

# Implementing an Observation Support System for the Data Assimilation Research Testbed



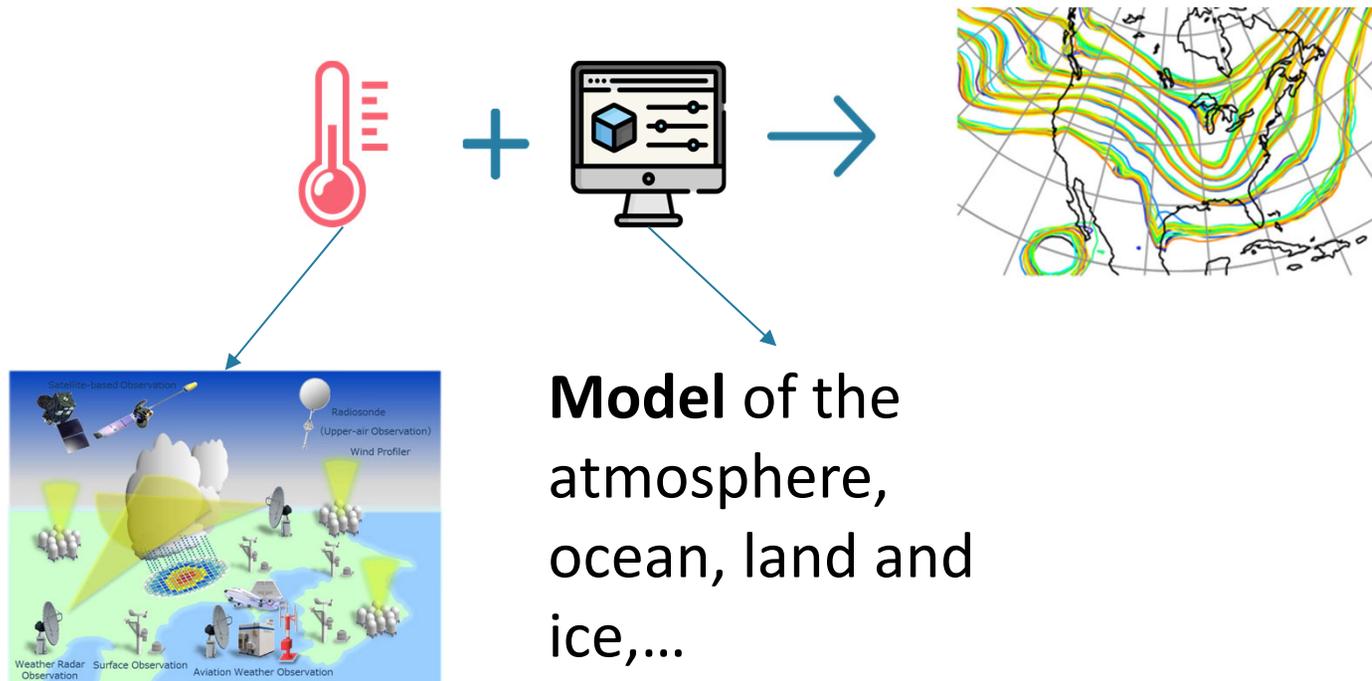
**JASON NGO**  
Haverford College  
August, 2019



# DATA ASSIMILATION

What is Data Assimilation?

- Observations statistically combined with Model Forecast to produce an analysis
- Widely used in numerical weather prediction



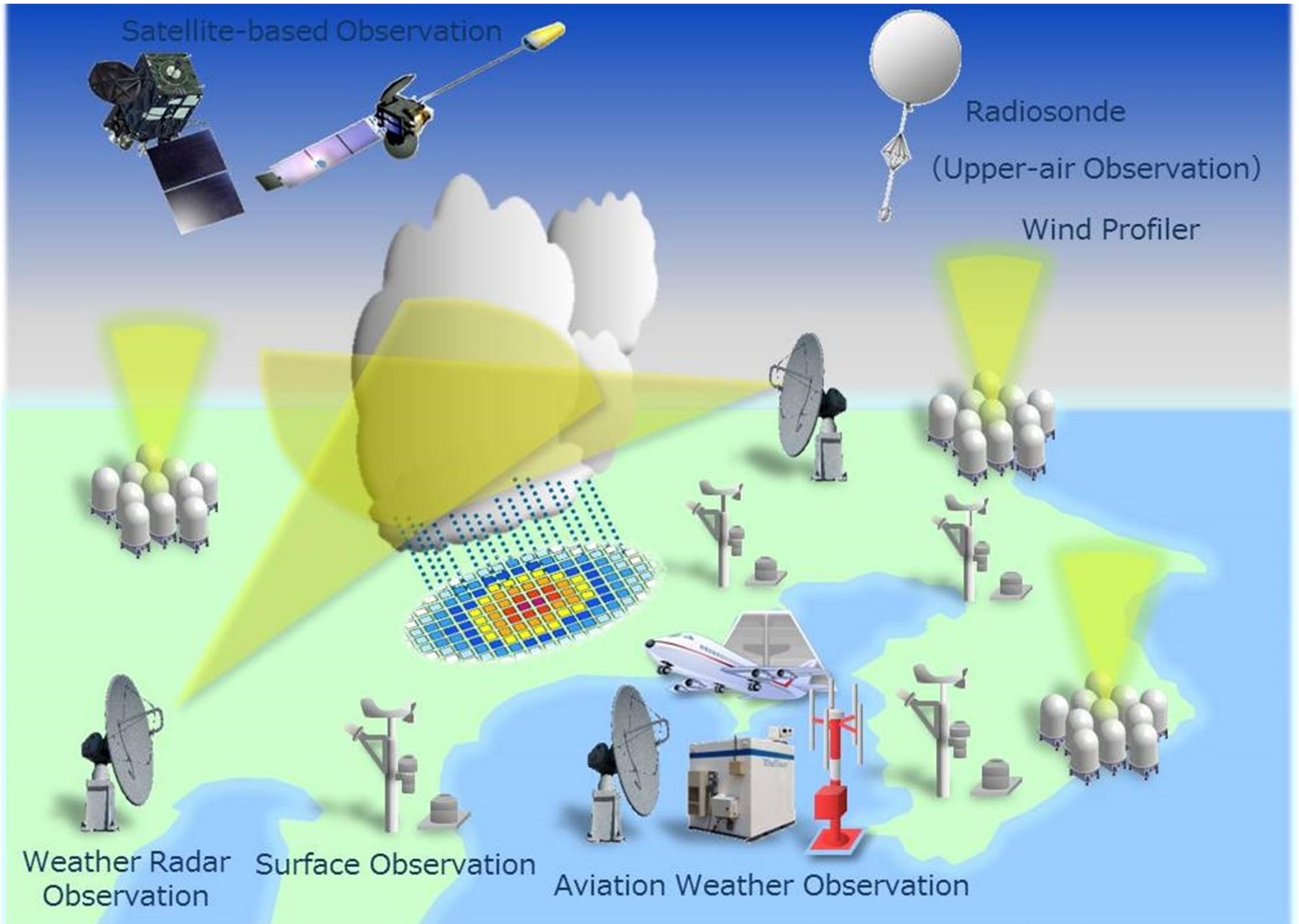
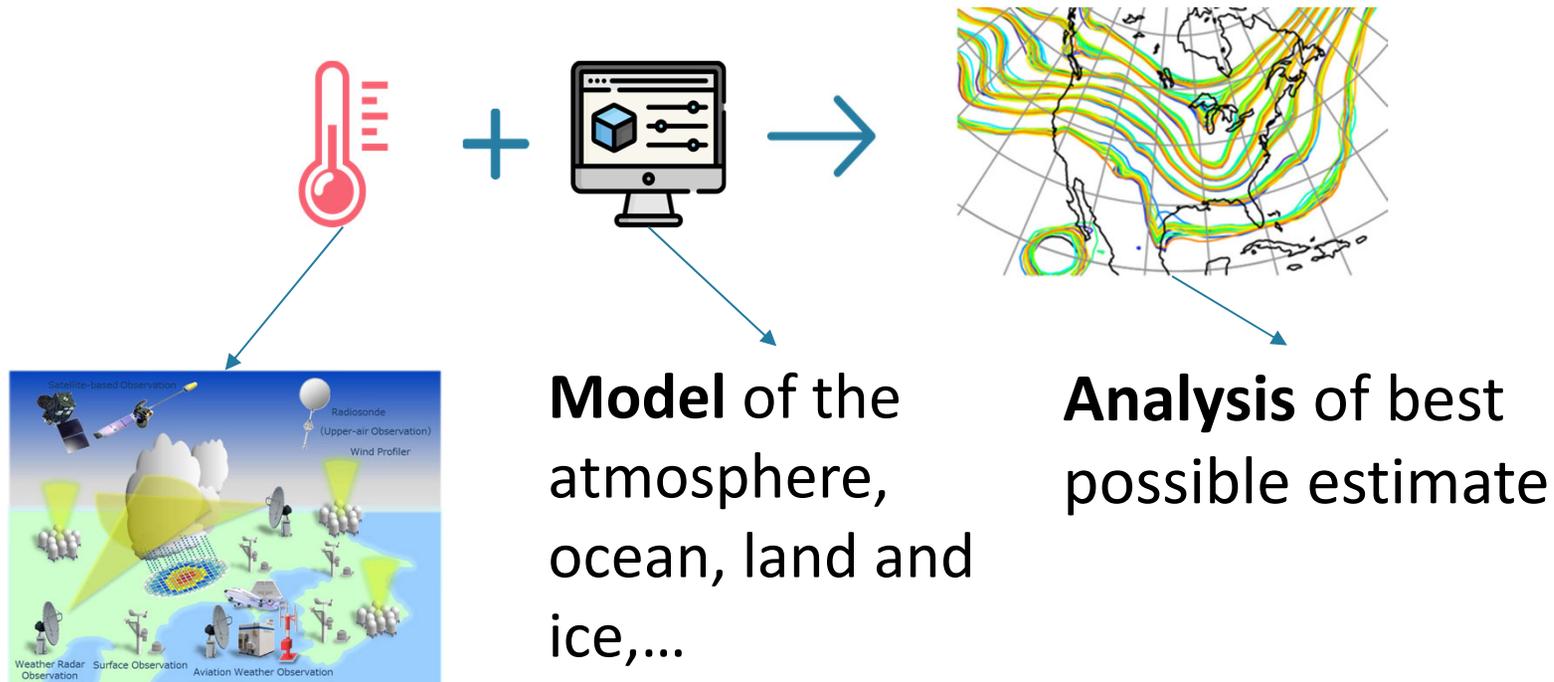


Figure 2: Weather Observations. Japan Meteorological Agency.

# DATA ASSIMILATION

What is Data Assimilation?

- Observations statistically combined with Model Forecast to produce an analysis
- Widely used in numerical weather prediction



# DATA ASSIMILATION RESEARCH TESTBED (DART)

- DART is a data assimilation open-source software developed by DAREs, NCAR
- DART provides modelers and observational scientists with powerful DA tools that are easy to implement and customize

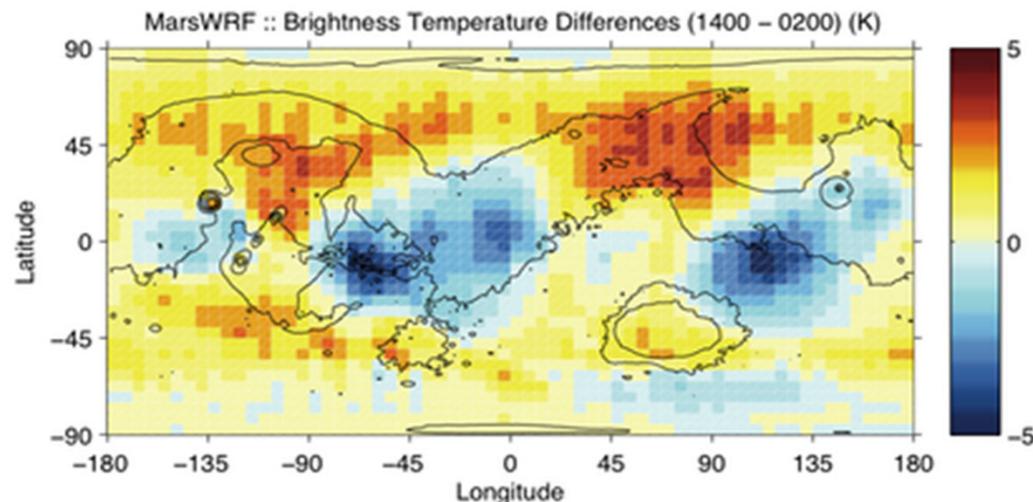


Figure 3: Brightness Temperature Difference within the Mars atmosphere. *DART, WRF, MARS.*

# MOTIVATION – DATABASE PROBLEM

- DART currently stores observation information in `obs_sequence` files.
  - Pros: contains exactly needed information, supports extensible metadata
  - Cons: difficult to query, non-standard

