pyDARN

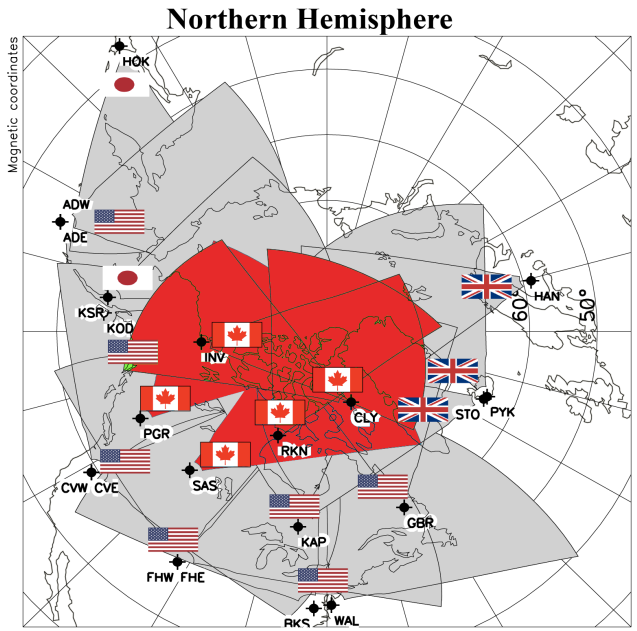
Marina Schmidt

June 17, 2019

Table of contents

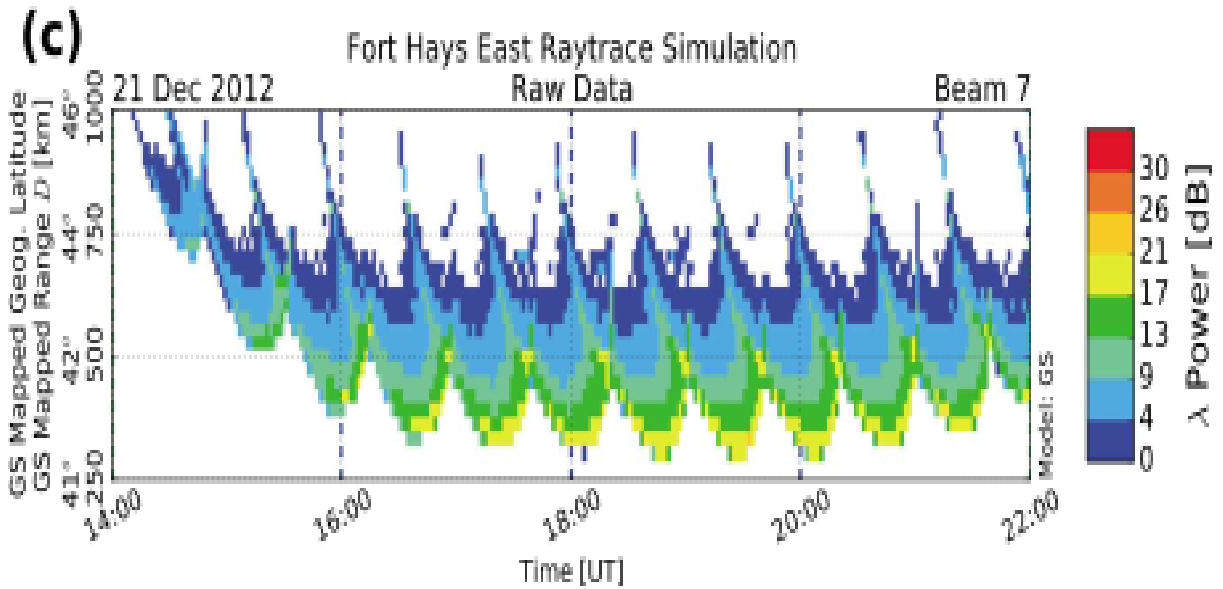
SuperDARN

Super Dual Auroral Radar Network - is an international network of high-frequency (HF) radars located throughout the northern and southern hemispheres.



