

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

Mauricio Sarrias
Cornell University

Abstract

Rchoice is a package in R for estimating models with individual heterogeneity for both cross-sectional and panel (longitudinal) data. In particular, the package allows binary, ordinal and count response, as well as continuous and discrete covariates. Individual heterogeneity is modeled by allowing the parameter associated with each observed variable (e.g., its coefficient) to vary randomly across individuals according to some pre-specified distribution. Simulated maximum likelihood method is implemented for the estimation of the moments of the distributions. In addition, functions for plotting the conditional individual-specific coefficients and their confidence interval are provided. This article is a general description of **Rchoice** and all functionalities are illustrated using real databases.

Keywords: discrete choice models, random parameters, simulated maximum likelihood, R, individual-specific estimates.

1. Introduction

Researchers from many fields have to deal with models where the dependent variable is binary, ordinal or count data. On the one hand, binary variables have two categories and are often used to indicate that an event has occurred or some action has taken place. For example, is a person suffering from diabetes? Is the person a smoker? Is the individual in the labor force? Did the citizen vote in the last election? On the other hand, an ordinal variable has multiple outcomes with an intrinsic ordering. For instance, surveys sometimes ask respondents to indicate their level of subjective well-being using a scale of the type *strongly agree*, *agree*, *disagree*, and *strongly disagree*. Count variables indicate the number of times that some event has occurred. How many articles has a scientist published? How often did a person visit the doctor in the last year? For these types of dependent variables, the traditional binary Logit/Probit, ordered Probit/Logit model, and Poisson model for count data have been applied to measure the impact of covariates on categorical outcomes. For a general overview of these models see for example [Long \(1997\)](#) and [Winkelmann and Boes \(2006\)](#).

One important modeling shortcoming of these methods is the inherent assumption of a fixed and unique coefficient for all individuals in the sample, which might not be realistic given that individuals are intrinsically heterogeneous. As an example, consider the effect of the number of young children on scientists' productivity measured as the number of published articles. Since having more kids may imply fewer hours available to work, we would expect, on aver-

age, a negative correlation between the number of children and the level of publications. This negative relationship is global, but it neglects the fact that for some scientists having more kids might imply having a more organized life-style and hence increase their productivity (see for example [Krapf, Ursprung, and Zimmermann 2014](#)). Thus, the negative coefficient might hide significant individual heterogeneity resulting in misleading inferences for a subgroup of individuals with a positive coefficient. Similarly, we might assume that the effect is negative for all scientists, but the magnitude of the detrimental effect might vary across the sample. In this case, there exists individual heterogeneity but only in the negative domain of the coefficient. Finally, we might also find that the coefficient is zero (not significant). In such case, we would conclude that having young children is not important for productivity. Nonetheless, this may be due to the fact that heterogeneity among scientists in the sample cancels out positive and negative effects. Under these circumstances a model that allows for individual heterogeneity may be more appropriate.

Thanks to the emergence of more powerful computers and the development of simulation-based models, much of the recent progress in regression models with limited dependent variables has focused on more realistic behavioral models that allow individual heterogeneity in the parameters. The modeling strategy to accommodate such heterogeneity is assuming that coefficients vary randomly across individuals according to some continuous distribution, denoted by $g(\boldsymbol{\theta})$. Since the distribution is unknown and must be specified a priori by the researcher, this type of individual heterogeneity is usually labeled as ‘unobserved heterogeneity’ in the literature. All the information of the unobserved heterogeneity is captured by the parameters of the distribution $\boldsymbol{\theta}$, which usually represent the mean and variance of the coefficient. The goal is to estimate those parameters in order to get a profile of individual heterogeneity.

Although the random parameter approach has been widely applied to the Multinomial Logit model ([Train 2009](#); [Hensher and Greene 2003](#)), there are just a few applications for ordinal ([Falco, Maloney, Rijkers, and Sarrias 2015](#); [Greene and Hensher 2010b](#)) and count models ([Gourieroux, Monfort, and Trognon 1984](#); [Greene 2007](#); [Anastasopoulos and Mannering 2009](#)). The scarcity of applications to models other than the Multinomial Logit appears to be driven by the lack of statistical software that enables to estimate random parameter models for binary, ordinal, and count response. To my knowledge, only the commercial programs LIMDEP ([Greene 2015a](#)) and NLOGIT ([Greene 2015b](#)) are able to estimate those types of models in a concise and flexible manner. The **Rchoice** ([Sarrias 2015](#)) package for R ([R Core Team 2015](#)) is intended to make these estimation methods available to the general public and practitioners in a friendly and flexible way.

The aim of this paper is to present the functionalities of **Rchoice** for estimating ordered, count and binary choice models with random parameters. Its current version 0.3 allows estimating cross-sectional and panel (longitudinal) data models and includes also new functionalities to obtain the standard errors of the variance-covariance matrix of the random parameters. All models in **Rchoice** are estimated using simulated maximum likelihood (SML), which is very flexible for estimating models with a large number of random parameters ([Train 2009](#)). An additional characteristic of **Rchoice** is the ability to retrieve the individual conditional estimates of either the random parameters or compensating variations.

Rchoice is also intended to complement other related packages in R. For example, there exist several packages to estimate binary, count and ordered models with fixed parameters. The `glm` function (R Core Team 2015) allows to estimate different kind of discrete choice models such as Poisson and binary models. The function `probit` from the **micEcon** (Henningsson 2014) package allows to estimate Probit model. Moreover, the function `polr` from the package **MASS** (Venables and Ripley 2002) allows to estimate ordered Probit and Logit models. The advantage of **Rchoice** is that allows more flexibility in the optimization routines, which might improve the convergence speed. Furthermore, **Rchoice** offers an alternative approach to fitting random effects models in the context of panel data to those already programmed in **pglm** (Croissant 2013) and **lme4** (Doran, Bates, Bliese, Dowling *et al.* 2007). Regarding random parameter models, **mlogit** (Croissant *et al.* 2012), **RSGHB** (Dumont, Keller, and Carpenter 2014) and **gmnl** (Sarrias and Daziano 2015) allow estimating models with individual heterogeneity in the context of Multinomial Logit model in a very similar fashion to **Rchoice**. Other packages such as **FlexMix** (Grün and Leisch 2008) and **PReMiuM** (Liverani, Hastie, Azizi, Papathomas, and Richardson 2015) allow to estimate models with individual heterogeneity by assuming that $g(\boldsymbol{\theta})$ is discrete or a mixture of distributions. All these packages available in R cover almost all possibilities of estimating discrete choice models with random parameters or individual heterogeneity. The **Rchoice** is available from the Comprehensive R Archive Network (CRAN) at <http://CRAN.R-project.org/package=Rchoice>.

This paper is organized as follows. Section 2 briefly explains the methodology for models with random parameters. Section 3 gives a general description of the functions in **Rchoice**. Section 4 shows how **Rchoice** handles the random coefficients. All functionalities of **Rchoice** using real databases are presented in Section 5. Section 6 explains some computational issues that may arise when estimating random parameter models using SML. Finally, Section 7 concludes.

