Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Estimating Models of Strategic Interaction in R

Brenton Kenkel
University of Rochester

Curtis S. Signorino
University of Rochester

Abstract

Come up with this.

Keywords: one, two, three.

```
R> options(useFancyQuotes = FALSE)
R> library("games")
R> packageVersion("games")

[1] '0.7.0'
```

1. Introduction

R (R Development Core Team 2010)

2. Strategic statistical models

cite curt's work, plus all applications in the literature

introduce leblang as running example

3. Specification and estimation

3.1. Modeling player utilities

The archetypal use of a strategic model is to estimate the effect of observed factors on players' utility for each possible outcome. To avoid an overabundance of parameters and potential

citations

inefficiency, analysts will typically want to make some exclusion restrictions—i.e., to leave some regressors out of some utility equations.[1] This necessitates the use of multiple model formulas, which we handle via the **Formula** package (Zeileis and Croissant 2010). The variables to include in each utility are specified using the standard `formula` syntax, and each set is separated by a vertical bar (`|`). For example, in the `egame12` model, an analyst may want to use the specification

$$\begin{aligned}
U_{11} &= \beta_{11,0} + \beta_{11,1} x_1 \\
U_{13} &= 0 \\
U_{14} &= \beta_{14,0} + \beta_{14,1} x_1 + \beta_{14,2} x_2 \\
U_{24} &= \beta_{24,0} + \beta_{24,2} x_2,
\end{aligned}$$

where x_1 and x_2 are observed variables. The appropriate `Formula` syntax is `y ~ x1 | 0 | x1 + x2 | x2`.

In some of the more complex models, such as `egame123` with its eight utility equations, writing the model formulas manually may be daunting or prone to error. We provide two options for easing the process. First, users may specify the model formulas as a list; the fitting functions then use the internal function `checkFormulas` to convert it to the appropriate `Formula` object.

```
R> f1 <- list(u11 = y ~ x1, u13 = ~0, u14 = ~x1 + x2, u24 = ~x2)
R> games:::checkFormulas(f1)

y ~ x1 | 0 | x1 + x2 | x2
<environment: 0x929e084>
```

(Elements of the list need not be named; in fact, the names are ignored.) Second, the function `makeFormulas` provides interactive prompts for constructing the model formulas step by step. The user needs to supply the name of the model he or she intends to fit and a character vector containing outcome descriptions. For the Leblang data, the appropriate call would look like `makeFormulas(egame12, outcomes = c("no attack", "devaluation", "defense"))`. The following menu will appear at the R console:

```
Equation for player 1's utility from no attack:

1: fix to 0
2: intercept only
3: regressors, no intercept
4: regressors with intercept

Selection:
```

If `3` or `4` is selected, the user will be prompted to enter a space-separated list of variables to include in the utility equation of interest. We use functions from **stringr** (Wickham 2010)

[1] A necessary condition for identification in a strategic model is that no regressor appear in all of a player's utility equations for the outcomes reachable after her move (Lewis and Schultz 2003). The fitting functions and `makeFormulas` enforce this condition. One way to accomplish it is to fix each player's utility to 0 for one outcome. This comes without loss of generality, since Von Neumann–Morgenstern utilities are unique only up to an affine transformation **(cite)**.

in parsing the input. The same menu will then be displayed for player 1's utility from devaluation, player 1's utility from defense, and player 2's utility from defense. The final prompt will ask for the name of the variable (or variables; see Section 3.2 below on dependent variable specification) containing information on the observed outcomes. The function will then return the `Formula` specification corresponding to the given input, which can be supplied as the `formulas` argument of the appropriate fitting function.

3.2. Model fitting

different y specification

3.3. Assessing convergence

4. Analyzing fitted models

References

Lewis JB, Schultz KA (2003). "Revealing Preferences: Empirical Estimation of a Crisis Bargaining Game with Incomplete Information." *Political Analysis*, **11**(4), 345–367.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org/`.

Wickham H (2010). ***stringr**: Make it easier to work with strings.* R package version 0.4, URL `http://CRAN.R-project.org/package=stringr`.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. URL `http://www.jstatsoft.org/v34/i01/`.

Affiliation:

Brenton Kenkel
315A Harkness Hall
Department of Political Science
University of Rochester
Rochester, NY 14627
Email: `brenton.kenkel@gmail.com`

Journal of Statistical Software `http://www.jstatsoft.org/`
published by the American Statistical Association `http://www.amstat.org/`

Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd