Implementing an Observation Support System for the Data Assimilation Research Testbed

JASON NGO
Haverford College

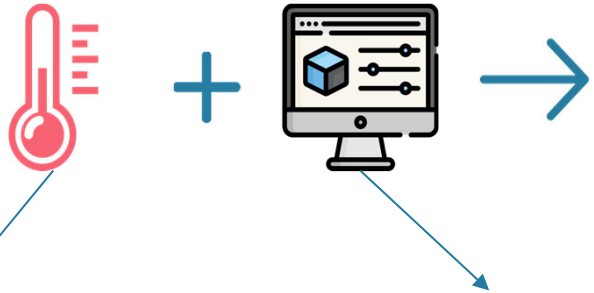
August, 2019

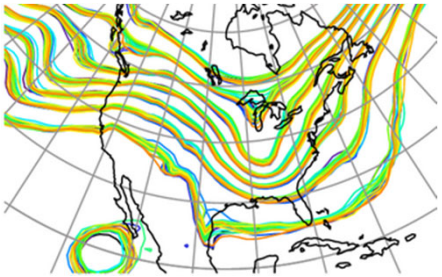
This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

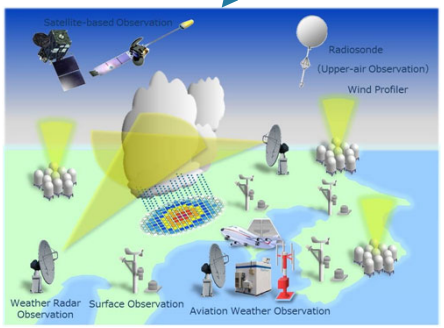
DATA ASSIMILATION

What is Data Assimilation?

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







Model of the atmosphere, ocean, land and ice,…

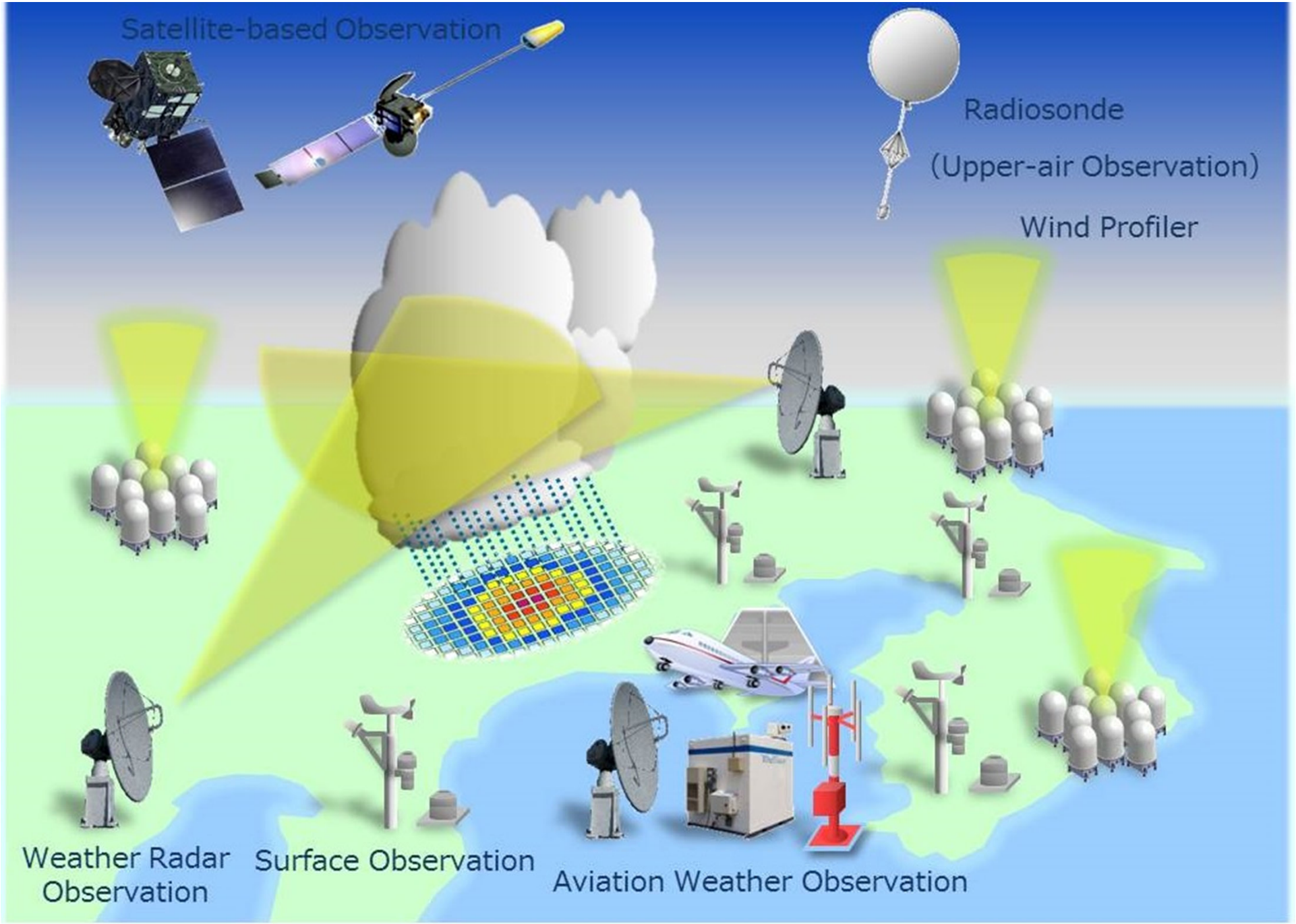
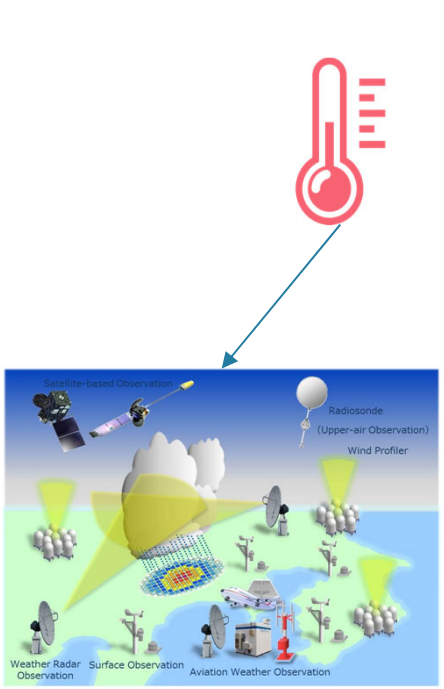


Figure 2: Weather Observations. *Japan Meteorological Agency.*

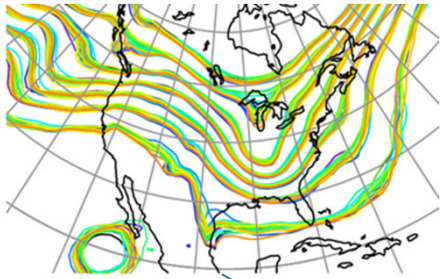
DATA ASSIMILATION

What is Data Assimilation?

