

Session 4: R Packages

A sampler; also, 'Ranadu'

Al Cooper

RAF Sessions on R and RStudio

What is a package?

"Base" functions

- Most of what we have been reviewing is in the base package
Always available, always loaded.
- Many functions, like `plot()`, are in other standard packages like 'graphics'
- Want to see everything available on CRAN?
See this CRAN URL; better starting point is this URL

RStudio: see the 'Packages' button:

- 1 Most are inactive in the sense that they are not using memory or available. To use:
 - (a) check the box;
 - (b) `require(signal)` or `library(ggplot2)`;
 - (c) also `beanplot::beanplot`; often useful
- 2 On barolo, all the standard EOL packages. Setting `.Renv` appropriately gives you access to the packages of others.

A few to note:

Recently used:

- 1 ncdf4: basic netCDF functions
- 2 ggplot2 and ggthemes
- 3 signal (includes filtering)
- 4 devtools: helpful constructing packages
- 5 nleqslv: solve non-linear equations
- 6 knitr: intermix text and R code
- 7 maps and mapproj
- 8 shiny: interactive apps
- 9 zoo: na.approx for interpolation

Data-access functions:

Data <- getNetCDF (): loads data.frame with requested variables
V <- standardVariables (): defines a common set
DataDirectory (): "/scr/raf_data/" on barolo
i <- getIndex (): find index for a specified time
r <- setRange (): set a range of indices to a specified time interval
TellAbout (V): lists some characteristics of V

R code and response:

```
Project <- "DEEPWAVE"  
Flight <- "rf15"  
fname <- sprintf("%s%s/%s%s.nc", DataDirectory(), Project,  
  Project, Flight) # or fname <- '...'  
Data <- getNetCDF(fname, standardVariables(c("GGALT", "PITCH")),  
  Start = 40000, End = 53000, F = 15) # loads data.frame  
names(Data) # shows variables in Data  
[1] "Time" "ATX" "DPXC" "EWX" "GGALT" "LATC" "LONC" "MACHX"  
[9] "MR" "PALT" "PSXC" "QCXC" "TASX" "WDC" "WSC" "WIC"  
[17] "PITCH" "RF"
```

R code and response:

```
TellAbout (Data)
```

```
[1] "Variable class is data.frame, length = 18, dim = "
```

```
[2] "5401"
```

```
[3] "18"
```

Time		ATX		DPXC	
Min.	:2014-07-03 04:00:00	Min.	:-55.67	Min.	:-63.12
1st Qu.:	:2014-07-03 04:22:30	1st Qu.:	:-54.48	1st Qu.:	:-61.02
Median	:2014-07-03 04:45:00	Median	:-31.59	Median	:-50.41
Mean	:2014-07-03 04:45:00	Mean	:-38.89	Mean	:-50.40
3rd Qu.:	:2014-07-03 05:07:30	3rd Qu.:	:-30.35	3rd Qu.:	:-40.83
Max.	:2014-07-03 05:30:00	Max.	:-12.03	Max.	:-20.51

EWX		GGALT		LATC		LONC	
Min.	:0.01239	Min.	:2929	Min.	:-45.94	Min.	:170.7
1st Qu.:	:0.01633	1st Qu.:	:5767	1st Qu.:	:-45.40	1st Qu.:	:171.7
Median	:0.06023	Median	:5774	Median	:-44.71	Median	:172.4
Mean	:0.10355	Mean	:6729	Mean	:-44.68	Mean	:172.4
3rd Qu.:	:0.17339	3rd Qu.:	:8693	3rd Qu.:	:-43.88	3rd Qu.:	:173.3
Max.	:1.20097	Max.	:8817	Max.	:-43.45	Max.	:173.8

MACHX		MR		PALT		PSXC	
Min.	:0.4112	Min.	:0.01808	Min.	:3170	Min.	:295.7

More about getNetCDF ():

- 1 The first variable returned is "Time". This is converted from the time variable used in netCDF files (seconds after a specified reference time) to 'POSIX'-format time that is understood by R.
 - (a) Gives appropriate labels in plots vs time.
 - (b) Includes date; no ambiguity if data.frames are merged.
 - (c) Requires interpretation; not a simple index. This works:

```
Data$ATX[Data$Time==as.POSIXct("2014-07-04 08:33:19",  
tz='UTC')]
```


– but see 'getIndex', an easier way to reference one time
- 2 Handles high-rate files by returning 25 values per second in flat arrays. Where variables are lower rate, interpolation is used, Savitzky-Golay with 4th-order polynomials spanning 3 s centered on each 25-Hz point, so all variables are 25-Hz.
- 3 Data\$RF is included to be able to merge resulting files and still identify data from individual flights: `Data[RF==15,]` gives only measurements from that flight.