Data Products and Plots

FITACF file → Range-Time Parameter plots (RTP)



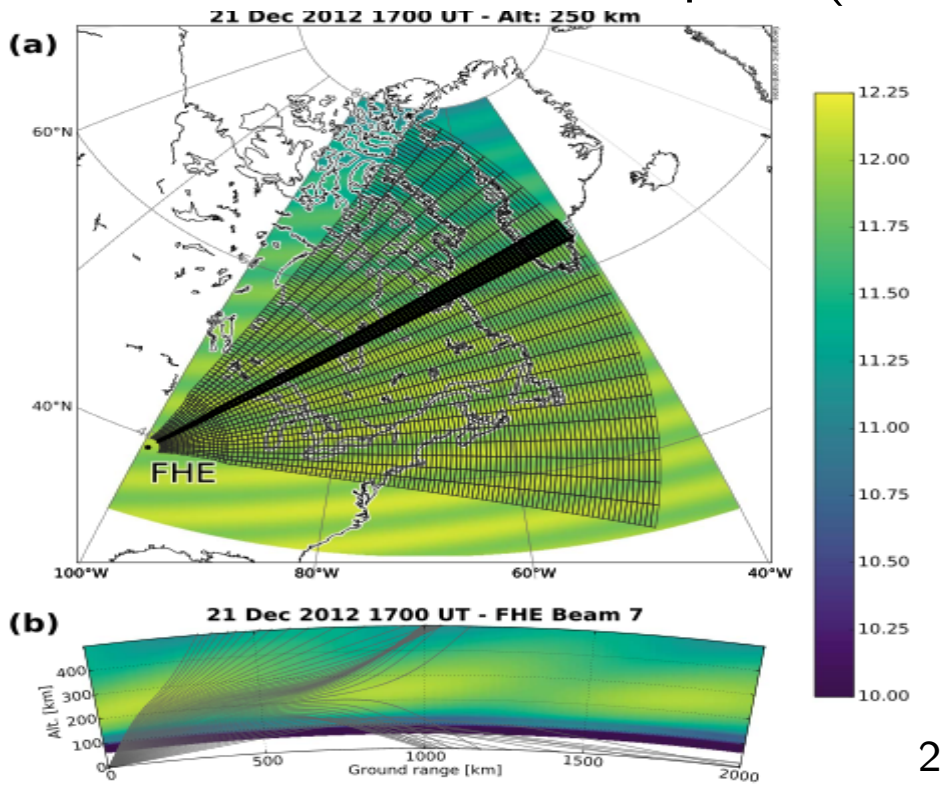
[1]

Commonly used to investigate: TID • Ionospheric patches • ULF waves

[1] Frissell et al. 2015

Data Products and Plots

FITACF file → Field of View plots (FOV)

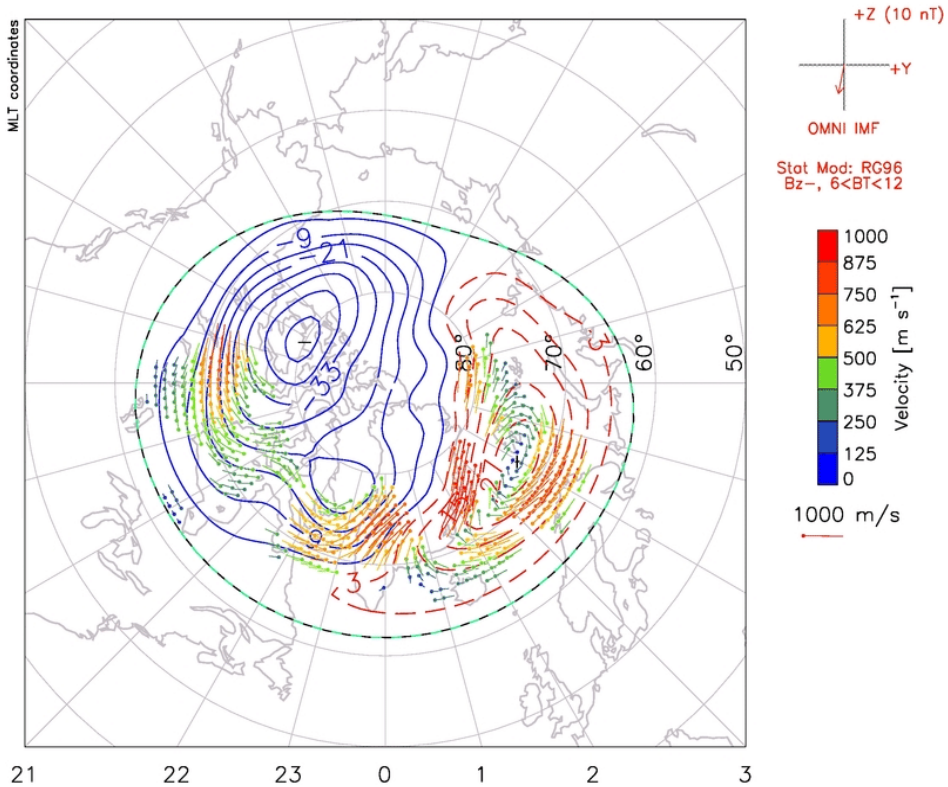


[2]

