

Some R 2.5.1 Preliminaries

Convenience features to make life a bit easier
when working with R

Two important environment variables

```
> Sys.getenv( "R_HOME" )
```

```
R_HOME
```

```
"C:\\PROGRA~1\\R\\R-25~1.0"
```

```
> Sys.getenv( "HOME" )
```

```
HOME
```

```
"C:\\Documents and Settings\\Bill Venables\\My Documents"
```

```
> getwd( )
```

```
[1] "C:/Documents and Settings/Bill Venables/My Documents/R  
Modelling"
```

S_HOME

- Usually located at
`C:/Program Files/R/R-2.x.y`
- Main subdirectories:
 - `bin` (executables)
 - `include` (for programming)
 - `lib` (for programming)
 - `library` (standard libraries)
 - `doc` (documentation)
 - `etc` (initialisation, &c)

Initialisation

- System wide:
 - commands in file `R_HOME/etc/Rprofile.site`
- Global to a particular user:
 - commands in file `R_HOME/.Rprofile`
- Particular to a chapter:
 - commands in file `<wd>/.Rprofile`
 - commands in function `.First`

Simple structure of the working directory

- Parent directory (folder) containing:
 - **.RData** saved image,
 - created and maintained by R
 - **.Rprofile** if required.
 - Automatically executed when S-PLUS starts
 - **File.RData** saved objects for loading
 - Data files (***.xls**, ***.csv**, ***.txt**, ...),
 - Scripts, (***.R**, ***.q**)
 - Output files

Editing the GUI Preferences

Rgui Configuration Editor

Single or multiple windows ☐ MDI ☒ SDI ☒ MDI toolbar ☐ MDI statusbar

Pager style ☐ multiple windows ☒ single window Language for menus and messages

Font ☒ TrueType only size style

Console rows columns Initial left top
☒ set options(width) on resize? buffer bytes lines
☒ buffer console by default?

Pager rows columns

Graphics windows: initial left top

Console and Pager Colours

| Background | Output text | User input | Titles in pager |
|---|---|--------------------------------------|--|
| <input type="text" value="MistyRose3"/> | <input type="text" value="NavajoWhite3"/> | <input type="text" value="purple3"/> | <input type="text" value="DarkOrchid3"/> |
| <input type="text" value="MistyRose4"/> | <input type="text" value="NavajoWhite4"/> | <input type="text" value="purple4"/> | <input type="text" value="DarkOrchid4"/> |
| <input type="text" value="moccasin"/> | <input type="text" value="navy"/> | <input type="text" value="red"/> | <input type="text" value="DarkRed"/> |

Some possibilities for `.Rprofile` files

```
### extracts from my user .Rprofile
options(show.signif.stars = FALSE,
       length = 999999)
### for this course
require(MASS, quietly = T)
require(nnet, quietly = T)
```

The ASOR package

- Allows data sets to be stored out of memory,
- Objects remain visible – 'promises' are set to load the data on demand.
- Still somewhat experimental...
- Exercise
 - Set up a working directory for this course
 - Install the ASOR package into your machine
 - Load all the data sets provided for this course
 - Store them in an ASOR library
 - Set up a .First function to Attach() the library on commencement of the R session

R, An Introduction to the System

- R is a *language* for manipulating objects
- What sort of *language*?
 - An *interactive, object-oriented, function* language
- What sort of *objects*?
 - Data sets, vectors, matrices, arrays, lists, functions, &c.
- What sort of *manipulations*?
 - Input and output,
 - Construction, computations,
 - Graphical display,
 - Assignments, &c.

The R Session

- Input and output of data sets
 - From text/SAS/Excel/... files, ODBC connexions,
...
- Methods of interaction:
 - Command line,
 - Native R script file
- Some editor (Tinn-R, WinEdit, Emacs/ESS, ...) and R working in tandem – most powerful of all

Working with the file system

- Set aside a special directory (Microsoft: "folder") for all the files you will need, or generate, for this course.
- We call this the "working directory".
- Start R and using `setwd(...)` make this directory your working directory.
- Save your image, creating an `.RData` file in your working directory
- To start R with this as your working directory, go to that directory with the file explorer and double-click on the `.RData` file
- Sub-directories for special kinds of file (e.g. graphics) are a good idea. Neatness counts.

Always keep an eye on the search path

- List of places from which objects are sought by the evaluator. May include packages, attached lists or data frames, or the 'global environment'

> **search()**

```
[1] ".GlobalEnv"          "package:stats"
[3] "package:graphics"    "package:grDevices"
[5] "package:utils"        "package:datasets"
[7] ".R_Data"              ".R_Utills"
[9] "package:methods"      "Autoloads"
[11] "package:base"
```

R packages needed for the course

- ASOR (available from <http://verde.esalq.usp.br/~jorge/clarice/venables/>)
 - MASS
 - mgcv
 - lattice, grid, rgl
 - nlme, lme4, Matrix
 - RODB, foreign
 - rpart, tree, gbm, randomForest
-
- Choose a .br CRAN mirror and install the ones in red.
The ones in blue should be already there.

Demonstration 1: Installing and Using ASOR

- Not strictly needed, but useful for large data sets
- Allows large objects to be stored out of memory in a special `.R_Store` sub-directory
- **Attach()** – make the stored files visible on the search path
- **Store()** – take objects out of memory and store as files
- **Objects()** – list the stored objects
- **Remove()** – remove objects from the file store

Reading in data

- The simplest way is probably using .csv files
- Data sets for the course will come from several places.
- Some data sets are already available on the course web site

<http://verde.esalq.usp.br/~jorge/clarice/venables/>

- More will be supplied as the course progresses, as need be

Demonstration 2: Reading in and Storing

- Download the CSVFiles.zip and unzip into your working directory
- Start Tinn-R and open the script `makeInputData.R`
- See how it works.
- Start R and execute the script. This will generate another script called `inputData.R`
- Execute `inputData.R` (not as a source file)
- This should start (if need be) an `.R_Store` directory of stored objects

Use of R for modelling

- "S is a *language* for *manipulating objects*"
- When using S (i.e. the R program) use objects to transmit information as completely as possible.
- Avoid using
`attach(data_set)`
...
`detach(data_set)`
- Make frequent use of `with(...)`, `transform(...)` and `subset(...)`

Demonstration 3: using `with()` & friends

- Like making a copy of the data object visible *ahead* of the search path.
- Ensures that variables come from the right place
`value <- with(data, expression)`
- or
`with(data, do_something)`
- `transform()` allows data frames to be extended
- `subset()` allows selection of rows or columns of data frames in a graceful way

Quick example

- The janka data – a leading example

```
with(janka,  
      plot(Density, Hardness,  
            xlim = range(0, Density),  
            ylim = range(0, Hardness),  
            las=0))
```