

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

eViz

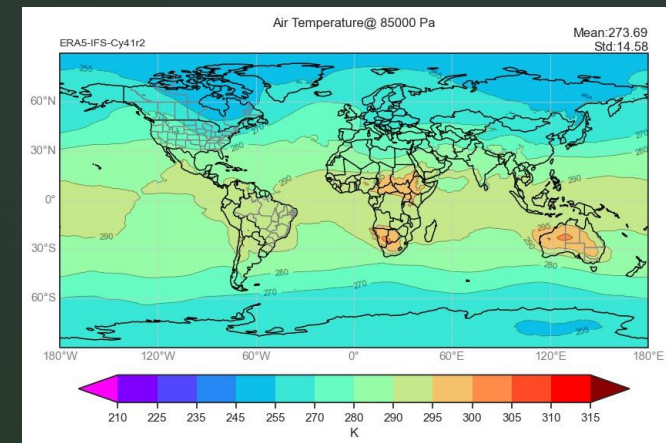
Carlos Cruz¹, Vanessa Valenti¹, Deepthi Raghunandan³, Deon Kouatchou³, Jules Kouatchou¹

Megan Damon¹, Emma Knowland², Viral Shah²

¹ASTG, ²GMAO, ³NASA/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 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
```

```
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 ID 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'][:10]
surface_temp = ds['temp'][:10]
precip_water = ds['precip_water'][:10]
winds_300 = ds['winds_300'][:10]

# 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'], sigma=1.5, order=0)
heights_500 = ndimage.gaussian_filter(ds['heights_500'], sigma=1.5, order=0)
```

```
(2): import vcs, cdms2
import os
import cdat_info

(4): # vcs.download_sample_data_files()

# The vcs_canvas is the root object of VCS
vcs_canvas = vcs.init()

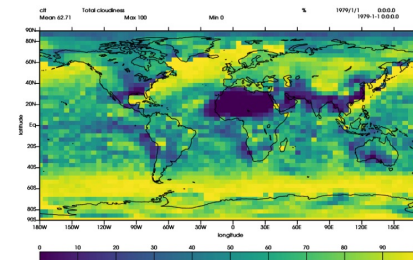
cdms_file = cdms2.open(os.path.join(cdat_info.get_sampledata_path(), "clt.nc"))

# We'll pull a variable out of the netCDF file
clt_variable = cdms_file("clt")

# And then we'll plot it using the default graphics method (a boxfill) and the default
vcs_canvas.plot(clt_variable)

# To output to a .png file, you can just do this:
# vcs_canvas.png("clt.png")

(8): vcs_canvas.plot(clt_variable)
```



```
from metpy.plots import add_metpy_logo

# Create the map subplots
fig, axes = plt.subplots(2, 2, figsize=(12, 12))

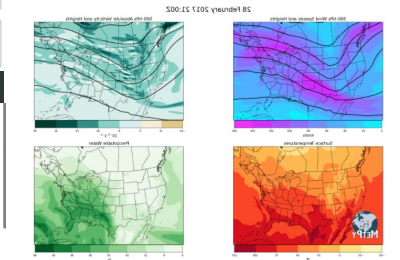
# Top-left: Vorticity at 500 hPa
vort_500 = ds['vort_500'][:10]
vort_500 = vort_500 * 1e5
axes[0,0].contour(vort_500, levels=[-1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

# Top-right: Surface temperature
temp = ds['temp'][:10]
temp = temp.metpy.convert_units('degf')
axes[0,1].contourf(temp, levels=[50, 60, 70, 80, 90, 100])

# Bottom-left: Precipitation at water
precip = ds['precip_water'][:10]
precip = precip.metpy.convert_units('inches')
axes[1,0].contourf(precip, levels=[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])

# Bottom-right: Winds at 300 hPa
winds = ds['winds_300'][:10]
winds = winds.metpy.convert_units('knots')
axes[1,1].contourf(winds, levels=[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])

# Add a color bar for the winds
cbar = plt.colorbar(axes[1,1].contourf, location='right', extend='both')
cbar.set_label('Winds (knots)')
```



Complex interfaces

**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

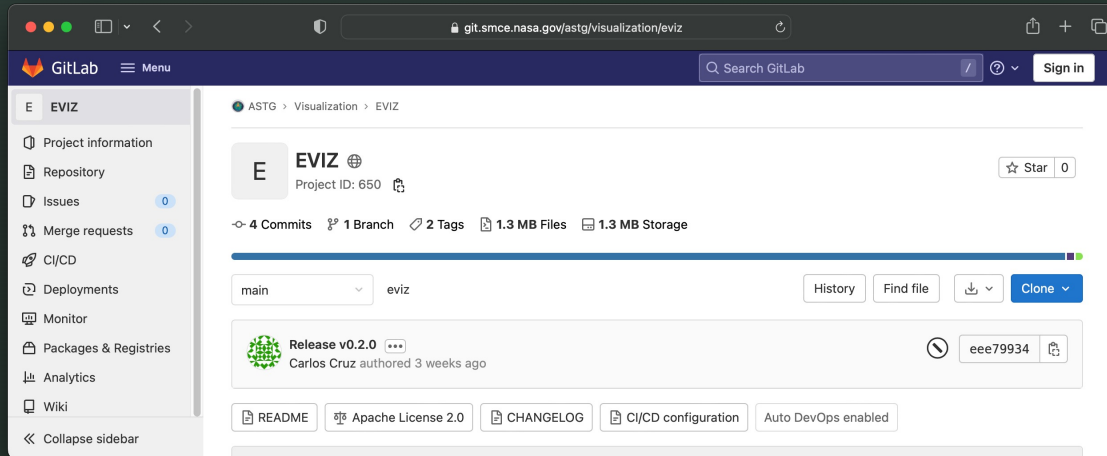
Deep data
exploration w/
highly interactive
plotting

Diagnosing
recently spun up
model

Comparing model
results w/
benchmark runs
or reanalysis data

OO design

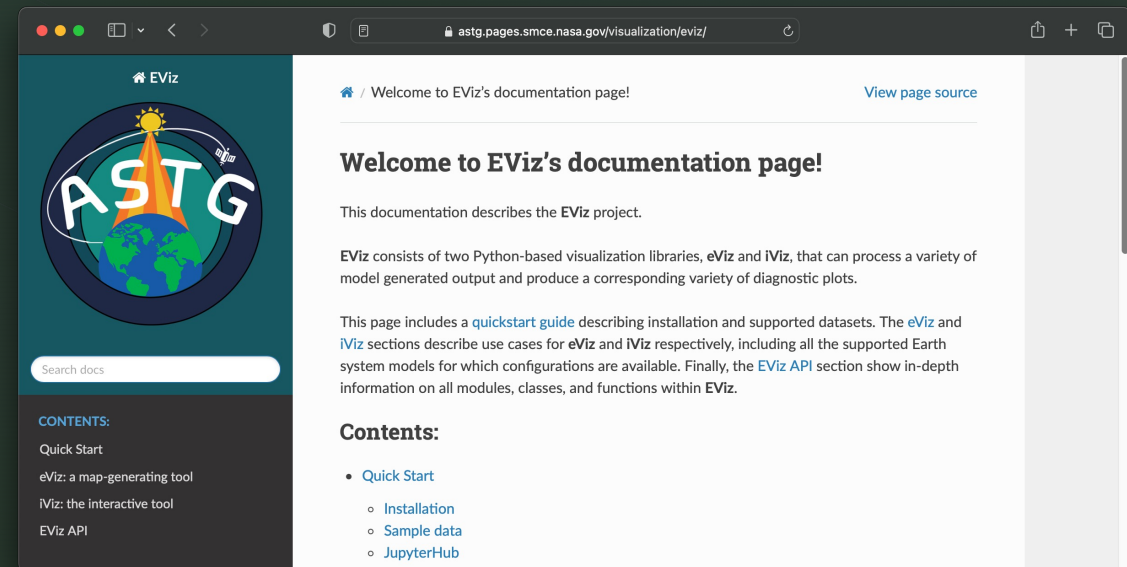
Open source



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

Git repo and documentation
hosted on SMCE

Current release: v0.2.2
For MAC and Linux



<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
```