2. Methodology

2.1. Models with random parameters

In order to develop and motivate the idea behind random parameter models, consider the following latent process

$$\begin{aligned} y_{it}^* &= \mathbf{x}_{it}'\boldsymbol{\beta}_i + \epsilon_{it}, & i = 1, \dots, n; t = 1, \dots, T_i \\ \boldsymbol{\beta}_i &\sim g(\boldsymbol{\beta}_i|\boldsymbol{\theta}), \end{aligned} \tag{1}$$

where y_{it}^* is a latent (unobserved) process for individual i in period t , \mathbf{x}_{it} is a vector of covariates, and ϵ_{it} is the error term. Note that the conditional probability density function (pdf) of the latent process $f(y_{it}^*|\mathbf{x}_{it}, \boldsymbol{\beta}_i)$ is determined once the nature of the observed y_{it} and the population pdf of ϵ_{it} is known: if y_{it} is binary and ϵ_{it} is distributed as normal, then the latent process becomes the traditional Probit model; if y_{it} is an ordered categorical variable and ϵ_{it} is logistically distributed, then the traditional ordered Logit model arises. Formally, the pdf for binary, ordered, and Poisson model are, respectively

$$f(y_{it}^* | \mathbf{x}_{it}, \boldsymbol{\beta}_i) = \begin{cases} [F(\mathbf{x}'_{it}\boldsymbol{\beta}_i)]^{y_{it}} [1 - F(\mathbf{x}'_{it}\boldsymbol{\beta}_i)]^{1-y_{it}} \\ \prod_{j=1}^J [F(k_j - \mathbf{x}'_{it}\boldsymbol{\beta}_i) - F(k_{j-1} - \mathbf{x}'_{it}\boldsymbol{\beta}_i)]^{y_{itj}} \\ \frac{1}{y_{it}!} \exp[-\exp(\mathbf{x}'_{it}\boldsymbol{\beta}_i)] \exp(\mathbf{x}'_{it}\boldsymbol{\beta}_i)^{y_{it}} \end{cases} \quad (2)$$

For the binary and ordered models, $F(\cdot)$ represents the cumulative distribution function (cdf) of the error term, which $F(\epsilon) = \Phi(\epsilon)$ for Probit and $F(\epsilon) = \Lambda(\epsilon)$ for Logit.¹ For the ordered model, κ_j represents the threshold for alternative $j = 1, \dots, J - 1$, such that $\kappa_0 = -\infty$ and $\kappa_J = +\infty$.

In the structural model (1), we allow the vector coefficient $\boldsymbol{\beta}_i$ to be different for each individual in the population. In other words, the marginal effect on the latent dependent variable is individual-specific. Nevertheless, we do not know how these parameters vary across individuals. All we know is that they vary according to the population pdf $g(\boldsymbol{\beta}_i | \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the moments of the distribution such as the mean and the variance, which must be estimated. A fully parametric model arises once $g(\boldsymbol{\beta}_i | \boldsymbol{\theta})$ and the distribution of ϵ are specified. In Section 4 I detail the distributions allowed by **Rchoice**.

For simplicity in notation, assume that the coefficient vector is independent normal distributed, so that $\beta_k \sim N(\beta_k, \sigma_k^2)$ for the k th element in $\boldsymbol{\beta}_i$. Note that each coefficient can be written as $\beta_{ki} = \beta_k + \sigma_k \omega_i$, where $\omega_i \sim N(0, 1)$, or in vector form as $\boldsymbol{\beta}_i = \boldsymbol{\beta} + \mathbf{L}\boldsymbol{\omega}_i$, where \mathbf{L} is a diagonal matrix that contains the standard deviation parameters, σ_k . All the information about the individual heterogeneity for each individual attribute is captured by the standard deviation parameter σ_k . If $\sigma_k = 0$, then the model is reduced to the fixed parameter model, but if it is indeed significant then it would reveal that the relationship between x_{itk} and y_{it} is heterogeneous and focusing just on the central tendency β_k alone would veil useful information. It is useful to note that the random effect model is a special case in which only the constant is random (see Section 5.3).

2.2. Extensions: observed heterogeneity and correlation

One important and straightforward extension of the random parameter model is to allow the coefficients to be correlated. In this case, \mathbf{L} is a lower triangular matrix (also known as the Cholesky matrix), which produces the covariance matrix of the random parameters, $\mathbf{L}\mathbf{L}^\top = \boldsymbol{\Sigma}$. Furthermore, observed heterogeneity can be also accommodated by allowing the parameter heterogeneity to be partly systematic in terms of observed variables. This type of model is also known as a hierarchical model (Greene and Hensher 2010a). Formally, the parameter vector can be written as

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{s}_i + \mathbf{L}\boldsymbol{\omega}_i, \quad (3)$$

where $\boldsymbol{\Pi}$ is a matrix of parameters, \mathbf{s}_i is a vector of covariates that do not vary across time, and $\boldsymbol{\omega} \sim N(\mathbf{0}, \mathbf{I})$. Then, the mean of the parameters is $E(\boldsymbol{\beta}_i) = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{s}_i + \mathbf{L}E(\boldsymbol{\omega}) = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{s}_i$

¹For the Logit model $\Lambda(\epsilon) = \exp(\epsilon)/(1 + \exp(\epsilon))$.

and its covariance is $\text{VAR}(\boldsymbol{\beta}_i) = \mathbf{E}(\mathbf{L}\boldsymbol{\omega}(\boldsymbol{\omega}\mathbf{L})^\top) = \mathbf{L}\mathbf{E}(\boldsymbol{\omega}\boldsymbol{\omega}^\top)\mathbf{L} = \mathbf{L}\mathbf{I}\mathbf{L} = \mathbf{L}\mathbf{L}^\top = \boldsymbol{\Sigma}$.

As an illustration of the formulation above, suppose that we are interested in modeling some latent process, which is explained by two variables. The parameters associated with each observed variable, β_{1i} and β_{2i} , are not fixed, but rather they vary across individuals and are correlated. Furthermore, suppose that the means of the random parameters depend upon some observed individual socio-economic variables: S_i, B_i and C_i . Then, the parameters can be written as:

$$\begin{aligned}\beta_{1,i} &= \beta_1 + \pi_{1,1}S_i + \pi_{1,2}B_i + \pi_{1,3}C_i + s_{11}\omega_{1,i} \\ \beta_{2,i} &= \beta_2 + \pi_{2,1}S_i + \pi_{2,2}B_i + \pi_{2,3}C_i + s_{21}\omega_{1,i} + s_{22}\omega_{2,i}\end{aligned}$$

or in vector form:

$$\begin{pmatrix} \beta_{1,i} \\ \beta_{2,i} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \pi_{1,1} & \pi_{1,2} & \pi_{1,3} \\ \pi_{2,1} & \pi_{2,2} & \pi_{2,3} \end{pmatrix} \begin{pmatrix} S_i \\ B_i \\ C_i \end{pmatrix} + \begin{pmatrix} s_{11} & 0 \\ s_{21} & s_{22} \end{pmatrix} \begin{pmatrix} \omega_{1,i} \\ \omega_{2,i} \end{pmatrix}.$$

Since the mean of the random parameter is a function of observed variables, this specification relaxes the assumption of homogeneity across individuals in terms of the means. The unobserved part accounts for all other individual-specific factors that cannot be captured for the observed variables. Note that the variance-covariance matrix of the random parameters is:

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

and the conditional mean vector is $\mathbf{E}(\boldsymbol{\beta}_i|\mathbf{s}_i) = \boldsymbol{\beta} + \mathbf{I}\mathbf{s}_i$, which varies across individuals because of \mathbf{s}_i . Illustrations with real data for the model with correlated random parameters and with observed heterogeneity are presented in Section 5.2 and Section 5.4, respectively.

2.3. Estimation

In this section, I explain the simulated maximum likelihood (SML) procedure used by **Rchoice** to estimate models with random parameters. For a more complete treatment of SML see for example [Gourieroux and Monfort \(1997\)](#); [Lee \(1992\)](#); [Hajivassiliou and Ruud \(1994\)](#) or [Train \(2009\)](#).