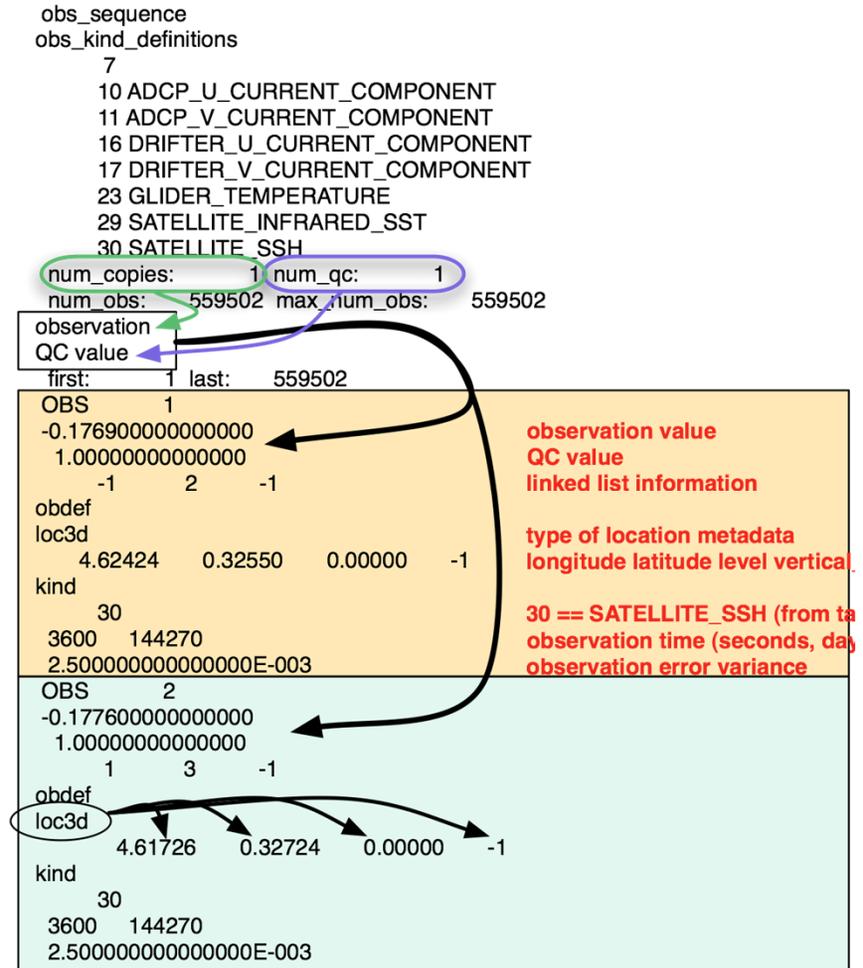


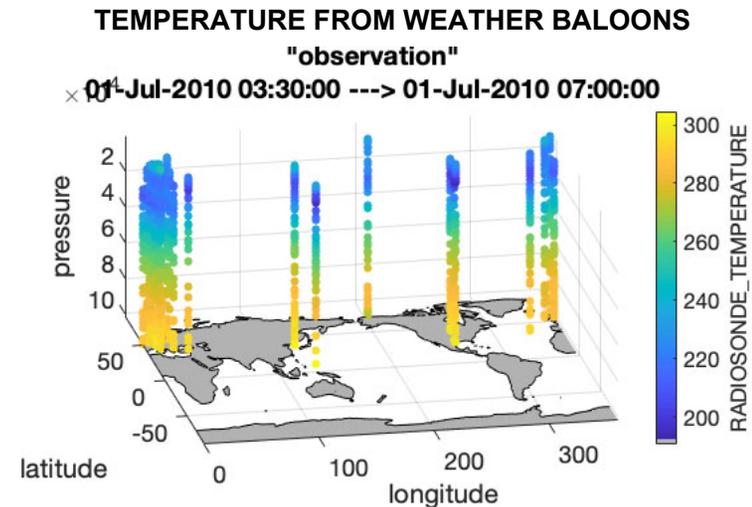
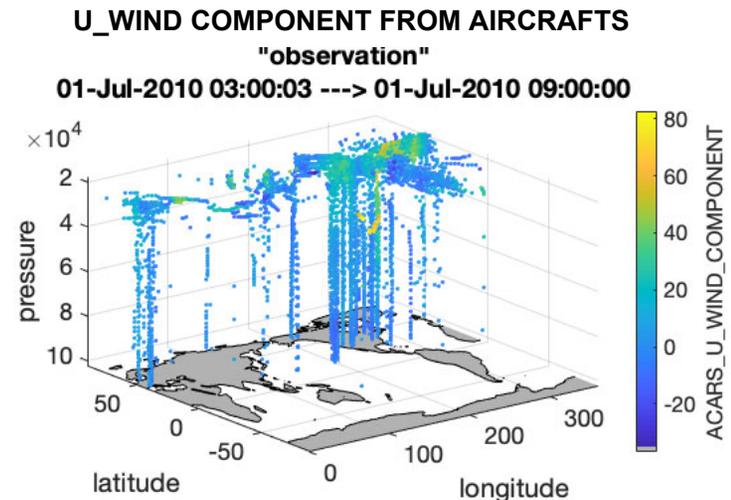
Figure 4: `obs_sequence` file structure.

# MOTIVATION – VISUALIZATION INFRASTRUCTURE

The 3D point-data obs\_sequence files are not supported by current visualization software.

→ Convert to netCDF files & implement MATLAB functions to plot:

- Pros: customizable, pretty
- Cons: expensive MATLAB license, difficult to maintain, does not support extensible metadata



Plotted with MATLAB

# SOLUTION – NETCDF CONVERTER

Convert to **standard** netCDF with Fortran & Python scripts

- Preserve point-data structure

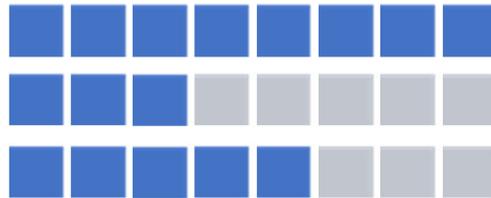
```
types:
    extensible Metadata;
dimensions:
    obs;
variables:
    time(obs);
    lat(obs);
    lon(obs);
    vert (obs);
    salinity(obs);
    Metadata(obs);
// global attributes
groups: dummy {
    obs_id = 0,2,3...
    // attributes
}
    netCDF file structure
```

# SOLUTION – NETCDF CONVERTER

Convert to **standard** netCDF with Fortran & Python scripts

- Preserve point-data structure
- Support extensible metadata with varying-length variables

Fixed-length arrays



Varying-length arrays



**types:**

```
extensible Metadata;
```

**dimensions:**

```
obs;
```

**variables:**

```
time(obs);
```

```
lat(obs);
```

```
lon(obs);
```

```
vert (obs);
```

```
salinity(obs);
```

```
Metadata(obs);
```

```
// global attributes
```

```
groups: dummy {
```

```
obs_id = 0,2,3...
```

```
// attributes
```

```
}
```

```
netCDF file structure
```

# SOLUTION – NETCDF CONVERTER

Convert to **standard** netCDF with Fortran & Python scripts

- Preserve point-data structure
- Support extensible metadata with varying-length variables
- Support for data subsets with netCDF groups

**types:**

```
    extensible Metadata;
```

**dimensions:**

```
    obs;
```

**variables:**

```
    time(obs);
```

```
    lat(obs);
```

```
    lon(obs);
```

```
    vert (obs);
```

```
    salinity(obs);
```

```
    Metadata(obs);
```

```
// global attributes
```

```
groups: dummy {
```

```
    obs_id = 0,2,3...
```

```
    // attributes
```

```
}
```

netCDF file structure

# SOLUTION – NETCDF CONVERTER

Convert to **standard** netCDF with Fortran & Python scripts

- Preserve point-data structure
- Support extensible metadata with varying-length variables
- Support for data subsets with netCDF groups
- Follow netCDF metadata standards (CF Convention)

**types:**

```
    extensible Metadata;
```

**dimensions:**

```
    obs;
```

**variables:**

```
    time(obs);
```

```
    lat(obs);
```

```
    lon(obs);
```

```
    vert (obs);
```

```
    salinity(obs);
```

```
    Metadata(obs);
```

```
// global attributes
```

