Workshop

Session 1: Preliminaries

Bill Venables, CSIRO, Australia

UseR! 2012

Nashville

11 June, 2012

Contents

1 Some preliminaries: useful protocols and gratuitous advice

1.1 Use the file system

In working with **R**, *use the file system*. A good protocol is

- For each new project, set up a *working directory* which will contain all the files needed for that project in one place.
- The working directory may contain sub-directories for natural entities such as `Data`, `Fig`, `Archive`, `Scripts`, `Code`, &c
- It is better to start **R** *in* the working directory rather than start elsewhere use the GUI or *`setwd()`* to go there.
- Use the `SOAR` package (next slide) to keep objects available from one session to the next, and discard others. Keep it clean,

1.2 The SOAR package

- Use for keeping objects from one session to the next. Especially useful for *large* objects, or objects requiring a lot of time to generate.
- Keeps `.RData` files of stores objects in a sub-directory of the working directory, `./.R_Cache`.
- *`Store(...)`*: place objects in cache, removing from memory, but still visible as *promises*,
- *`Objects()`* (or *`Ls()`*): list cache contents,
- *`Attach()`*: place the cache on the search path as promises
- *`Remove(...)`*: delete objects from the cache, permanently.

An additions function, *`Search()`*, gives and enhanced view of the current search path.

1.3 Keep related objects together

Bad:

```
> rm(list = ls())
> library(SOAR)
> set.seed(1234)
> x <- sort(runif(500, 0, 10))
> beta <- runif(5, -1, 1)
> eta <- cbind(1, poly(x, 4)) %*% beta
> y <- exp(eta + rnorm(500, 0, 0.1))
> mu <- exp(eta + 0.1^2/2)
> dummyData <- data.frame(x=x, mu=mu, y=y, eta=eta)
> Store(dummyData)
> ls()

[1] "beta" "eta"  "mu"   "x"    "y"

> Ls()

[1] "dummyData"
```

Good:

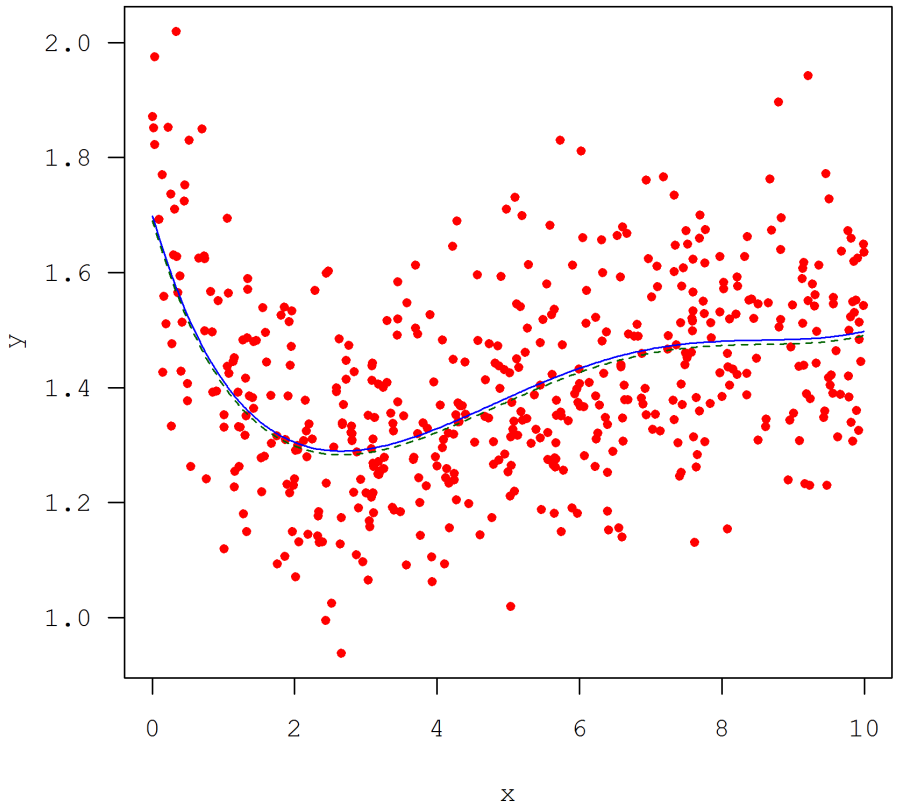
```
> library(SOAR)
> rm(list = ls())
> set.seed(1234)
> dummyData <- within(data.frame(x = sort(runif(500, 0, 10))), {
    beta <- runif(5, -1, 1)
    eta <- cbind(1, poly(x, 4)) %*% beta
    y <- exp(eta + rnorm(500, 0, 0.1))
    mu <- exp(eta + 0.1^2/2)
    rm(beta)
  })
> Store(dummyData)
> ls()

character(0)

> head(dummyData, 2)

           x       mu        y       eta
1 0.006121558 1.698634 1.871466 0.5248243
2 0.021467118 1.692014 1.851811 0.5209193
```

```
> plot(y ~ x, dummyData, pch=20, col="red")
> lines(mu ~ x, dummyData, col="blue")
> lines(exp(eta) ~ x, dummyData, col="darkgreen", lty="dashed")
```



1.4 Do not use *attach()*

- Visibility is key to ensuring **R** gets the right object.