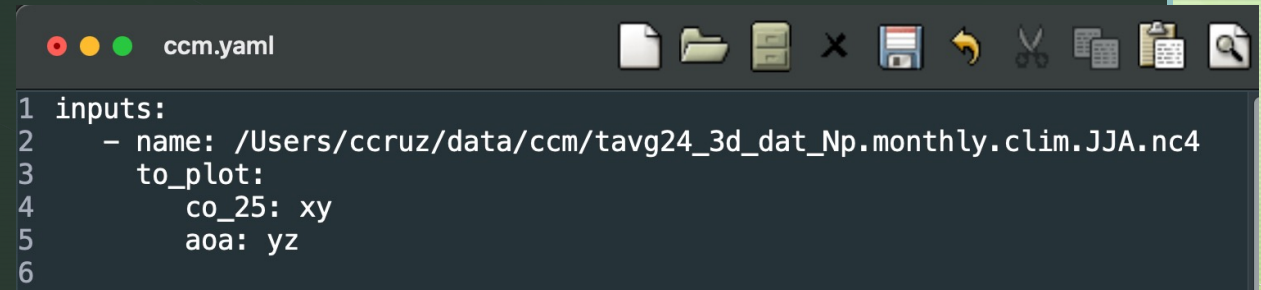
```
conda env create -f cicd/environment.yaml
```

```
conda activate viz
```

```
python eviz.py -s ccm
```

```
python iviz.py -s ccm
```

CCM data source



```
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
```

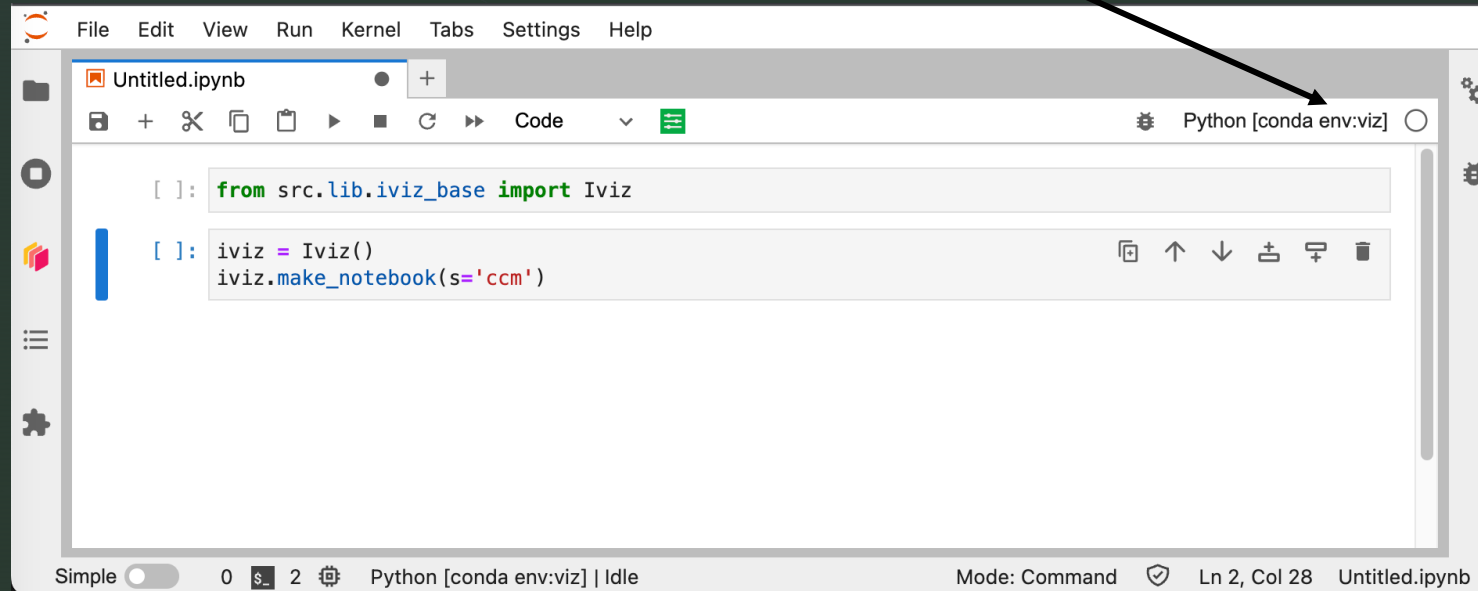
```
python iviz.py -s ccm <<< not recommended on DISCOVER
```

