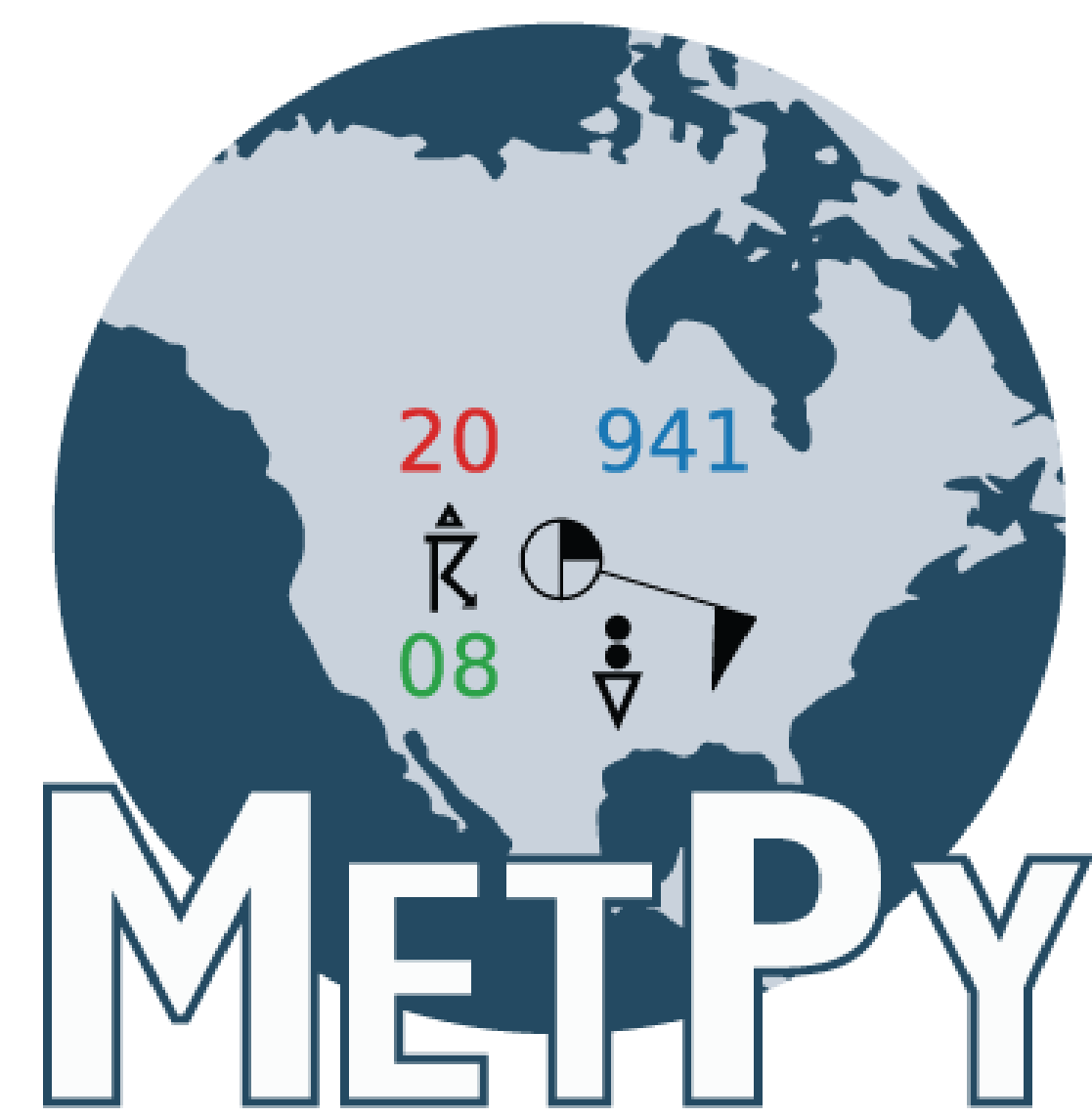


Enabling Declarative Syntax while using Matplotlib's pcolormesh in MetPy

By: Nathaniel Martinez

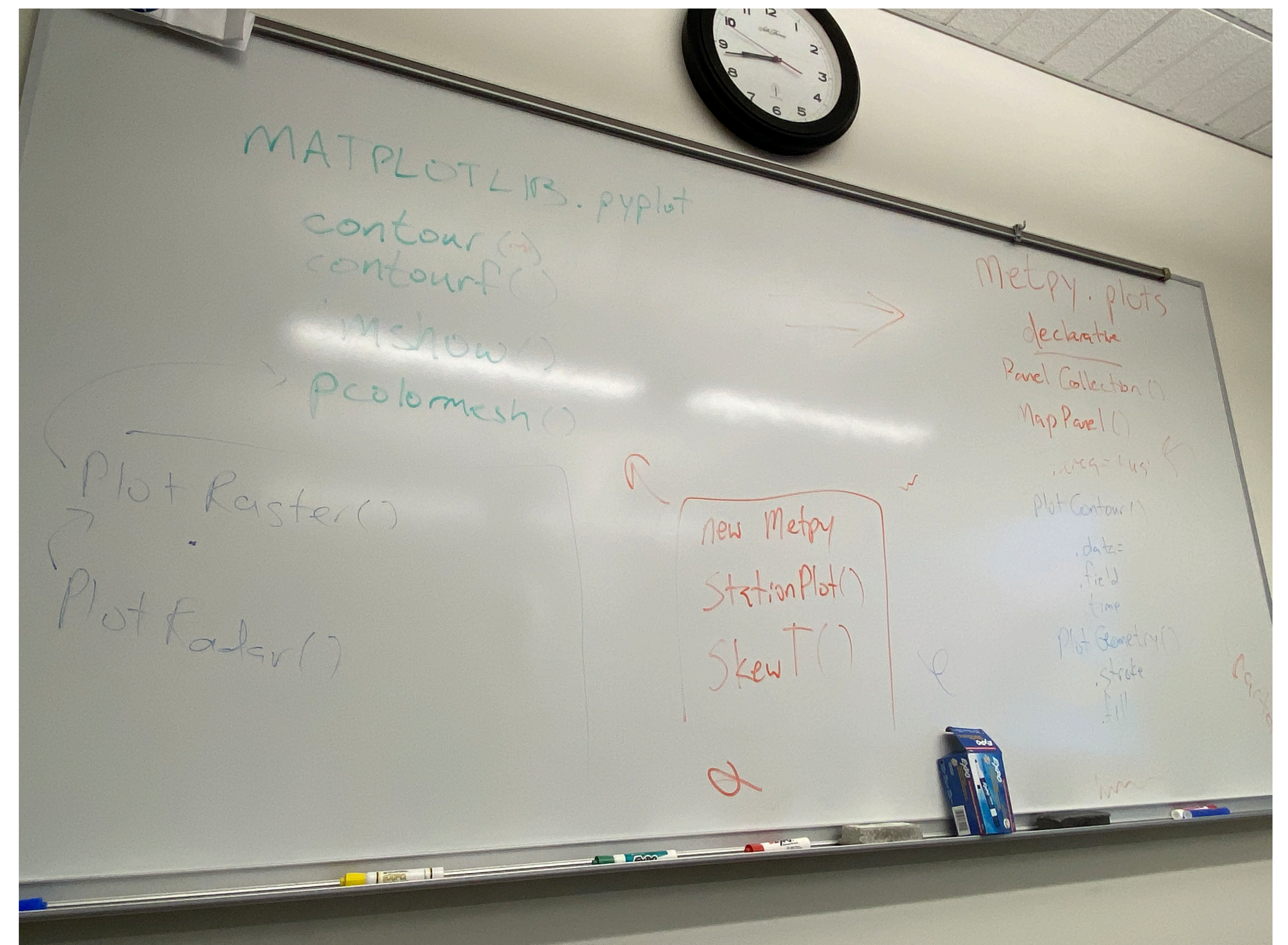


About Me!

- Rising 4th year at UChicago
- Computer Science & Environmental Science
- Interning since the end of May
 - Software development process
 - Adding declarative syntax for pcolormesh()
 - Documentation and example improvements

Software Development Process

- Planning
 - Identify the issue to be fixed or new functionality to be added
 - Identify stakeholders' software needs
- Design
 - Compile a design resolving the issues identified while planning



Software Development Process

- Implementation

- Draw from the design to write the code to fulfill the identified criteria

- Testing

- Create sufficient tests to ensure all new or edited code is verified to work properly

```
test_gempak.py
test_gini.py
test_metar.py
test_nexrad.py
test_station_data.py
test_tools.py
plots
  > __pycache__
  > baseline
test_cartopy_utils.py
test_ctables.py
test_declarative.py
test_mapping.py
test_mpl.py
test_skewt.py
test_station_plot.py
test_util.py
test_wx_symbols.py
units
test_cbook.py
test_deprecation.py
test_packaging.py
test_testing.py
test_xarray.py
```

Software Development Process

• Integration

- Create a pull request for the new code into the code repository, resolving any conflicts

Code changes using `azimuth_range_to_lat_lon` #2538

Merged dopplershift merged 2 commits into `Unidata:main` from `nmartinez233:azimuth_changes` 19 days ago

Conversation 13 Commits 2 Checks 25 Files changed 3

nmartinez233 commented on Jun 15 · edited

Changes made to resolve issue #2511:

- Included `azimuth_range_to_lat_lon` in the NEXRAD level 2 + 3 file examples
- Plotted counties on example images
- Updated documentation to reflect `max_range` is in kilometers

26 checks passed

- ✓ macOS 3.8 Details
- ✓ 3.8 requirements.txt Details
- ✓ CodeQL Details
- ✓ macOS 3.8 Details
- ✓ Flake8 Details
- ✓ 3.8 requirements.txt Details
- ✓ Windows 3.9 Details
- ✓ 3.9 requirements.txt Details

Declarative Syntax

- Simplifies plotting process
 - No need to call Matplotlib functions directly
 - Add support for pcolormesh

```
data = xr.open_dataset(get_test_data('narr_example.nc', as_file_obj=False))

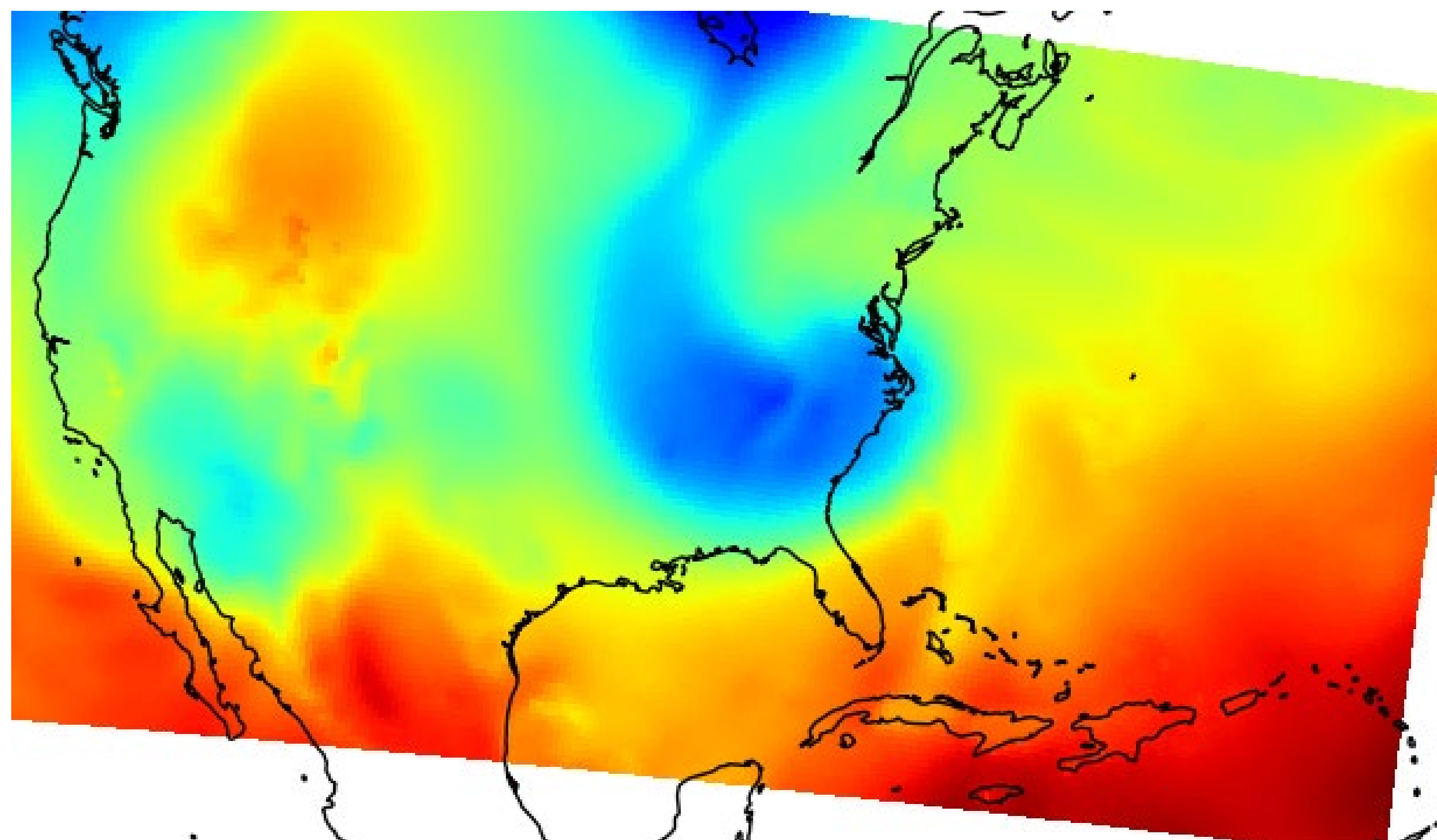
raster = RasterPlot()
raster.data = data
raster.field = 'Temperature'
raster.level = 700 * units.hPa

panel = MapPanel()
panel.area = 'us'
panel.projection = 'lcc'
panel.layers = ['coastline']
panel.plots = [raster]

pc = PanelContainer()
pc.size = (8.0, 8)
pc.panels = [panel]
pc.draw()
```

