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-section data. We implement simulated maximum likelihood methods for the estimation of the coefficients which can assume a variety of distributions such as the **mlogit** package does. 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.

1. Introduction

A growing number of empirical studies involve the assessment of influences on a choice among binary, count or ordered discrete alternatives. Those models are well known. However, one of the traditional modeling shortcomings is their inability to control for possible unobserved heterogeneity that may exist across individuals. For instance, one might assume that some variable, such as income or education, does not affect equally the utility of individuals, therefore there may be a deviation from the mean of the respective coefficient. Binary, count and ordered choice models with random parameters extension allows heterogeneity among individuals assuming some distribution on the parameters.

In this document we present the package **Rchoice** for program R. **Rchoice** is a package for estimating a variety of Ordered and Binary Choice Models with observed and unobserved heterogeneity in the coefficients. The estimation procedure is based on Maximum Simulated Likelihood (MSL) which allows to control for observed and unobserved heterogeneity in a very flexible way. [1] To our knowledge, only LIMDEP (Greene 2002) is able to estimate these type of models in a concise and flexible manner. Therefore, this package intends to make available these estimation methods to the general public and practitioners in a friendly and flexible way.

[1]For multinomial discrete choice models with random parameters see **mlogit** package in R (Croissant *et al.* 2012)

2. Maximum Simulated Likelihood Estimation

One of the major capability of the **Rchoice** package is to estimate Binary, Count and Ordered models with random parameters. They are estimated by **Rchoice** using Simulated Maximum Likelihood (SML). 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); Hajivassiliou and Ruud (1986); Train (2009); Cameron and Trivedi (2005)

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

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

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

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

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 $\widehat{f}(y|\mathbf{x}, \boldsymbol{\theta})$ of $f(y|\mathbf{x}, \boldsymbol{\theta})$ to form a likelihood function.

But, how can we obtain $\widehat{f}(y|\mathbf{x}, \boldsymbol{\theta})$? A good approximation can be obtained by **Monte Carlo integration**. [2] This procedure provides and alternative to deterministic numerical integration. Here we can *simulate* the integration using random draws from the distribution $g(\boldsymbol{\beta}|\boldsymbol{\theta})$. For 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:

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

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

$$\widehat{\boldsymbol{\theta}}_{MSL} \equiv \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\arg\max} \quad \sum_{i=1}^{N} \log \widehat{f}(y|\mathbf{x}_i, \boldsymbol{\beta}_{ir}, \boldsymbol{\theta})$$

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

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

Preposition 1 (Distribution of MSL 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:*

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

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

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

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

3. Technical Aspects of Maximum Simulated Likelihood

In this section, we show the general technical aspect of the MSL. This will allow us to accommodate any index-type regression model such as probit, logit, ordered and count 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:

$$f(y_i|\mathbf{x}_i, \boldsymbol{\beta}_{ir}, \boldsymbol{\theta}) = P_{ir}(\boldsymbol{\theta}) = P_{ir}$$

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

$$\log L_s = \sum_{i=1}^{N}\log\left(\frac{1}{R}\sum_{r=1}^{R}P_{ir}(\boldsymbol{\theta})\right) \tag{2}$$

If we define:

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

then, the maximum likelihood can be written as:

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

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_{ir}. As typical, the index model represents a latent process of the form:

$$U^*_{ir}(\boldsymbol{\theta}) = \mathbf{z}_i'\boldsymbol{\delta} + \mathbf{x}_i'\boldsymbol{\beta}_{ir} + \epsilon_i, \tag{4}$$

where $\mathbf{z}_i$ is a vector of variables with fixed parameters $\boldsymbol{\delta}$; $\mathbf{x}_i$ is a vector of variables with random coefficients $\boldsymbol{\beta}_{ir}$; and ϵ_i is the error term. For simplicity, assume that $\boldsymbol{\beta}_{ir} \sim N(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, 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. 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. 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 $\mathrm{Var}(\boldsymbol{\beta}_i) = E(\mathbf{L}\boldsymbol{\omega}(\boldsymbol{\omega})') = \mathbf{L}E(\boldsymbol{\omega}\boldsymbol{\omega}')\mathbf{L} = \mathbf{L}\mathbf{I}\mathbf{L} = \mathbf{L}\mathbf{L}' = \boldsymbol{\Sigma}$.

Example 1 (Representation of Correlated Random Parameters). *Suppose two correlated random parameters, β_1 and β_2, whose mean 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:

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

and the conditional mean vector is:

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

Rchoice allows to specify different distribution for the random parameters. See section 4.

