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 tikal, all the standard EOL packages. Setting .Renviron appropriately gives you access to the packages of others.

A few to note:

Recently used:

1. ncdf: 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

'Ranadu'

Data-access functions:

Data <- getNetCDF ( ): loads data.frame with requested variables
V <- standardVariables ( ): defines a comment set
DataDirectory ( ): "/scr/raf_data/" on tikal
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("GGALTB", "PITCH")),
    Start = 40000, End = 53000, F = 15)  # loads data.frame
names(Data)  # shows variables in Data
 [1] "Time"    "ATX"     "DPXC"    "EWX"     "GGALT"   "LATC"    "LONC"
 [8] "MACHX"   "MR"      "PALT"    "PSXC"    "QCXC"    "TASX"    "WDC"
[15] "WSC"     "WIC"     "GGALTB"  "PITCH"   "RF"
```

'Ranadu'

R code and response:

```
TellAbout (Data)
[1] "Variable class is data.frame, length = 19, dim = "
[2] "5401"
[3] "19"
      Time                          ATX               DPXC
 Min.   :2014-07-03 04:00:00   Min.   :-56.10   Min.   :-63.14
 1st Qu.:2014-07-03 04:22:30   1st Qu.:-54.99   1st Qu.:-61.04
 Median :2014-07-03 04:45:00   Median :-32.03   Median :-50.43
 Mean   :2014-07-03 04:45:00   Mean   :-39.31   Mean   :-50.41
 3rd Qu.:2014-07-03 05:07:30   3rd Qu.:-30.70   3rd Qu.:-40.84
 Max.   :2014-07-03 05:30:00   Max.   :-12.48   Max.   :-20.52
                                                NA's   :3
      EWX                GGALT            LATC              LONC
 Min.   :0.01236   Min.   :2929   Min.   :-45.94   Min.   :170.7
 1st Qu.:0.01630   1st Qu.:5767   1st Qu.:-45.40   1st Qu.:171.7
 Median :0.06014   Median :5774   Median :-44.71   Median :172.4
 Mean   :0.10342   Mean   :6729   Mean   :-44.68   Mean   :172.4
 3rd Qu.:0.17322   3rd Qu.:8693   3rd Qu.:-43.88   3rd Qu.:173.3
 Max.   :1.19992   Max.   :8817   Max.   :-43.45   Max.   :173.8
 NA's   :3
```

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'
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 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.

Other 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
## 1 849.0 1625  0.4 -2.8   79 3.68    0    0 286.6 297.4 287.3
## 2 848.0 1634  1.2 -3.8   69 3.42  356    0 287.6 297.6 288.2
## 3 846.0 1653  2.4 -3.6   65 3.48  348    1 289.0 299.3 289.6
## 4 843.0 1682  4.0 -4.0   56 3.39  336    2 291.0 301.1 291.6
## 5 833.0 1778  5.8 -6.2   42 2.90  296    5 293.9 302.8 294.4
## 6 827.8 1829  6.5 -5.8   41 3.01  275    6 295.1 304.3 295.7
```

Other ways of getting data into R: HTML pages

readHTMLTable(URL, ...)

Example: RTD schedule for route 228 southbound:

```
require(XML)
Loading required package:  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 1120A 1130A
2 1220P 1230P
3  120P  130P
4  220P  230P
5  321P  331P
6  352P  402P
7  422P  432P
8  452P  502P
9  522P  532P
```

Algorithm Functions

Now available:

MurphyKoop (DP, P)
DPfromE (E)
MixingRatio
PotentialTemperature
EquivalentPotentialTemperature
WetEquivalentPotentialTemperature
VirtualTemperature
VirtualPotentialTemperature
MachNumber
TrueAirspeed
PCorFunction
KingProbe

AirTemperature
calcAttack
GV_AOAfromRadome
GV_YawFromRadome
ButterworthFilter
ComplementaryFilter
Gravity
PressureAltitude
RecoveryFactor
SpecificHeats
StandardConstant

Convenience and Special Functions:

Now available:

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

Special (available):

DemingFit ( )
AdiabaticTandLWC ( )

Plotting routines (available):

plotWAC ( )
lineWAC ( )
theme_WAC ( )
plotTrack ( )

Development projects:

1. ggplotWAC ( )
2. size distributions: CDP etc.
3. Soundings:
   - (a) Skew-T based on Davies-Jones pseudo-adiabatic lines
   - (b) Paluch and Betts plots
4. Spectral-analysis and autocorrelation functions

NEXT TIME: Generating Figures