Bad:

```
> attach(dummyData)
> beta <- coef(lm(eta ~ poly(x, 4)))    ## OK so far
> LModel <- lm(log(y) ~ poly(x, 4))     ## object incomplete
> rbind(beta = beta, beta_hat = coef(LModel))

         (Intercept) poly(x, 4)1 poly(x, 4)2 poly(x, 4)3 poly(x, 4)4
beta       0.3378383   0.6020866   0.7109118  -0.8997202   0.3488682
beta_hat   0.3402111   0.5147677   0.9063047  -0.9512177   0.3534272
```

The object *LModel* relies on the context for its meaning. From where does the data come?

These are *much* worse. Meaning unclear and prediction impossible.

```
> rough_1 <- lm(log(dummyData$y) ~ poly(dummyData$x, 4))    ## BAD!
> rough_2 <- lm(log(dummyData[,3]) ~ poly(dummyData[,1],4)) ## Horrible!
```

Good:

```
> beta <- with(dummyData,
               qr.coef(qr(cbind(1, poly(x, 4))), eta))
> LModel <- lm(log(y) ~ poly(x, 4), dummyData)
> rbind(beta = as.vector(beta), beta_hat = coef(LModel))

         (Intercept) poly(x, 4)1 poly(x, 4)2 poly(x, 4)3 poly(x, 4)4
beta       0.3378383   0.6020866   0.7109118  -0.8997202   0.3488682
beta_hat   0.3402111   0.5147677   0.9063047  -0.9512177   0.3534272
```

The object *LModel* now has information on where the data comes from.

Keep an eye on the search path as you work and keep it tidy:

```
> Search()  ## From the SOAR package - enhanced

   name               lib.loc
01 .GlobalEnv
02 dummyData
03 .R_Cache           .
04 package:SOAR       C:/Lib/R
05 package:stats      R_HOME/library
06 package:graphics   R_HOME/library
07 package:grDevices  R_HOME/library
08 package:utils      R_HOME/library
09 package:datasets   R_HOME/library
10 package:methods    R_HOME/library
11 Autoloads
12 package:base       R_HOME/library

> detach("dummyData")
```

1.5 Working protocols

- Find a front-end to **R** with which you feel comfortable. None is ideal (as yet). The following two are cross-platform.
  - `Rstudio` is probably best for beginner;
  - `Emacs + ESS` has a very steep learning curve;
- Establish your primary data sources early.
- *Use scripts!* This is very important.
- *Do not use absolute file names in scripts!* Your file names should be relative to the working directory.
- Establish, via scripts, a clear path from your primary data sources to **R**, and be prepared for changes.
- Use `SOAR` (or equivalent) to hold objects over temporarily from one session to the next, but *do not* rely on the saved object versions.

- Keep your global environment clean and your saved `.RData` file *small*. This will make startup quicker and keep your memory size in check.

A final look at the dummy example:

```
> ci <- confint(LModel)
> data.frame(ci, beta = beta,
             OK = ifelse(ci[, 1] < beta & beta < ci[,2], "yes", "no"),
             check.names = FALSE)
                  2.5 %     97.5 %       beta  OK
(Intercept)  0.3311377  0.3492845  0.3378383 yes
poly(x, 4)1  0.3118799  0.7176555  0.6020866 yes
poly(x, 4)2  0.7034169  1.1091925  0.7109118 yes
poly(x, 4)3 -1.1541055 -0.7483299 -0.8997202 yes
poly(x, 4)4  0.1505394  0.5563150  0.3488682 yes
```

2 Using R: some familiar concepts

- **R** is a *language* for *manipulating objects*.
- In **R**, *everything* is an object and *every object* has a *class*.
- The **R** evaluator is (recursively) given an object manipulation task (function call) and *names* of objects to use.
- Where objects are located is governed by the *scoping* rules, which ultimately lead to the global environment and *search path*.
- Generic manipulations (`print`, `plot`, `summary`, &c) have their detailed operation determined by the *class* of the objects on which they act.

References

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with* **S** (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.

Session information

- R version 2.15.0 (2012-03-30), `i386-pc-mingw32`
- Locale: `LC_COLLATE=English_Australia.1252`, `LC_CTYPE=English_Australia.1252`, `LC_MONETARY=English_Australia.1252`, `LC_NUMERIC=C`, `LC_TIME=English_Australia.1252`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: SOAR 0.99-10
- Loaded via a namespace (and not attached): tools 2.15.0