18 January 2016

TO: DataReview File
FROM: Al Cooper
SUBJECT: Using the DataReview Shiny app

Introduction, installation, and initial run

This is one route to installation of this package on a new system and running it as a Shiny app. This assumes that recent versions of R and RStudio are already installed and that 'git' is installed; if not, see the guidance in the RanaduManual.pdf located in the Ranadu directory on tikal or barolo or at this URL: https://drive.google.com/open?id=0B1kIUH45ca5AYnNfcDZuQm9aV2c.

If you have already installed Ranadu according to those instructions, you have an RStudio directory in your home directory. If not, create one.

Then follow these steps:

1. Run rstudio from your RStudio directory
2. Download the DataReview project from the github repository, as follows:
   (a) In the 'File' menu (top left), select 'New Project'
   (b) Select 'Version Control'
   (c) Select 'Git'
   (d) Type this for the Repository URL: https://github.com/WilliamCooper/DataReview.git
   (e) The Project directory name should be DataReview
   (f) Make sure it is created as a subdirectory of ~/RStudio
   (g) Click 'CreateProject'. RStudio should then download the repository in the new directory and switch you to the new project.
3. Update the 'Ranadu' package if necessary: (If you don't have it, see the RanaduManual.pdf above.)
   (a) In RStudio, switch to the Ranadu project. You can use the drop-down tab at the upper right corner of RStudio, or you can use 'File', 'Open Project', go to the ~RStudio/Ranadu directory, and select Ranadu.Rproj.
   (b) Select the 'git' tab in the top right pane.
   (c) Select the 'pull' arrow. If it succeeds, the repository should update and you will get a pop-up window that you can dismiss via the 'Close' button. If it doesn't succeed, you can either find a git expert to help or remove and re-install the entire Ranadu repository following the RanaduManual instructions.

(d) Once you have the repository in place, select the 'Build' tab (next to 'Git' in the top right pane, then the 'Build & Reload' item in the 2nd line at the left side. This should build and install the package. If it is already up-to-date, you may see that button and can skip this step.

4. Now install the 'shiny' package:

   (a) Select the 'packages' tab in the lower-right pane.

   (b) Click 'install'

   (c) Make sure the top drop-down menu says 'Repository (CRAN)'

   (d) type 'shiny' in the middle entry window

   (e) the default 'Install to library' will not need changing.

   (f) click 'install'

5. Now you need two preparatory steps:

   (a) You need a netCDF data file in an appropriate directory. Appropriate directories include: whatever is pointed to by $DATA_DIR, /scr/raf_data/, /home/data/, /home/Data, /Data, ~/Data . Ranadu will search these in this order. To function right, as presently configured there should be a netCDF files present named ORCAStf01.nc .

   (b) The default is set up to start with project ORCAS and flight rf01. If this is not what is wanted, change 'global.R' in DataReview to have the desired initial project name where it is defined (at about line 183), and change the default project and flight number in 'ui.R' by making the desired project first in the choices for the project list (at about line 11) and the Value for the 'Flight' variable (at about line 16) to be whatever is desired instead of the default 1. For ORCAS, you should not need to do this.

6. Now run by typing 'shiny::runApp()' (without the quote marks) into the console, the lower left pane in RStudio. This will probably appear in a pop-up window, but you can configure to make the display appear in the RStudio 'Viewer' window (in the bottom right pane; this works best if you maximize the viewer pane by clicking on the taller of the two rectangular icons at the top right of the bottom-right pane), or you can configure to have the display appear in a frame in your browser.

Overview of the program structure

The program uses a series of plot functions, defined in the project subdirectory 'PlotFunctions' to generate the plots. These are mostly the same as used in the previous 'Review.R' script, and each takes as an argument the data.frame generated by reading the netCDF file. There are 23 plot files

and two more that are configurable skeletons; for more, use the model in the directory to construct additional plot functions named RPlot26.R to RPlot30.R.

The shiny app itself uses four files:

1. global.R – run once at start-up. This is where initial configuration values are determined, and it also defines a number of functions that are used by other components. One function you might want to change is the limitData function, now set to eliminate measurements from the plots whenever TASX < 110 or abs(ROLL) > 5; this is intended to eliminate regions like the start and end of flights when flaps may be deployed or turns where wind measurements might be affected, and it is activated by a check-box in the DataReview panel. You could follow the pattern used now to change what you want to exclude when this check-box is active.

2. ui.R – controls the user interface, so it defines the interactive components and how they are displayed on the screen.

3. server.R – contains code segments that are activated when particular interactive components as defined in ui.R are changed. This generates the displayed plots.

4. Configuration.R – the configuration file, with separate sections for each project. The ORCAS segment is at the top. This file is 'sourced' at the beginning of the run and anytime the defined Project changes. It sets a list 'VRPlot' that determines the variables to be used in calls to the plot functions.