Let $\mathbf{y}_i = \{y_{i1}, y_{i2}, \dots, y_{iT_i}\}$ be the sequence of choices made by individual i . Assuming that individuals are independent across time, the joint pdf, given $\boldsymbol{\beta}_i$, can be written as

$$\mathbf{P}(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\beta}_i) = \prod_{t=1}^{T_i} f(y_{it}^*|\mathbf{x}_{it}, \boldsymbol{\beta}_i), \quad (4)$$

where $f(y_{it}^*|\mathbf{x}_{it}, \boldsymbol{\beta}_i)$ is given in (2) for each model. Since $\boldsymbol{\beta}_i$ is unobserved, we need to integrate it out of the joint density, the unconditional pdf will be the weighted average of the conditional

probability (4) evaluated over all possible values of β , which depends on the parameters of the distribution of β_i :

$$P_i(\theta) = \int_{\beta_i} P(\mathbf{y}_i | \mathbf{X}_i, \beta_i) g(\beta_i) d\beta_i. \quad (5)$$

The probability in (5) has no close-form solution, that is, it is difficult to integrate out the random parameter and hence it is difficult to perform maximum likelihood (ML) estimation. However, ML estimation may still be possible if we instead use a good approximation $\tilde{P}_i(\theta)$ of $P_i(\theta)$ to form a likelihood function.

But, how can we obtain $\tilde{P}_i(\theta)$? A good approximation can be obtained by **Monte Carlo integration**. This procedure provides an alternative to deterministic numerical integration. Here we can ‘*simulate*’ the integration using random draws from the distribution $g(\beta_i)$.² For a given value of the parameters θ , a value of β_i is drawn from its distribution. Using this draw of β_i , $P_i(\theta)$ from Equation (5) is calculated. This process is repeated for many draws, and the average over the draws is the simulated probability. Formally, the simulated probability for individual i is

$$\tilde{P}_i(\theta) = \frac{1}{R} \sum_{r=1}^R \tilde{P}(\mathbf{y}_i | \mathbf{X}_i, \beta_{ir}), \quad (6)$$

where $\tilde{P}(\mathbf{y}_i | \mathbf{X}_i, \beta_{ir})$ is the simulated probability for individual i in period t evaluated at the r th draw of β_i , and R is the total number of draws. Given independence over i , the SML estimator is the value θ that maximizes:

$$\hat{\theta}_{SML} \equiv \arg \max_{\theta \in \Theta} \sum_{i=1}^N \log \tilde{P}_i(\theta).$$

Lee (1992) and Hajivassiliou and Ruud (1994) show that under regularity conditions, the SML estimator is consistent and asymptotically normal. When the number of draws, R , rises faster than the square root of the number of observations, the estimator is asymptotically equivalent to the maximum likelihood estimator.

Rchoice uses **maxLik** package (Henningsen and Toomet 2011) to perform ML and SML estimation procedures. All models with random parameters are estimated using SML, while models with fixed parameters are estimated by ML. Furthermore, **Rchoice** uses analytical gradients and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (the default) to iteratively solve the SML. For further details about the computation of SML see Section 6.

3. An overview of Rchoice package

After installation, **Rchoice** is loaded by typing:

²Another numerical approximation is Gauss-Hermite quadrature. However, it has been documented that for models with more than 3 random parameters SML performs better (Train 2009).

```
R> library("Rchoice")
```

The main function in the package is `Rchoice`, which enables us to estimate the models. The arguments for this function are the following:

```
R> args(Rchoice)
```

```
function (formula, data, subset, weights, na.action, family,
  start = NULL, ranp = NULL, R = 40, haltons = NA, seed = 10,
  correlation = FALSE, panel = FALSE, index = NULL, mvar = NULL,
  print.init = FALSE, init.ran = 0.1, gradient = TRUE, ...)
NULL
```

The `formula` argument is a symbolic description of the model to be estimated and consists of two parts. The first one is reserved for standard variables with fixed and random parameters. The second part is reserved for variables that enter the mean of the random parameters (\mathbf{s}_i from Equation (3)). The usage of the second part is further explained in Section 5.4. The models are specified by the argument `family`. The models estimated by `Rchoice` and the corresponding family function are presented in Table 1.

<i>Model</i>	<i>Function</i>
Poisson	<code>family = poisson</code>
Binary Probit	<code>family = binomial("probit")</code>
Binary Logit	<code>family = binomial("logit")</code>
Ordered Probit	<code>family = ordinal("probit")</code>
Ordered Logit	<code>family = ordinal("logit")</code>

Table 1: Models estimated by **Rchoice** and the corresponding family function.

The main arguments for the control of the random parameters are `ranp`, `R`, `haltons`, `seed`, `correlation` and `mvar`. The argument `ranp` is a vector that specifies the distribution of the random parameters. The distributions allowed by `Rchoice` are presented in Section 4 and illustrated in Section 5.2. `R` specifies the number of draws used for the simulation of the probabilities. `haltons` indicates the type of draws used in the simulation procedure and it is further explained in Section 4. The argument `seed` sets the seed for the pseudo-random draws, being 10 the default. If `correlation = TRUE`, the correlated random parameter model presented in Section 2.2 is estimated. For an example see Section 5.2. Finally, the argument `mvar` is a named list that indicates how the variables that modify the mean of the random parameters enter in each random parameter. This feature is illustrated in Section 5.4.

The main arguments for panel or longitudinal data bases are `panel` and `index`. If `panel = TRUE`, then the panel structure of the data is taken into account. `index` is a string variable indicating the ‘id’ for individuals in the data. Models with panel structure are presented in Section 5.3.

Finally, the arguments `print.init`, `init.ran` and `gradient` are intended to control the optimization procedure. If `print.init = TRUE`, then the initial values for the optimization procedure are printed. The argument `init.ran` sets the initial values for the standard deviation of the random parameters, $\sigma_k, \forall k$. Numerical instead of analytical gradients are used in the optimization of either ML or SML if `gradient = FALSE`. Other arguments that control the optimization can be passed to **maxLik** package. A more detail discussion of these arguments are presented in Section 6.

Other important functions in **Rchoice** are presented in Table 2.

<i>Function</i>	<i>Section</i>	<i>Description</i>
<code>vcov</code>	Section 5.2	Method function that extracts the covariance matrix of the coefficients or the random parameters. It also allows getting the standard errors for the variance-covariance matrix of the random parameters.
<code>effect.Rchoice</code>	Section 5.5	Function to obtain the individuals' conditional estimate of the random parameters or compensating variations.
<code>plot</code>	Section 5.5	Plot the individuals' conditional estimates of the random parameters or compensating variations.

Table 2: Other important functions in the **Rchoice** package.

4. Drawing from densities

As mentioned in Section 2.1, the distribution of the random parameters $g(\beta_i)$ can take any shape, but it must be chosen a priori by the researcher. This is a critical step in applied work. As stated by Hensher and Greene (2003), “*distributions are essentially arbitrary approximations to the real behavioral profile. The researcher chooses a specific distribution because he has a sense that the “empirical truth” is somewhere in their domain*”. Therefore, some prior theoretical knowledge about the expected domain of the coefficients may lead to a more appropriate choice of the distributions. The distributions allowed by **Rchoice** are the following: normal, triangular, uniform, log-normal, truncated normal and Johnson S_b . These distributions are explained in further detail below.

Another important issue is that good performance of SML requires a very large number of draws. The main drawback to this approach is that with large samples and complex models, the maximization of SML can be very time consuming. Researchers have gained speed with no degradation in simulation performance through the use of smaller number of Halton draws (Bhat 2001; Train 2000). The Halton sequence is constructed based on a deterministic method that uses prime numbers as its base (see Train 2009, for further details). 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). Figure 1 shows two sequences of 1,000 Halton draws (Panel A) and two sequences of 1,000 pseudo-random draws (Panel B). The Halton draws are based on prime numbers 2 and 3. It can be observed that the Halton draws have better coverage of the unit square than the pseudo-random draws. This characteristic of the Halton sequence ensures a better coverage of the multidimensional area of integration and reduces the computation time of the SML.