**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



The screenshot shows a Jupyter Notebook interface. The top menu bar includes File, Edit, View, Run, Kernel, Tabs, Settings, and Help. The notebook is titled 'Untitled.ipynb'. The kernel is set to 'Python [conda env:viz]'. The code in the notebook is as follows:

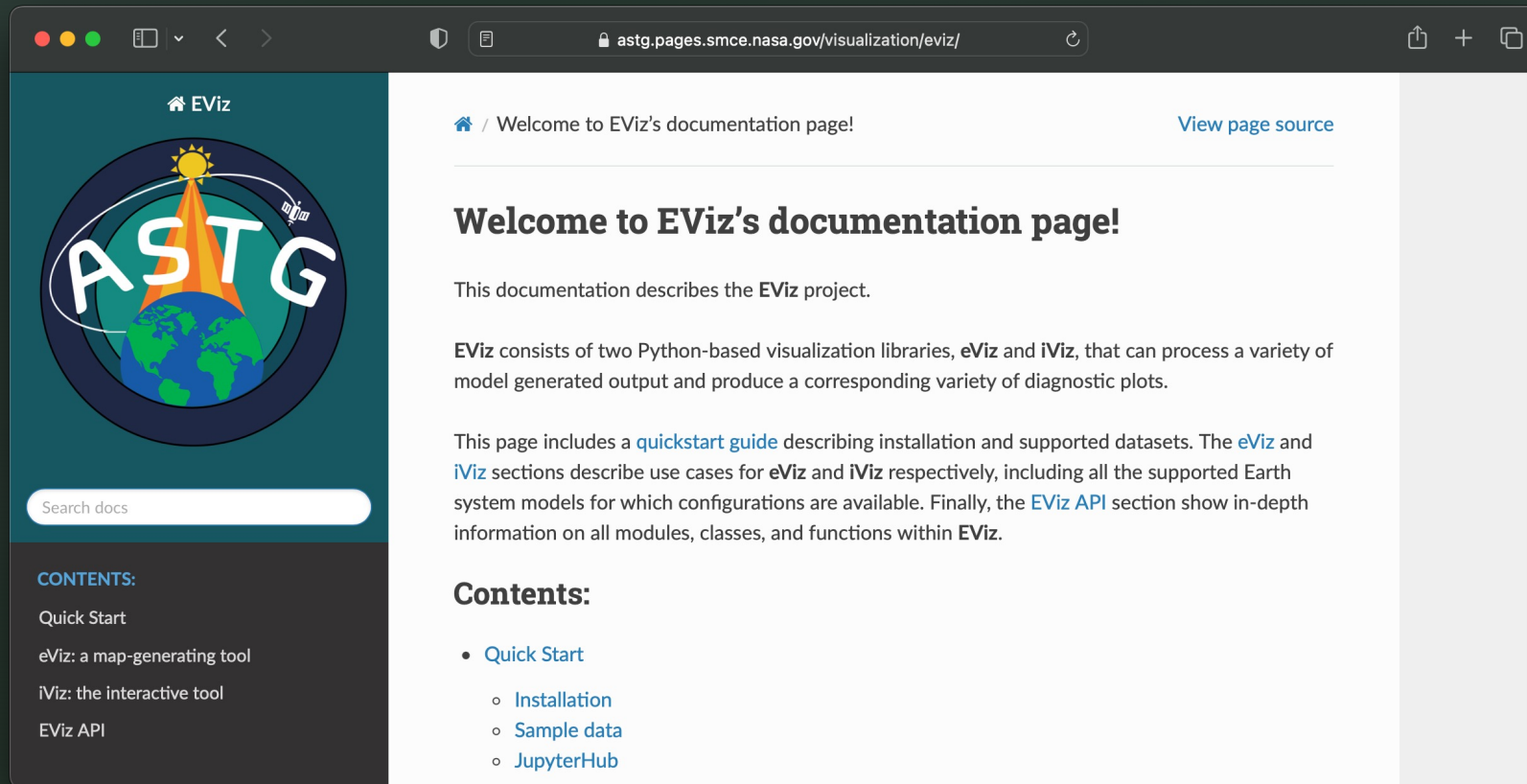
```
[ ]: from src.lib.iviz_base import Iviz

[ ]: iviz = Iviz()
      iviz.make_notebook(s='ccm')
```

An arrow points from the text 'select the viz kernel' to the kernel name 'viz' in the interface.

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