0]
winds_300 = ds['winds_300'][0]
```

```
# Do unit conversions to what we wish to plot
vort_500 = vort_500 * 1e5
surface_temp = surface_temp.metpy.convert_units('degF')
precip_water = precip_water.metpy.convert_units('inches')
winds_300 = winds_300.metpy.convert_units('knots')
```

```
# Smooth the height data
heights_300 = ndimage.gaussian_filter(ds['heights_300'][0], sigma=1.5, order=0)
heights_500 = ndimage.gaussian_filter(ds['heights_500'][0], sigma=1.5, order=0)
```





Other key limitations:

Rigid frameworks that are difficult to extend or adapt to changing requirements

Model specific or not able to deal with multiple data formats

Two tools

eViz: a static plotting tool with a Command Line Interface (CLI) that is configurable with a YAML file.
(**Plotting is Matplotlib* based**)

iViz: a highly interactive visualization Graphical User Interface (GUI) that launches in a web browser via Jupyter Notebook or CLI.
(**Plotting is Holoviews* based**)

*Python packages

Two tools

eViz: a static plotting tool with a Command Line Interface (CLI) that is configurable with a YAML file.

iViz: a highly interactive visualization Graphical User Interface (GUI) that launches in a web browser via Jupyter Notebook or CLI.

- **Easy to use**
- **Model agnostic**
- **Portable**
- **HPC friendly**
- **Extensible**

Deep data exploration w/ highly interactive plotting

Diagnosing recently spun up model

Comparing model results w/ benchmark runs or reanalysis data

YAML based configuration
Support multiple data sources