(not-Ranadu) Ways of getting data into R: tables

read.table ()

- Easy way to read data in text spreadsheet form:
export from Excel in CSV format
read.table with the same separator as the argument
- other options include 'header' and 'skip'
- The 'file' argument can also be a complete URL. This URL with the code below will download the current Denver sounding as a data.frame.

```
Names <- read.table(file=URL_UW, skip=7, nrows=1)
```

```
A <- read.table (file=URL_UW, skip=13, nrows=70,  
  col.names=as.vector(t(Names))); head(A)
```

```
##      PRES HGHT TEMP DWPT RELH MIXR DRCT SKNT  THTA  THTE  THTV  
## 1  823.0 1625 17.6 -0.4   30 4.54   20   10 307.4 321.7 308.2  
## 2  822.0 1635 17.0  0.0   32 4.68   21   10 306.9 321.6 307.7  
## 3  802.9 1829 15.2 -0.8   33 4.52   45   13 307.0 321.2 307.8  
## 4  773.8 2134 12.3 -2.0   37 4.28   65   12 307.2 320.7 308.0  
## 5  745.9 2438  9.5 -3.3   41 4.05   90   14 307.4 320.2 308.1  
## 6  718.8 2743  6.7 -4.5   45 3.83  120   10 307.5 319.6 308.2
```

(not-Ranadu) Ways of getting data into R: HTML pages

```
readHTMLTable(URL, ...)
```

Example: RTD schedule, route 228 southbound at the RAF hangar:

```
suppressMessages(require(XML))
Schedule <- readHTMLTable(U, header = FALSE, which = 1,
  skip.rows = 1:10)
names(Schedule) <- c("Stop1", "2", "3", "4", "5", "6", "7",
  "(RAF)", "BPNR", " ")
head(Schedule[, 8:9], 9)
```

	(RAF)	BPNR
1	120P	--
2	220P	--
3	321P	--
4	352P	402P
5	422P	--
6	452P	--
7	522P	--
8	552P	--
9	622P	--

Available in Ranadu:

MurphyKoop (DP, P)

DPfromE (E)

MixingRatio

PotentialTemperature

EquivalentPotentialTemperature

WetEquivalentPotentialTemperature

VirtualTemperature

VirtualPotentialTemperature

MachNumber

TrueAirspeed

PCorFunction

KingProbe

AdiabaticTandLWC

AirTemperature

calcAttack

GV_AOAfromRadome

GV_YawFromRadome

ButterworthFilter

ComplementaryFilter

Gravity

PressureAltitude

RecoveryFactor

SpecificHeats

StandardConstant

CAPE

WindProcessor

Convenience and Special Functions:

Now available:

DataDirectory ()
GetAttributes (V)
getIndex (Time, HHMMSS)
r <- setRange (Time, Start, End)
getRAFDData ()
getStartEnd(Time)
ncsubset ()
binStats ()
TellAbout (V)
ValueOf ()
ValueOfAll ()

Special (available):

DemingFit ()
AdiabaticTandLWC ()

Plotting routines (available):

plotWAC ()
ggplotWAC ()
lineWAC ()
theme_WAC ()
plotTrack ()
skew-T based on Davies-Jones
pseudo-adiabatic lines
Paluch and Betts plots

Development projects:

- 1 size distributions: CDP etc.
- 2 Spectral-analysis and autocorrelation functions (available now but reliant on Xanadu, needs re-coding)

Standard help functions:

```
?Ranadu::getNetCDF  
?ggplotWAC  
?Ranadu::Ranadu  
etc
```

The manuals for Ranadu and the Ranadu Shiny App

- 1 See the manuals in the directory specified by the R function `path.package('Ranadu')`.
- 2 See the version on GitHub at this URL:
<https://github.com/WilliamCooper/Ranadu/blob/master/inst/RanaduManual.pdf>