**groups:** dummy {

```
    obs_id = 0,2,3...
```

```
    // attributes
```

```
}
```

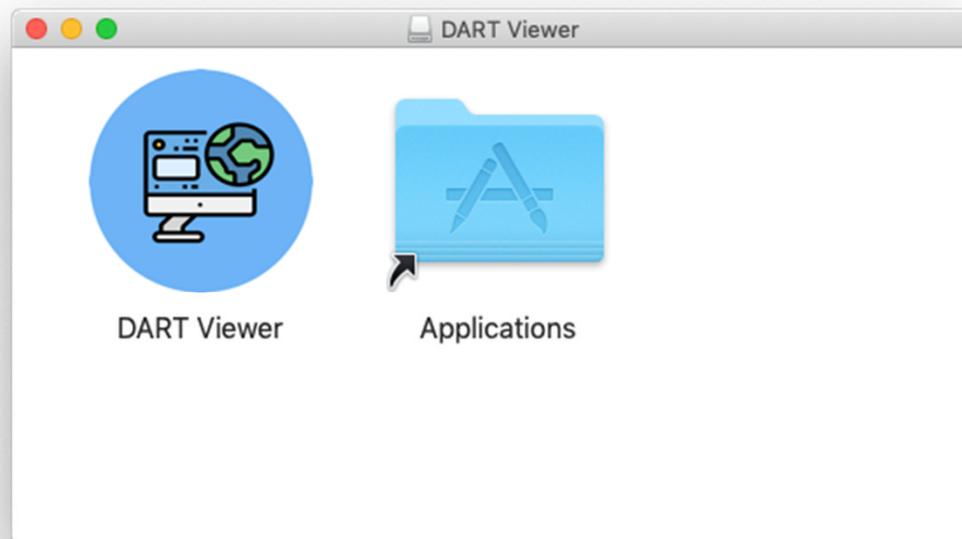
```
netCDF file structure
```

# SOLUTION – GRAPHICAL USER INTERFACE (GUI)

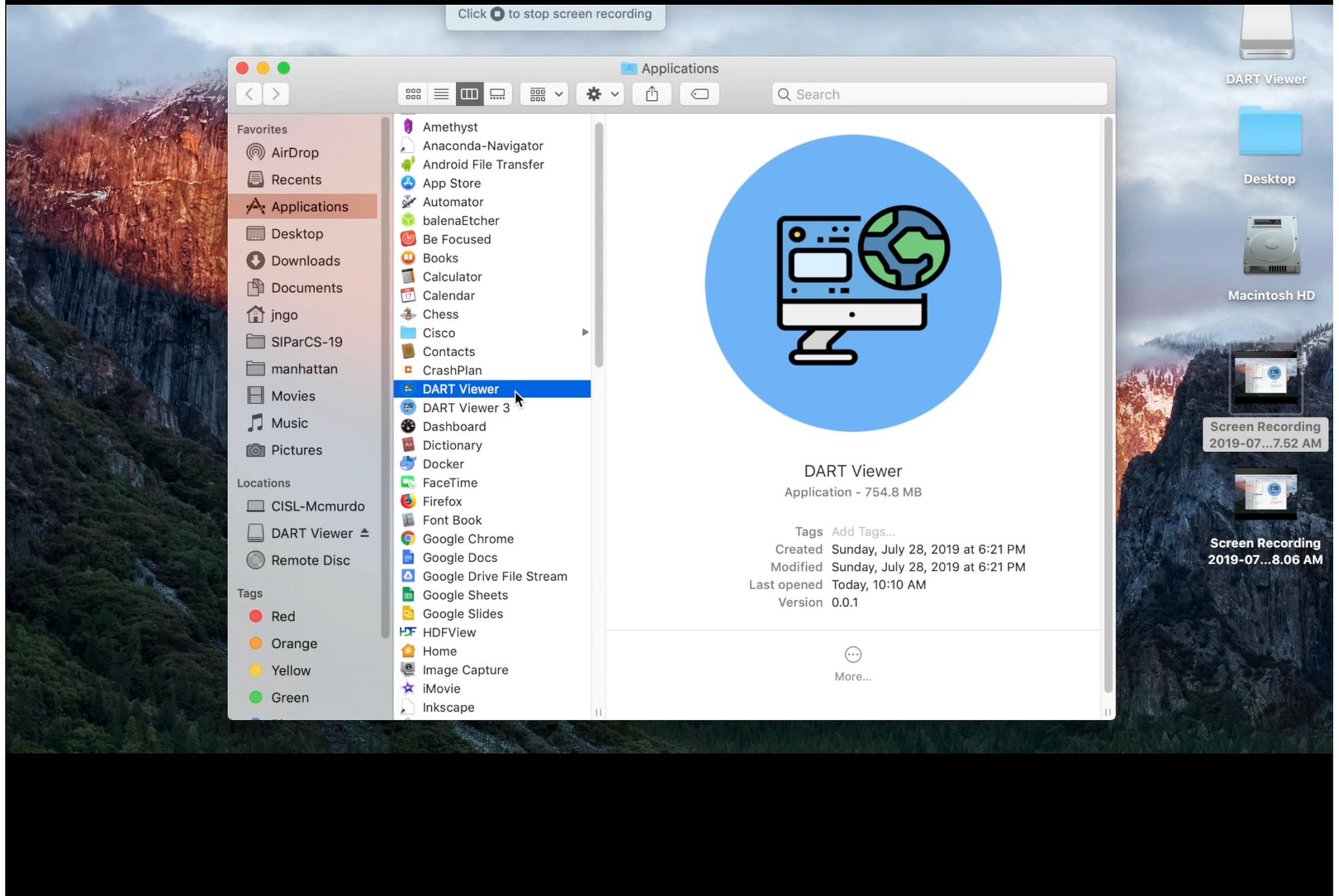
**Frontend:** Built with Qt, a cross-platform software development environment

GUI pros:

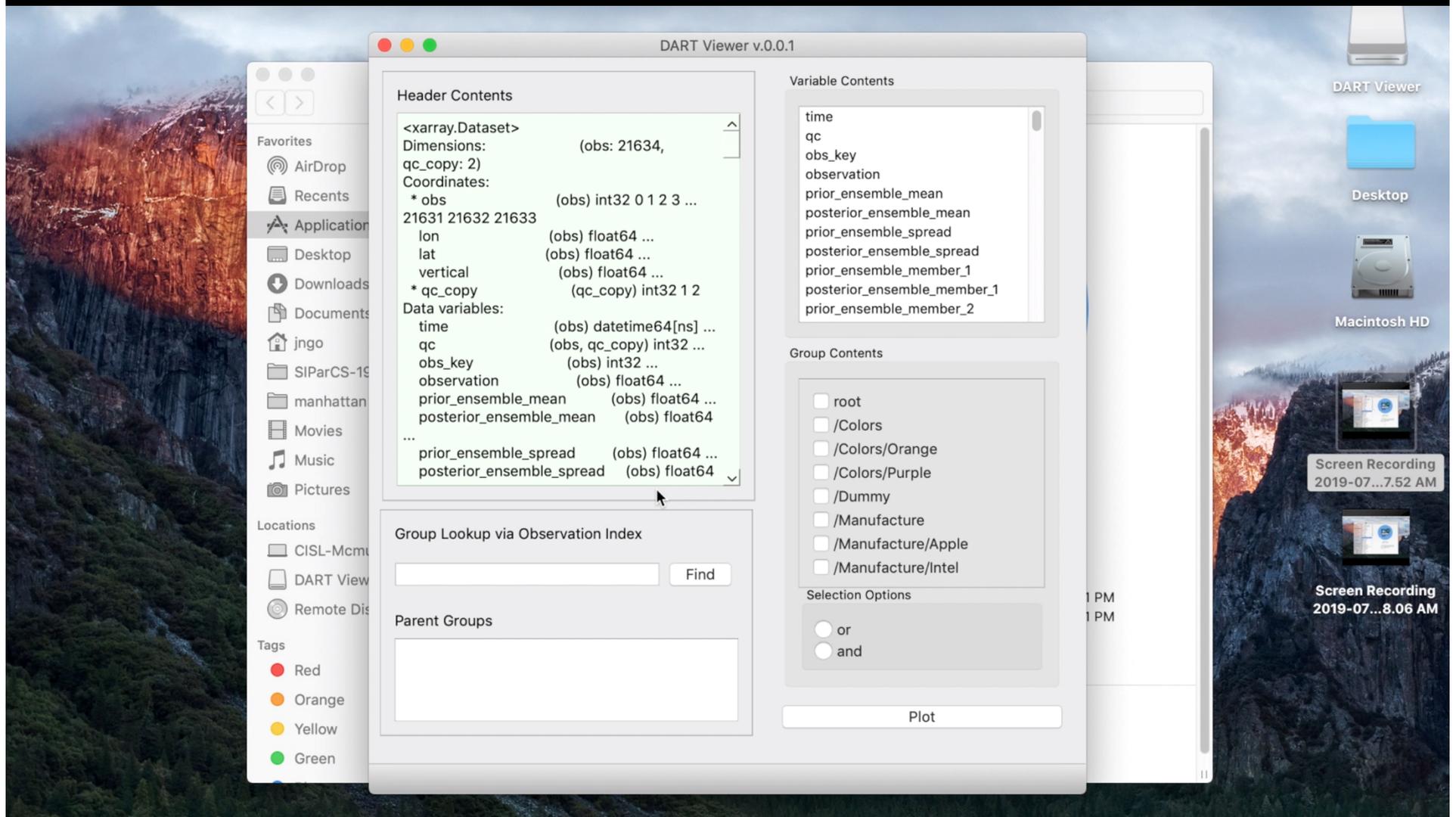
- Requires no programming languages
- Free
- Mac/Linux compatible



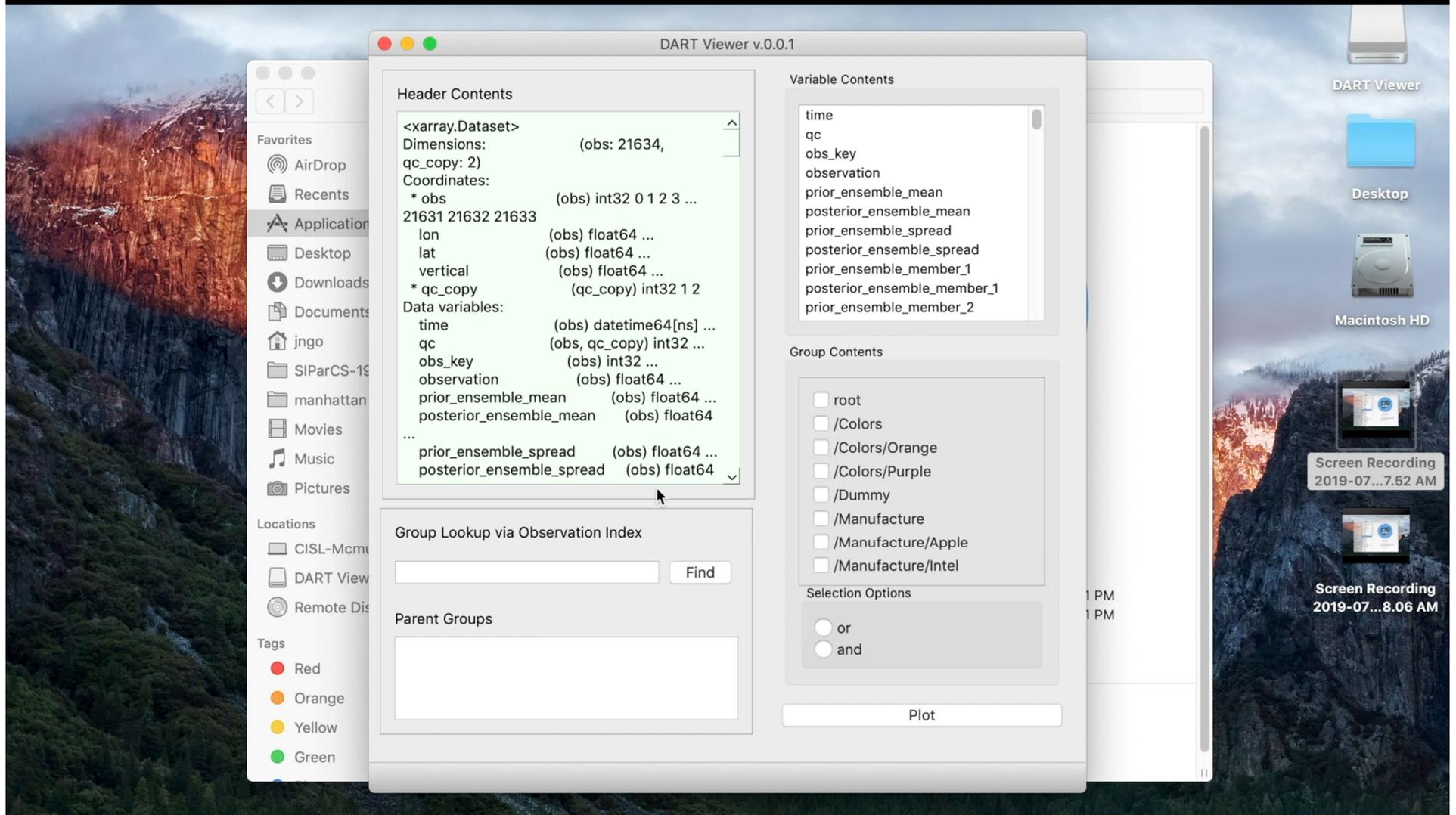
# SOLUTION – GRAPHICAL USER INTERFACE (GUI)



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)

The screenshot shows the DART Viewer v.0.0.1 application window on a Mac desktop. The desktop background is a scenic mountain landscape. On the desktop, there are icons for 'DART Viewer', 'Desktop', 'Macintosh HD', and two 'Screen Recording' files from 2019-07-...7.52 AM and 2019-07-...8.06 AM. The DART Viewer window is the central focus, displaying the following information:

**Header Contents**

```
<xarray.Dataset>
Dimensions:          (obs: 21634,
qc_copy: 2)
Coordinates:
  * obs              (obs) int32 0 1 2 3 ...
21631 21632 21633
  lon               (obs) float64 ...
  lat               (obs) float64 ...
  vertical          (obs) float64 ...
  * qc_copy         (qc_copy) int32 1 2
Data variables:
  time              (obs) datetime64[ns] ...
  qc                (obs, qc_copy) int32 ...
  obs_key           (obs) int32 ...
  observation       (obs) float64 ...
  prior_ensemble_mean (obs) float64 ...
  posterior_ensemble_mean (obs) float64 ...
  prior_ensemble_spread (obs) float64 ...
  posterior_ensemble_spread (obs) float64 ...
  ...
  prior_ensemble_spread (obs) float64 ...
  posterior_ensemble_spread (obs) float64 ...
```

**Variable Contents**

```
time
qc
obs_key
observation
prior_ensemble_mean
posterior_ensemble_mean
prior_ensemble_spread
posterior_ensemble_spread
prior_ensemble_member_1
posterior_ensemble_member_1
prior_ensemble_member_2
```

**Group Contents**

- root
- /Colors
- /Colors/Orange
- /Colors/Purple
- /Dummy
- /Manufacture
- /Manufacture/Apple
- /Manufacture/Intel

**Selection Options**

or  
 and

**Group Lookup via Observation Index**

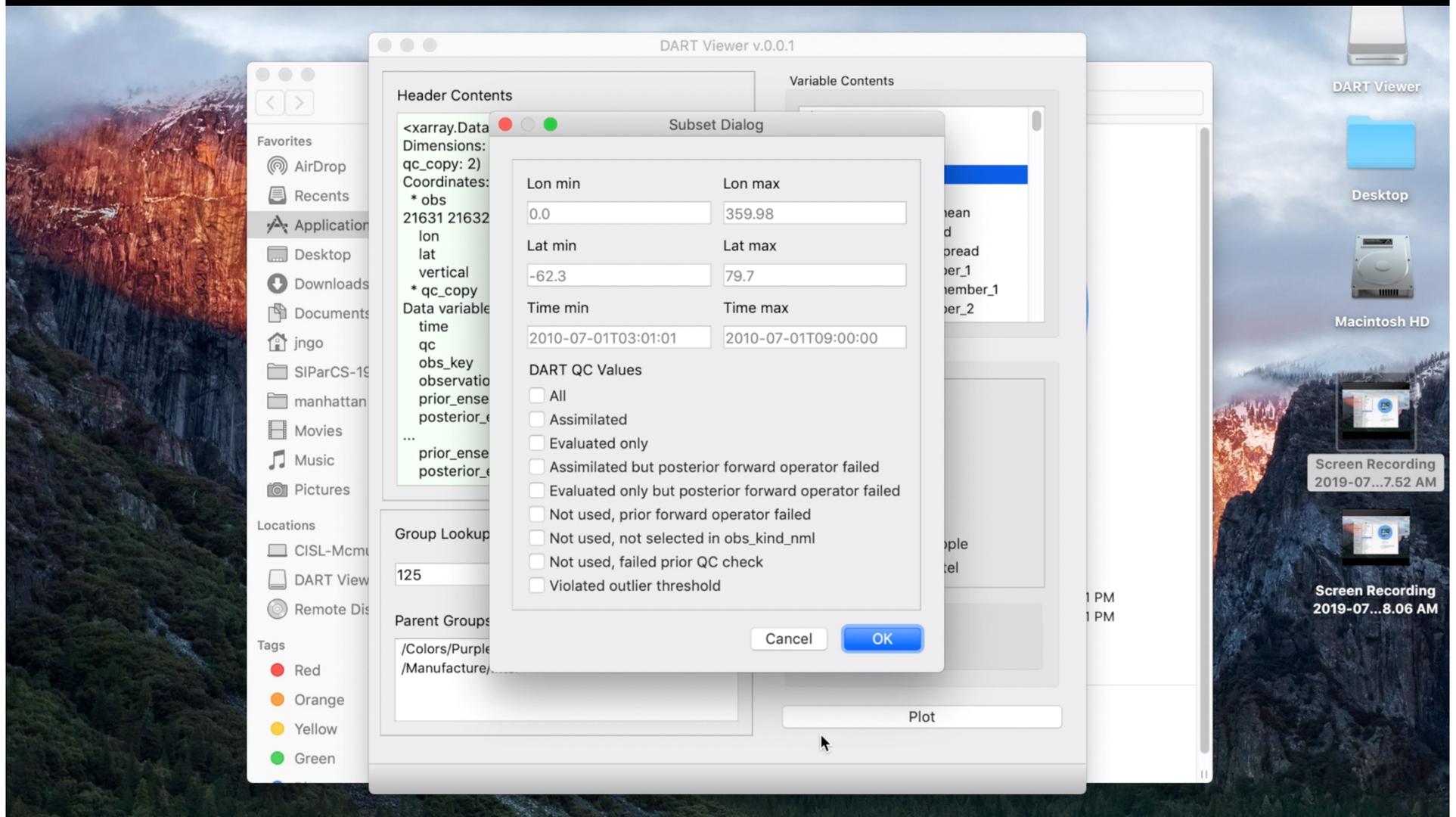
125 Find

**Parent Groups**

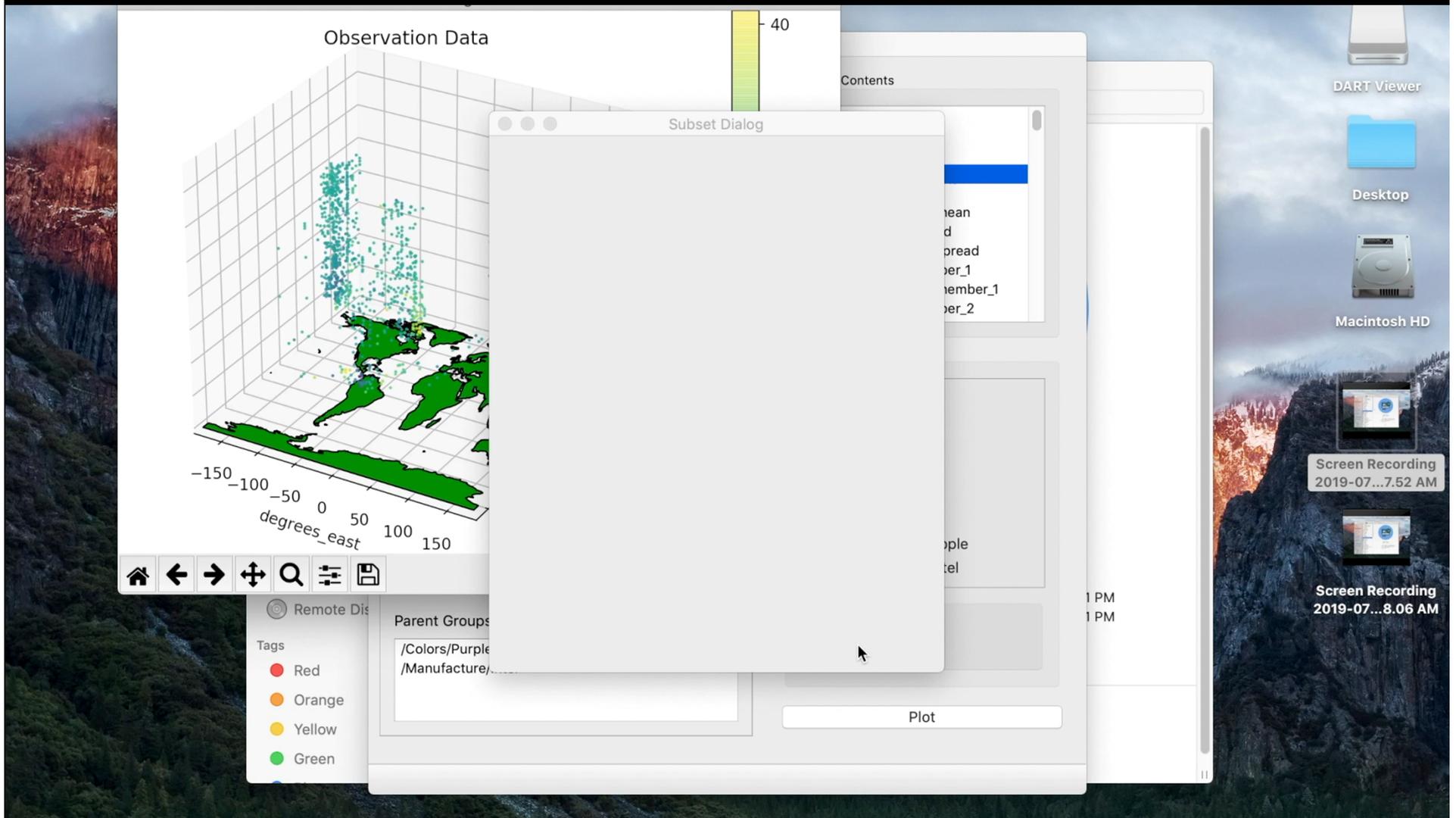
```
/Colors/Purple
/Manufacture/Intel
```

**Plot**

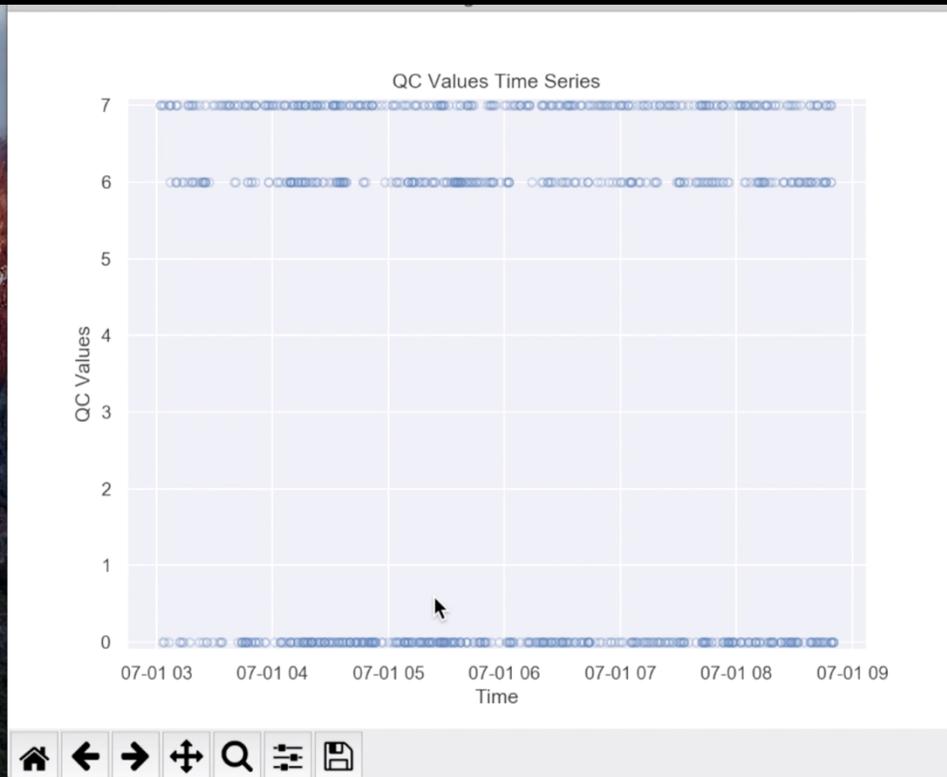
# SOLUTION – GRAPHICAL USER INTERFACE (GUI)



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)