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

$$P_{ir}(\boldsymbol{\theta}) = F(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) \tag{5}$$

where $q_i = 2y_i - 1$[3]. Furthermore, if the model is probit, then:

$$\begin{aligned} F(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) &= \Phi(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) \\ f(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) &= \phi(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) \end{aligned}$$

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

$$\begin{aligned} F(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) &= \Lambda(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) = \frac{\exp(q_i \cdot U^*_{ir}(\boldsymbol{\theta}))}{1+\exp(q_i \cdot U^*_{ir}(\boldsymbol{\theta}))} \\ f(q_i \cdot U^*_{ir}(\boldsymbol{\theta})) &= \Lambda(q_i \cdot U^*_{ir}(\boldsymbol{\theta}))\left[1-\Lambda(q_i \cdot U^*_{ir}(\boldsymbol{\theta}))\right] \end{aligned}$$

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

$$P_{ir}(\boldsymbol{\theta}) = \frac{\exp(-\exp(U^*_{ir}(\boldsymbol{\theta})))\exp(U^*_{ir}(\boldsymbol{\theta}))^{y_i}}{y_i!} \tag{6}$$

and for the ordered model, we have:

$$P_{ir}(\boldsymbol{\theta}) = F(\kappa_j - U^*_{ir}(\boldsymbol{\theta})) - F(\kappa_{j-1} - U^*_{ir}(\boldsymbol{\theta})) \tag{7}$$

where:

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

This last reparametrization ensure the ordering of the thresholds.

3.2. Gradient of the Simulated Maximum Likelihood

SML procedure is very time consuming. Providing the gradient to the maximization procedure can considerable reduce the time to achieve convergence. **Rchoice** package provides the

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

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:

$$\frac{\partial \log L_s}{\partial \boldsymbol{\theta}} = \sum_{i=1}\left(\frac{\frac{1}{R}\sum_{r=1}^{R}\left(\frac{\partial P_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)}{\frac{1}{R}\sum_{r=1}^{R}P_{ir}(\boldsymbol{\theta})}\right) \tag{8}$$

For the derivative term,

$$\begin{aligned}\frac{\partial P_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= P_{ir}\frac{\partial \log P_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\ &= P_{ir}(\boldsymbol{\theta})\mathbf{g}_{ir}(\boldsymbol{\theta}).\end{aligned} \tag{9}$$

where we use the fact that $\frac{\partial \log p}{\partial \theta} = \frac{1}{p}\frac{\partial p}{\partial \theta}$. Now, inserting 9 into 8 we get:

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

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,

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

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

$$\frac{\partial \log P_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{g}_{ir} = \lambda_{ir}(\boldsymbol{\theta})\left(\frac{\partial U^*_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right), \tag{12}$$

where:

$$\lambda_{ir} = \frac{q_i f(q_i U^*_{ir}(\boldsymbol{\theta}))}{F(q_i U^*_{ir}(\boldsymbol{\theta}))},$$

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

$$\frac{\partial \log P_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{g}_{ir} = (y_i - \exp(U^*_{ir}(\boldsymbol{\theta})))\left(\frac{\partial U^*_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right).$$

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

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

and by:

$$\frac{\partial \log P_{ir}(\boldsymbol{\theta})}{\partial \boldsymbol{\alpha}_k} = \frac{d_{j,k} f_{ir,j}(\boldsymbol{\theta}) - d_{j-1,k} f_{ir,j-1}(\boldsymbol{\theta})}{F_{ir,j}(\boldsymbol{\theta}) - F_{ir,j-1}(\boldsymbol{\theta})}\left(\frac{\partial \boldsymbol{\kappa}_j}{\partial \boldsymbol{\alpha}_k} - \frac{\partial \boldsymbol{\kappa}_{j-1}}{\partial \boldsymbol{\alpha}_k}\right)$$

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

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

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

4. Drawing from Densities

SML procedure requires to draw pseudo-random numbers from the specified distribution for simulation. A good performance of SML requires very large number of draws. 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**. [4]

[4]In terms of programming, we have modified the functions from **mlogit** (Croissant *et al.* 2012) to allow variation in the mean by observed covariates (Hierarchical model).