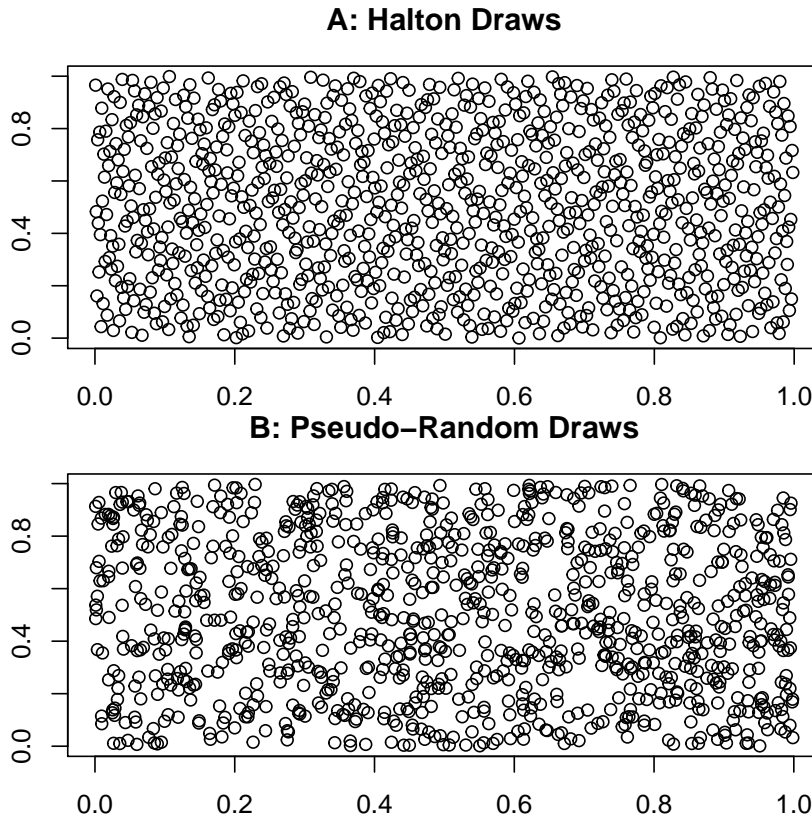


Figure 1: Halton vs pseudo-random draws.

Rchoice handles Halton or pseudo-random draws in the following way. Suppose that there are K random parameters. Then, the K elements of ω_{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.

An important attribute of the Halton values is that they are also distributed in the $(0, 1)$ interval as shown in Figure 1. 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 distribution:

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

where β_k and σ_k are estimated. Then, $\beta_{k,i} \sim N(\beta_k, \sigma_k^2)$. Since the domain of the normal distribution is $(-\infty, +\infty)$, assuming a given coefficient to follow a normal distribution is equivalent to making an a priori assumption that there is a proportion of individuals with a positive coefficient and another proportion with negative ones. For example, the proportion of positive coefficients can be computed as $\Phi(\widehat{\beta}_k/\widehat{\sigma}_k)$. The main disadvantage of the normal distribution is that it has infinite tails, which may result in some individuals having implausible extreme coefficients. If this is the case, the triangular or uniform distribution may be more appropriate.

2. 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. The pdf of this distribution looks like a triangle, and its main advantage is that it has a definite upper and lower limits resulting in shorter tails than the normal distribution.

3. Uniform distribution:

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

where β_k and σ_k are estimated. In this case, the parameter for each individual is equally likely to take on any value in some interval. Note also that the uniform distribution with bounds 0 and 1 is very suitable when there exists individual heterogeneity in a dummy variable. For this case, the restriction $\beta_k = \sigma_k = 1/2$ can be applied.

4. 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, $\beta_{k,i} \sim \log N(\beta_k, \sigma_k^2)$. The support of the log-normal distribution is $(0, +\infty)$, therefore the coefficient is allowed to have individual heterogeneity in the positive domain only. If the some coefficient is expected a priori to be negative for all the individuals, one can create the negative of the variable and then include this new variable in the estimation. This allows the coefficient to be negative without imposing a sign change in the estimation procedure (Train 2009).

5. Truncated (at 0) normal distribution:

$$\beta_{k,ir} = \begin{cases} \beta_k + \sigma_k w_{k,ir} & \text{if } \beta_{k,ir} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$w_{k,ir} \sim N(0, 1)$$

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. This distribution is useful when the researcher has a priori belief that for some individuals the marginal latent effect of the variable is null.

6. Johnson's S_b distribution:

$$\beta_{k,ir} = \frac{\exp(\beta_k + \sigma_k w_{k,ir})}{1 + \exp(\beta_k + \sigma_k w_{k,ir})}$$

$$w_{k,ir} \sim N(0, 1)$$

where β_k and σ_k are estimated. This distribution gives coefficients between 0 and 1, which is also very suitable for dummy variables. If the researcher needs the coefficient to be between 0 and d , then the variable can be multiplied by d before estimation. The main advantage of the Johnson S_b is that it can be shaped like log-normal distribution, but with thinner tails below the bound.

Rchoice allows to the user to specify both types of random draws by the argument `haltons`: pseudo-random draws (`haltons = NULL`) and Halton draws (`haltons = NA`) as default. For the Halton draws, the default is to use the first K 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 number of elements dropped for each serie. For example, if $K = 2$, 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

In this section I present some of the capabilities of **Rchoice**. Section 5.1 shows how to estimate Poisson, binary and ordered response models with fixed parameters. Examples of how to estimate models with random parameters are presented in Section 5.2. Section 5.3 explains how to estimate models with random parameters with panel or longitudinal data. In Section 5.4, models with both observed and unobserved heterogeneity are estimated. Finally, Section 5.5 shows how the conditional estimates for each individual can be plotted.

5.1. Standard models

In this section, I show the capabilities of **Rchoice** to estimate Poisson, binary and ordered regression models without random parameters. The main objective of this section is to show how **Rchoice** can interact with other packages in R.

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

```
R> data("Articles")
R> 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:

```
R> 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 `family = poisson`:

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

```
Model: poisson
Model estimated on: Mon May 11 10:57:16 2015
```

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

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

```
Coefficients:
      Estimate Std. Error z-value Pr(>|z|)
constant  0.304617  0.102982   2.958  0.0031 **
fem       -0.224594  0.054614  -4.112 3.92e-05 ***
mar        0.155243  0.061375   2.529  0.0114 *
kid5      -0.184883  0.040127  -4.607 4.08e-06 ***
```

```

phd      0.012823   0.026397   0.486   0.6271
ment     0.025543   0.002006  12.733  < 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
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 the Newton-Raphson algorithm in 7 iterations. 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 the 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** (Zeileis 2006):

```

R> library("sandwich")
R> library("lmtest")
R> coeftest(poisson.fixed, vcov = sandwich)

```

t test of coefficients:

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

```

R> vcov.stata <- vcovHC(poisson.fixed, type = "HC0") *
+                 nObs(poisson.fixed) / (nObs(poisson.fixed) - 1)
R> coeftest(poisson.fixed, 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 the `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 the **lmtest** package (Zeileis and Hothorn 2002). For example, we can test the nonlinear hypothesis that the ratio between the `phd` and `ment` coefficient is zero³, $H_0 : \text{phd}/\text{ment} = 0$, by:

```
R> library("car")
R> deltaMethod(poisson.fixed, "phd/ment")
```

	Estimate	SE
phd/ment	0.5020048	1.043031

The main argument to estimate other models is `family` (see Table 1). This provides a convenient way to specify the details of the models used by **Rchoice**. For Probit models, the user should specify `family = binomial("probit")`, and for Logit `family = binomial("logit")`. In the following example, I 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.