[2] Frissell et al. 2015

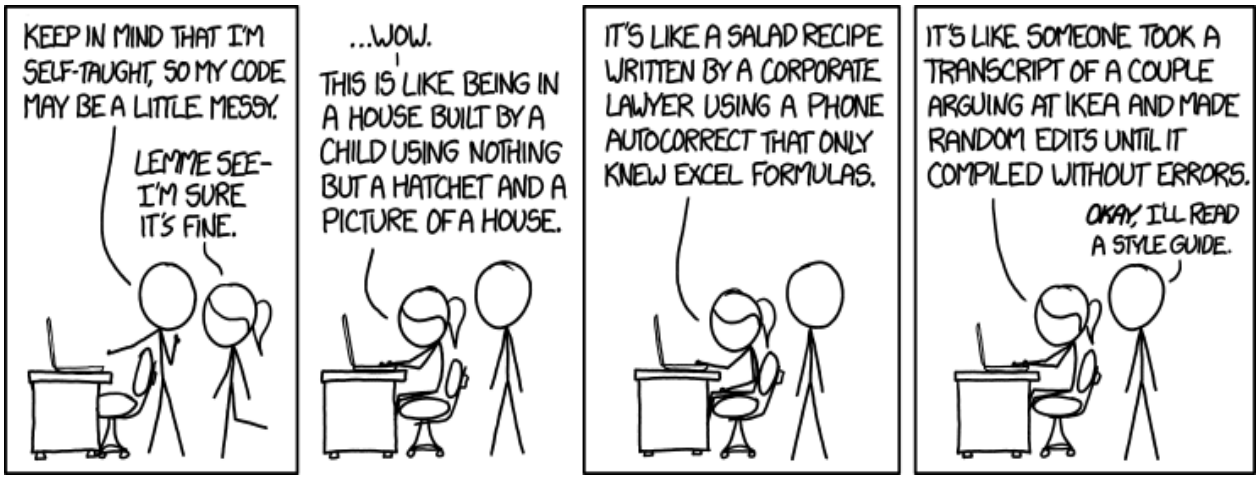
Data Products and Plots

Map file → Convection



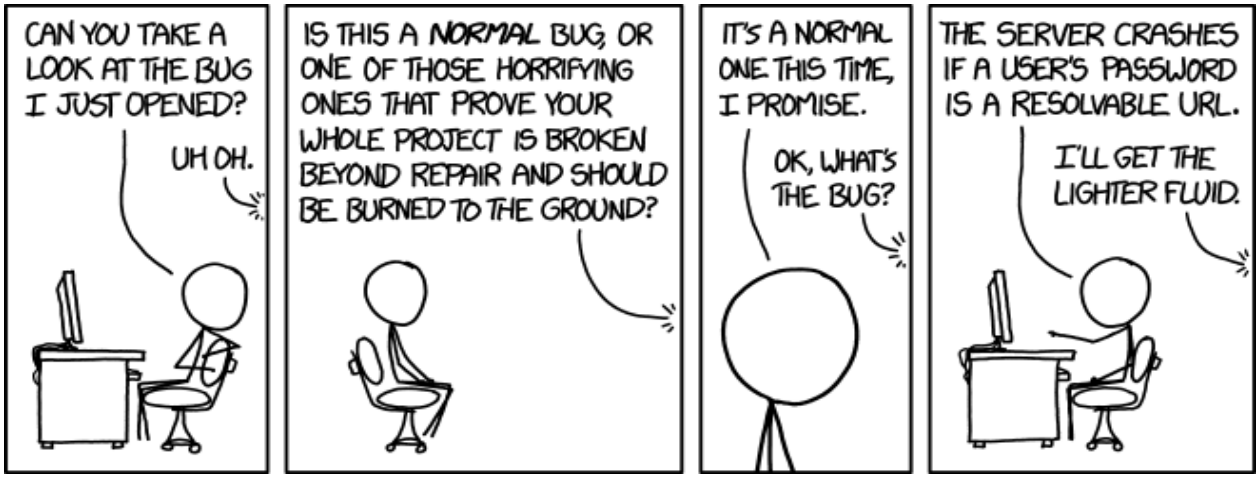
Commonly used to investigate: sub-storms • Storms • Auroral activity

Software Practices



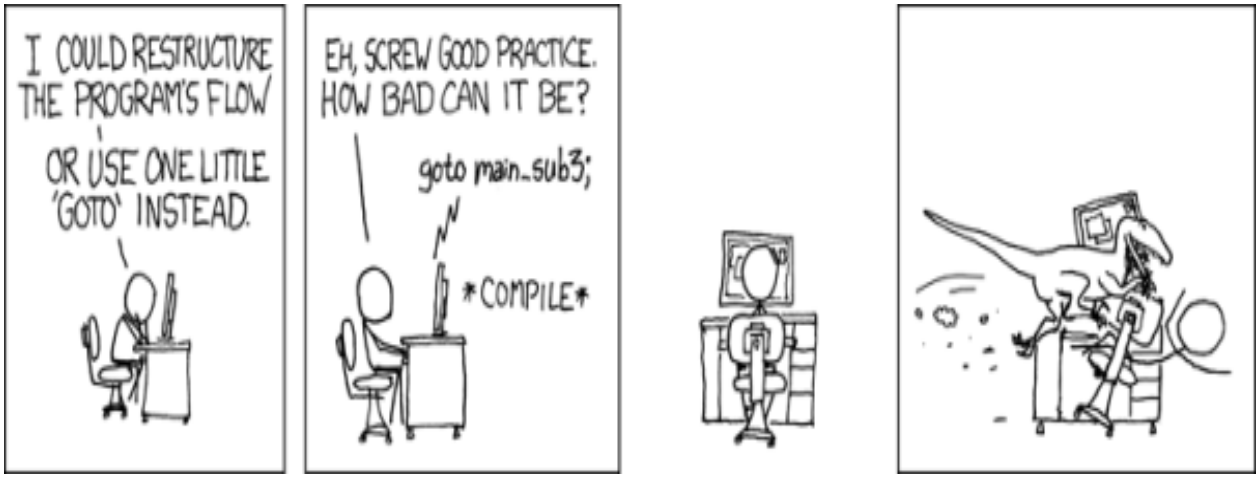
- Consistency
  - Style Guides - ex) PEP8 for python
  - Policies - How to structure the code?
  - Rules of the Road - What is the code development work flow?
  - Function/variable naming
- Code review - proof reading code to ensure practices and standards are being met
- Code readability - variable and function names that make sense to the audience

