

Discrete Choice Models with Random Parameters in R: The Rchoice Package

Mauricio Sarrias
Cornell University

Abstract

Rchoice is a package for R which enables the estimation of a variety of Binary, Count and Ordered models with unobserved and observed heterogeneity in the parameters for cross-sectional and panel data. We implement simulated maximum likelihood methods for the estimation of the coefficients which can assume a variety of distributions. This document is a general description of **Rchoice** and all functionalities are illustrated using real databases.

Keywords: discrete choice models, simulated maximum likelihood, R, econometric, random parameters, hierarchical models.

1. Introduction

A growing number of empirical studies seek to measure the impact of covariates on nominal discrete alternatives. Those models are well known. However, one of the traditional modeling shortcomings is their inability to control for possible observed or unobserved heterogeneity that may exist across individuals. For instance, one might assume that some variable does not affect equally the utility of individuals, therefore there may be a deviation from the mean of the respective coefficient. This type of modelling has been widely used in Multinomial Logit Model known also as Mixed Logit Model (see [Train 2009](#); [Hensher and Greene 2003](#)) and can be estimated in R using **mlogit** ([Croissant *et al.* 2012](#)).

Observed and unobserved heterogeneity can easily be extended to binary, ordered and count data (see [Gourieroux, Monfort, and Trognon 1984](#); [Ng, Carpenter, Goldstein, and Rasbash 2006](#); [Greene 2007, 2012](#); [Greene and Hensher 2010a,b](#)). Allowing parameter values to vary across the population according to some pre-specified distribution overcomes the problem of having a fixed-representative coefficient for all individuals. Furthermore, one might assume that this heterogeneity is not only due to unobserved factors, but also to observed individual characteristics such as socioeconomic variables.

In this document we present the package **Rchoice** for R. **Rchoice** is a package for estimating ordered, count and binary choice models with observed and unobserved heterogeneity in the coefficients. The estimation procedure is based on Simulated Maximum Likelihood (SML) which allow controlling for observed and unobserved heterogeneity in a very flexible way. To our knowledge, only LIMDEP ([Greene 2002](#)) is able to estimate these type of models in a

concise and flexible manner. Therefore, this package is intended to make these estimation methods available to the general public and practitioners in a friendly and flexible way.

This new version of **Rchoice** (version 0.2) allows estimating Panel Data models and makes the inclusion of hierarchical variables more flexible. It also includes new helper functions to get the standard errors of the variance-covariance matrix of the random parameters.

This paper is organized as follows. In section 2 we briefly explain the Simulated Maximum Likelihood Procedure. Section 3 we discuss some technical aspects of Simulated Maximum Likelihood estimations and the formulas used by **Rchoice**. In Section 4 we explain how **Rchoice** handles creates the random coefficients. Section 5 we show the all functionalities of **Rchoice** are illustrated using real databases. Section 6 concludes.

2. Simulated Maximum Likelihood Estimation

In this section we briefly explain some basic ideas of SML procedure. For a more complete treatment of SML see for example [Gourieroux and Monfort \(1997\)](#); [Lee \(1992\)](#); [Hajivassiliou and Ruud \(1986\)](#) or [Train \(2009\)](#).

A random parameter model or random coefficient model permits regression parameter to vary across individuals according to an arbitrarily specified distribution. A fully parametric random parameter model specifies the dependent variable y_i conditional on regressors \mathbf{x}_i and given parameters β_i to have conditional density $f(y_i|\mathbf{x}_i, \beta_i)$, where β_i are iid with density $g(\beta_i|\theta)$. Inference is based on the density of y_i conditional on \mathbf{x}_i and given θ :

$$f(y|\mathbf{x}_i, \theta) = \int f(y|\mathbf{x}, \beta)g(\beta, \theta)d\beta$$

This integral will not have a closed-form solution except in some special cases. For example, we can assume normally distributed random parameters, with $\beta_i \sim N(\mu, \Sigma)$. Then $\beta_i = \mu + \Sigma^{-1/2}\mathbf{v}_i$, where $\mathbf{v}_i \sim N(\mathbf{0}, \mathbf{I})$, thus:

$$f(y|\mathbf{x}, \theta) = \int_{-\infty}^{\infty} f(y|\mathbf{x}, \mu, \Sigma) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\mathbf{v}^2\right) d\mathbf{v} \quad (1)$$

Note that (1) has no close-form solution, that is, it is difficult to integrate out the random parameter and hence it difficult to perform ML estimation. However ML estimation may still be possible if we instead use a good approximation $\hat{f}(y|\mathbf{x}, \theta)$ of $f(y|\mathbf{x}, \theta)$ to form a likelihood function.

But, how can we obtain $\hat{f}(y|\mathbf{x}, \theta)$? A good approximation can be obtained by **Monte Carlo integration**.¹ This procedure provides and alternative to deterministic numerical integration. Here we can ‘*simulate*’ the integration using random draws from the distribution $g(\beta|\theta)$. For

¹Another numerical approximation is Gauss-Hermite quadrature. However, it has been documented that for models with more than 3 random parameters SML performs better.

example, the researcher specifies the function form $g(\boldsymbol{\beta}|\boldsymbol{\theta})$ and wants to estimate the parameter $\boldsymbol{\theta}$. The Monte Carlo approximation is:

$$\hat{f}(y|\mathbf{x}_i, \boldsymbol{\beta}_i, \boldsymbol{\theta}) = \frac{1}{R} \sum_{r=1}^R \tilde{f}(y|\mathbf{x}, \boldsymbol{\beta}_{ir}, \boldsymbol{\theta}),$$

where $\boldsymbol{\beta}_{ir}$ is the r th draw of $\boldsymbol{\beta}$ from $g(\boldsymbol{\beta}_i|\boldsymbol{\theta})$ for individual i . Given independence over i , the SML is the value $\boldsymbol{\theta}$ that maximizes:

$$\hat{\boldsymbol{\theta}}_{SML} \equiv \arg \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{i=1}^N \log \hat{f}(y|\mathbf{x}_i, \boldsymbol{\beta}_{ir}, \boldsymbol{\theta})$$

The following preposition gives the asymptotic distribution of SML estimator. For a complete derivation of the asymptotic properties of the SML and a more comprehensive view see [Lee \(1992\)](#) or [Gourieroux and Monfort \(1997\)](#).

Proposition 1 (Distribution of SML Estimator). *Assume the following:*

1. *The data are from a simple random sample from a dgp with likelihood function $f(y|\mathbf{x}, \boldsymbol{\theta}_0)$ that satisfies the regularity conditions so that the ML estimator is consistent and asymptotically normal with variance matrix $\mathbf{A}^{-1}(\boldsymbol{\theta}_0)$, where:*

$$\left[\frac{1}{N} \sum_{i=1}^N \frac{\partial \log f(y|\mathbf{x}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \bigg|_{\boldsymbol{\theta}_0} \right] \xrightarrow{p} \mathbf{A}(\boldsymbol{\theta}_0)$$

2. *The likelihood function f is estimated using the simulator \hat{f} with \tilde{f} unbiased for f .*

*Then the **simulated maximum likelihood** estimator is asymptotically equivalent to the ML estimator if $R \rightarrow \infty$, $N \rightarrow \infty$ and $\sqrt{N}/R \rightarrow \infty$, and it has a limit normal distribution with:*

$$\sqrt{N}(\hat{\boldsymbol{\theta}}_{SML} - \boldsymbol{\theta}) \xrightarrow{d} N(0, \mathbf{A}^{-1}(\boldsymbol{\theta}_0))$$

3. Technical Aspects of Simulated Maximum Likelihood

In this section, we show the general technical aspect of the SML. This will allow us to accommodate any index-type regression model such as Probit, Logit, Ordered and Poisson model. This section relies heavily on chapter 15 of [Greene \(2012\)](#) and chapter 10 of [Train \(2009\)](#).