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

Model: binomial

Model estimated on: Mon May 11 10:57:16 2015

Call:

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

³This ratio can be thought as a compensating variation between both variables. See for example Section 5.5 for further details of compensating variation with individual heterogeneity.

Frequencies of categories:

```
y
      0      1
0.4322 0.5678
The estimation took: 0h:0m:0s
```

Coefficients:

	Estimate	Std. Error	z-value	Pr(> z)	
constant	2.781983	0.441876	6.296	3.06e-10	***
k5	-0.880688	0.113436	-7.764	8.22e-15	***
k618	-0.038656	0.040454	-0.956	0.339304	
age	-0.037701	0.007612	-4.953	7.31e-07	***
wc	0.481150	0.135271	3.557	0.000375	***
hc	0.077440	0.124733	0.621	0.534700	
lwg	0.371645	0.087605	4.242	2.21e-05	***
linc	-0.451494	0.100748	-4.481	7.42e-06	***

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

Optimization of log-likelihood by Newton-Raphson maximisation

Log Likelihood: -451.9

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 the 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,...,4.⁴

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

Model: ordinal

Model estimated on: Mon May 11 10:57:17 2015

Call:

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

⁴See `help(Health)`.

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	z-value	Pr(> z)	
kappa.1	2.094209	0.090487	23.144	< 2e-16	***
kappa.2	4.198067	0.105952	39.623	< 2e-16	***
kappa.3	4.961227	0.115706	42.878	< 2e-16	***
constant	2.170194	0.778474	2.788	0.00531	**
age	-0.030637	0.004517	-6.782	1.19e-11	***
educ	0.065848	0.018456	3.568	0.00036	***
married	-0.336312	0.110278	-3.050	0.00229	**
hhkids	0.220423	0.099485	2.216	0.02672	*
linc	0.176496	0.098799	1.786	0.07403	.

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

Optimization of log-likelihood by BFGS maximisation

Log Likelihood: -2663

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, I show how to estimate those kinds of models for count data using cross-sectional data. For binary and ordered models the syntax is the same provided that the `family` argument is correctly specified.

Continuing with the previous Poisson model, now I will assume that the effect of `kid5`, `phd` and `ment` are not fixed, but rather heterogeneous across the population. Specifically, I will assume that the coefficients for those variables are independent normally distributed, that is, I will not allow correlation among them. In particular,

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

where $\omega_{k,ir} \sim N(0,1)$. Then, the Poisson model with random parameter is estimated using the following syntax:

```
R> 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, I have specified that all of them are normal distributed using "n". The shorthand for the remaining distributions that can be used are:⁵

- 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 as default, but if users want pseudo-random draws they can specify `haltons = NULL` (see Section 4).

```
R> summary(poisson.ran)
```

```
Model: poisson
```

```
Model estimated on: Mon May 11 10:57:43 2015
```

```
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:26s
```

```
Coefficients:
```

	Estimate	Std. Error	z-value	Pr(> z)	
constant	0.225583	0.132500	1.703	0.08866	.
fem	-0.218498	0.070558	-3.097	0.00196	**
mar	0.156431	0.079121	1.977	0.04803	*
mean.kid5	-0.197775	0.063472	-3.116	0.00183	**
mean.phd	-0.029942	0.037217	-0.805	0.42109	
mean.ment	0.031110	0.003814	8.158	4.44e-16	***
sd.kid5	0.285310	0.089104	3.202	0.00136	**

⁵These shorthands are the same as those used by `mlogit` and `gmnl` packages.

```
sd.phd      0.165405    0.016585    9.973 < 2e-16 ***
sd.ment     0.015876    0.003535    4.491 7.11e-06 ***
```

```
---
```

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

```
Optimization of log-likelihood by BFGS maximisation
```

```
Log Likelihood: -1574
```

```
Number of observations: 915
```

```
Number of iterations: 81
```

```
Exit of MLE: successful convergence
```

```
Simulation based on 40 Halton draws
```

It is important to check the exit 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:

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

```
mean.kid5
```

```
0.2440946
```

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 whether $H_0 = \sigma_{\text{kid5}} = \sigma_{\text{phd}} = \sigma_{\text{ment}} = 0$. This can be done by using the function `waldtest` or `lrtest` from package **lmtest**:

```
R> waldtest(poisson.fixed, poisson.ran)
```

```
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      909
2      906  3 280.14 < 2.2e-16 ***
```

```
---
```

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

```
R> lrtest(poisson.fixed, poisson.ran)
```

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    6 -1651.1
2    9 -1574.2  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:

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

Both models `poisson.ran` and `poisson.ran2` can be compared using `mtable` from **memisc** (Elff 2014) package:

```
R> library("memisc")
```

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

```
R> Ttable
```

Calls:

```
model 1: Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
  family = poisson, ranp = c(kid5 = "n", phd = "n", ment = "n"),
  method = "bfgs", iterlim = 2000)
model 2: Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
  family = poisson, ranp = c(kid5 = "u", phd = "t", ment = "cn"),
  method = "bfgs", iterlim = 2000)
```

```
=====
              model 1  model 2
-----
constant      0.226    0.254
              (0.132)  (0.132)
fem           -0.218** -0.216**
              (0.071)  (0.069)
mar            0.156*   0.149
              (0.079)  (0.076)
mean.kid5     -0.198** -0.221***
              (0.063)  (0.064)
mean.phd      -0.030   -0.104*
              (0.037)  (0.041)
mean.ment      0.031***  0.028***
              (0.004)  (0.003)
sd.kid5       0.285**   0.508***
```

	(0.089)	(0.123)
sd.phd	0.165***	0.226***
	(0.017)	(0.025)
sd.ment	0.016***	0.022***
	(0.004)	(0.003)

N	915	915
Log-likelihood	-1574.166	-1575.816
BIC	3209.702	3213.003
AIC	3166.332	3169.632
=====		

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, I estimate the model `poissonc.ran`, but assuming that the random parameters are correlated: $\beta_i \sim N(\beta, \Sigma)$ for a general matrix Σ . The main argument for this model is `correlation = TRUE`:

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

```
Model: poisson
Model estimated on: Mon May 11 10:59:16 2015
```

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

```
The estimation took: 0h:1m:1s
```

```
Coefficients:
```

	Estimate	Std. Error	z-value	Pr(> z)	
constant	0.235301	0.131432	1.790	0.073409	.
fem	-0.228057	0.070992	-3.212	0.001316	**
mar	0.150374	0.079625	1.889	0.058954	.
mean.kid5	-0.229971	0.063024	-3.649	0.000263	***
mean.phd	-0.032431	0.037128	-0.874	0.382384	
mean.ment	0.033804	0.003751	9.012	< 2e-16	***
sd.kid5.kid5	0.279620	0.091789	3.046	0.002317	**
sd.kid5.phd	0.084343	0.055691	1.514	0.129908	
sd.kid5.ment	-0.025400	0.005943	-4.274	1.92e-05	***
sd.phd.phd	-0.143787	0.028258	-5.088	3.61e-07	***
sd.phd.ment	-0.002123	0.007752	-0.274	0.784153	

```
sd.ment.ment  0.011351  0.007372  1.540 0.123616
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Optimization of log-likelihood by BFGS maximisation
Log Likelihood: -1571
Number of observations: 915
Number of iterations: 211
Exit of MLE: successful convergence
Simulation based on 40 Halton draws
```

The output prints the mean of the random parameters along with the lower-triangular Cholesky factor \mathbf{L} . We can extract the variance-covariance matrix, Σ , and the correlation matrix of the random parameters using S3 method `vcov` in the following way:

```
R> vcov(poissonc.ran, what = "ranp", type = "cov")
```

```
          kid5          phd          ment
kid5  0.07818737  0.023583914 -0.0071022904
phd   0.02358391  0.027788312 -0.0018369769
ment -0.00710229 -0.001836977  0.0007785002
```

```
R> vcov(poissonc.ran, what = "ranp", type = "cor")
```

```
          kid5          phd          ment
kid5  1.0000000  0.5059604 -0.910334
phd   0.5059604  1.0000000 -0.394951
ment -0.9103340 -0.3949510  1.000000
```

The argument `what` indicates which covariance matrix has to be extracted. The default is `coefficient`, and the `vcov` behaves as usual. If `what = "ranp"` the covariance matrix of the random parameters is returned as default. The argument `type` indicates what matrix of the random parameters should be returned. Among other things, the output shows that `ment` is negatively related with `phd` and `kid5`. Specifically, we can see that the correlation between `phd` and `ment` is around -0.4. We can also test whether the variances of the random parameters are significant using Delta Method. To do so, we can use the `se = TRUE` argument in the `vcov` function, which is a wrapper of `deltamethod` function from `msm` (Jackson 2011) package:

```
R> vcov(poissonc.ran, what = "ranp", type = "cov", se = TRUE)
```

```
Elements of the variance-covariance matrix
```

```
          Estimate Std. Error z-value Pr(>|z|)
v.kid5.kid5  0.07818737  0.05133227  1.5232 0.127718
v.kid5.phd   0.02358391  0.01152073  2.0471 0.040650 *
```

```

v.kid5.ment -0.00710229  0.00245237 -2.8961  0.003778 **
v.phd.phd   0.02778831   0.00896418  3.0999  0.001936 **
v.phd.ment -0.00183698  0.00186657 -0.9841  0.325044
v.ment.ment 0.00077850   0.00032116  2.4240  0.015349 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Note that the covariance between `ment` and `phd` even though is negative is not significant. To get the standard errors of the standard deviations for the random parameters, we might use:

```
R> vcov(poissonc.ran, what = "ranp", type = "sd", se = TRUE)
```

Standard deviations of the random parameters

```

      Estimate Std. Error z-value Pr(>|z|)
kid5 0.2796201  0.0917893  3.0463  0.002317 **
phd  0.1666983  0.0268875  6.1999  5.652e-10 ***
ment 0.0279016  0.0057552  4.8481  1.247e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

5.3. Random parameters models with panel data

The current version of this package also handles panel data by estimating random effect (RE) models. Suppose that the i observation \mathbf{y}_i has unconditional joint density $f(\mathbf{y}_i^*|\mathbf{X}_i, \alpha_i, \boldsymbol{\beta})$ and the random effect has density $\alpha_i \sim g(\alpha_i|\boldsymbol{\theta})$, where $g(\alpha_i|\boldsymbol{\theta})$ does not depend on observables. Thus, the unconditional joint density for the i th observation is

$$f(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\beta}, \boldsymbol{\theta}) = \int_{\alpha_i} \left[\prod_{t=1}^{T_i} f(y_{it}^*|\mathbf{x}_{it}, \alpha_i, \boldsymbol{\beta}) \right] g(\alpha_i|\boldsymbol{\theta}) d\alpha_i$$

This model can be estimated using the package **pglm** (Croissant 2013), which uses **Gauss-Hermite quadrature** to approximate the integration in the probability. However, note that this model can be seen as a random parameter model where the constant is random. Therefore, users might estimate a simple RE model with **Rchoice** by typing `ranp = (constant = 'n')`.

In the following example I will estimate a binary Probit model with RE and random parameters using Unions database from the **pglm** package.

```

R> data("Unions", package = "pglm")
R> Unions$lwage <- log(Unions$wage)

```

The model is estimated using the following syntax:


```
R> 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, I assumed that `lwage` is distributed as triangular, while the constant is assumed to be normal distributed. This is the same as assuming that $\alpha_i \sim N(0, \sigma_\alpha^2)$.

There are two main arguments for the panel estimation. The argument `panel = TRUE` indicates that the data is a panel. This implies that users should indicate the variable that corresponds to the 'id' of the individuals in the `index` argument. In the this example, the 'id' is given by the variable "id".

Finally, the argument `print.init = TRUE` indicates that the initial values used by `Rchoice` will be displayed. This argument is useful if one wants to see the magnitude of the initial values for the parameters.

```
R> summary(union.ran)
```

Model: binomial

Model estimated on: Mon May 11 10:59:55 2015

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:39s

Coefficients:

	Estimate	Std. Error	z-value	Pr(> z)	
age	0.05407	0.32802	0.165	0.869	
exper	-0.09174	0.02098	-4.372	1.23e-05	***
ruralyes	0.22735	0.17537	1.296	0.195	
mean.constant	-1.30073	0.24497	-5.310	1.10e-07	***
mean.lwage	0.01089	0.15419	0.071	0.944	
sd.constant	1.19918	0.16110	7.444	9.79e-14	***
sd.lwage	1.04202	0.10745	9.698	< 2e-16	***

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

Optimization of log-likelihood by BFGS maximisation

Log Likelihood: -738.6

Number of observations: 2000

Number of iterations: 77

Exit of MLE: successful convergence

Simulation based on 10 Halton draws

The results indicate that $\sigma_\alpha = 1.2$ and is significant. Furthermore, the finding of a significant standard deviation yet insignificant mean for `lwage` attests to the existence of substantial heterogeneity; positive and negative coefficients in the sample compensate for each other, such that the coefficient of 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:

```
R> 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)
R> summary(oprobit.ran)
```

5.4. Random parameter model with observed heterogeneity

In this section I illustrate how to estimate a Poisson random parameter model with observed heterogeneity. In the following example, I assume that there exists not only unobserved heterogeneity in the coefficients for `phd` and `ment`, but also observed heterogeneity in the mean of the coefficients. Specifically, I assume that the coefficients vary across individuals according to:

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

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

```
R> 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`.

```
R> summary(poissonH.ran)
```

Model: poisson

Model estimated on: Mon May 11 11:00:50 2015

Call:

```
Rchoice(formula = art ~ fem + mar + kid5 + phd + ment | fem +
+       phd, data = Articles, family = poisson, ranp = c(kid5 = "n",
+       phd = "n", ment = "n"), R = 10, mvar = list(phd = c("fem"),
+       ment = c("fem", "phd")), method = "bfgs", iterlim = 2000)
```

The estimation took: 0h:0m:55s

Coefficients:

	Estimate	Std. Error	z-value	Pr(> z)	
constant	0.222646	0.181920	1.224	0.22100	
fem	-0.572068	0.217337	-2.632	0.00848	**
mar	0.176966	0.075165	2.354	0.01855	*
mean.kid5	-0.259958	0.061981	-4.194	2.74e-05	***
mean.phd	-0.019339	0.056808	-0.340	0.73354	
mean.ment	0.048094	0.009960	4.829	1.37e-06	***
phd.fem	0.133917	0.068995	1.941	0.05226	.
ment.fem	-0.006846	0.007134	-0.960	0.33721	

```

ment.phd -0.004857  0.002902 -1.674  0.09415 .
sd.kid5   0.431139  0.085080  5.067 4.03e-07 ***
sd.phd    0.133125  0.015333  8.682 < 2e-16 ***
sd.ment   0.014867  0.003050  4.874 1.09e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Optimization of log-likelihood by BFGS maximisation
Log Likelihood: -1577
Number of observations: 915
Number of iterations: 99
Exit of MLE: successful convergence
Simulation based on 10 Halton draws

```

The estimated parameters indicate that gender matters only for `phd`'s mean coefficient. Moreover, the results indicate that the prestige of the PhD reduce the effect of `ment`. We can test whether the interaction variables are jointly significant by using `lrtest` function:

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

Likelihood ratio test

