Introduction to the R Statistical Computing Environment
R Programming

John Fox
McMaster University
ICPSR 2014

Programming Basics

Topics

- Function definition
- Basic matrix operations
- Control structures:
  - Conditionals: if, ifelse, switch
  - Iteration: for, while, repeat
  - Recursion
- Loops:
  - Avoiding loops: apply and its relatives
  - Making loops more efficient
- Non-trivial programming examples
- Simulation
- Debugging and profiling R code
- Object-oriented programming
Choose initial estimates of the regression coefficients, such as \( b_0 = 0 \).

At each iteration \( t \), update the coefficients:

\[
b_t = b_{t-1} + (X'V_{t-1}X)^{-1}X'(y - p_{t-1})
\]

where

- \( X \) is the model matrix, with \( x_i' \) as its \( i \)th row;
- \( y \) is the response vector (containing 0’s and 1’s);
- \( p_{t-1} \) is the vector of fitted response probabilities from the previous iteration, the \( i \)th entry of which is

\[
p_{i,t-1} = \frac{1}{1 + \exp(-x_i'b_{t-1})}
\]

- \( V_{t-1} \) is a diagonal matrix, with diagonal entries \( p_{i,t-1}(1 - p_{i,t-1}) \).

Step 2 is repeated until \( b_t \) is close enough to \( b_{t-1} \). The estimated asymptotic covariance matrix of the coefficients is given by \((X'VX)^{-1}\).

Another approach is to let a general-purpose optimizer do the work of maximizing the log-likelihood,

\[
\log_e L = \sum y_i \log_e p_i + (1 - y_i) \log_e (1 - p_i)
\]

Optimizers work by evaluating the gradient (vector of partial derivatives) of the ‘objective function’ (the log-likelihood) at the current estimates of the parameters, iteratively improving the parameter estimates using the information in the gradient; iteration ceases when the gradient is sufficiently close to zero.

For the logistic-regression model, the gradient of the log-likelihood is

\[
\frac{\partial \log_e L}{\partial b} = \sum (y_i - p_i)x_i
\]
The covariance matrix of the coefficients is the inverse of the matrix of second derivatives. The matrix of second derivatives, called the *Hessian*, is
\[
\frac{\partial \log e L}{\partial b \partial b'} = X'VX
\]

The `optim` function in R, however, calculates the Hessian numerically (rather than using an analytic formula).

---

**Debugging and Profiling R Code**

- Tools integrated with the RStudio IDE:
  - Locating an error: `traceback()`
  - Setting a breakpoint and examining the local environment of an executing function: `browser()`
  - A simple interactive debugger: `debug()`
  - A post-mortem debugger: `debugger()`

- Measuring time and memory usage with `system.time` and `Rprof`
Object-Oriented Programming
The S3 Object System

- S3 versus S4 objects
- How the S3 object system works
- Method dispatch, for object of class "class":
  \[
  \text{generic}(\text{object}) \implies \text{generic.class}(\text{object}) \implies \text{generic.default}(\text{object})
  \]
  - For example, summarizing an object mod of class "lm":
    \[
    \text{summary}(\text{mod}) \implies \text{summary.lm}(\text{mod})
    \]
- Objects can have more than one class, in which case the first applicable method is used.
  - For example, objects produced by glm() are of class c("glm", "lm") and therefore can inherit methods from class "lm".
- Generic functions:
  \[
  \text{generic} \leftarrow \text{function}(\text{object}, \text{other-arguments}, ...) \ \text{UseMethod("generic")}
  \]
  - For example, summary \leftarrow function(object, ...)
    \[
    \text{UseMethod("summary")}
    \]