Suppose that there are K_2 random parameters. Then, the K_2 elements of $\boldsymbol{\omega}_{ir}$ are drawn as follows. We begin with a K_2 random vector $\boldsymbol{\omega}_{ir}$ that is:

- K_2 independent draws from the standard uniform $(0, 1)$ distribution or
- K_2 independent draws from the mth Halton sequence, where m is the mth prime number in the sequence of K_2 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, $\beta_{k,i} \sim N(\boldsymbol{\beta}_k, \sigma_k^2)$

2. Truncated normal Parameter:

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

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

3. Log-Normal Distribution:

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

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

4. Uniform:

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

where β_k and σ_k are estimated.

5. Triangular distribution:

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

where β_k and σ_k are estimated.

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 draw. 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))`.

Note that log-normal and truncated normal give positive coefficients only. If the user wants a variable to have only negative coefficients, he should create the negative of the variable.

5. Poisson Regression Model Examples

5.1. Standard Poisson Model

In 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).

In this section, we show the capabilities of **Rchoice** to estimate Poisson regression model with and without random parameters.

Rchoice is loaded by typing:

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

In order 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) suggest 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. In order to see this, we estimate a Poisson regression model use the `Rchoice` function specifying `link = "poisson"`:

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

##
## Starting values Parameters:
## (Intercept)         fem         mar        kid5         phd        ment
##     1.33425    -0.38089     0.26320    -0.29144    -0.01137     0.06149

summary(poisson)

##
## Model: poisson
## Model estimated on: Mon Mar 24 11:54:29 2014
##
## Call:
## Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
##     link = "poisson", method = "nr")
##
```

```
## The estimation took: 0h:0m:0s
##
## Coefficients:
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  0.30462    0.10298    2.96   0.0031 **
## fem         -0.22459    0.05461   -4.11  3.9e-05 ***
## mar          0.15524    0.06137    2.53   0.0114 *
## kid5        -0.18488    0.04013   -4.61  4.1e-06 ***
## phd          0.01282    0.02640    0.49   0.6271
## ment         0.02554    0.00201   12.73  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by Newton-Raphson maximisation
## Log Likelihood: -1650
## 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.

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**:

```
require(sandwich)
require(lmtest)
```

```
coeftest(poisson, vcov = sandwich)

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.30462    0.14652    2.08  0.03790 *
## fem         -0.22459    0.07166   -3.13  0.00178 **
## mar          0.15524    0.08193    1.89  0.05843 .
## kid5        -0.18488    0.05596   -3.30  0.00099 ***
```

```
## phd          0.01282    0.04196    0.31  0.76001
## ment         0.02554    0.00382    6.69  3.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In order 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|)
## (Intercept)  0.30462    0.14660    2.08   0.0380 *
## fem         -0.22459    0.07170   -3.13   0.0018 **
## mar          0.15524    0.08197    1.89   0.0586 .
## kid5        -0.18488    0.05599   -3.30   0.0010 ***
## phd          0.01282    0.04199    0.31   0.7601
## ment         0.02554    0.00382    6.69    4e-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 $H_0 : \texttt{phd/ment} = 0$ by:

```
require(car)
```

```
deltaMethod(poisson, "phd/ment")

##          Estimate    SE
## phd/ment    0.502 1.043
```

5.2. Random Parameter Poisson Model

Now, 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 among 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:

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

Then, in order to estimate this model, we can write:

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

##
## Starting values Parameters:
## (Intercept)         fem         mar   mean.kid5    mean.phd   mean.ment
##     0.30461    -0.22458     0.15526    -0.18489     0.01282     0.02554
##     sd.kid5      sd.phd     sd.ment
##     0.00000     0.00000     0.00000
```

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 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 a default, but if the user wants pseudo-random draws he can specify `haltons = NULL`. As explained before, the default maximization algorithm for SML is BGFS.

```
summary(poisson.ran)

##
## Model: poisson
## Model estimated on: Mon Mar 24 11:55:03 2014
##
## Call:
## Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
##     link = "poisson", ranp = c(kid5 = "n", phd = "n", ment = "n"),
##     method = "bfgs", iterlim = 2000)
##
## The estimation took: 0h:0m:34s
##
## Coefficients:
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  0.25256    0.13245    1.91  0.05654 .
## fem         -0.22629    0.07097   -3.19  0.00143 **
```

```
## mar          0.15765    0.07861    2.01  0.04490 *
## mean.kid5   -0.20753    0.06282   -3.30  0.00095 ***
## mean.phd    -0.03306    0.03664   -0.90  0.36702
## mean.ment    0.03000    0.00374    8.02  1.1e-15 ***
## sd.kid5      0.28760    0.08585    3.35  0.00081 ***
## sd.phd       0.15981    0.01711    9.34  < 2e-16 ***
## sd.ment      0.01455    0.00379    3.84  0.00013 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -1580
## Number of observations: 915
## Number of iterations: 67
## Exit of MLE: successful convergence
## Simulation based on 40 Halton draws
```