3.1. Simulated Maximum Likelihood

In order to develop a general set of results, it is convenient to write each single density in the simulated function as:

$$f(y_{it}|\mathbf{x}_{it}, \boldsymbol{\beta}_{ir}, \boldsymbol{\theta}) = P_{itr}(\boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is the vector that collects all the parameters. The simulated log-likelihood is :

$$\log L_s = \sum_{i=1}^N \log \left(\frac{1}{R} \sum_{r=1}^R \prod_{t=1}^{T_i} P_{itr}(\boldsymbol{\theta}) \right) \quad (2)$$

If we define:

$$P_{ir}(\boldsymbol{\theta}) = \prod_{t=1}^{T_i} P_{itr}(\boldsymbol{\theta}),$$

and:

$$P_i(\boldsymbol{\theta}) = \frac{1}{R} \sum_{r=1}^R P_{ir}(\boldsymbol{\theta}),$$

then, the maximum likelihood can be written as:

$$\log L_s = \sum_{i=1}^N \log P_i(\boldsymbol{\theta}) \quad (3)$$

With this notation, we will be able to accommodate richer specifications of the index function and discrete choice models by simply changing the specification of $P_{itr}(\boldsymbol{\theta})$. As is typically the case, the index model represents a latent process of the form:

$$U_{itr}^*(\boldsymbol{\theta}) = \mathbf{z}_{it}'\boldsymbol{\delta} + \mathbf{x}_{it}'\boldsymbol{\beta}_{ir} + \epsilon_{it}, \quad (4)$$

where \mathbf{z}_{it} is a vector of variables with fixed parameters $\boldsymbol{\delta}$; \mathbf{x}_{it} is a vector of variables with random coefficients $\boldsymbol{\beta}_{ir}$; and ϵ_{it} is the error term. For simplicity, assume that $\boldsymbol{\beta}_i \sim N(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, then the random vector of coefficients can be written as:

$$\boldsymbol{\beta}_{ir} = \boldsymbol{\beta} + \mathbf{L}\boldsymbol{\omega}_{ir}$$

where $\boldsymbol{\omega}_{ir}$ is a vector of random draws from normal standard distribution. If the random parameters are correlated normal, then \mathbf{L} is a lower triangular which produces the covariance matrix of the random parameters, $\mathbf{L}\mathbf{L}' = \boldsymbol{\Sigma}$; otherwise, the matrix \mathbf{L} is a diagonal matrix of standard deviations. The random effect model is a special case in which only the constant is random. A hierarchical model is obtained by allowing the parameter heterogeneity to be partly systematic in terms of observed variables:

$$\boldsymbol{\beta}_{ir} = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{s}_i + \mathbf{L}\boldsymbol{\omega}_{ir},$$

where $\boldsymbol{\Pi}$ is a matrix of parameters and \mathbf{s}_i is a vector of covariates that do not vary across time.² Then, $E(\boldsymbol{\beta}_i) = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{s}_i + \mathbf{L}E(\boldsymbol{\omega}) = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{s}_i$ and its covariance is $\text{Var}(\boldsymbol{\beta}_i) = E(\mathbf{L}\boldsymbol{\omega}(\boldsymbol{\omega}\mathbf{L})') = \mathbf{L}E(\boldsymbol{\omega}\boldsymbol{\omega}')\mathbf{L} = \mathbf{L}\mathbf{L}' = \boldsymbol{\Sigma}$.

²Note that the [Mundlak \(1978\)](#) and [Chamberlain \(1980\)](#) approach to modelling fixed effects is also accommodated by letting $\mathbf{s}_i = \bar{\mathbf{x}}_i$, in the equation for the overall constant term ([Greene and Hensher 2010b](#))

Example 1 (Representation of Correlated Random Parameters). Suppose two correlated random parameters, β_1 and β_2 , whose means depend upon variables S, B and C . Then:

$$\begin{aligned}\beta_{1,ir} &= \bar{\beta}_1 + \pi_{1,1}S_i + \pi_{1,2}B_i + \pi_{1,3}C_i + s_{11}\omega_{1,ir} \\ \beta_{2,ir} &= \bar{\beta}_2 + \pi_{2,1}S_i + \pi_{2,2}B_i + \pi_{2,3}C_i + s_{21}\omega_{1,ir} + s_{22}\omega_{2,ir}\end{aligned}$$

or in vector form:

$$\begin{pmatrix} \beta_{1,ir} \\ \beta_{2,ir} \end{pmatrix} = \begin{pmatrix} \bar{\beta}_1 \\ \bar{\beta}_2 \end{pmatrix} + \begin{pmatrix} \pi_{1,1} & \pi_{1,2} & \pi_{1,3} \\ \pi_{2,1} & \pi_{2,2} & \pi_{2,3} \end{pmatrix} \begin{pmatrix} S_i \\ B_i \\ C_i \end{pmatrix} + \begin{pmatrix} s_{11} & 0 \\ s_{21} & s_{22} \end{pmatrix} \begin{pmatrix} \omega_{1,ir} \\ \omega_{2,ir} \end{pmatrix}$$

In this case, the variance-covariance matrix of the random parameters is:

$$\Sigma = \mathbf{L}\mathbf{L}' = \begin{pmatrix} s_{11} & 0 \\ s_{21} & s_{22} \end{pmatrix} \begin{pmatrix} s_{11} & s_{21} \\ 0 & s_{22} \end{pmatrix} = \begin{pmatrix} s_{11}^2 & s_{11}s_{22} \\ s_{21}s_{22} & s_{21}^2 + s_{22}^2 \end{pmatrix},$$

and the conditional mean vector is:

$$E(\beta_i | \mathbf{s}_i) = \beta + \Pi \mathbf{s}_i$$

Finally, depending on the nature of the dependent variable and the distribution of the error term, the probability of a given outcome can be specified. It is well known that, if the dependent variable is binary, then the probability for each individual in each draw is:

$$P_{itr}(\theta) = F[q_{it} \cdot U_{itr}^*(\theta)] \quad (5)$$

where $q_{it} = 2y_{it} - 1$.³ Furthermore, if the model is Probit, then:

$$\begin{aligned}F[q_{it} \cdot U_{itr}^*(\theta)] &= \Phi[q_{it} \cdot U_{itr}^*(\theta)] \\ f[q_{it} \cdot U_{itr}^*(\theta)] &= \phi[q_{it} \cdot U_{itr}^*(\theta)]\end{aligned}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and the pdf for the standard normal distribution. Likewise, if the model is logit, then:

$$\begin{aligned}F[q_{it} \cdot U_{itr}^*(\theta)] &= \Lambda[q_{it} \cdot U_{itr}^*(\theta)] = \frac{\exp(q_{it} \cdot U_{itr}^*(\theta))}{1 + \exp(q_{it} \cdot U_{itr}^*(\theta))} \\ f[q_{it} \cdot U_{itr}^*(\theta)] &= \Lambda[q_{it} \cdot U_{itr}^*(\theta)] [1 - \Lambda(q_{it} \cdot U_{itr}^*(\theta))]\end{aligned}$$

For the Poisson model, the probability for individual i and the r th draw is:

$$P_{itr}(\theta) = \frac{\exp(-\exp(U_{itr}^*(\theta))) \exp(U_{itr}^*(\theta))^{y_{it}}}{y_{it}!} \quad (6)$$

³As explained by Greene (2012), if the distribution is symmetric, as the normal and logistic are, then $1 - F(\mathbf{x}'\beta) = F(\mathbf{x}'\beta)$. Then, $\log L = \sum_i F(q_i \mathbf{x}_i \beta)$

and for the Ordered model, we have:

$$P_{itr}(\boldsymbol{\theta}) = F[\kappa_j - U_{itr}^*(\boldsymbol{\theta})] - F[\kappa_{j-1} - U_{itr}^*(\boldsymbol{\theta})] \quad (7)$$

where:

$$\kappa_j = \kappa_{j-1} + \exp(\alpha_j)$$

This last reparametrization ensures the ordering of the thresholds.

3.2. Gradient of the Simulated Maximum Likelihood

The SML procedure is very time consuming. If the model is very complex, the algorithm may take hours to converge. Providing the gradient to the maximization procedure can considerably reduce the time to achieve convergence. The **Rchoice** package provides the gradient for all random parameter models.