In addition, there is a 'Review.R' script in the project directory that can be used to include batch production of the PDF-format plots.

Data sources

The program can use two types of data files, netCDF files (ending in '.nc') and R save files (ending in '.Rdata'). These must reside where the DataDirectory () function will find them. The first item in the DataDirectory search path is the environmental variable '$DATA_DIR', so if there are problems finding the data setting this environmental variable should solve them.

The netCDF files are the conventional data product from nimbus. Advantages of using these files is that all the variables and information as processed is available. The disadvantage is that they are much larger than the corresponding Rdata files, and applications like that on Amazon Web Services may use too much storage and take longer to transfer the files.

The Rdata files can be produced by the DataReview program, as described below; see the 'R' button. These files contain all the variables included in the project configuration, but not others that might be of interest as additions to plots. The advantage is that the files are much smaller, typically <10% of the netCDF-file size, and load and process faster. They also transfer via FTP

faster. The standard PDF-producing run will work with these files. The disadvantage is that only the standard variables are included so some of the flexibility to modify plots can't be realized with these files. One partial solution is, before making the Rdata files, to modify the Configuration.R file for the project to reference the additional variables you might want.

If both types of file are present for a particular project and flight, the netCDF file will be used.

A simple normal run:

1. select the 'DataReview' project in RStudio. Make sure it is configured to start with the working directory set to the project directory; i.e., in the console window, 'getwd()' should return the 'DataReview' project directory.
2. start by typing the console command 'shiny::runApp()'
3. Change the flight number to the flight desired.
4. Sequence through the plots by using the incrementing buttons or by clicking your cursor in the 'plot' entry window and using the up-arrow or down-arrow buttons.

Features:

1. *The top-left navigation choices:* The normal choice is "project, flight, and plot" which displays choices for these entries and also has buttons for batch production of plots. The second choice, "time range, restrictions" has additional controls for setting the time range and some data restrictions, as described below. The items immediately below refer to options in the first navigation panel.
   (a) *The project dropdown menu:* Select the project as needed. The initialization process should restrict the projects to those where entries are available in the Configuration.R file and where either PROJECTrf01.nc or PROJECTtf01.nc is available. The accompanying 'save config' button is discussed later.
   (b) *The flight selection window:* Use the radio buttons to select among rf, tf, or ff for research, test, or ferry flights. If the selected flight is not available, a message will appear in the plot window and you can make another selection.
   (c) *The plot selection window:* Plot numbers from 1 to 44 can be selected. Changes can be made via the incrementing buttons at the side of the entry, or by placing the cursor in the selection window and pressing the up or down arrows to move forward or backward in the sequence, or by selecting the value and typing a new value (e.g., to change immediately from plot 1 to plot 26). Note that these numbers do not correspond to the RPlot##() functions that make the plots; many of those functions make several of these plots.

(d) *The pushbutton labeled 'PDF' with an Adobe symbol:* Click on this button (the left button in the set of three buttons at the top-right of the display) to make a series of pdf-format plots that are saved in the DataReview directory. If the 'rstudio' library is available, these will be displayed after construction. It takes about a minute to run through the plots (varying of course with computer speed), so there is a short delay. Other features in the 'DataReview' window will be obeyed; for example, if a time range is selected or the 'apply restrictions' checkbox is checked (see below), the plots will be constructed with those features. The plot file will be overwritten if the button is pressed again.

(e) *The pushbutton labeled 'PNG' with a graphics-image display:* Click on this button (the middle button at the top right of the display) to construct a series of PNG-format plots in the subdirectory PNG. These will not be displayed but will be available for other uses. Any plots with the same name as the default (which is, e.g., ORCASrf01Plot01.png, etc.) will be overwritten without warning, so retrieve files that are needed before making another set of plots. To save a single PNG, use the 'show in new window' button (when running in the RStudio viewer pane) to get an html version of the display, and save the image embedded in that display by right-clicking on that image and selecting 'save image'.

(f) *The pushbutton labeled 'R' with an archive-file image (right button):* Selecting this button will write an .Rdata-format file containing the data.frame in use to the selected project directory. The DataReview shiny app will read this only if the corresponding netCDF file is not present. These files are much smaller than the full netCDF file and will process much faster and, when transferred to other systems, involve much smaller transfers. On EOL computers like barolo, be aware that these are written to the standard data directory (by default /scr/raf_data/{Project}) so they should be removed or moved elsewhere after being generated.

(g) *The 'add var' selection menu:* Not all plots respond to this selector. It will be discussed again below because it is used on conjunction with the 'variables' list.