The result shows that indeed the three coefficients are random in the sample. We can also say that :

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

## mean.kid5
##    0.2353
```

a 23% of the individuals have a positive coefficient for `kid5`. 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 effect.

Suppose that now we want to test if $H_0 = \sigma_{\texttt{kid5}} = \sigma_{\texttt{phd}} = \sigma_{\texttt{ment}} = 0$. 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   203     <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   1577
## 2   6   1651 -3   148     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both test 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"))

##
## Starting values Parameters:
## (Intercept)         fem         mar   mean.kid5    mean.phd   mean.ment
##     0.30461    -0.22458     0.15526    -0.18489     0.01282     0.02554
##     sd.kid5      sd.phd     sd.ment
##     0.00000     0.00000     0.00000

summary(poisson.ran2)

##
## Model: poisson
## Model estimated on: Mon Mar 24 11:59:48 2014
##
## Call:
## Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
##     link = "poisson", ranp = c(kid5 = "u", phd = "t", ment = "cn"),
##     method = "bfgs", iterlim = 2000)
##
## The estimation took: 0h:4m:45s
##
## Coefficients:
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  0.22946    0.13160    1.74   0.0812 .
## fem         -0.20804    0.06947   -2.99   0.0027 **
## mar          0.14610    0.07723    1.89   0.0585 .
## mean.kid5   -0.34485    0.06644   -5.19  2.1e-07 ***
## mean.phd     0.04838    0.03567    1.36   0.1750
## mean.ment    0.02942    0.00398    7.39  1.5e-13 ***
## sd.kid5      0.13994    0.05802    2.41   0.0159 *
## sd.phd       0.22193    0.02212   10.03  < 2e-16 ***
## sd.ment      0.02036    0.00377    5.41  6.4e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -1570
## Number of observations: 915
## Number of iterations: 307
## Exit of MLE: successful convergence
## Simulation based on 40 Halton draws
```

Now, we estimate the model `poisson.ran`, but assuming that the random parameters are correlated:

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

##
## Starting values Parameters:
##  (Intercept)          fem          mar    mean.kid5     mean.phd
##      0.30461     -0.22458      0.15526     -0.18489      0.01282
##    mean.ment sd.kid5.kid5  sd.kid5.phd sd.kid5.ment  sd.phd.phd
##      0.02554      0.00000      0.00000      0.00000     0.00000
##  sd.phd.ment sd.ment.ment
##      0.00000      0.00000

summary(poissonc.ran)

##
## Model: poisson
## Model estimated on: Mon Mar 24 12:01:32 2014
##
## Call:
## Rchoice(formula = art ~ fem + mar + kid5 + phd + ment, data = Articles,
##     link = "poisson", ranp = c(kid5 = "n", phd = "n", ment = "n"),
##     correlation = TRUE, method = "bfgs", iterlim = 2000)
##
## The estimation took: 0h:1m:43s
##
## Coefficients:
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)   0.24230    0.13035    1.86   0.0630 .
## fem          -0.20528    0.07018   -2.92   0.0034 **
## mar           0.15420    0.07799    1.98   0.0480 *
## mean.kid5    -0.21050    0.06134   -3.43   0.0006 ***
## mean.phd     -0.03418    0.03762   -0.91   0.3637
```

```
## mean.ment     0.03095    0.00380    8.15  4.4e-16 ***
## sd.kid5.kid5  0.13641    0.05909    2.31   0.0210 *
## sd.kid5.phd  -0.16717    0.02158   -7.75  9.5e-15 ***
## sd.kid5.ment  0.00757    0.00350    2.16   0.0306 *
## sd.phd.phd    0.01105    0.05813    0.19   0.8493
## sd.phd.ment  -0.00681    0.00805   -0.85   0.3978
## sd.ment.ment  0.01809    0.00332    5.45  5.1e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -1570
## Number of observations: 915
## Number of iterations: 160
## Exit of MLE: successful convergence
## Simulation based on 40 Halton draws
```

We can extract the $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}'$ 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  0.018607 -0.02280  0.0010323
## phd  -0.022803  0.02807 -0.0013403
## ment  0.001032 -0.00134  0.0004308

cor.Rchoice(poissonc.ran)

##         kid5     phd    ment
## kid5  1.0000 -0.9978  0.3646
## phd  -0.9978  1.0000 -0.3854
## ment  0.3646 -0.3854  1.0000
```