Next, the formulas used by **Rchoice** to obtain the gradient are given. To obtain the derivatives, we begin with:

$$\frac{\partial \log L_s}{\partial \boldsymbol{\theta}} = \sum_{i=1}^N \left(\frac{\frac{1}{R} \sum_{r=1}^R \left(\frac{\partial \prod_{t=1}^{T_i} P_{itr}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)}{\frac{1}{R} \sum_{r=1}^R \prod_{t=1}^{T_i} P_{itr}(\boldsymbol{\theta})} \right) \quad (8)$$

For the derivative term,

$$\frac{\partial \prod_{t=1}^{T_i} P_{itr}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = P_{ir}(\boldsymbol{\theta}) \sum_{t=1}^{T_i} \mathbf{g}_{itr}(\boldsymbol{\theta}) \quad (9)$$

where we use the fact that $\frac{\partial \log p}{\partial \boldsymbol{\theta}} = \frac{1}{p} \frac{\partial p}{\partial \boldsymbol{\theta}}$ and $\mathbf{g}_{itr} = \partial \log P_{itr}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$. Now, inserting (9) into (8) we get:

$$\frac{\partial \log L_s}{\partial \boldsymbol{\theta}} = \sum_{i=1}^N \left(\frac{\sum_{r=1}^R P_{ir}(\boldsymbol{\theta}) \mathbf{g}_{ir}(\boldsymbol{\theta})}{\sum_{r=1}^R P_{ir}(\boldsymbol{\theta})} \right). \quad (10)$$

where $\sum_{t=1}^{T_i} \mathbf{g}_{itr}(\boldsymbol{\theta}) = \mathbf{g}_{ir}$. Define the weight $Q_{ir}(\boldsymbol{\theta}) = P_{ir}(\boldsymbol{\theta}) / \sum_{r=1}^R P_{ir}(\boldsymbol{\theta})$ so that $0 < Q_{ir}(\boldsymbol{\theta}) < 1$ and $\sum_{r=1}^R Q_{ir}(\boldsymbol{\theta}) = 1$. Then,

$$\frac{\partial \log L_s}{\partial \boldsymbol{\theta}} = \sum_{i=1}^N \sum_{r=1}^R Q_{ir}(\boldsymbol{\theta}) \mathbf{g}_{ir}(\boldsymbol{\theta}) = \sum_{i=1}^N \bar{\mathbf{g}}_i(\boldsymbol{\theta}). \quad (11)$$

Example 2 (Gradients formulas used by Rchoice). For binary models, taking first derivatives on the log of equation (5) we get:

$$\frac{\partial \log P_{itr}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{g}_{itr} = \lambda_{itr}(\boldsymbol{\theta}) \left[\frac{\partial U_{itr}^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right], \quad (12)$$

where:

$$\lambda_{itr} = \frac{q_{it} \cdot f[q_{it} \cdot U_{itr}^*(\boldsymbol{\theta})]}{F[q_{it} \cdot U_{itr}^*(\boldsymbol{\theta})]},$$

and $f = \frac{\partial F(\cdot)}{\partial \boldsymbol{\theta}}$. If the model is the Poisson regression model, then \mathbf{g}_{itr} is given by:

$$\frac{\partial \log P_{itr}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{g}_{itr} = [y_{it} - \exp(U_{itr}^*(\boldsymbol{\theta}))] \left(\frac{\partial U_{itr}^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right).$$

For the ordered model, let $\boldsymbol{\theta}$ the vector collecting all the parameters except for the thresholds parameters. Then, \mathbf{g}_{itr} is given by:

$$\frac{\partial \log P_{itr}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = - \frac{f_{itr,j}(\boldsymbol{\theta}) - f_{itr,j-1}(\boldsymbol{\theta})}{F_{itr,j}(\boldsymbol{\theta}) - F_{itr,j-1}(\boldsymbol{\theta})} \left(\frac{\partial U_{itr}^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right),$$

and by:

$$\frac{\partial \log P_{itr}(\boldsymbol{\theta})}{\partial \boldsymbol{\alpha}_k} = \frac{d_{j,k} f_{itr,j}(\boldsymbol{\theta}) - d_{j-1,k} f_{itr,j-1}(\boldsymbol{\theta})}{F_{itr,j}(\boldsymbol{\theta}) - F_{itr,j-1}(\boldsymbol{\theta})} \left(\frac{\partial \kappa_j}{\partial \boldsymbol{\alpha}_k} - \frac{\partial \kappa_{j-1}}{\partial \boldsymbol{\alpha}_k} \right)$$

with $\delta_{j,k} = 1$ if $j = k$ and 0 otherwise. Finally, if parameters are uncorrelated, then:

$$\frac{\partial U_{itr}^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \mathbf{z}_{it} \\ \mathbf{x}_{it} \\ \mathbf{s}_i \otimes \mathbf{x}_{it} \\ \boldsymbol{\omega}_{ir} \bullet \mathbf{x}_{it} \end{pmatrix}.$$

The **Rchoice** uses this formulas to compute the gradient and uses the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (the default) to iteratively solve the MSL.

4. Drawing from Densities

The SML procedure requires drawing pseudo-random numbers from the specified distribution for simulation. A good performance of SML requires very large number of draws. As explained above, the main drawback to this approach is that with large samples and complex models, the maximization of $\log L_s$ can be very time consuming. Researchers have gained speed with no degradation in simulation performance through the use of small number of Halton draws (Bhat 2001; Train 2000). The idea is that, instead of taking independent random draws, simulation can potentially be improved by selecting evaluation points more systematically and with better coverage (Sándor and Train 2004). In this section, we detail how draws are computed by **Rchoice**.

Suppose that there are K random parameters. Then, the K elements of $\boldsymbol{\omega}_{ir}$ are drawn as

follows. We begin with a K random vector ω_{ir} , that is:

- K independent draws from the standard uniform $(0, 1)$ distribution or
- K independent draws from the m th Halton sequence, where m is the m th prime number in the sequence of K prime numbers beginning with 2.

An important attribute of the Halton values is that they are also distributed in the $(0, 1)$ interval. Then, the primitive draw (Pseudo or Halton draws) is then transformed to the distribution specified by the user as follows:

- $u_{k,ir} \sim U(0, 1)$: primitive draw from halton or pseudo-random number generator
- $w_{k,ir} = \Phi^{-1}(u_{k,ir}) \sim N(0, 1)$

Using these two primitive draws, **Rchoice** creates the random parameters as follows:

1. Normal Parameter:

$$\begin{aligned}\beta_{k,ir} &= \beta_k + \sigma_k w_{k,ir} \\ w_{k,ir} &\sim N(0, 1)\end{aligned}$$

where β_k and σ_k are estimated. Then, $\beta_{k,i} \sim N(\beta_k, \sigma_k^2)$

2. Truncated normal Parameter:

$$\begin{aligned}\beta_{k,ir} &= \begin{cases} \beta_k + \sigma_k w_{k,ir} & \text{if } \beta_{k,ir} > 0 \\ 0 & \text{otherwise} \end{cases} \\ w_{k,ir} &\sim N(0, 1)\end{aligned}$$

where β_k and σ_k are estimated. Then, $\beta_{k,i} \sim N(\beta_k, \sigma_k^2)$ with the share below zero massed at zero

3. Log-Normal Distribution:

$$\begin{aligned}\beta_{k,ir} &= \exp(\beta_k + \sigma_k w_{k,ir}) \\ w_{k,ir} &\sim N(0, 1)\end{aligned}$$

where β_k and σ_k are estimated. Then, $\beta_{k,i} \sim \log N(\beta_k, \sigma_k^2)$

4. Uniform:

$$\begin{aligned}\beta_{k,ir} &= \beta_k + \sigma_k(2 \times u_{k,ir} - 1) \\ u_{k,ir} &\sim U(0, 1)\end{aligned}$$

where β_k and σ_k are estimated.

5. Triangular distribution:

$$\begin{aligned}\beta_{k,ir} &= \beta_k + \sigma_k v_{k,ir} \\ v_{k,ir} &\sim 1(u_{k,ir} < 0.5) (\sqrt{2u_{k,ir}} - 1) + 1(u_{k,ir} \geq 0.5) \left(1 - \sqrt{2(1 - u_{k,ir})}\right)\end{aligned}$$

where β_k and σ_k are estimated.

6. Johnson's S_b :

$$\begin{aligned}\beta_{k,ir} &= \frac{\exp(\beta_k + \sigma_k w_{k,ir})}{1 + \exp(\beta_k + \sigma_k w_{k,ir})} \\ w_{k,ir} &\sim N(0, 1)\end{aligned}$$

where β_k and σ_k are estimated.