```

Model 1: art ~ fem + mar + kid5 + phd + ment
Model 2: art ~ fem + mar + kid5 + phd + ment | fem + phd
#Df LogLik Df Chisq Pr(>Chisq)
1  9 -1574.2
2 12 -1576.9  3 5.4385  0.1424

```

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 derive an individual-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:⁶

$$\hat{\beta}_i = \hat{E}(\beta_i | \text{data}_i) = \sum_{r=1}^R \left(\frac{\hat{P}(\mathbf{y}_i | \mathbf{X}_i, \beta_i)}{\sum_{r=1}^R \hat{P}(\mathbf{y}_i | \mathbf{X}_i, \beta_i)} \right) \hat{\beta}_{ir},$$

where:

$$\hat{\beta}_{ir} = \hat{\beta} + \hat{\Pi} \mathbf{s}_i + \hat{\mathbf{L}} \omega_{ir}.$$

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:

⁶Note that this simulator is also known as the posterior distribution of the individual parameters.

$$\widehat{E}(\beta_i^2 | \text{data}_i) = \sum_{r=1}^R \left(\frac{\widehat{P}(\mathbf{y}_i | \mathbf{X}_i, \beta_i)}{\sum_{r=1}^R \widehat{P}(\mathbf{y}_i | \mathbf{X}_i, \beta_i)} \right) \widehat{\beta}_{ir}^2,$$

and then computing the square root of the estimated variance,

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

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 plotting 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:

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

The graph produced using this command is visualized in Figure 2. We may also plot the individual confident interval for the conditional means for the first 20 individuals using:

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

Figure 3 displays the individual conditional means and their respectively confident interval.

The method `plot` for objects of class `Rchoice` is a wrapper of `effect.Rchoice` function. This function retrieves the individual's conditional mean of both the parameters and the compensating variations. For example, one can get the individual's conditional mean and standard errors plotted in Figure 2 typing:

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

The argument `effect` is a string indicating what conditional mean should be computed. In this example, we are requiring the conditional expectation of the individual coefficients `effect = "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.

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

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

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

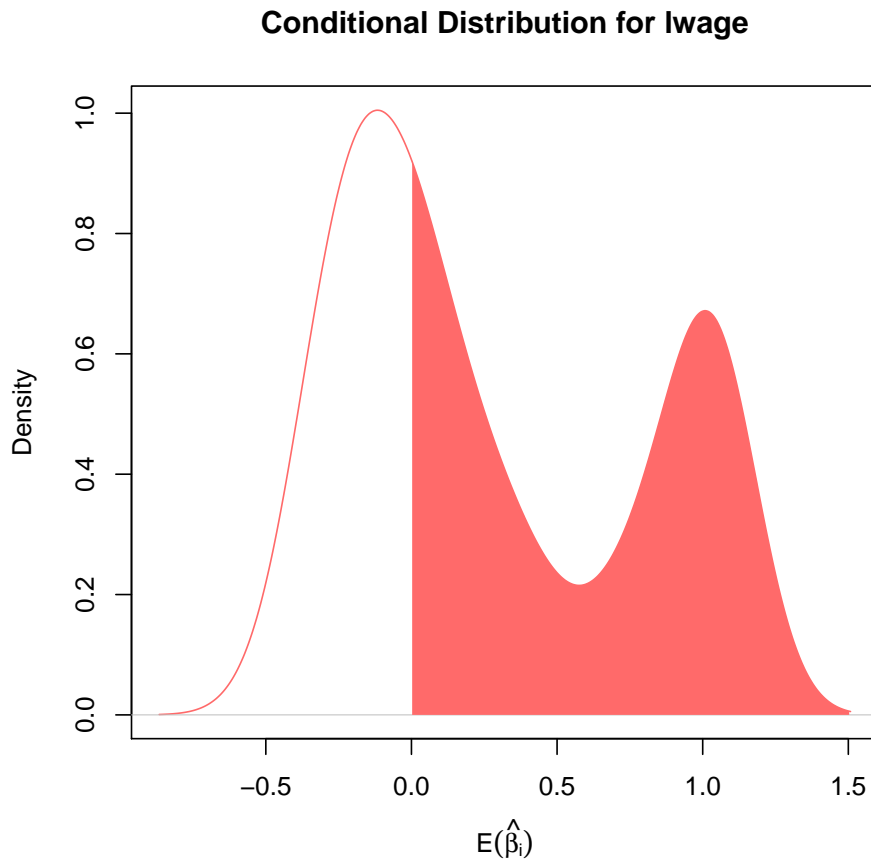


Figure 2: Kernel density of the individual's conditional mean.

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 individual's conditional mean of the **compensating variations** using both `plot` and `effect.Rchoice`. Compensating variation is the variation in two regressors such that the latent variable 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 $y_i^* = 0$. 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

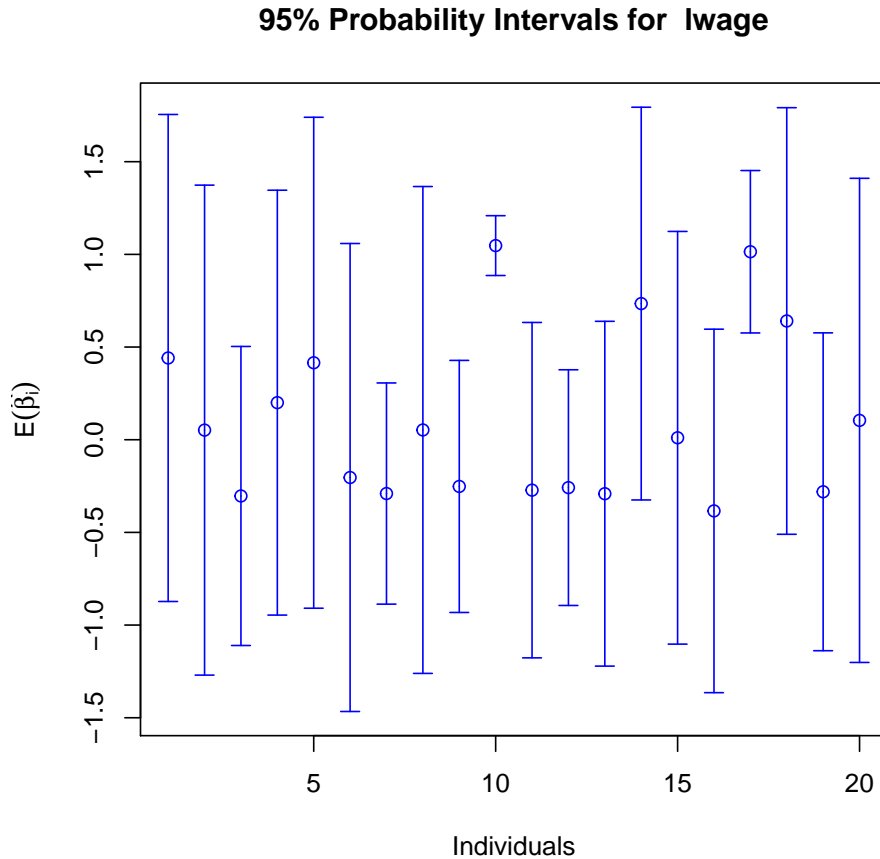


Figure 3: Individual confident interval for the conditional means.

of the coefficient is random and its distribution follows from the joint distribution of the coefficients.

6. Computational issues

The estimated parameters in any model estimated using SML depend on at least four factors. The first of them is the random number seed. If the random draws used in the estimation are pseudo-random draws, instead of Halton draws, then the parameter might change if the seed is changed. As default, **Rchoice** sets `seed = 10`. Second, the number of draws are very important for the asymptotic properties of the SML (see Section 2.3). In the examples provided in this document, I used just a few draws for time restrictions. However, in real applied work, researchers must use a greater number of draws, especially if pseudo-random draws are used in the estimation. A good starting point is 500 draws (see for example [Bhat 2001](#); [Sándor and Train 2004](#)).

Starting values are also crucial to achieve convergence, specially for the correlated random

parameter model with deterministic variation in the mean. Note that version 0.1 of **Rchoice** sets the initial values at 0, while versions 0.2 and 0.3 set them at 0.1 as default. Users might change this with the argument `init.ran` or setting a new vector of starting values with the argument `start`.

Finally, the choice of optimization method is another important factor that influences model convergence. As explained before, **Rchoice** uses the function `maxLik` in order to maximize the log-likelihood function, which permits to estimate models by the Newton-Raphson (NR), BFGS and Berndt-Hall-Hall-Hausman (BHHH) procedures. I have found that BFGS often works best in the sense it is more likely to converge than the alternative algorithm. That is the main reason why this algorithm is set as default. If this method does not converge, users might re-estimate the model using another algorithm. For example, the user might type `method = "nr"` for the NR method or `method = "bhhh"` for the BHHH method in the **Rchoice** function. **Rchoice** uses the numerical Hessian if `method = "nr"` and the model is estimated with random parameters, thus it can be very slow compared to the other methods. It is worth mentioning that BHHH is generally faster than the other procedures, but it can failure 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 fact, users should check the variables and re-scale or recode them if necessary. Furthermore, it is also convenient to use the argument `print.level = 2`, for example, to trace the optimization procedure in real time. For more information about arguments for optimization type `help(maxLik)`.

7. Conclusions