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



Model of the atmosphere, ocean, land and ice,…



Analysis of best possible estimate

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

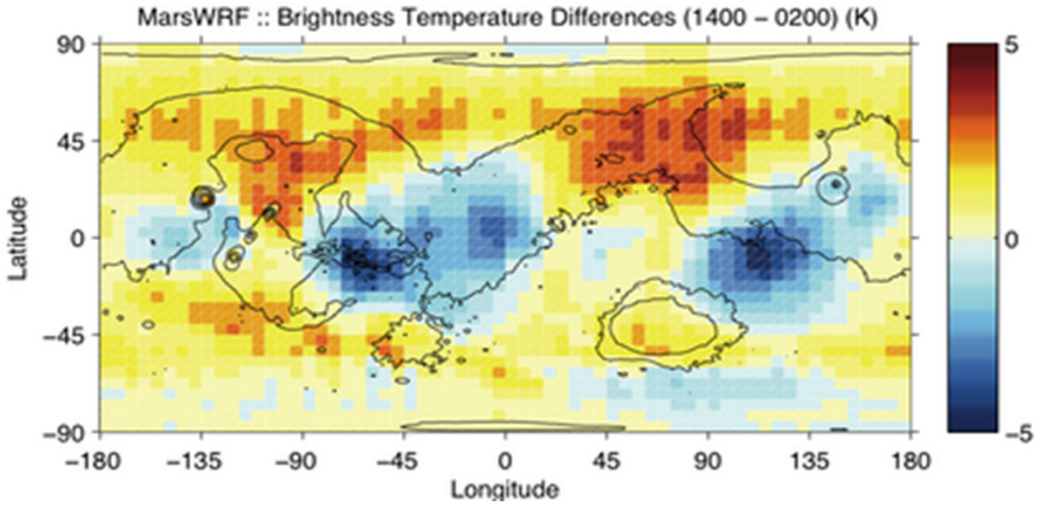


Figure 3: Brightness Temperature Difference within the Mars atmosphere. *DART, WRF, MARS*.

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

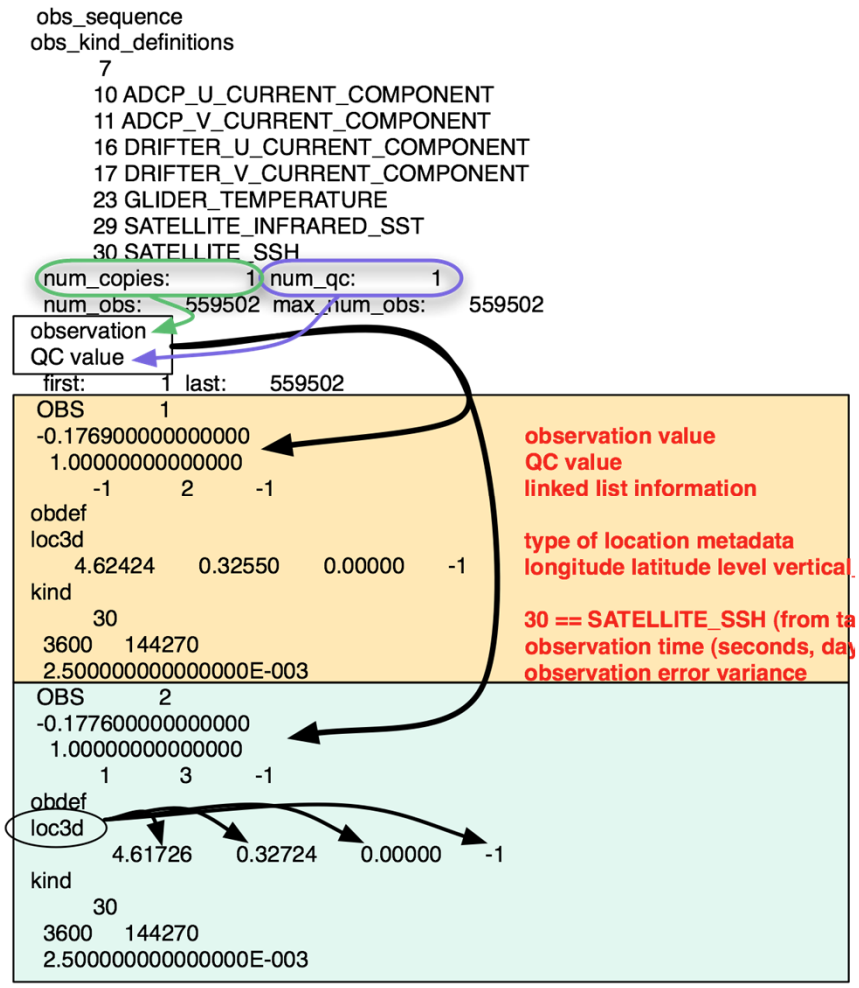


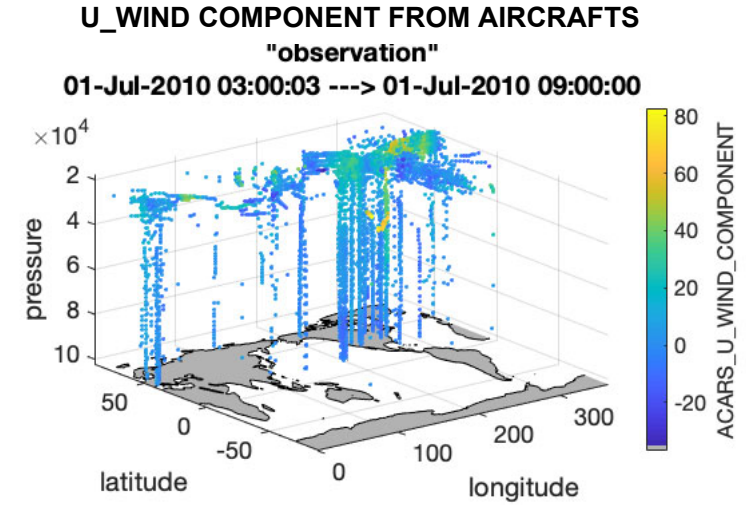
Figure 4: `obs_sequence` file structure.