In applied work, one issue that causes concern is the choice of the distribution of the random parameters. As explained by [Hensher and Greene \(2003\)](#), distributions are essentially arbitrary approximations to the real behavioral profile. The researcher choose a specific distribution because he has a sense that the “empirical truth” is somewhere in their domain. For example, the log-normal form is often used if the coefficient needs to be specific non-negative sign. On the contrary, the normal and triangular distributions are useful when there is no certainty of the sign of the coefficient. The problem with the normal distribution is that its domain is $(-\infty, \infty)$ which may results in very extreme coefficients. The triangular distribution may solve this problem because it possesses shorter tails.

Rchoice allows to the user to specify two type of random draws by the argument **haltons**: pseudo-random draws (**haltons** = **NULL**) and Halton draws (**haltons** = **NA**) as default. If **haltons** = **NULL**, the seed is set to **set.seed(123)**. The user can change this by the **seed** argument. For the Halton draws, the default is to use the first K_2 primes numbers starting with 3. Within each series, the first 100 draws are discarded, as the first draws tend to be highly correlated across different draws. The user can also change the prime number and the element dropped for each serie. For example, if $K_2 = 2$, and the user wants to use the primes numbers 5 and 31 along with dropping the first 10 draws, he could specify **haltons** = **list("prime" = c(5,31), "drop" = c(10,10))**.

5. Applications Using Rchoice

5.1. Standard Models

In this section, we show the capabilities of **Rchoice** to estimate Poisson, Binary and Ordered

regression model without random parameters.⁴ The main objective of this section is to show how **Rchoice** can interact with other packages in R.

Rchoice is loaded by typing:

```
library("Rchoice")
```

To show how to estimate Poisson regression models using **Rchoice**, we will use data on scientific productivity (Long 1990, 1997). We load the data using

```
data("Articles")
head(Articles,3)
```

	art	fem	mar	kid5	phd	ment
1	0	0	1	0	2.52	7
2	0	1	0	0	2.05	6
3	0	1	0	0	3.75	6

To see more information about the data, one can use:

```
help(Articles)
```

The work by Long (1990) suggests that gender, marital status, number of young children, prestige of the graduate program, and the number of articles written by a scientist's mentor could affect a scientist's level of publication. To see this, we estimate a Poisson regression model and use the **Rchoice** function specifying `link = poisson`:

```
poisson <- Rchoice(art ~ fem + mar + kid5 + phd + ment,
                   data = Articles,
                   family = poisson)
summary(poisson)
```

Model: poisson

Model estimated on: Wed Dec 17 10:39:21 2014

Call:

```
Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
        family = poisson, method = "nr")
```

⁴In R there exist several package to estimate binary, count and ordered models. `glm` function allows to estimate different kind of discrete choice models such as Poisson and binary models. The function `probit` from the package **micEcon** allows to estimate probit model. Moreover, the function `polr` from the package **MASS** allows to estimate ordered models (Venables and Ripley 2002). The advantage of **Rchoice** is that allows more flexibility in the optimization routines which improves the convergence speed. **Rchoice** uses the function `maxLik` in order to maximize the log-likelihood function, which permits to estimate models by the Newton-Raphson (NR), BGFS and Berndt-Hall-Hall-Hausman (BHHH) procedures (see Henningsen and Toomet 2011).

```

The estimation took: 0h:0m:0s

Coefficients:
      Estimate Std. Error t-value Pr(>|t|)
constant  0.3046168  0.1029822  2.9580  0.003097 **
fem       -0.2245942  0.0546138 -4.1124 3.915e-05 ***
mar        0.1552434  0.0613747  2.5294  0.011425 *
kid5      -0.1848827  0.0401272 -4.6074 4.077e-06 ***
phd        0.0128226  0.0263972  0.4858  0.627141
ment       0.0255427  0.0020061 12.7327 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by Newton-Raphson maximisation
Log Likelihood: -1651.1
Number of observations: 915
Number of iterations: 7
Exit of MLE: gradient close to zero

```

The output shows that the log-likelihood function is estimated using NR algorithm in 7 iterations. If the user wants to estimate the model using another algorithm he should type `method = "bfgs"` for the BGFS method or `method = "bhhh"` for BHHH method.⁵

In terms of interpretation, we can say that, being a female scientist decreases the expected number of articles by a factor of 0.8 ($= \exp(-.225)$), holding all other variables constant. Or equivalently, being a female scientist decreases the expected number of articles by 20% ($= 100 [\exp(-.225) - 1]$), holding all other variables constant. Prestige of PhD department is not important for productivity.

Another capability of **Rchoice** is its interaction with other packages in R. For example, we can compute the robust standard error by using the package **sandwich**:

```

library(sandwich)
library(lmtest)
coeftest(poisson, vcov = sandwich)

t test of coefficients:

      Estimate Std. Error t value  Pr(>|t|)

```

⁵BHHH is generally faster than the other procedures, but it can blow up if the variables have very different scale. The larger the ratio between the largest standard deviation and the smallest standard deviation, the more problems the user will have with the estimation procedure. Given this, we encourage the users check the variables and re-scale or recode them if necessary. **Rchoice** uses the numerical hessian if `method = 'nr'` and the model is estimated with random parameters, thus it can be very slow compare to the other methods.

```

constant  0.3046168  0.1465197  2.0790  0.0378958  *
fem       -0.2245942  0.0716622 -3.1341  0.0017793  **
mar        0.1552434  0.0819292  1.8948  0.0584297  .
kid5      -0.1848827  0.0559633 -3.3036  0.0009917  ***
phd        0.0128226  0.0419642  0.3056  0.7600096
ment       0.0255427  0.0038178  6.6905  3.884e-11  ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

To get the same robust standard errors as STATA ([StataCorp 2011](#)), we need to make a small sample correction:

```

vcov.stata <- vcovHC(poisson, type = "HC0") * nObs(poisson)/(nObs(poisson)-1)
coeftest(poisson, vcov = vcov.stata)

t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
constant    0.3046168  0.1465999  2.0779  0.038001 *
fem         -0.2245942  0.0717014 -3.1324  0.001790 **
mar          0.1552434  0.0819740  1.8938  0.058567 .
kid5        -0.1848827  0.0559939 -3.3018  0.000998 ***
phd          0.0128226  0.0419871  0.3054  0.760137
ment         0.0255427  0.0038198  6.6868  3.977e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

where the correction is $n/(n - 1)$.

Rchoice also interacts with `linearHypothesis` and `deltaMethod` functions from **car** ([Fox, Bates, Firth, Friendly, Gorjanc, Graves, Heiberger, Monette, Nilsson, Ogle *et al.* 2009](#)) and the `lrtest` and `waldtest` functions from **lmtest** package ([Zeileis and Hothorn 2002](#)). For example, we can test $H_0 : \text{phd}/\text{ment} = 0$ by:

```

library(car)
deltaMethod(poisson, "phd/ment")

             Estimate      SE
phd/ment  0.5020048  1.043031

```

The main argument to estimate other models is `family`. For probit models, the user should specify `family = binomial("probit")`, and for Logit `family = binomial("logit")`. In the following example, we use the **Workmroz** data base to estimate a binary Probit model, where the dependent variable `lfp` equals 1 if wife is in the paid labor force, and 0 otherwise.

```

data("Workmroz")
probit <- Rchoice(lfp ~ k5 + k618 + age + wc + hc + lwg + linc,
                  data = Workmroz,
                  family = binomial('probit'))
summary(probit)

Model: binomial
Model estimated on: Wed Dec 17 10:39:21 2014

Call:
Rchoice(formula = lfp ~ k5 + k618 + age + wc + hc + lwg + linc,
        data = Workmroz, family = binomial("probit"), method = "nr")

Frequencies of categories:
y
    0      1
0.43218 0.56782
The estimation took: 0h:0m:0s

Coefficients:
              Estimate Std. Error t-value Pr(>|t|)
constant    2.7819830   0.4418758   6.2958 3.057e-10 ***
k5          -0.8806883   0.1134365  -7.7637 8.216e-15 ***
k618        -0.0386560   0.0404545  -0.9555 0.3393037
age         -0.0377010   0.0076118  -4.9529 7.310e-07 ***
wc           0.4811500   0.1352711   3.5569 0.0003752 ***
hc           0.0774402   0.1247331   0.6208 0.5347000
lwg          0.3716453   0.0876052   4.2423 2.213e-05 ***
linc        -0.4514938   0.1007483  -4.4814 7.415e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by Newton-Raphson maximisation
Log Likelihood: -451.91
Number of observations: 752
Number of iterations: 4
Exit of MLE: gradient close to zero