Software Practices



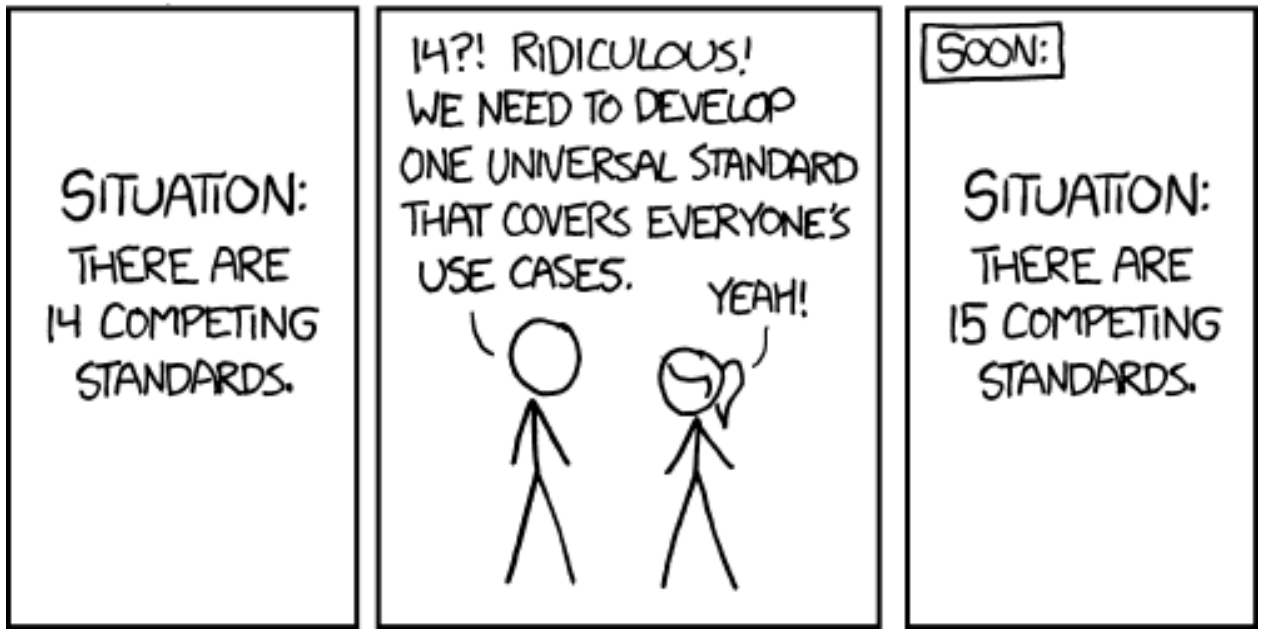
- Testing
  - Unit tests - testing a single unit of a function
  - Integration testing - testing multiple functions together
  - Bench marking - Performance analysis (speed, memory usage)
  - Coverage testing - testing all lines of code

Software Practices



- Flexibility/Extend-ability - To add/remove features without problems (de-coupling) and building off other code/packages/libraries
- Documentation

Software Practices



- Scope - Not trying to do all the science!
  - Goal statement
  - Policies
  - Decisions made by a team

What is pyDARN?

pyDARN

is a python library for SuperDARN data visualizations.