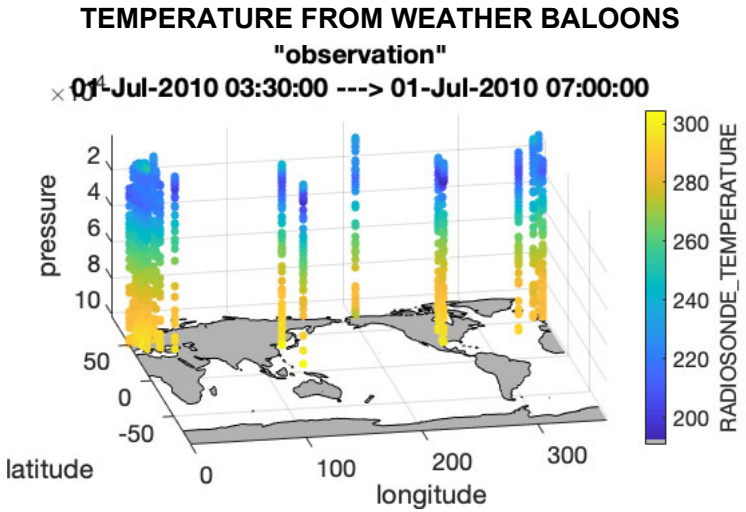
MOTIVATION – VISUALIZATION INFRASTRUCTURE

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
}
```

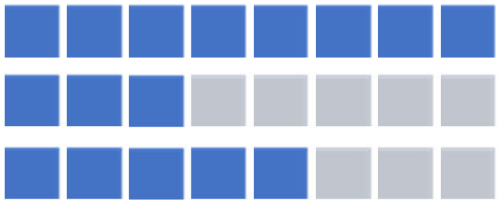
netCDF file structure

SOLUTION – NETCDF CONVERTER

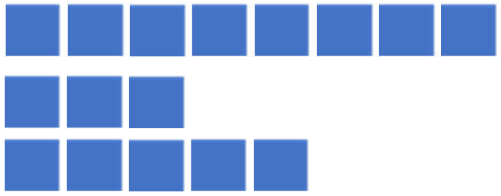
Convert to **standard** netCDF with Fortran & Python scripts

- 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
}
```

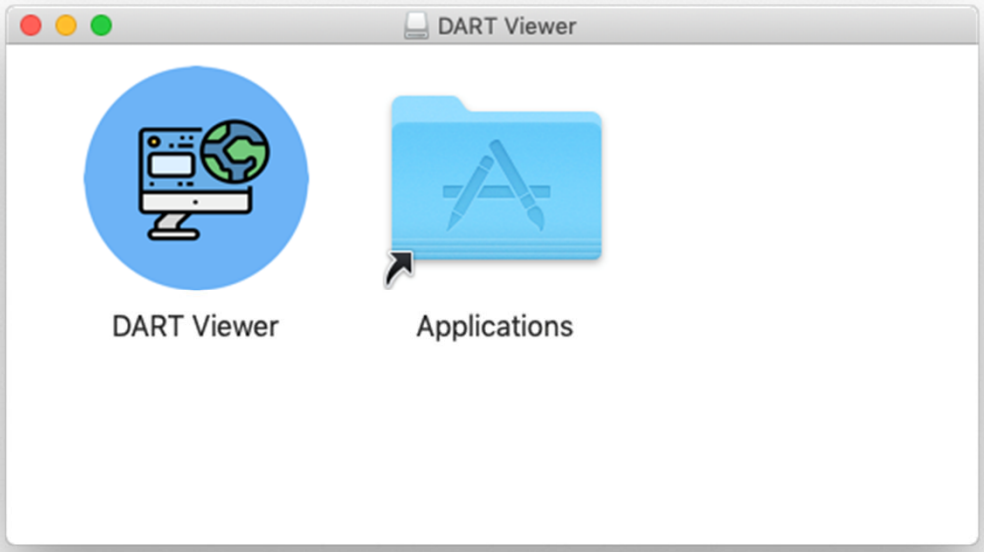
netCDF file structure

SOLUTION – GRAPHICAL USER INTERFACE (GUI)

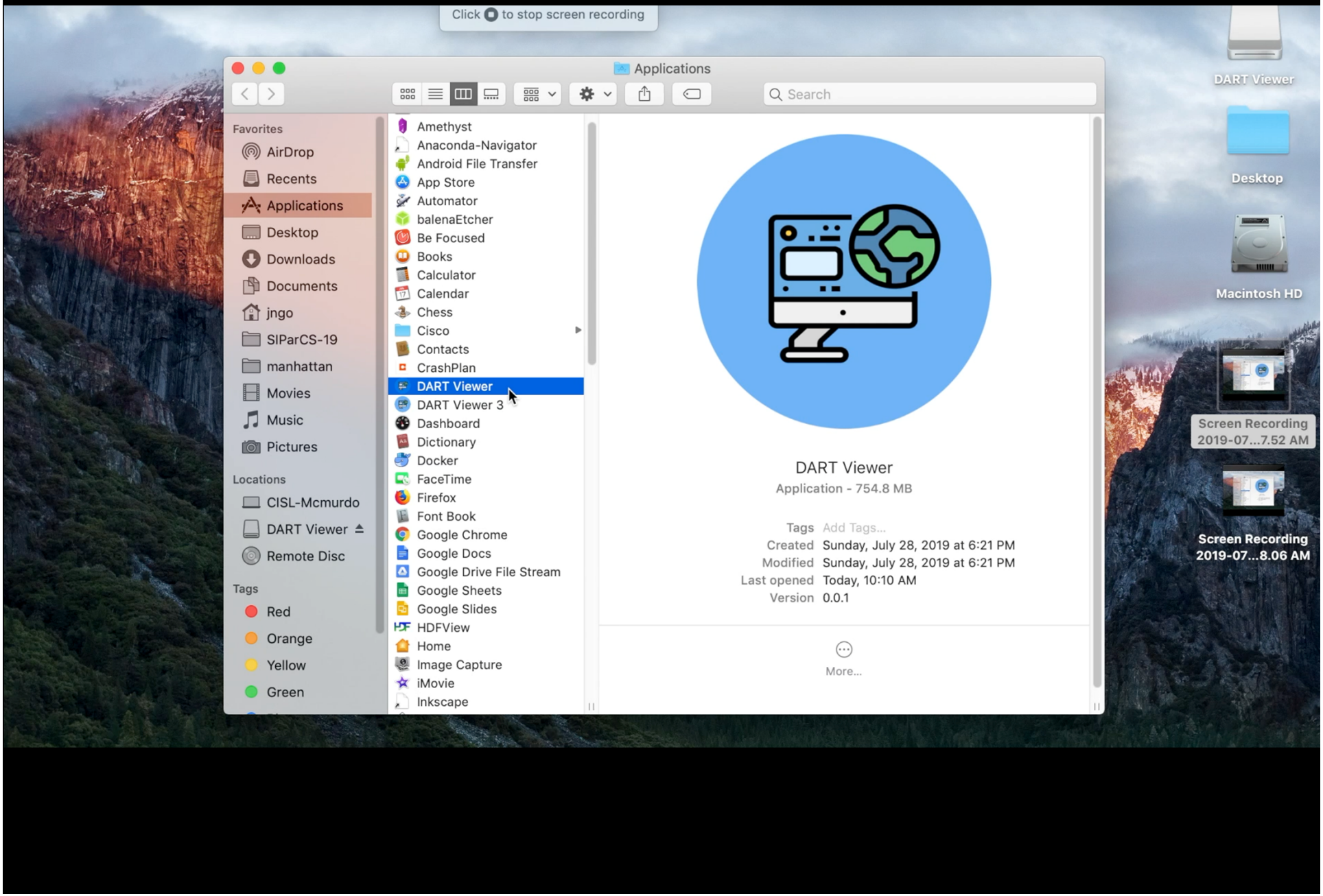
Frontend: Built with Qt, a cross-platform software development environment

GUI pros:

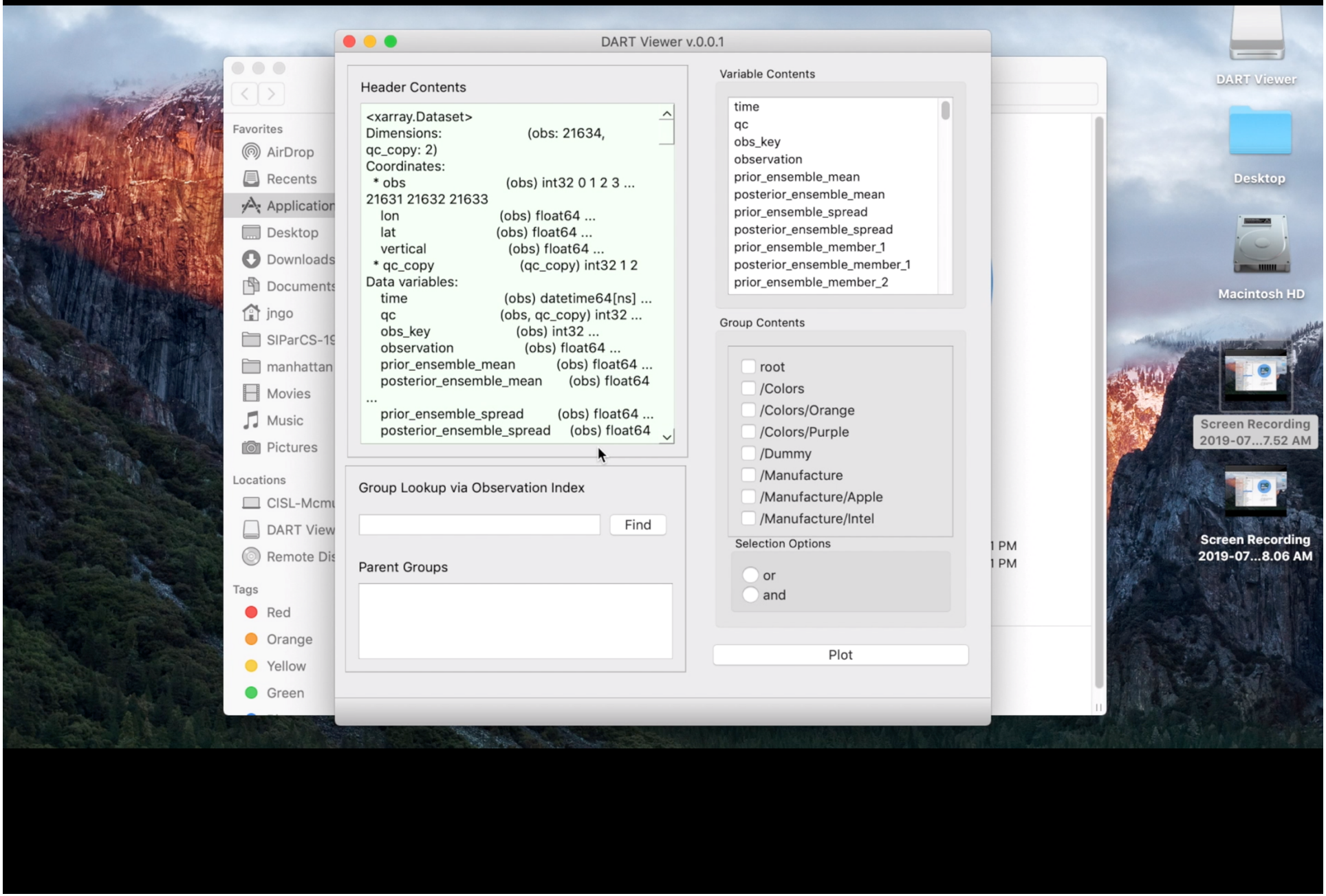
- Requires no programming languages
- Free
- Mac/Linux compatible



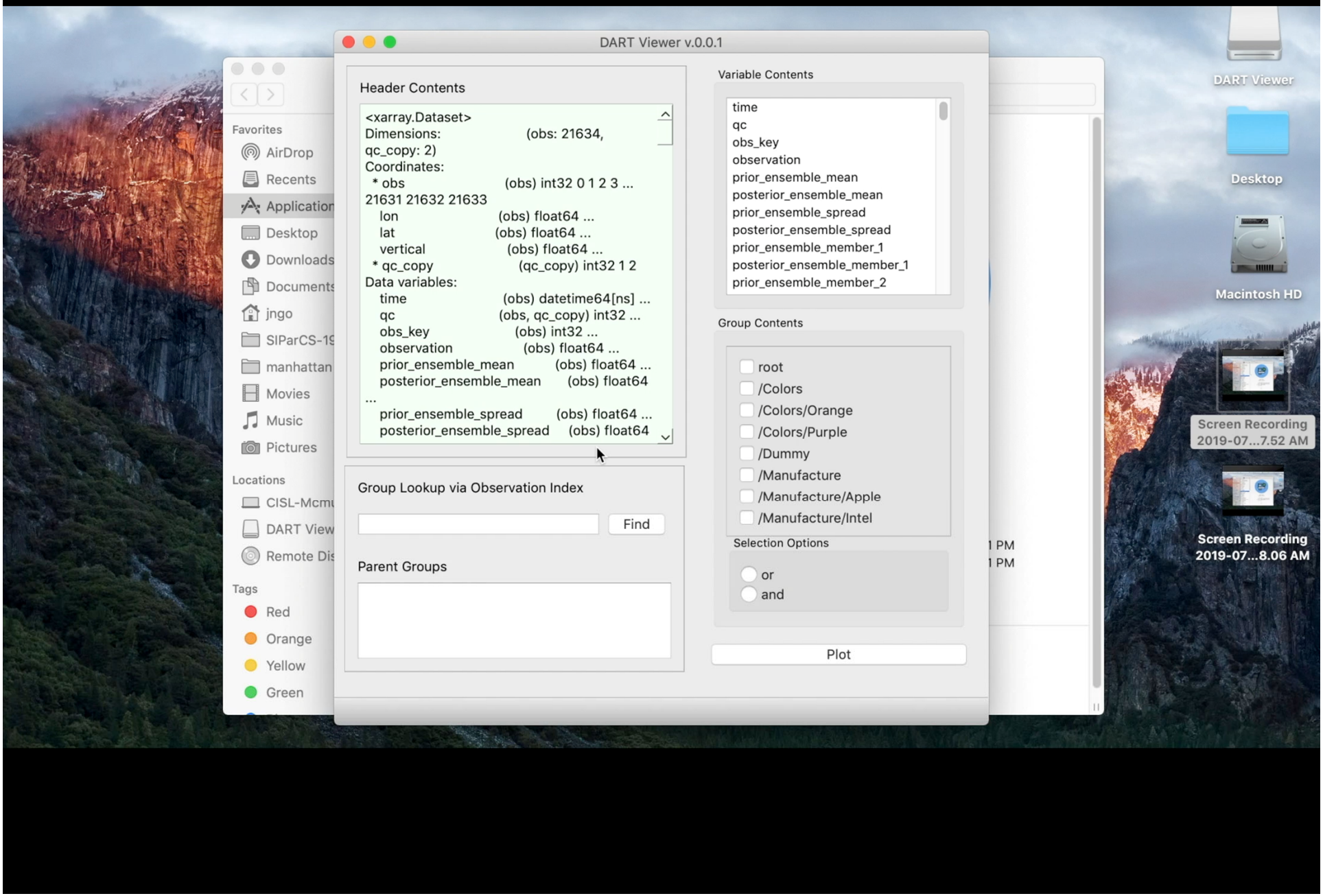
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



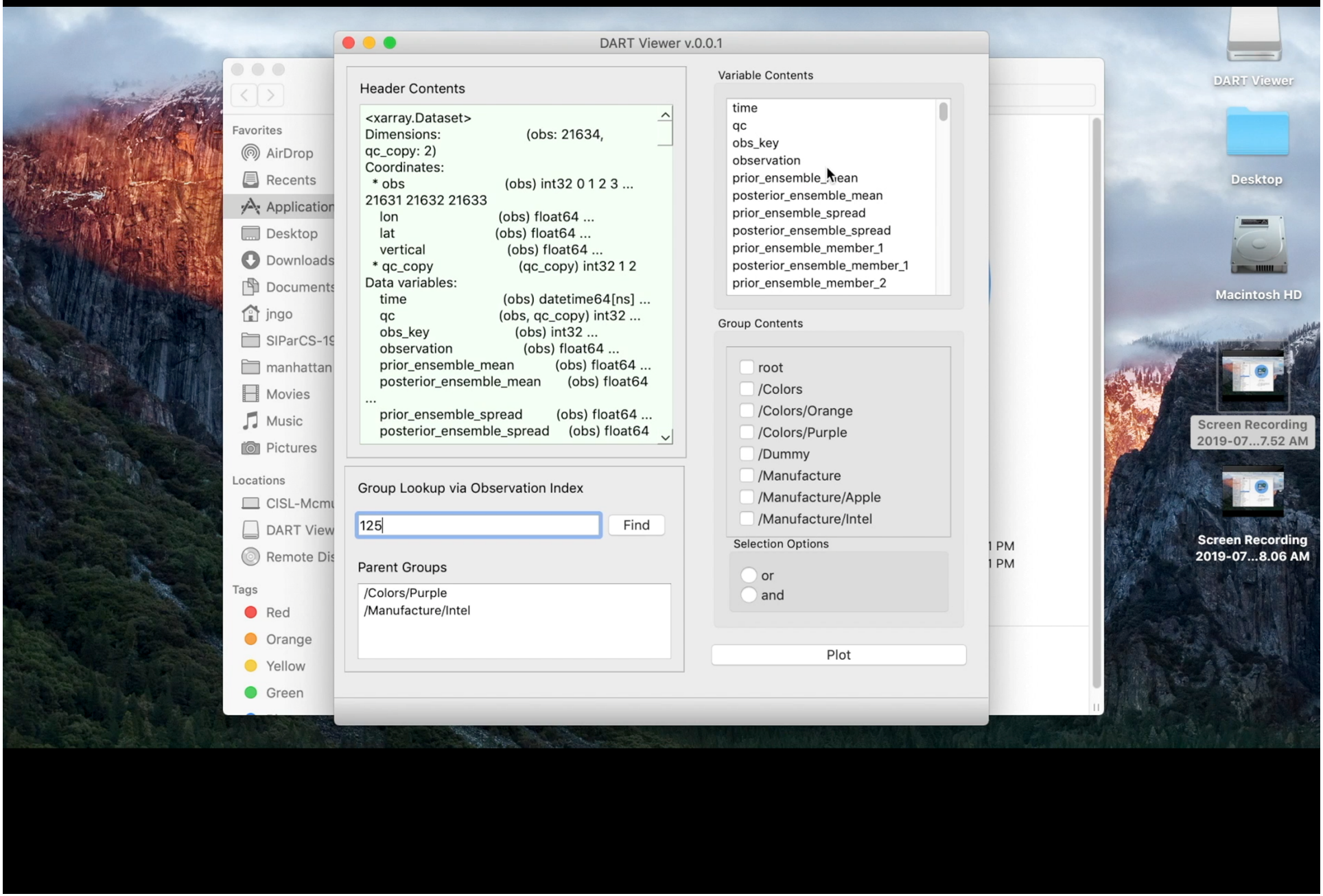
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



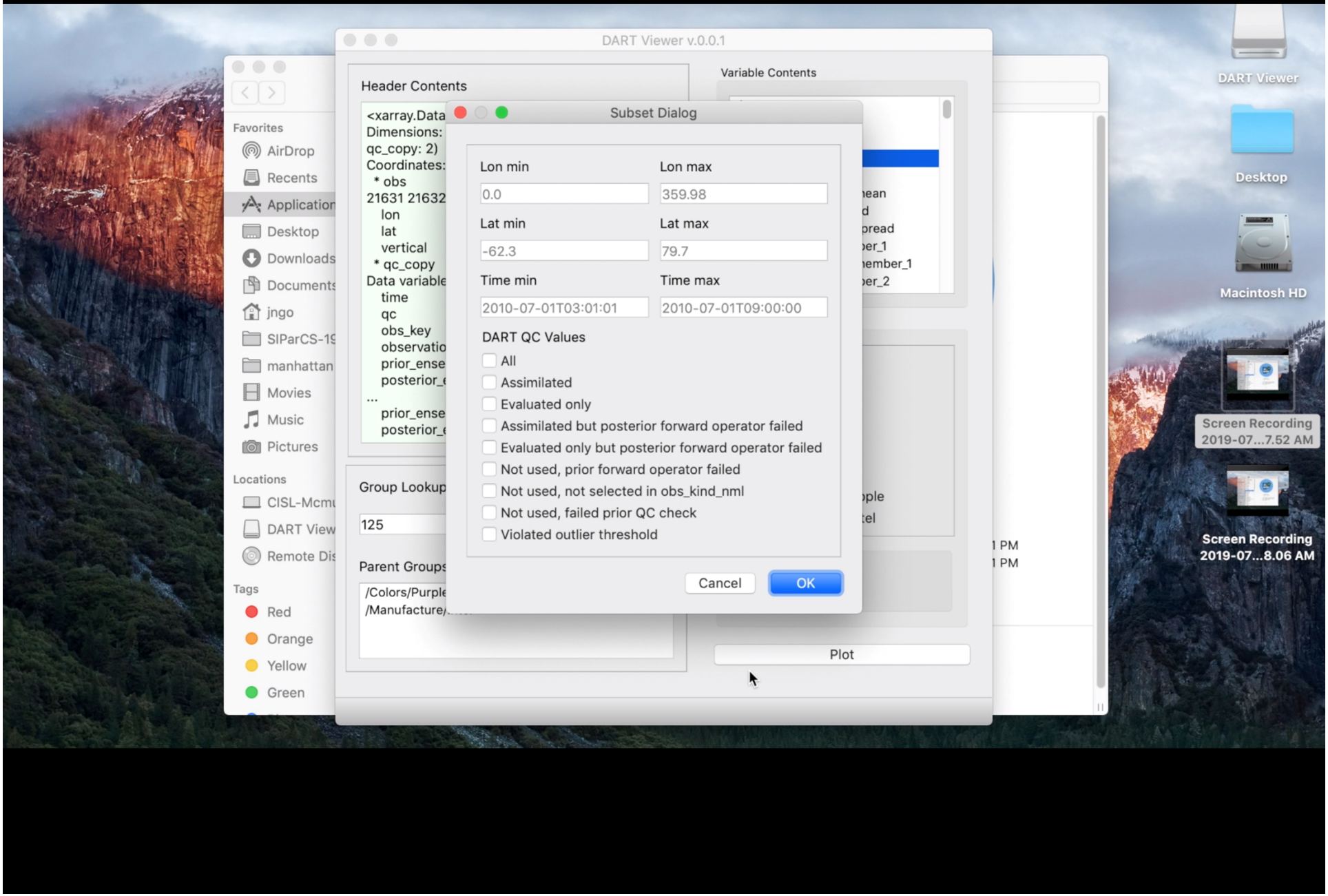
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



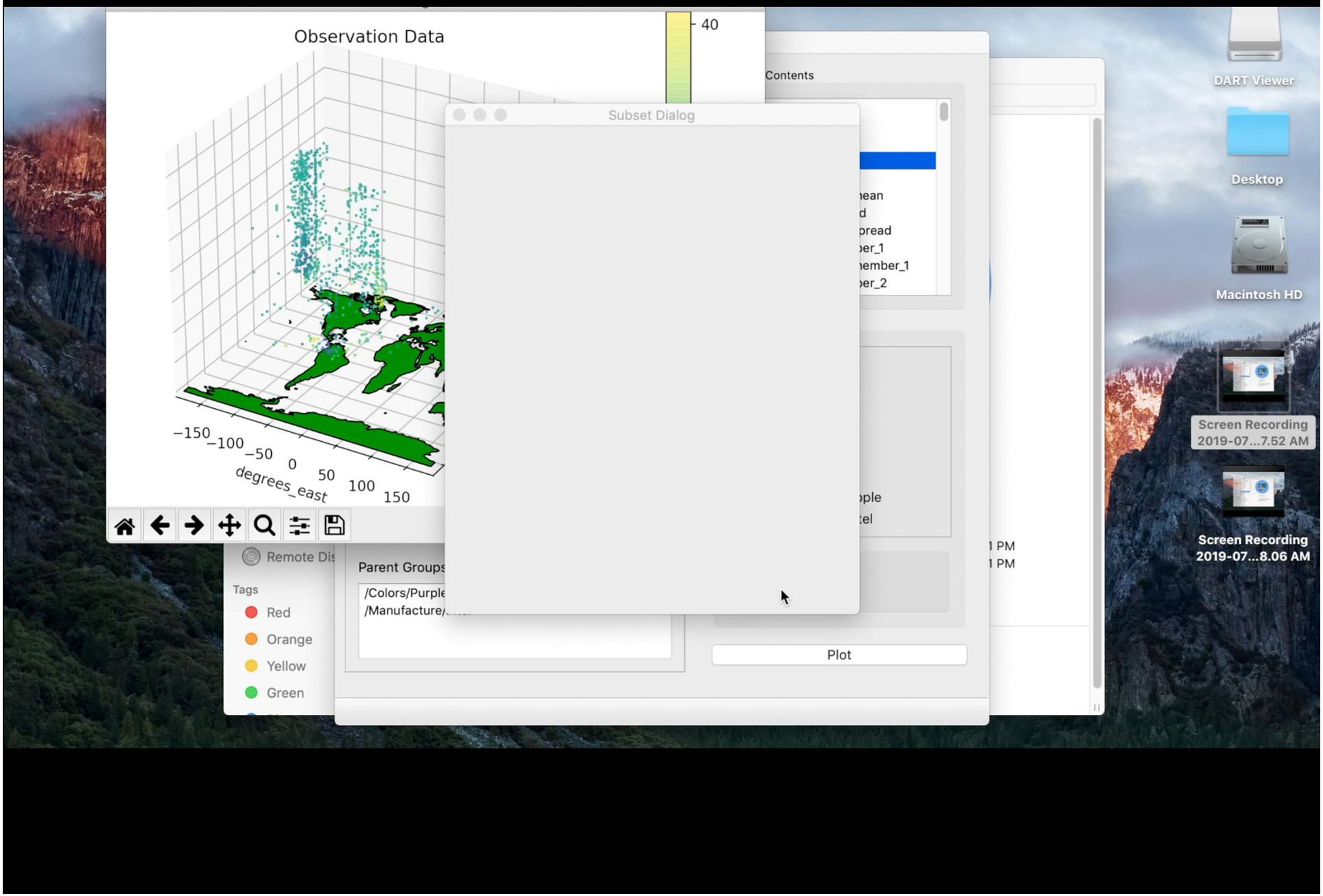
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