```

Ordered Probit and Logit models are estimated in the same way. In this case we use **Health** database and create the logarithm of household income. The dependent variable, **newhsat**, is a categorical variable that indicates the self reported health assessment of individuals recorded with values 0,1,...,4.

```

data("Health")
Health$linc <- log(Health$hhinc)
ologit <- Rchoice(newhsat ~ age + educ + married + hhkids + linc,
                  data = Health[1:2000, ],
                  family = ordinal('logit'))
summary(ologit)

Model: ordinal
Model estimated on: Wed Dec 17 10:39:21 2014

Call:
Rchoice(formula = newhsat ~ age + educ + married + hhkids + linc,
        data = Health[1:2000, ], family = ordinal("logit"), method = "bfgs")

Frequencies of categories:
y
  0    1    2    3    4
0.0600 0.2675 0.4545 0.1010 0.1170
The estimation took: 0h:0m:0s

Coefficients:
              Estimate Std. Error t-value Pr(>|t|)
kappa.1      2.0942090   0.0904874  23.1436 < 2.2e-16 ***
kappa.2      4.1980669   0.1059515  39.6225 < 2.2e-16 ***
kappa.3      4.9612273   0.1157057  42.8780 < 2.2e-16 ***
constant     2.1701941   0.7784738   2.7878 0.0053075 **
age          -0.0306368   0.0045175  -6.7819 1.186e-11 ***
educ          0.0658477   0.0184559   3.5678 0.0003599 ***
married     -0.3363123   0.1102781  -3.0497 0.0022909 **
hhkids       0.2204229   0.0994854   2.2156 0.0267168 *
linc         0.1764959   0.0987995   1.7864 0.0740337 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximisation
Log Likelihood: -2662.6
Number of observations: 2000
Number of iterations: 67
Exit of MLE: successful convergence

```

5.2. Random Parameters Models with Cross Sectional Data

The main advantage of **Rchoice** over other packages is that it allows estimating models with random parameters. In this section, we show how to estimate those kinds of models for Bi-

nary, Ordered and Poisson models using cross-sectional data.

First, we estimate a Poisson regression model with random parameters. In this case, we will assume that the effect of `kid5`, `phd` and `ment` are not fixed, but rather heterogeneous across the population. Specifically, we will assume that the coefficients for those variables are independent normally distributed, that is, we will not allow correlation among them:

$$\begin{aligned}\beta_{\text{kid5},i} &= \beta_{\text{kid5}} + \sigma_{\text{kid5}}\omega_{\text{kid5},ir} \\ \beta_{\text{phd},i} &= \beta_{\text{phd}} + \sigma_{\text{phd}}\omega_{\text{phd},ir} \\ \beta_{\text{ment},i} &= \beta_{\text{ment}} + \sigma_{\text{ment}}\omega_{\text{ment},ir}\end{aligned}$$

where $\omega_{k,ir} \sim N(0, 1)$. Then, in order to estimate this model, we can write:

```
poisson.ran <- Rchoice(art ~ fem + mar + kid5 + phd + ment,
  data = Articles,
  family = poisson,
  ranp = c(kid5 = "n", phd = "n", ment = "n"))
```

It is important to discuss the arguments for the `Rchoice` function. First, the argument `ranp` indicates which variables are random in the formula and their distributions. In this case, we have specified that all of them are normal distributed using "n". The rest of the distributions are:

- Normal = "n",
- Log-Normal = "ln",
- Truncated Normal = "cn",
- Uniform = "u",
- Triangular = "t",
- Johnson's S_b = "sb"

The number of draws are not specified. Therefore, `Rchoice` will set `R = 40` as default. The user can change this by changing the `R` argument. The type of draws are Halton draws default, but if the user wants pseudo-random draws he can specify `haltons = NULL`. Finally, it is always important checking the exit of the estimation. As explained before, the default maximization algorithm for SML is BGFS.⁶

```
summary(poisson.ran)
```

⁶For more information about arguments for the optimization type `help(maxLik)`.

```

Model: poisson
Model estimated on: Wed Dec 17 10:39:48 2014

Call:
Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
        family = poisson, ranp = c(kid5 = "n", phd = "n", ment = "n"),
        method = "bfgs", iterlim = 2000)

The estimation took: 0h:0m:27s

Coefficients:
              Estimate Std. Error t-value Pr(>|t|)
constant    0.2255825   0.1324995   1.7025 0.088659 .
fem         -0.2184985   0.0705576  -3.0967 0.001957 **
mar          0.1564309   0.0791214   1.9771 0.048030 *
mean.kid5   -0.1977748   0.0634716  -3.1160 0.001833 **
mean.phd    -0.0299420   0.0372167  -0.8045 0.421091
mean.ment    0.0311104   0.0038136   8.1578 4.441e-16 ***
sd.kid5      0.2853104   0.0891038   3.2020 0.001365 **
sd.phd       0.1654054   0.0165846   9.9734 < 2.2e-16 ***
sd.ment      0.0158755   0.0035353   4.4905 7.105e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximisation
Log Likelihood: -1574.2
Number of observations: 915
Number of iterations: 81
Exit of MLE: successful convergence
Simulation based on 40 Halton draws

```

It is important checking the exist of the estimation. In our example, the output informs us that the convergence was achieved successfully. The results also show that the standard deviations of the coefficients are highly significant, indicating that parameters do indeed vary in the population. Since the parameters are normally distributed, we can also say that :

```

pnorm(coef(poisson.ran)["mean.kid5"]/coef(poisson.ran)["sd.kid5"])

mean.kid5
0.2440946

```

a 24% of the individuals have a positive coefficient for `kid5`. In other words, for about 76% of PhD students, having children less than 6 years old reduces their productivity. Note also that the mean coefficient for `phd` is 0 (not significant). This is due to the fact that the unobserved heterogeneity among scientists in the sample cancel out positive and negative effects. These observations are not possible with a Poisson regression with fixed parameters.

Suppose that now we want to test if $H_0 = \sigma_{\text{kid5}} = \sigma_{\text{phd}} = \sigma_{\text{ment}} = 0$. This can be done by using the function `waldtest` or `lrtest` from package **lmtest**:

```
waldtest(poisson.ran, poisson)

Wald test

Model 1: art ~ fem + mar + kid5 + phd + ment
Model 2: art ~ fem + mar + kid5 + phd + ment
  Res.Df Df   Chisq Pr(>Chisq)
1      906
2      909 -3 280.14  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

lrtest(poisson.ran, poisson)

Likelihood ratio test

Model 1: art ~ fem + mar + kid5 + phd + ment
Model 2: art ~ fem + mar + kid5 + phd + ment
  #Df LogLik Df   Chisq Pr(>Chisq)
1    9 -1574.2
2    6 -1651.1 -3 153.78  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both tests reject the null hypothesis. We can also specify different distribution of the parameters by using the S3 method `update`:

```
poisson.ran2 <- update(poisson.ran,
                      ranp = c(kid5 = "u", phd = "t", ment = "cn"),
                      R = 10)
```

Both models `poisson.ran` and `poisson.ran2` can be compared using `mtable` from **memisc**:

```
library(memisc)

mtable("model 1"= poisson.ran, "model 2" = poisson.ran2,
       summary.stats = c("N", "Log-likelihood", "BIC", "AIC"))
```

The previous model specifies the coefficients to be independently distributed while one would expect correlation. For example, the effect of the prestige of PhD department could be positive correlated with the number of publications by mentor. Now, we estimate the model `poisson.ran`, but assuming that the random parameters are correlated. That is, we specified $\beta_i \sim N(\beta, \Sigma)$ for general Σ :

```

poissonc.ran <- Rchoice(art ~ fem + mar + kid5 + phd + ment,
                        data = Articles,
                        ranp = c(kid5 = "n", phd = "n", ment = "n"),
                        family = poisson,
                        correlation = TRUE,
                        R = 10)

summary(poissonc.ran)

Model: poisson
Model estimated on: Wed Dec 17 10:40:35 2014

Call:
Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
        family = poisson, ranp = c(kid5 = "n", phd = "n", ment = "n"),
        R = 10, correlation = TRUE, method = "bfgs", iterlim = 2000)

