Session 8: Special Cases Relevant to RAF

Miscellaneous Useful Routines and Tips

Al Cooper

[1]Research Aviation Facility, Earth Observing Laboratory
National Center for Atmospheric Research

Presentation prepared 3/8/2015

Outline

MAKING DATA REVIEW MORE COMPREHENSIVE/EASIER

A starting point rather than an end point!

The Goal:

Generate a set of plots and statistics so that all measurements accessible from the netCDF files can be reviewed by examining the output. Review of the plots should help identify where study in more depth is needed while enabling those in the field to keep constant watch on the measurements.

Planning ahead!

- Create a structure consisting of plot functions and a main script that calls these functions.
- Functions can be added or subtracted for specific projects.
- Special runs possible with subsets of plots or times.
- Use github as the repository to enable easy updates in the field.

CURRENT STATUS FOR WINTER

Program Review.R

- In use for data review in the field
- part of batch processing; run automatically
- generates both pdf and html files with plots
- plots saved to the field catalog
- Memo describing use and interpretation:
  see this link
- Program and memo-generating program reside on tikal
- Also archived on github here (as Review.zip, part of larger archive)

BASIC OPERATIONS ON NETCDF FILES

Some useful things to do:

1. Add a variable with attributes
2. Modify a variable and/or attributes
3. Delete a variable
4. Create subset files for economy and for reproducibility archives

Examples:

1. Apply Schuler pitch correction
2. Smooth / interpolate / filter using centered filters
3. Check calculations of, e.g., wind
4. Set values missing
5. Add height-above-terrain variable
6. Add LAMS variables
7. Correct attributes without reprocessing

BASIC STRUCTURE

Steps:

1. load libraries
2. open file
3. manipulate
4. close

code:

```
library(ncdf)
NF = open.ncdf ()
## add variable "A", values A
vA <- var.def.ncdf ("A",...)
newf <- var.add.ncdf (NF, vA)
put.var.ncdf (newf, "A", A)
close.ncdf (newf)
```

ADD A VARIABLE

Add "HOT/HAT" = height of/above terrain

```
copy file to duplicate
read needed LATC, LONC, GGALT from duplicate
HOT < HeightOfTerrain (D$LATC, D$LONC)
HAT <- D$GGALT - HOT
NF <- open.ncdf (...) ## duplicate file
varHOT <- var.def.ncdf (“HOT”, “m”,
   netCDFfile$dim[“Time”], -32767.,
   “Elevation of the Earth’s surface below the GV”)
varHAT <- --similar--
newf <- var.add.ncdf (NF, varHOT)
att.put.ncdf (newf, “HOT”, ... )  # add attribute
newf <- var.add.ncdf (newf, varHAT)
att.put.ncdf (newf, “HAT”, ... )
put.var.ncdf (newf, “HOT”, HOT)
put.var.ncdf (newf, “HAT”, HAT)
close.ncdf (newf)
```

MODIFY A VARIABLE

Set HOT missing if HOT < 20

```
copy file to duplicate
get HOT from dup. netCDF file (getNetCDF)
HOT[HOT < 20] <- -32767.
NF <- open.ncdf (...)   # use duplicate
put.var.ncdf (NF, "HOT", HOT)
close.ncdf (NF)
```

CREATE A SUBSET NETCDF FILE

Easy solutions for small subsets:

- For specific time and variables, use Ranadu::ncsubset
- NCKS (used by ncsubset)

Removing variables or time

- Ranadu::getNetCDF associates all attributes with the returned data.frame (recent change): Complete correspondence between data.frame and netCDF file.
- Ranadu::makeNetCDF creates a netCDF file from a data.frame
- Simply edit data.frame before calling makeNetCDF, e.g., by eliminating a time interval or a variable (example: Data$WI <- NULL). Also can add a variable and appropriate attributes. When subsetting, it may be necessary to make appropriate changes to the attributes.

RECENT ADDITIONS TO Ranadu:

New functions:

1. skewT: construct a plotted thermodynamic diagram
   (a) uses updated thermodynamics and water vapor functions
   (b) added to Review.R
2. AdiabaticTandLWC: calculate values for adiabatic ascent
3. binStats.R: useful for plotting error-bar plots
   (a) classifies values of one variable into bins of another
   (b) returns means and standard deviations for the bins
   (c) used in circle-fit analyses
4. Lagrange interpolation routine
5. PitchCorrection: applies Schuler-oscillation correction to pitch
6. WindProcessor: Calculates wind from basic measurements.
7. makeNetCDF: creates a netCDF file from an R data.frame.

SIGNIFICANT OTHER CHANGES:

Modifications to data.frames from getNetCDF

The attributes from the parent netCDF file are now preserved as attributes of the data.frame (for global attributes) or as attributes of the individual variables (for variable attributes).

- When data.frames are saved for reproducibility, this provides an added level of documentation of what data were used.
- Checking attributes is now easy in R: e.g., attributes(Data$AKRD)
- It is possible to create near-duplicates of the original netCDF file from the data.frame, using Ranadu::makeNetCDF

Other changes:

1. plotTrack: added option for a plot that drifts with the wind
2. plotWAC: new calling options. automatic generation of legends
3. RecoveryFactor now includes heated probes

NEW ROUTINES

Program to enable analysis of circle maneuvers:

- See the routine CircleManeuver.R in ~/RStudio/WINTER
- Documentation: CircleManeuver.pdf
- Incorporated extensively in WindUncertainty.pdf

Develop PCOR to match PSFDC to PSFC

- recalQCF.Rnw in ~cooperw/RStudio/WINTER (on tikal)
- documentation: ~cooperw/RStudio/WINTER/recalQCF.pdf

SOME THINGS I HAVE FOUND USEFUL

- **rbind:** concatenate data.frames by rows. Example: Construct one data.frame with all the measurements from a project to use when determining sensitivity coefficients, etc.
- **smoothing and interpolation:** Example: see use in Review.R
- **restrictions on data:**
  example: DataV <- Data[Data$TAS > 130,]
- **generating variable names:** for Data$ATX, with V="ATX"
  eval (parse (text=sprintf ("Data$%s", V)))
- **saving data:** can speed repeated execution:
  save(Data, file="./Data.Rdata"); load ("./Data.Rdata")
- **str(object):** very useful for seeing the 'structure' of an object
- **object.size (object):** how big is the object in memory?
  (dim, length, nrow: complementary information but not size)
- **adding to a list, where c() can mix list entries:**
  vlist[[length(vlist)+1]] <- v
- **remove or add a variable from a data.frame:**
  Data$Var <- NULL; Data["Var"] <- Var

USEFUL PRACTICES

1. Save the data.frame so running can be reproducible.
2. Put segments of code into 'chunks' and load them into scripts, to make it easy to re-use them. 'source()' and 'read_chunk' are useful for this.
3. Use and re-use functions for common tasks like smoothing or summarizing fit results. RStudio makes this particularly easy.
4. I find it's usually easier to use base plots first, then change to ggplot for more attractive plots.
5. Use round(x,digits) and format (x, nsmall) to format output; use options and obts_chunk$set to set global options.
6. If you are going to save a data.frame, make sure the attributes get transfered to it if necessary. These are lost on sub-setting. See HELP with getNetCDF for how to do this.

Hate typing <- ? In RStudio, use [Alt]-