Two tools

eViz: a static plotting tool with a Command Line Interface (CLI) that is configurable with a YAML file.

iViz: a highly interactive visualization Graphical User Interface (GUI) that launches in a web browser via Jupyter Notebook or CLI.

- Easy to use
- Model agnostic
- Portable
- HPC friendly
- Extensible

- Deep data exploration w/ highly interactive plotting

- Diagnosing recently spun up model
- Comparing model results w/ benchmark runs or reanalysis data

Modern Python

Two tools

eViz: a static plotting tool with a Command Line Interface (CLI) that is configurable with a YAML file.

iViz: a highly interactive visualization Graphical User Interface (GUI) that launches in a web browser via Jupyter Notebook or CLI.

- **Easy to use**
- **Model agnostic**
- **Portable**
- **HPC friendly**
- **Extensible**

Deep data exploration w/ highly interactive plotting

- **Diagnosing recently spun up model**
- **Comparing model results w/ benchmark runs or reanalysis data**

Can diagnose models earlier (run time) and more often.

Two tools

eViz: a static plotting tool with a Command Line Interface (CLI) that is configurable with a YAML file.

iViz: a highly interactive visualization Graphical User Interface (GUI) that launches in a web browser via Jupyter Notebook or CLI.

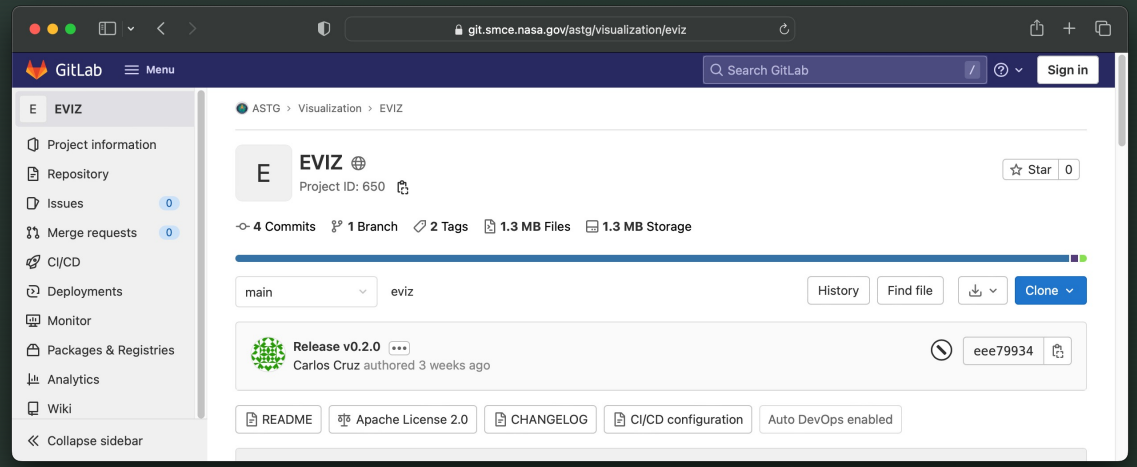
- Easy to use
- Model agnostic
- Portable
- HPC friendly
- Extensible

- Deep data exploration w/ highly interactive plotting

- Diagnosing recently spun up model