The estimation took: 0h:0m:46s

Coefficients:
              Estimate Std. Error t-value Pr(>|t|)
constant      0.2490701  0.1272589   1.9572  0.050325 .
fem           -0.2207883  0.0691399  -3.1934  0.001406 **
mar            0.1410103  0.0793115   1.7779  0.075415 .
mean.kid5     -0.1996965  0.0566969  -3.5222  0.000428 ***
mean.phd      -0.0356327  0.0358630  -0.9936  0.320429
mean.ment      0.0346114  0.0036420   9.5034 < 2.2e-16 ***
sd.kid5.kid5  -0.1815531  0.0722360  -2.5133  0.011960 *
sd.kid5.phd   -0.1803741  0.0235439  -7.6612  1.843e-14 ***
sd.kid5.ment   0.0297596  0.0045700   6.5119  7.421e-11 ***
sd.phd.phd    -0.0638155  0.0287141  -2.2224  0.026253 *
sd.phd.ment   -0.0163012  0.0055910  -2.9156  0.003550 **
sd.ment.ment   0.0028148  0.0064630   0.4355  0.663179
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximisation
Log Likelihood: -1579.4
Number of observations: 915
Number of iterations: 189
Exit of MLE: successful convergence
Simulation based on 10 Halton draws

```

The output prints the mean of the random parameters along with the lower-triangular Choleski factor \mathbf{L} . We can extract the $\mathbf{\Sigma} = \mathbf{LL}'$ matrix of variance-covariance matrix and the correlation matrix of the random parameters using `cov.Rchoice` and `cor.Rchoice`:

```
cov.Rchoice(poissonc.ran)
```

	kid5	phd	ment
kid5	0.032961545	0.032747485	-0.005402947
phd	0.032747485	0.036607232	-0.004327588
ment	-0.005402947	-0.004327588	0.001159286

```
cor.Rchoice(poissonc.ran)
```

	kid5	phd	ment
kid5	1.0000000	0.9427374	-0.8740405
phd	0.9427374	1.0000000	-0.6643044
ment	-0.8740405	-0.6643044	1.0000000

Among other things, the output shows that the three parameters are positively correlated. Specifically, we can see that the correlation between `phd` and `ment` is around -0.4. We can also test if the variances of the random parameters are significant using Delta Method. For this, we can use the `se.cov.Rchoice` function, which is a wrapper of `deltamethod` function from `msm` package. For example:

```
se.cov.Rchoice(poissonc.ran)
```

Elements of the variance-covariance matrix

	Estimate	Std. Error	t-value	Pr(> t)
v.kid5.kid5	0.03296155	0.02622936	1.2567	0.208875
v.kid5.phd	0.03274749	0.01248677	2.6226	0.008727 **
v.kid5.ment	-0.00540295	0.00242844	-2.2249	0.026091 *
v.phd.phd	0.03660723	0.00817813	4.4762	7.597e-06 ***
v.phd.ment	-0.00432759	0.00141756	-3.0528	0.002267 **
v.ment.ment	0.00115929	0.00028675	4.0429	5.281e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

To get the standar errors of the standard deviations for the random parameters, we might use:

```
se.cov.Rchoice(poissonc.ran, sd = TRUE)
```

Standard deviations of the random parameters

	Estimate	Std. Error	t-value	Pr(> t)
kid5	0.1815531	0.0722360	2.5133	0.01196 *

```
phd    0.1913302  0.0213718  8.9525 < 2.2e-16 ***
ment  0.0340483  0.0042109  8.0857 6.661e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.3. Random Parameters Models with Panel Data

This new version of this package also handles panel data by estimating Random Effect Models (RE). The RE model specifies:

$$\epsilon_{it} = v_{it} + u_i$$

where v_{it} and u_i are independent variables with

$$\begin{aligned} E(v_{it}|\mathbf{X}) &= 0; \text{Cov}(v_{it}, v_{is}|\mathbf{X}) = \text{Var}(v_{it}|\mathbf{X}) = 1, & \text{if } i = j \text{ and } t = s; 0 \text{ otherwise,} \\ E(u_i|\mathbf{X}) &= 0; \text{Cov}(u_i, u_j|\mathbf{X}) = \text{Var}(u_i|\mathbf{X}) = \sigma_u^2, & \text{if } i = j; 0 \text{ otherwise,} \\ \text{Cov}(v_{it}, u_j) &= 0 & \text{for all } i, t, j, \end{aligned}$$

and \mathbf{X} indicates all the exogenous data in the sample, \mathbf{x}_{it} for all i and t . This model can be estimated using the package **pglm** (Croissant 2013), which uses **quadrature** to approximate the integration in the probability.

Rchoice also allows estimating RE effect models along with random parameters. Note that assuming that the constant is random is equivalent to a RE model. Therefore, the user might estimate a simple RE model by typing `ranp = (constant = 'n')`.

In this example we estimate a Probit Model with RE and random parameters using **Unions** database from the **pglm** package.

```
data('Unions', package = 'pglm')
Unions$lwage <- log(Unions$wage)
```

The model is estimated using the following syntax:

```
union.ran <- Rchoice(union ~ age + exper + rural + lwage,
  data = Unions[1:2000, ],
  family = binomial('probit'),
  ranp = c(constant = "n", lwage = "t"),
  R = 10,
  panel = TRUE,
  index = "id",
  print.init = TRUE)
```

Starting Values:

age	exper	ruralyes	mean.constant	mean.lwage
-0.009678358	-0.017142802	0.292871899	-1.380275299	0.420245667
sd.constant	sd.lwage			
0.100000000	0.100000000			

In this case, we assume that `lwage` is distributed as triangular, while the constant is assumed to be normal distributed. This is the same as assuming that $u_i \sim N(0, \sigma_u^2)$.

There are two main arguments for the panel estimation. The argument `panel = TRUE` indicates that the data is a panel. This implies that the user should indicate the variable that corresponds to the ID of the individuals in the `index` argument.

Finally, the argument `print.init = TRUE` indicates that the initial values used by `Rchoice` will be displayed.

```
summary(union.ran)
```

Model: binomial

Model estimated on: Wed Dec 17 10:41:19 2014

Call:

```
Rchoice(formula = union ~ age + exper + rural + lwage, data = Unions[1:2000,
], family = binomial("probit"), ranp = c(constant = "n",
lwage = "t"), R = 10, panel = TRUE, index = "id", print.init = TRUE,
method = "bfgs", iterlim = 2000)
```

Frequencies of categories:

y

	0	1
	0.7605	0.2395

The estimation took: 0h:0m:43s

Coefficients:

	Estimate	Std. Error	t-value	Pr(> t)
age	0.054069	0.328016	0.1648	0.8691
exper	-0.091741	0.020984	-4.3720	1.231e-05 ***
ruralyes	0.227349	0.175374	1.2964	0.1948
mean.constant	-1.300731	0.244972	-5.3097	1.098e-07 ***
mean.lwage	0.010886	0.154192	0.0706	0.9437
sd.constant	1.199181	0.161099	7.4438	9.792e-14 ***
sd.lwage	1.042017	0.107448	9.6978	< 2.2e-16 ***

```

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximisation
Log Likelihood: -738.64
Number of observations: 2000
Number of iterations: 77
Exit of MLE: successful convergence
Simulation based on 10 Halton draws

```

The results indicate that $\sigma_u = 1.1991808$ and is significant. On the other hand, the finding of a significant standard deviation yet insignificant mean for `lwage` attests to the existence of substantial heterogeneity; positive and negative coefficient in the sample compensate for each other, such that the coefficient on the mean is not significant.