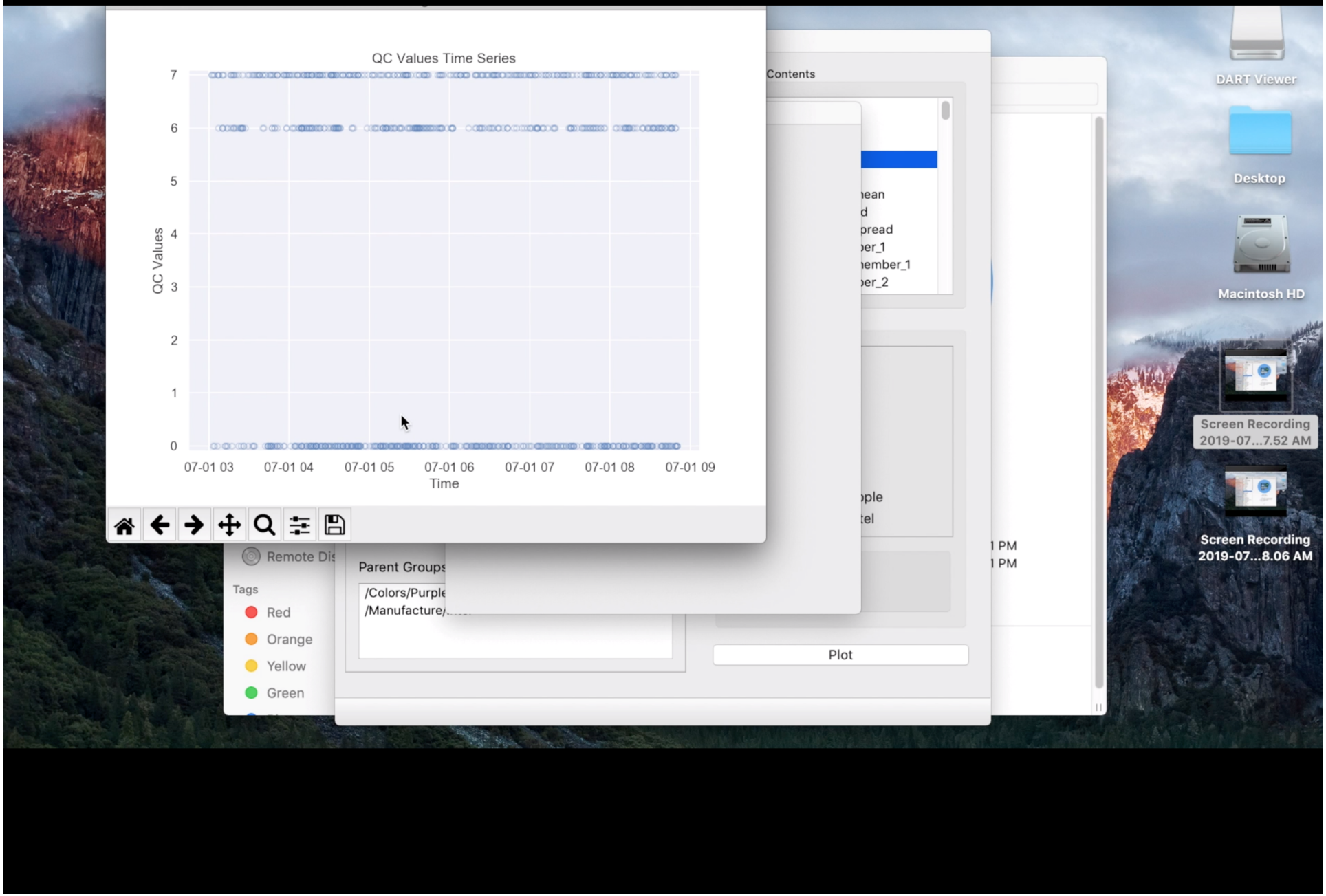
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



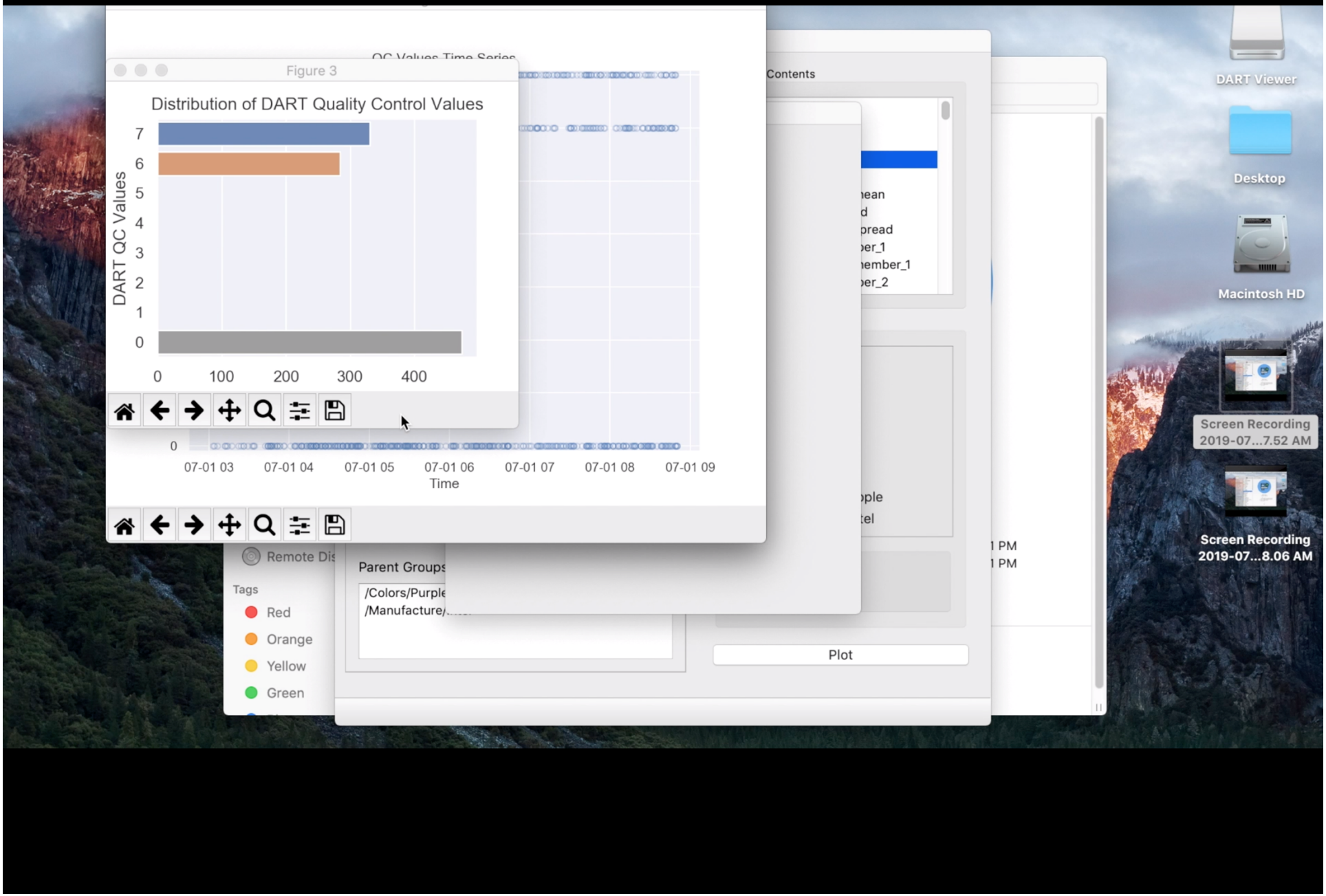
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