- Comparing model results w/ benchmark runs or reanalysis data

OO design

Open source

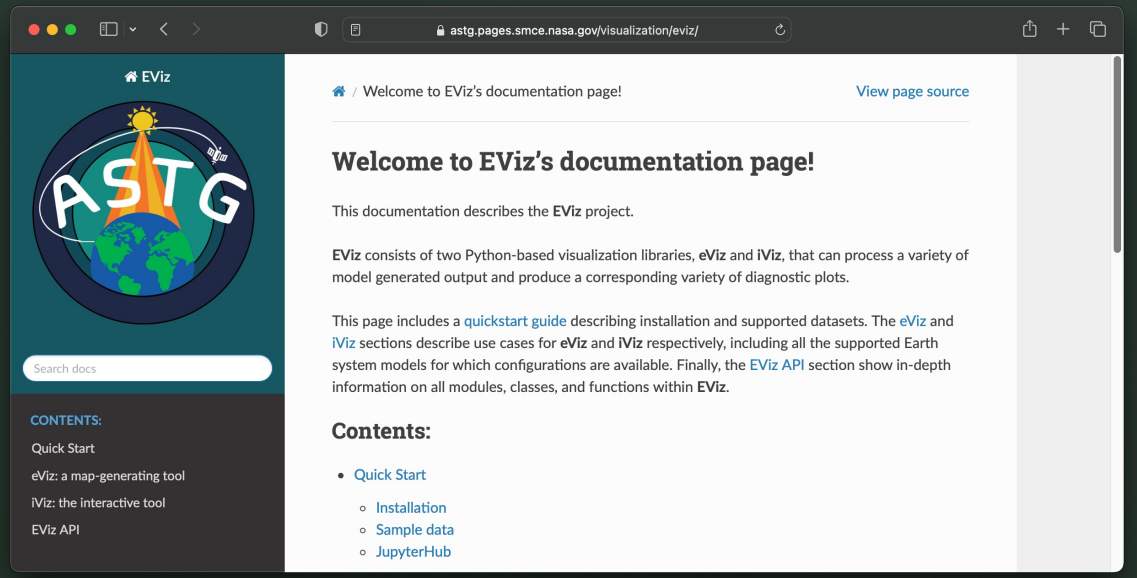


Current release: v0.2.2

For MAC and Linux

https://git.smce.nasa.gov/astg/visualization/eviz.git

Git repo and documentation hosted on SMCE



https://astg.pages.smce.nasa.gov/visualization/eviz/

Setup and use on a MAC

git clone ***https://git.smce.nasa.gov/astg/visualization/eviz.git***

cd eviz

conda env create –f cicd/environment.yaml

conda activate viz

python eviz.py –s ccm
python iviz.py –s ccm

CCM data **s**ource

ccm.yaml

```
1 inputs:
2    - name: /Users/ccruz/data/ccm/tavg24_3d_dat_Np.monthly.clim.JJA.nc4
3      to_plot:
4         co_25: xy
5         aoa: yz
6
```

Note: YAML files need to be configured before running eviz.py

Setup and use on DISCOVER

To use your own code base:

git clone ***https://git.smce.nasa.gov/astg/visualization/eviz.git***

cd eviz

module load anaconda

conda activate viz

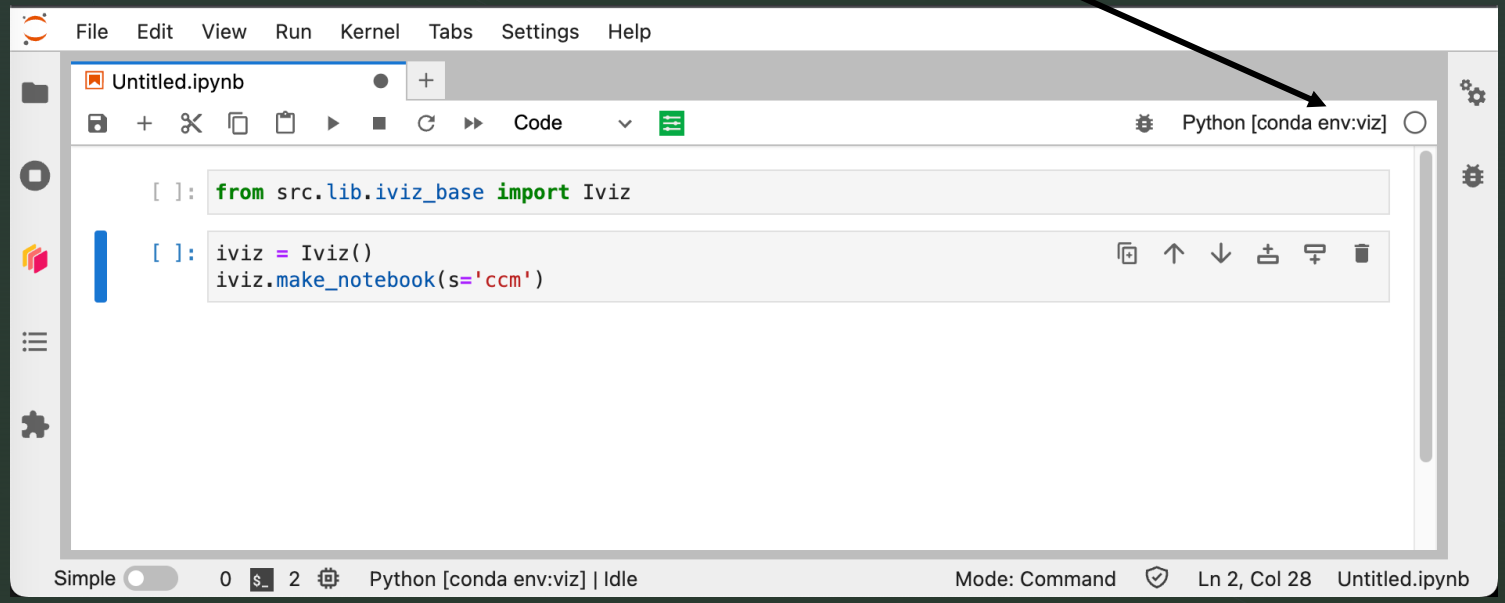
python eviz.py –s ccm
python iviz.py –s ccm <<< not recommended on DISCOVER

YAML files are already configured to visualize sample data

Setup and use on DISCOVER

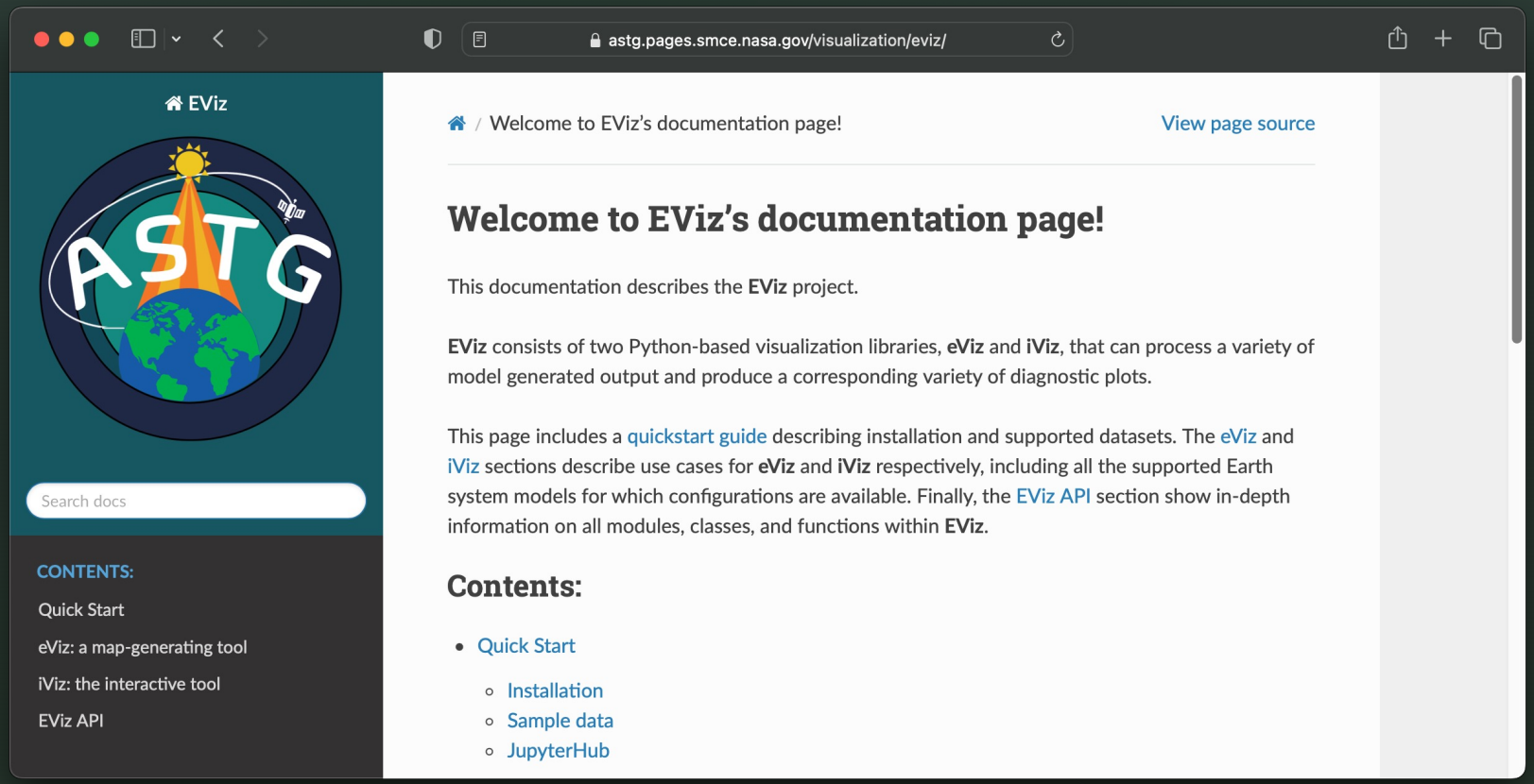
To use iViz on DISCOVER first log in to JupyterHub

- ***https://jh-discover.nccs.nasa.gov/***
- Open up a notebook and select the ***viz*** kernel



YAML files are already configured to visualize sample data

For more information, please visit the Eviz documentation page



https://astg.pages.smce.nasa.gov/visualization/eviz/

Demo