As in the previous cases, an Ordered Probit Model with RE and random parameters can be estimated in the same way, but changing the distribution with the `family` argument.

```

oprobit.ran <- Rchoice(newhsat ~ age + educ + married + hhkids + linc,
  data = Health[1:2000, ],
  family = ordinal('probit'),
  ranp = c(constant = "n", hhkids = "n", linc = "n"),
  panel = TRUE,
  index = "id",
  R = 100,
  print.init = TRUE)
summary(oprobit.ran)

```

5.4. Hierarchical Poisson Random Parameter Model

In this section we show how to estimate a Hierarchical Poisson Random Parameter Model. In this case, we assume that there exist not only unobserved heterogeneity in the coefficients for `phd` and `ment`, but also observed heterogeneity in the mean of the coefficients. Specifically, we assume that:

$$\begin{aligned}
 \beta_{\text{kid5},ir} &= \beta_{\text{kid5}} + \sigma_{\text{kid5}}\omega_{\text{kid5},ir} \\
 \beta_{\text{phd},ir} &= \beta_{\text{phd}} + \pi_{\text{phd},\text{fem}}\text{fem} + \sigma_{\text{phd}}\omega_{\text{phd},ir} \\
 \beta_{\text{ment},ir} &= \beta_{\text{ment}} + \pi_{\text{ment},\text{fem}}\text{fem} + \pi_{\text{ment},\text{phd}}\text{phd} + \sigma_{\text{ment}}\omega_{\text{ment},ir}
 \end{aligned}$$

The formulation above implies that, for example, `ment`'s coefficient (or marginal effect on latent productivity) varies by gender and `phd`.

```

poissonH.ran <- Rchoice(art ~ fem + mar + kid5 + phd + ment | fem + phd,
  data = Articles,
  ranp = c(kid5 = "n", phd = "n", ment = "n"),

```

```
mvar = list(phd = c("fem"), ment = c("fem", "phd")),
family = poisson,
R = 10)
```

Rchoice manages the variables in the hierarchical model by the **formula** object. Note that the second part of the **formula** is reserved for all the variables that enter in the mean of the random parameters. The argument **mvar** is a list that indicates how all this variables enter in each random parameter. For example **phd = c("fem")** indicates that the mean of **phd** coefficient varies according to **fem**.

```
summary(poissonH.ran)
```

We can test if the interaction variables are jointly significant by using **lrtest**:

```
lrtest(poissonH.ran, poisson.ran)
```

5.5. Plotting Conditional Means

It is important to note that the estimates of the model parameters provide the unconditional estimates of the parameter vector, but we can form a person specific conditional estimator (see [Train 2009](#); [Greene 2012](#)). The estimator of the conditional mean of the distribution of the random parameters, conditioned on the person specific data, is:

$$\hat{\beta}_i = \hat{E}(\beta_i | \text{data}_i) = \sum_{r=1}^R \hat{Q}_{ir} \hat{\beta}_{ir}$$

where:

$$\hat{\beta}_{ir} = \hat{\beta} + \hat{\Pi} s_i + \hat{\mathbf{L}} \omega_{ir}$$

Note that these are not actual estimates of β_i , but are estimates of the conditional mean of the distribution of the random parameters ([Greene, Harris, and Spencer 2014](#)). We can also estimate the standard deviation of this distribution by estimating:

$$\hat{E}(\beta_i^2 | \text{data}_i) = \sum_{r=1}^R \hat{Q}_{ir} \hat{\beta}_{ir}^2,$$

and then computing the square root of the estimated variance,

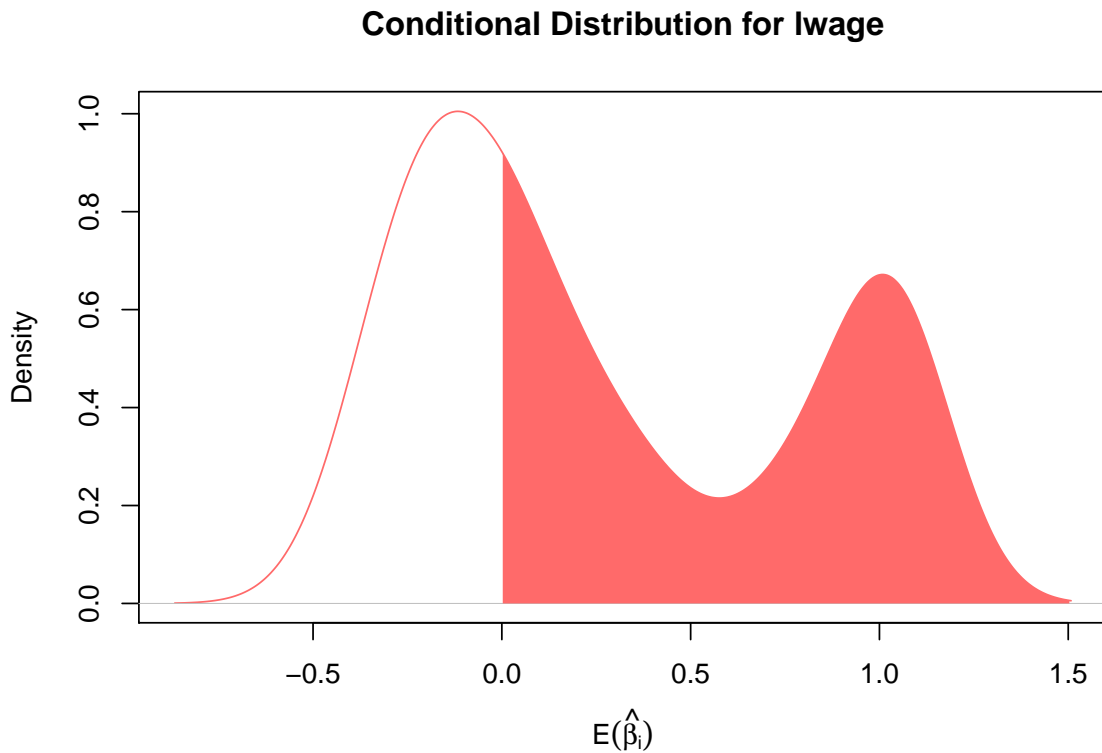
$$\sqrt{\hat{E}(\beta_i^2 | \text{data}_i) - \hat{E}(\beta_i | \text{data}_i)^2}$$

With the estimates of the conditional mean and conditional variance, we can then compute the limits of an interval that resembles a confidence interval as the mean plus and minus two estimated standard deviation. This will construct an interval that contains at least 95 percent of the conditional distribution of β_i ([Greene 2012](#)).

Rchoice allows to plot the histogram and kernel density of conditional means of random parameters using the function `plot`. For instance, the kernel of the conditional mean of $\beta_{\text{lwage},i}$ for `union.ran` model can be obtained by typing:

```
plot(union.ran, par = "lwage", type = "density")
```

Figure 1: Kernel Density of the Individual's Conditional Mean

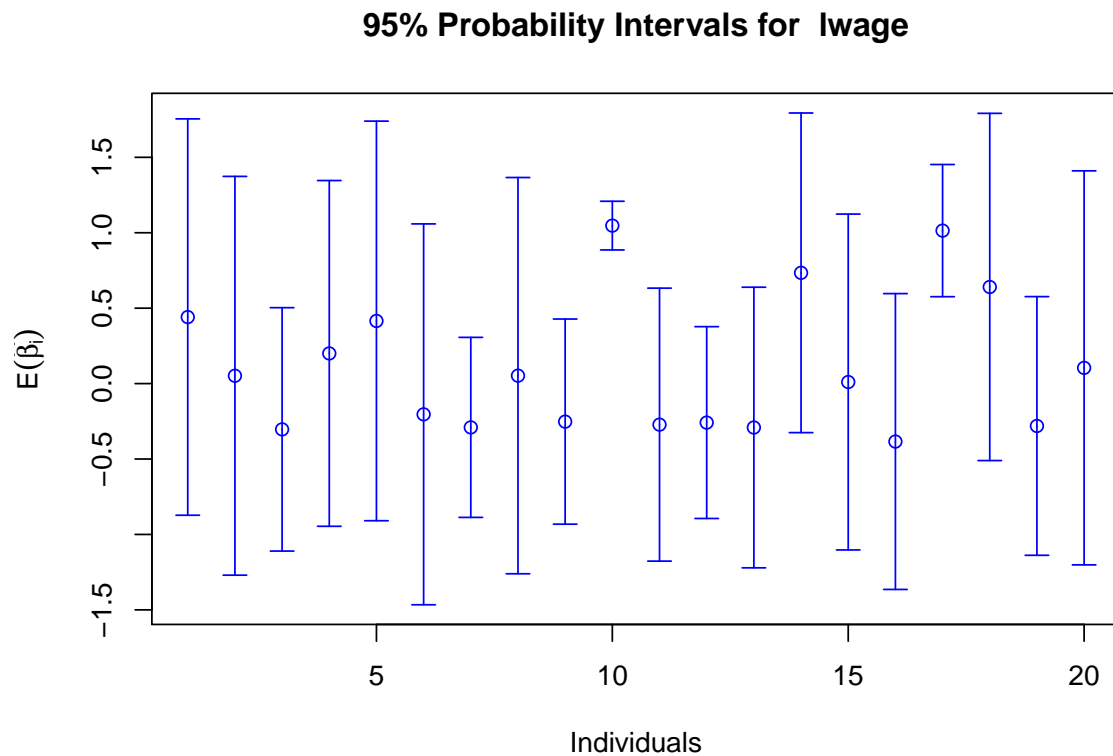


As [Greene \(2012\)](#) points out, even if the analysis departs from normal marginal distributions β_i , the sample distribution of the n estimated conditional means is not necessarily normal. Therefore, the kernel estimator based on the n estimators can have a variety of shapes.