Reading SuprDARN FITACF file

- Readability - self-documenting code

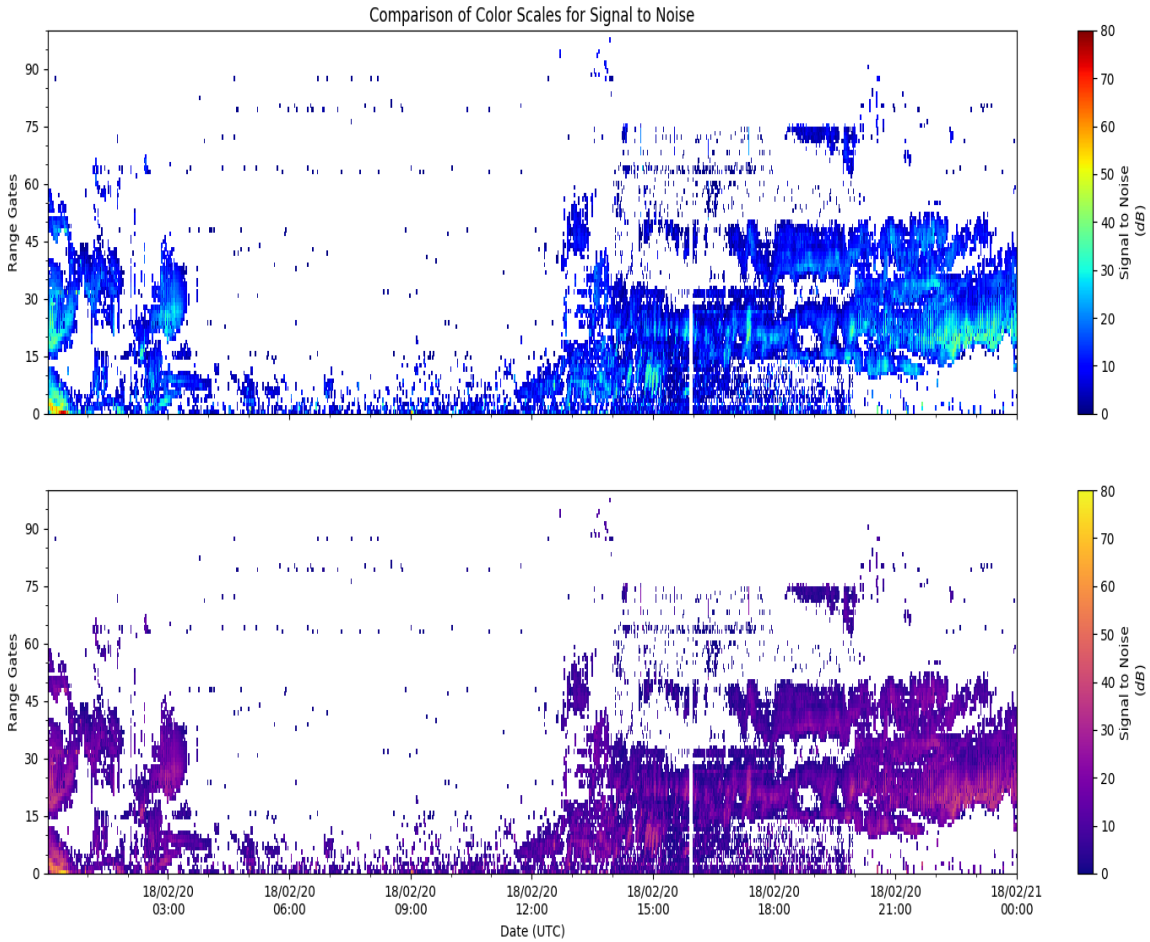
Reading FITACF file

```
import pydarn
import matplotlib.pyplot as plt
import matplotlib.dates as dates
from datetime import datetime

# Read in SuperDARN FITACF file
darn_read = pydarn.DarnRead("20180220.C0.rkn.fitacf")
data = darn_read.read_fitacf()
```