SOLUTION – GRAPHICAL USER INTERFACE (GUI)



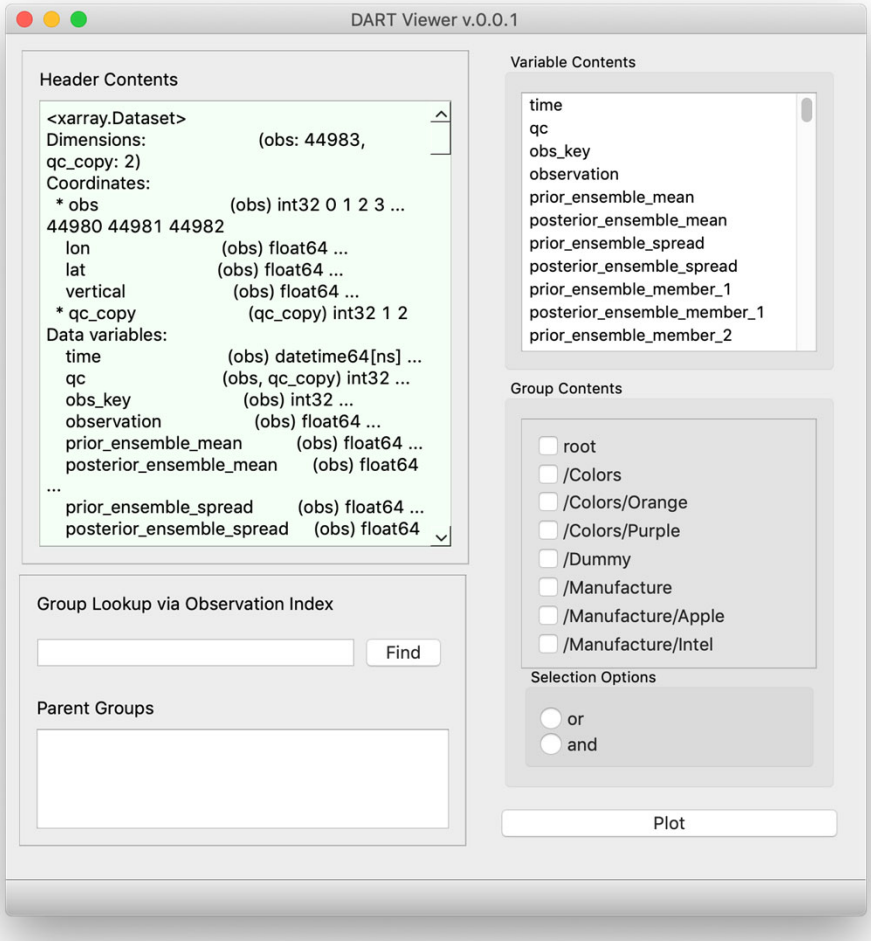
SOLUTION – GRAPHICAL USER INTERFACE (GUI)



SOLUTION – GRAPHICAL USER INTERFACE (GUI)

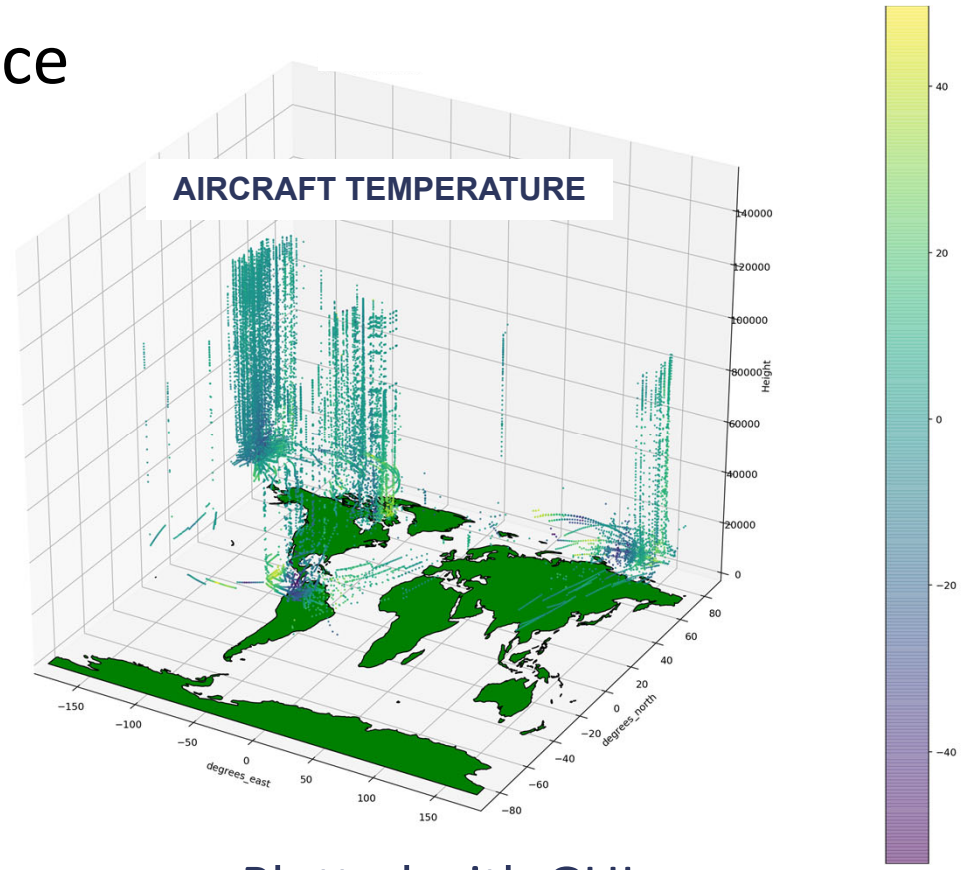
Python Backend

- 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
- or at the airport on Saturday

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NSF