We may also plot the individual confident interval for the conditional means for the first 20 individuals:

```
plot(union.ran, par = "lwage", ind = TRUE, id = 1:20, col = "blue")
```


Figure 2: Individual Confident Interval for the Conditional Means



The method `plot` for `Rchoice` class is a wrapper of `effect.Rchoice` function. This function is a helper function to obtain the conditional estimate of the individual random parameters or the compensating variations.

```
bi.wage <- effect.Rchoice(union.ran, par = "lwage", effect = "ce")
```

The argument `effect` is a string indicating what type of effect should be computed. In this example, we are requiring the the conditional expectation of the individual coefficients "ce". `effect.Rchoice` returns two list. The first one with the estimated conditional means for all the individuals, and the second one with the estimated standard errors of the conditional means.

```
summary(bi.wage$mean)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.4116	-0.1460	0.0718	0.2824	0.8643	1.0530

```
summary(bi.wage$sd.est)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000017	0.3440000	0.5372000	0.4526000	0.6290000	0.8440000

One might also get the individuals' **compensating variations** using both `plot` and `effect.Rchoice`. Compensating variation is the variation in two regressors such that the latent variables does not change. Let x_{il} denote the l th element in \mathbf{x}_i and β_l the corresponding parameter, and let m index the m th elements in both vectors \mathbf{x}_i and $\boldsymbol{\beta}$, respectively. Now consider a change in x_{il} and x_{im} at the same time, such that $U_i^* = 0$. This requires

$$0 = \beta_l \Delta x_{il} + \beta_{im} \Delta x_{im} \implies \frac{\Delta x_{il}}{\Delta x_{im}} = -\frac{\beta_{im}}{\beta_l}$$

where β_{im} is a random coefficient. This ratio (without the minus sign) is computed or plotted if the argument `effect = "cv"` in any of the two functions. The argument `par` is the variable whose coefficient goes in the numerator (β_{im}), and the argument `wrt` is a string indicated which coefficient goes in the denominator (β_l). Note that since β_{im} is random, the ratio of the coefficient is random and its distribution follows from the joint distribution of the coefficients.

6. Conclusions and future development

The **Rchoice** package contains most of the newly developed models in binary, count and ordered models with random parameters. **Rchoice** handles cross-section data with observed and unobserved heterogeneity. Allowing parameter values to vary across the population according to some pre-specified distribution overcomes the problem of having a fixed-representative coefficient for all individuals. The distribution supported by **Rchoice** are normal, log-normal, uniform, truncated normal and triangular distribution. It also allows to the user choose between Halton draws and pseudo-random numbers and correlated parameters.

The **Rchoice** package intends to make available those estimation methods to the general public and practitioners in a friendly and flexible way. In future versions, we expect to add functions that allows estimating latent class models. We also hope to include functions to compute marginal effects.

References

- Bhat CR (2001). "Quasi-random Maximum Simulated Likelihood Estimation of the Mixed Multinomial Logit Model." *Transportation Research Part B: Methodological*, **35**(7), 677–693.
- Chamberlain G (1980). "Analysis of Covariance with Qualitative Data." *The Review of Economic Studies*, pp. 225–238.
- Croissant Y (2013). *pglm: Panel Generalized Linear Model*. R package version 0.1-2, URL <http://CRAN.R-project.org/package=pglm>.
- Croissant Y, *et al.* (2012). "Estimation of Multinomial Logit Models in R: The **mlogit** Packages." *R package version 0.2-2*. URL <http://cran.r-project.org/web/packages/mlogit/vignettes/mlogit.pdf>.

- Fox J, Bates D, Firth D, Friendly M, Gorjanc G, Graves S, Heiberger R, Monette G, Nilsson H, Ogle D, *et al.* (2009). “**CAR**: Companion to Applied Regression, R Package Version 1.2-16.” *Online at <http://cran.r-project.org/web/packages/car/index.html> (accessed on August 2012).*
- Gourieroux C, Monfort A (1997). *Simulation-based Econometric Methods*. Oxford University Press.
- Gourieroux C, Monfort A, Trognon A (1984). “Pseudo Maximum Likelihood Methods: Applications to Poisson Models.” *Econometrica: Journal of the Econometric Society*, pp. 701–720.
- Greene W (2007). *Functional Form and Heterogeneity in Models for Count Data*. Now Publishers Inc.
- Greene W, Harris MN, Spencer C (2014). “Estimating the Standard Errors of Individual-Specific Parameters in Random Parameters Models.” *Technical report*.
- Greene WH (2002). *LIMDEP: Version 8.0: Econometric Modeling Guide*. Econometric Software.
- Greene WH (2012). *Econometric Analysis*. 7 edition. Prentice Hall. ISBN 0131395386.
- Greene WH, Hensher DA (2010a). *Modeling Ordered Choices: A Primer*. Cambridge University Press.
- Greene WH, Hensher DA (2010b). “Ordered Choices and Heterogeneity in Attribute Processing.” *Journal of Transport Economics and Policy (JTEP)*, **44**(3), 331–364.
- Hajivassiliou VA, Ruud PA (1986). “Classical Estimation Methods for LDV Models Using Simulation.” In RF Engle, D McFadden (eds.), *Handbook of Econometrics*, volume 4 of *Handbook of Econometrics*, chapter 40, pp. 2383–2441. Elsevier.
- Henningsen A, Toomet O (2011). “**maxLik**: A Package for Maximum Likelihood Estimation in R.” *Computational Statistics*, **26**(3), 443–458.
- Hensher DA, Greene WH (2003). “The Mixed Logit Model: The State of Practice.” *Transportation*, **30**(2), 133–176.
- Lee LF (1992). “On Efficiency of Methods of Simulated Moments and Maximum Simulated Likelihood Estimation of Discrete Response Models.” *Econometric Theory*, **8**(04), 518–552.
- Long JS (1990). “The Origins of Sex Differences in Science.” *Social Forces*, **68**(4), 1297–1316.
- Long JS (1997). *Regression Models for Categorical and Limited Dependent Variables*, volume 7. Sage.
- Mundlak Y (1978). “On the Pooling of Time Series and Cross Section Data.” *Econometrica: journal of the Econometric Society*, pp. 69–85.
- Ng ES, Carpenter JR, Goldstein H, Rasbash J (2006). “Estimation in Generalised Linear Mixed Models with Binary Outcomes by Simulated Maximum Likelihood.” *Statistical Modelling*, **6**(1), 23–42.

- Sándor Z, Train K (2004). “Quasi-random Simulation of Discrete Choice Models.” *Transportation Research Part B: Methodological*, **38**(4), 313–327.
- StataCorp (2011). “Stata Statistical Software: Release 12.” *College Station, TX: StataCorp LP*.
- Train K (2000). “Halton Sequences for Mixed Logit.” *Department of Economics, UCB*.
- Train K (2009). *Discrete Choice Methods with Simulation*. 2 edition. Cambridge university press.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer, New York. ISBN 0-387-95457-0, URL <http://www.stats.ox.ac.uk/pub/MASS4>.
- Zeileis A, Hothorn T (2002). “Diagnostic Checking in Regression Relationships.” *R News*, **2**(3), 7–10. URL <http://CRAN.R-project.org/doc/Rnews/>.

Affiliation:

Mauricio Sarrias
325 W. Sibley Hall
Department of City and Regional Planning
Cornell University
E-mail: msarrias86@gmail.com
URL: <http://www.msarrias.weebly.com>