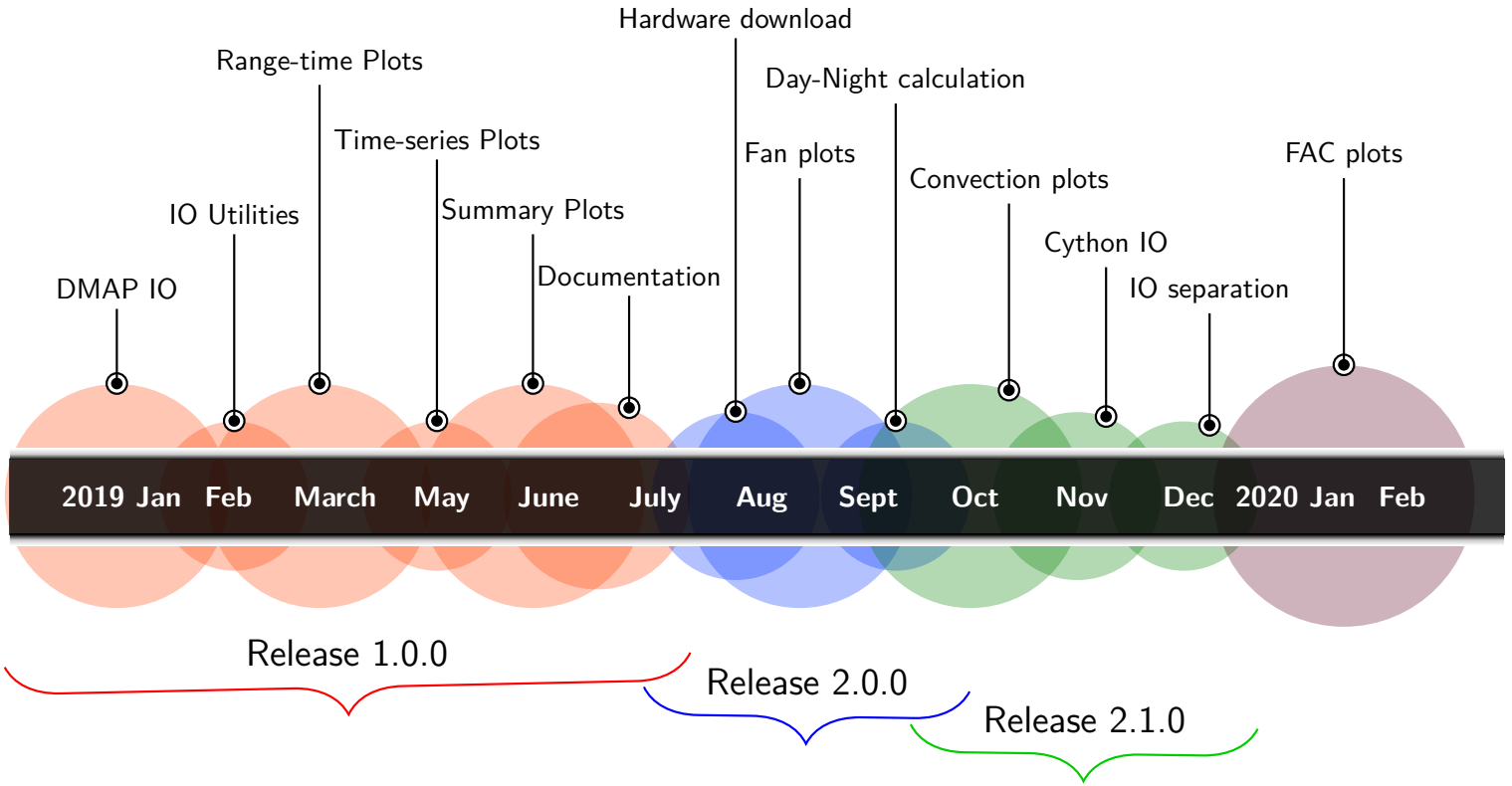
Plotting Range-Time Plots

```
# Create matplotlib figure and axes object
fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True)

# example of dependency injection where the matplotlib
# axes object is being injected into the code to extend
# the ability of matplotlib and utilize the axes object
pydarn.RTP.plot_range_time(data, beam_num=7, parameter='p_l', ax=ax1,
                           boundary=(0, 80),
                           color_bar_label="Signal to Noise\n ($dB$)")

# change the date format to HH:MM
ax2.xaxis.set_major_formatter(dates.DateFormatter('%H:%M'))

#change color map to plasma
pydarn.RTP.plot_range_time(data, beam_num=7, parameter='p_l',
                           color_map='plasma', boundary=(0,80),
                           ax=ax2, color_bar_label="Signal to Noise\n ($dB$)")

ax2.set_ylabel("Range Gates")
ax1.set_ylabel("Range Gates")
ax2.set_xlabel("Date (UTC)")
ax1.set_title("Comparison of Color Scales for Signal to Noise")
plt.show()
```

Plotting Range-Time Plots



pyDARN Timeline



Goals:

- Simplicity
- User/Developer friendly
- Narrow scope

Thank You

- Ashton Reimer - SRI International
- Keith Kotyk - SuperDARN Canada, University of Saskatchewan
- Angeline Burell - NRL