Raster Plots

- Plots a grid of values
 - Helpful in plotting key meteorological data
 - Temperature, wind speed, humidity, etc.
 - Potential for radar reflectivity plots



```
# Pull data out of the file
sweep = 0

# First item in ray is header, which has azimuth angle
az = np.array([ray[0].az_angle for ray in f.sweeps[sweep]])
diff = np.diff(az)
diff[diff > 180] -= 360.
diff[diff < -180] += 360.
avg_spacing = diff.mean()
az = (az[:-1] + az[1:]) / 2
az = np.concatenate(( [az[0] - avg_spacing], az, [az[-1] + avg_spacing] ))

# 5th item is a dict mapping a var name (byte string) to a tuple
# of (header, data array)
ref_hdr = f.sweeps[sweep][0][4][b'REF'][0]
ref_range = (np.arange(ref_hdr.num_gates + 1) - 0.5) * ref_hdr.gate_width + ref_hdr.first_gate
ref = np.array([ray[4][b'REF'][1] for ray in f.sweeps[sweep]])

rho_hdr = f.sweeps[sweep][0][4][b'RH0'][0]
rho_range = (np.arange(rho_hdr.num_gates + 1) - 0.5) * rho_hdr.gate_width + rho_hdr.first_gate
rho = np.array([ray[4][b'RH0'][1] for ray in f.sweeps[sweep]])
```

```
fig, axes = plt.subplots(1, 2, figsize=(15, 8))
add_metpy_logo(fig, 190, 85, size='large')
for var_data, var_range, ax in zip((ref, rho), (ref_range, rho_range), axes):
    # Turn into an array, then mask
    data = np.ma.array(var_data)
    data[np.isnan(data)] = np.ma.masked

    # Convert az, range to x, y
    xlocs = var_range * np.sin(np.deg2rad(az[:, np.newaxis]))
    ylocs = var_range * np.cos(np.deg2rad(az[:, np.newaxis]))

    # Plot the data
    ax.pcolormesh(xlocs, ylocs, data, cmap='viridis')
    ax.set_aspect('equal', 'datalim')
    ax.set_xlim(-40, 20)
    ax.set_ylim(-30, 30)
    add_timestamp(ax, f.dt, y=0.02, high_contrast=True)

plt.show()
```

Documentation Improvements

- Updating Documentation
 - Updated examples to reflect added functionality
 - Resolved example issues and demonstrated best practices for users to follow

```
# Grab azimuths and calculate a range based on number of gates
az = np.array(datadict['start_az'] + [datadict['end_az'][-1]])
rng = np.linspace(0, f.max_range, data.shape[-1] + 1)

# Grab azimuths and calculate a range based on number of gates,
# both with their respective units
az = units.Quantity(np.array(datadict['start_az'] + [datadict['end_az'][-1]]), 'degrees')
rng = units.Quantity(np.linspace(0, f.max_range, data.shape[-1] + 1), 'kilometers')

# Extract central latitude and longitude from the file
cent_lon = f.lon
cent_lat = f.lat

# Convert az, range to x, y
xlocs = rng * np.sin(np.deg2rad(az[:, np.newaxis]))
ylocs = rng * np.cos(np.deg2rad(az[:, np.newaxis]))
xlocs, ylocs = azimuth_range_to_lat_lon(az, rng, cent_lon, cent_lat)
```


Acknowledgments

- Thank you to Unidata, UCAR/NCAR/UCP for having me!
- Thank you to Drew Camron and Ryan May for their guidance and mentorship this summer!
- Thanks to Rhoen, Hassan, and the other interns for having a great summer together!