The **Rchoice** package contains most of the newly developed models in binary, count and ordered models with random parameters. The current version of **Rchoice** handles cross-sectional an panel data with observed and unobserved individual 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, triangular distribution and Johnson's S_b . It also allows the user to 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, I expect to add functions that allow estimating latent class models. I also hope to include functions to compute marginal effects.

References

- Anastasopoulos PC, Mannering FL (2009). "A Note on Modeling Vehicle Accident Frequencies with Random-Parameters Count Models." *Accident Analysis & Prevention*, **41**(1), 153–159. doi:<http://dx.doi.org/10.1016/j.aap.2008.10.005>.

- Bhat CR (2001). “Quasi-Random Maximum Simulated Likelihood Estimation of the Mixed Multinomial Logit Model.” *Transportation Research Part B: Methodological*, **35**(7), 677–693. doi:[http://dx.doi.org/10.1016/S0191-2615\(00\)00014-X](http://dx.doi.org/10.1016/S0191-2615(00)00014-X).
- 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>.
- Doran H, Bates D, Bliese P, Dowling M, *et al.* (2007). “Estimating the Multilevel Rasch Model: With the **lme4** Package.” *Journal of Statistical Software*, **20**(2), 1–18. URL <http://www.jstatsoft.org/v20/i02>.
- Dumont J, Keller J, Carpenter C (2014). *RSGHB: Functions for Hierarchical Bayesian Estimation: A Flexible Approach*. R package version 1.0.2, URL <http://CRAN.R-project.org/package=RSGHB>.
- Elff M (2014). *memisc: Tools for Management of Survey Data, Graphics, Programming, Statistics, and Simulation*. R package version 0.96-10, URL <http://CRAN.R-project.org/package=memisc>.
- Falco P, Maloney WF, Rijkers B, Sarrias M (2015). “Heterogeneity in Subjective Wellbeing: An Application to Occupational Allocation in Africa.” *Journal of Economic Behavior & Organization*, **111**(0), 137 – 153. doi:<http://dx.doi.org/10.1016/j.jebo.2014.12.022>.
- 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*. URL <http://cran.r-project.org/web/packages/car/index>.
- 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*, **52**(3), 701–720. URL <http://www.jstor.org/stable/1913472>.
- 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 (2012). *Econometric Analysis*. 7th edition. Prentice Hall.
- Greene WH (2015a). *LIMDEP: Version 10: Econometric Modeling Guide*. Econometric Software.
- Greene WH (2015b). *NLOGIT: Version 5: User’s Guide*. Econometric Software.

- 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. URL <http://www.jstor.org/stable/25801404>.
- Grün B, Leisch F (2008). “**FlexMix** Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters.” *Journal of Statistical Software*, **28**(4), 1–35. URL <http://www.jstatsoft.org/v28/i04/>.
- Hajivassiliou VA, Ruud PA (1994). “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 (2014). *micEcon: Microeconomic Analysis and Modelling*. R package version 0.6-12, URL <http://CRAN.R-project.org/package=micEcon>.
- 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. doi:10.1023/A:1022558715350.
- Jackson CH (2011). “Multi-State Models for Panel Data: The **msm** Package for R.” *Journal of Statistical Software*, **38**(8), 1–29. URL <http://www.jstatsoft.org/v38/i08/>.
- Krapf M, Ursprung HW, Zimmermann C (2014). “Parenthood and Productivity of Highly Skilled Labor: Evidence From the Groves of Academe.” *Technical report*, FRB of St. Louis Working Paper.
- 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. URL <http://www.jstor.org/stable/3532129>.
- Liverani S, Hastie DI, Azizi L, Papathomas M, Richardson S (2015). “**PRemiuM**: An R Package for Profile Regression Mixture Models Using Dirichlet Processes.” *Journal of Statistical Software*, **64**(7), 1–30. URL <http://www.jstatsoft.org/v64/i07>.
- Long JS (1990). “The Origins of Sex Differences in Science.” *Social Forces*, **68**(4), 1297–1316. URL <http://www.jstor.org/stable/2579146>.
- Long JS (1997). *Regression Models for Categorical and Limited Dependent Variables*, volume 7. Sage.
- R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>.
- Sándor Z, Train K (2004). “Quasi-Random Simulation of Discrete Choice Models.” *Transportation Research Part B: Methodological*, **38**(4), 313–327. doi:[http://dx.doi.org/10.1016/S0191-2615\(03\)00014-6](http://dx.doi.org/10.1016/S0191-2615(03)00014-6).

- Sarrias M (2015). *Rchoice: Discrete Choice (Binary, Poisson and Ordered) Models with Random Parameters*. R package version 0.3, URL <http://CRAN.R-project.org/package=Rchoice>.
- Sarrias M, Daziano R (2015). *gmn1: Multinomial Logit Models with Random Parameters*. R package version 1.0, URL <http://CRAN.R-project.org/package=gmn1>.
- StataCorp (2011). “STATA Statistical Software: Release 12.” *College Station, TX: StataCorp LP*.
- Train K (2000). “Halton Sequences for Mixed Logit.” *Department of Economics, University of California Berkeley*.
- Train K (2009). *Discrete Choice Methods with Simulation*. 2nd edition. Cambridge University Press.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. URL <http://www.stats.ox.ac.uk/pub/MASS4>.
- Winkelmann R, Boes S (2006). *Analysis of Microdata*. Springer Science & Business Media.
- Zeileis A (2006). “Object-Oriented Computation of Sandwich Estimators.” *Journal of Statistical Software*, **16**(i09). URL <http://www.jstatsoft.org/v16/i09>.
- 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
Department of City and Regional Planning
Cornell University
325 W. Sibley Hall
Ithaca NY 14853, United States of America
E-mail: msarrias86@gmail.com
URL: <http://www.msarrias.weebly.com>