2. *The 'time range, restrictions' navigation panel:*

(a) *The time slidebar:* The blue entries above the time slidebar show the start and end of the flight. Move the two sliders to limit the displayed time segment. The resolution is one minute, so you can't set the range finer than this. You can also select one of the sliders and use the left-arrow or right-arrow keys to step one minute for each click or in rapid repitition if the key is held down. You can also drag the dark-blue line between the sliders to move both together, e.g., to display a moving 10-min segment. The arrow in the lower right corner starts an animation sequence, where if a subset of the range is selected that subset will move sequentially through the flight.

(b) *The data-restrictions entries:* The four entries at the top right can be used to restrict the data. These restrictions are not imposed unless the 'apply restrictions' checkbox, discussed below, is checked. When it is, the following four constraints can be set:

i. 'tas min': the minimum TASX to be included. For all variables, data will not be included in plots and summaries when TASX is below the threshold set here.

ii. 'roll': the maximum absolute value of the roll to be included.

iii. 'Zmin-km': the minimum altitude to include, in units of km. All measurements from lower altitudes will be excluded.

iv. 'abs ROC': the maximum absolute value of the rate-of-climb to include. This test is based on VSPD because, although it is not the best ROC variable, the other available variables have varied, while VSPD has always been present.

3. *The left-sidebar functions:*

(a) *The left-sidebar top text line:* This is a descriptive title for the currently selected plot.

(b) *The left-sidebar 'plot class' selection list:* Use these selections to jump to sets of plots that provide information on particular types of measurements. For example, the 'wind' selection jumps to a set of plots that include winds, IRU performance, Schuler oscillation, and complementary-filter performance. The 'extras' at the end are placeholder plots that can be reconfigured to provide new plots.

(c) *The 'apply restrictions' checkbox:* Checking this box restricts the displayed measurements to those satisfying the constraints as set in the data-restriction top panel. These are intended to avoid periods of slow flight when flaps might be deployed and measurements in turns when some measurements like those of wind might be affected, so that those contributions don't affect the statistics that appear above some of the plots or in the 'stats' tab and other tabs (discussed below).

(d) *The 'variables' selection box in the left-sidebar:* Many of the plots have associated sets of measurements that could be used in those plots. For example, plot #3 shows measurements of temperature, so the set of variables that might be used here are shown in this selection list. Those actually used are highlighted. To make additional selections or to remove any selection, use a'CNTL-left-mouse' click with the cursor on the variable.

This is where the 'add var' drop-down menu near the top right corner can be used. If the algorithm used by the program to find candidate temperatures does not include a variable you want, you can select that variable from the 'add var' list and it will be added to the 'variables' list available for selection. For example, you might want to include DPXC on the plot of temperatures. Select DPXC from the 'add var' dropdown menu and it will appear in the 'variables' list. CNTL-left-mouse-click on it and it will be added to the display. Note that, if you know the name of the variable you want, you can save scrolling through the list by just typing the variable name in the top window.

This selection menu may be particularly useful in conjunction with the 'extras' plots (plots 43 and 44), where you can construct and save a new plot.

When you select a new variable that wasn't being used for these plots, you may see a pause as the program re-reads the netCDF file to get the new variable. Only variables being used are held in R memory as this shiny app runs.

If you want to make your new selection part of the standard configuration for this project, click the 'save config' button near the top left of the display. This must be done before changing to a new plot or the new configuration will be lost, and this must be done individually for each plot that is changed; changes are not preserved once the plot number is changed.

(e) *The 'see in ncplot' button:* Only use this button when running in RStudio on a computer with ncplot installed and with access to your display screen. In that case, clicking this button will bring up ncplot with only the variables in current use. This will not work from the barolo RStudio server because it does not forward the display from ncplot. It will work if you are running on your own computer and have ncplot installed there.

(f) *The 'see in Xanadu' button:* This will likely not be enabled unless you have the Xanadu program and access to your display screen.

(g) *The 'see maneuvers' button:* Clicking this button will start a search through the current file for maneuvers, and those will be printed on the console if you are running via RStudio. An algorithm tries to find candidates for speed-run, pitch, yaw, circle, and reverse-heading maneuvers.

4. *The main display window (right bottom panel in RStudio):* The plots that are generated in response to the user entries appear on the 'plot' tab in this window. The 'stats' tab displays mean values and standard deviations, and the 'listing' tab lists individual measurements. The 'histograms' tab shows histograms of the selected variables, and the 'soundings' tab shows box-and-whisker plots of selected variables vs altitude. All obey the 'apply restrictions' checkbox and the limitations of the time slidebar. You can therefore see plots isolated to particular times, statistics over a limited range of the flight, or listings for a range of times.

– End of Memo –

Reproducibility:

PROJECT:	DataReviewManual
ARCHIVE PACKAGE:	DataReviewManual.zip
CONTAINS:	attachment list below
PROGRAM:	DataReviewManual.Rnw
GIT:	https://github.com/WilliamCooper/DataReviewManual.git

Attachments: DataReviewManual.Rnw
DataReviewManual.pdf
SessionInfo