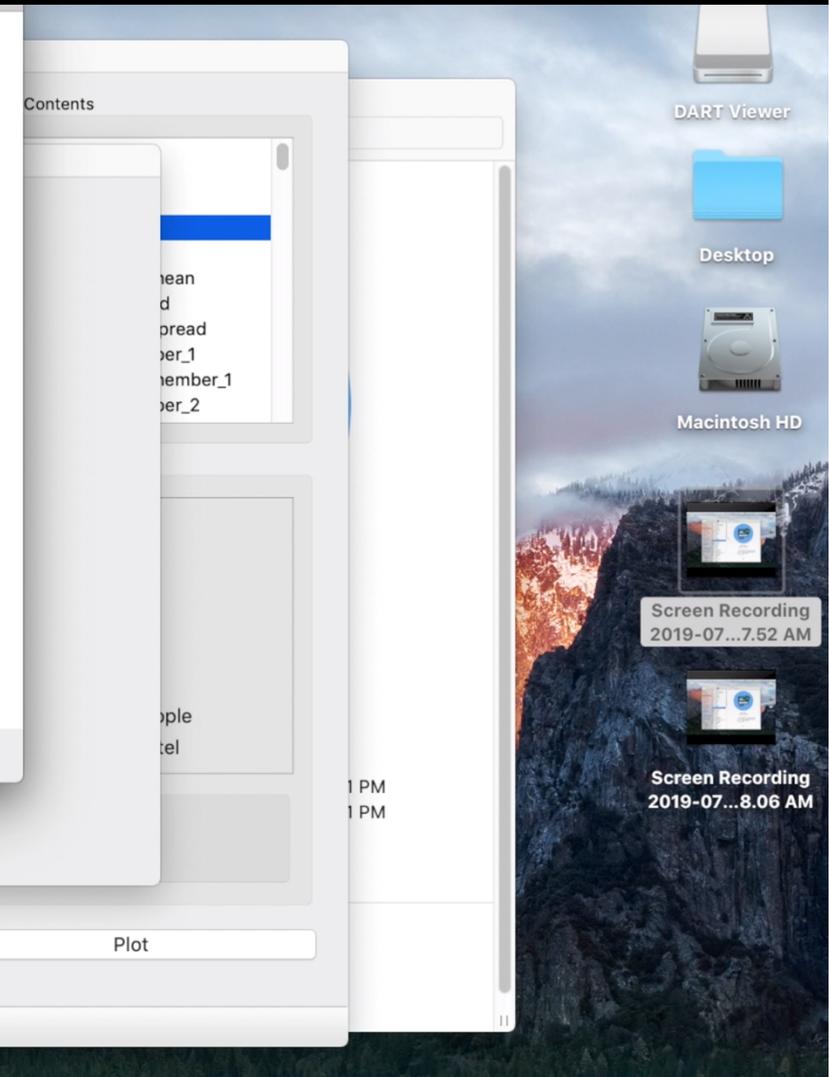


- Remote Dis
- Tags
- Red
  - Orange
  - Yellow
  - Green

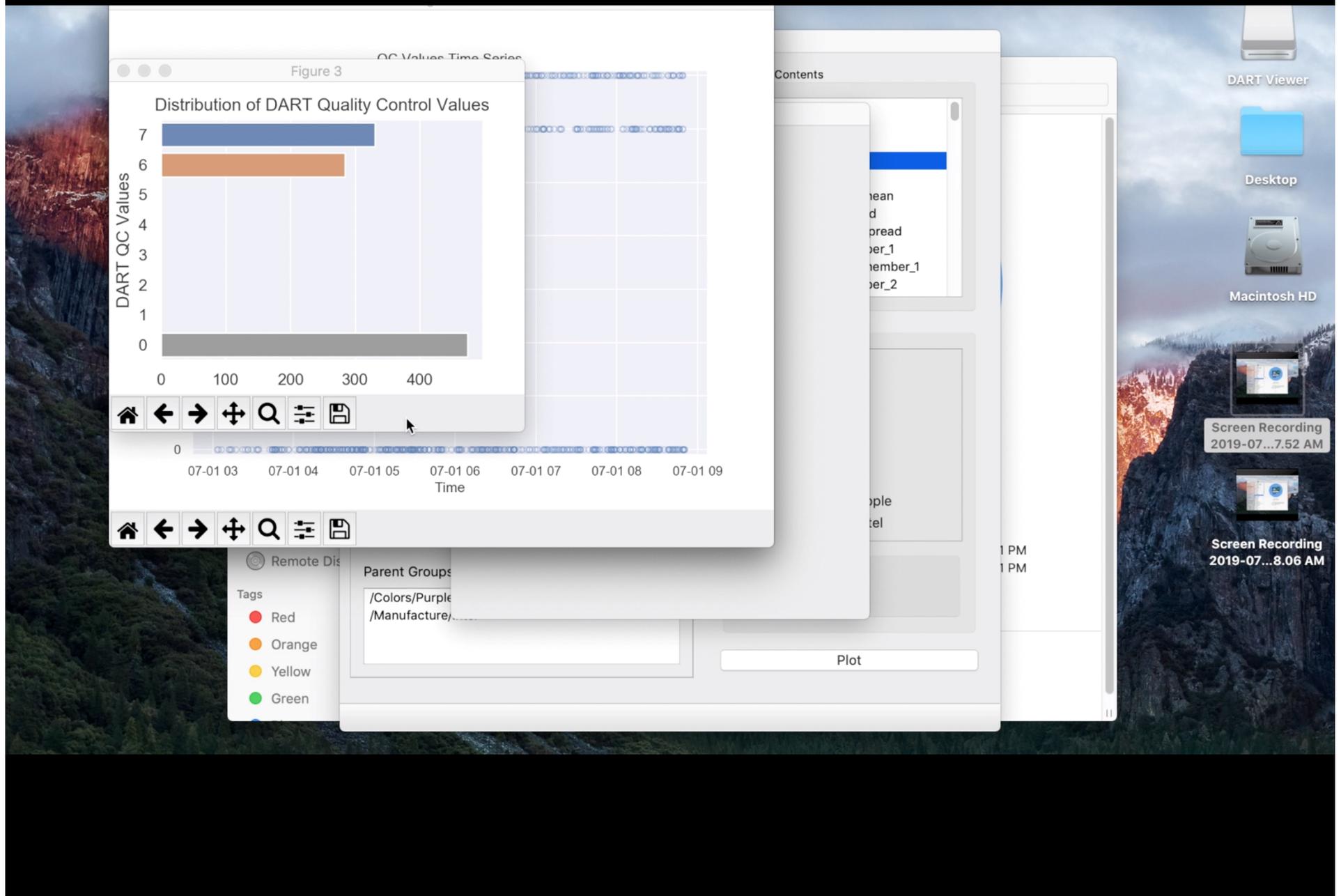
Parent Groups

- /Colors/Purple
- /Manufacture/...

Plot



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)



# SOLUTION – GRAPHICAL USER INTERFACE (GUI)

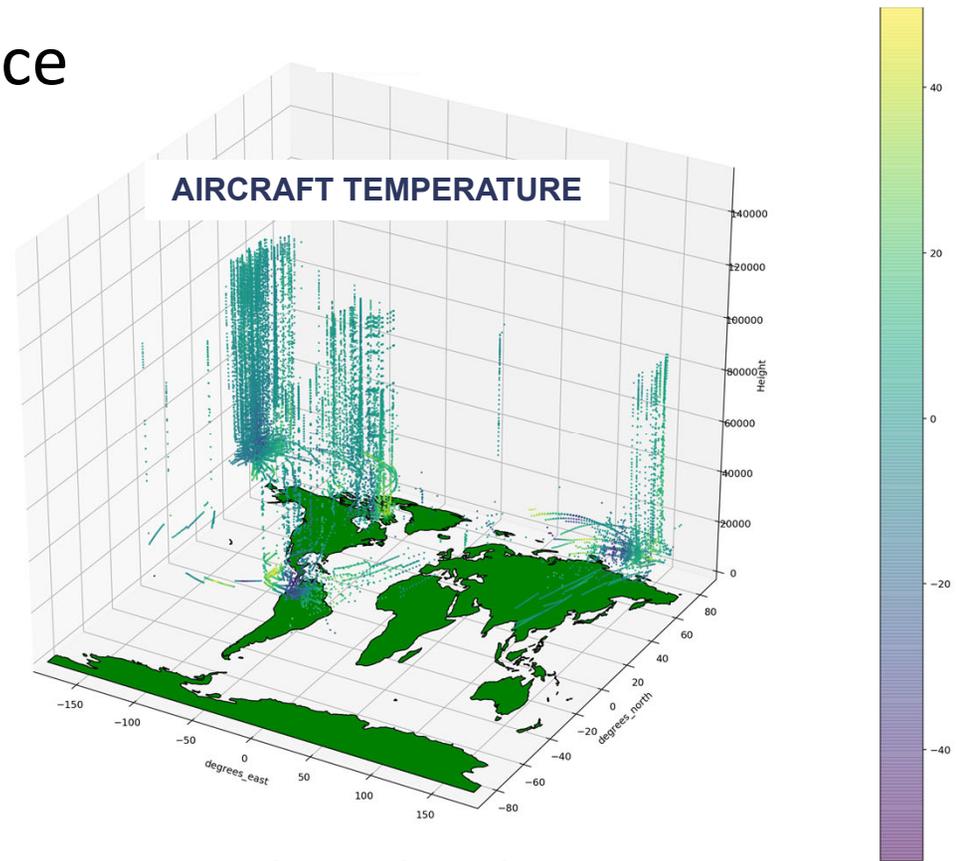
## Python Backend

- Database:
  - xarray
  - netcdf4
- Plot:
  - matplotlib
  - cartopy



# FUTURE WORK

- Automated testing capabilities
- Customizable plots
- More intuitive interface



Plotted with GUI

# ACKNOWLEDGEMENT

- Mentors: Jeff Anderson, Nancy Collins
- Data Assimilation Research Section
- Fellow SIParCS Interns
- CODE team

Any questions ?



You can find me at

- [aqngo@haverford.edu](mailto:aqngo@haverford.edu)
- or at the airport on Saturday