The R Statistical Computing Environment
Basics and Beyond
R Programming

John Fox

McMaster University

ICPSR/Berkeley 2016

Programming Basics
Topics

- Function definition
- Control structures:
  - Conditionals: `if`, `ifelse`, `switch`
  - Iteration: `for`, `while`, `repeat`
- Recursion

Beyond the Basics
Review of MLE of the Binary Logit Model: Estimation by Newton-Raphson

1. Choose initial estimates of the regression coefficients, such as $\mathbf{b}_0 = \mathbf{0}$.
2. At each iteration t, update the coefficients:

$$\mathbf{b}_t = \mathbf{b}_{t-1} + (\mathbf{X}'\mathbf{V}_{t-1}\mathbf{X})^{-1}\mathbf{X}'(\mathbf{y} - \mathbf{p}_{t-1})$$

   where
   - $\mathbf{X}$ is the model matrix, with $\mathbf{x}_i'$ as its ith row;
   - $\mathbf{y}$ is the response vector (containing 0's and 1's);
   - $\mathbf{p}_{t-1}$ is the vector of fitted response probabilities from the previous iteration, the ith entry of which is

$$p_{i,t-1} = \frac{1}{1 + \exp(-\mathbf{x}_i'\mathbf{b}_{t-1})}$$

   - $\mathbf{V}_{t-1}$ is a diagonal matrix, with diagonal entries $p_{i,t-1}(1 - p_{i,t-1})$.
3. Step 2 is repeated until $\mathbf{b}_t$ is close enough to $\mathbf{b}_{t-1}$. The estimated asymptotic covariance matrix of the coefficients is given by $(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}$.

Beyond the Basics
Review of MLE of the Binary Logit Model: Estimation by General Optimization

- 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

$$\frac{\partial \log_e L}{\partial \mathbf{b}} = \sum (y_i - p_i)\mathbf{x}_i$$

Beyond the Basics

Review of MLE of the Binary Logit Model: Estimation by General Optimization

- The covariance matrix of the coefficients is the inverse of the matrix of second derivatives. The matrix of second derivatives, called the *Hessian*, is

$$\frac{\partial \log_e L}{\partial \mathbf{b} \partial \mathbf{b}'} = \mathbf{X}'\mathbf{V}\mathbf{X}$$

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

Object-Oriented Programming

The S3 Object System

- S3 versus S4 objects
- How the S3 object system works
- Method dispatch, for *`object`* of class *`"class"`*: `generic(`*`object`*`)` $\Longrightarrow$ `generic.`*`class`*`(`*`object`*`)` $\Longrightarrow$ `generic.default(`*`object`*`)`
  - For example, summarizing an object `mod` of class `"lm"`: `summary(mod)` $\Longrightarrow$ `summary.lm(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: `generic <- function(object,` *`other-arguments`*`, ...) UseMethod("generic")`
  - For example, `summary <- function(object, ...) UseMethod("summary")`

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`