5.3. 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 `kid5`, `phd` and `ment`, but also observed heterogeneity in the mean. Specifically, we assume that:

$$
\begin{aligned}
\beta_{\texttt{kid5},i} &= \beta_{\texttt{kid5}} + \pi_1 \texttt{fem} + \sigma_{\texttt{kid5,kid5}}\omega_{\texttt{kid5},ir} \\
\beta_{\texttt{phd},i} &= \beta_{\texttt{phd}} + \pi_2 \texttt{fem} + \sigma_{\texttt{phd,kid5}}\omega_{\texttt{kid5},ir} + \sigma_{\texttt{phd,phd}}\omega_{\texttt{phd},ir} \\
\beta_{\texttt{ment},i} &= \beta_{\texttt{ment}} + \pi_3 \texttt{fem} + \sigma_{\texttt{ment,kid5}}\omega_{\texttt{kid5},ir} + \sigma_{\texttt{ment,phd}}\omega_{\texttt{phd},ir} + \sigma_{\texttt{ment,ment}}\omega_{\texttt{ment},ir}
\end{aligned}
$$

The formulation above implies that those three coefficients (or marginal effect on latent productivity) varies also if the individual is female or male. **Rchoice** manages the variables in the hierarchical model by the `formula` object: all the hierarchical variables are included after the | symbol. For example, we can estimate this model by typing:

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

## Warning:  Model assumes no constant in S variables...updating formula

##
## Starting values Parameters:
##   (Intercept)          fem          mar    mean.kid5     mean.phd
##       0.30461     -0.22458      0.15526     -0.18489      0.01282
##     mean.ment     kid5.fem      phd.fem     ment.fem sd.kid5.kid5
##       0.02554      0.00000      0.00000      0.00000      0.00000
##   sd.kid5.phd sd.kid5.ment   sd.phd.phd  sd.phd.ment sd.ment.ment
##       0.00000      0.00000      0.00000      0.00000      0.00000

summary(poissonH.ran)

##
## Model: poisson
## Model estimated on: Mon Mar 24 12:04:31 2014
##
## Call:
## Rchoice(formula = art ~ fem + mar + kid5 + phd + ment | fem,
##     data = Articles, link = "poisson", ranp = c(kid5 = "n", phd = "n",
##         ment = "n"), correlation = TRUE, method = "bfgs", iterlim = 2000)
##
## The estimation took: 0h:2m:59s
##
## Coefficients:
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)   0.35169    0.16573    2.12   0.0338 *
## fem          -0.56467    0.22595   -2.50   0.0125 *
## mar           0.13769    0.08021    1.72   0.0860 .
## mean.kid5    -0.22231    0.06952   -3.20   0.0014 **
## mean.phd     -0.07921    0.04844   -1.64   0.1020
## mean.ment     0.03709    0.00502    7.39  1.5e-13 ***
## kid5.fem     -0.02392    0.12058   -0.20   0.8428
## phd.fem       0.14298    0.07306    1.96   0.0504 .
## ment.fem     -0.00979    0.00930   -1.05   0.2924
## sd.kid5.kid5  0.40177    0.08427    4.77  1.9e-06 ***
```

```
## sd.kid5.phd   0.02284    0.04205    0.54   0.5871
## sd.kid5.ment  0.01065    0.00627    1.70   0.0895 .
## sd.phd.phd    0.17964    0.02476    7.25  4.0e-13 ***
## sd.phd.ment   0.01913    0.00639    2.99   0.0027 **
## sd.ment.ment  0.02227    0.00452    4.93  8.3e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -1570
## Number of observations: 915
## Number of iterations: 205
## Exit of MLE: successful convergence
## Simulation based on 40 Halton draws
```

The estimated parameters indicates that gender matters only for **phd** mean coefficient. We can test if the interaction variables are jointly significant by using **lrtest**:

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

## Likelihood ratio test
##
## Model 1: art ~ fem + mar + kid5 + phd + ment | fem - 1
## Model 2: art ~ fem + mar + kid5 + phd + ment
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1  15   1568
## 2  12   1572 -3  6.98      0.073 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.4. 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:

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

where:

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

We can also estimate the standard deviation of this distribution by estimating:

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

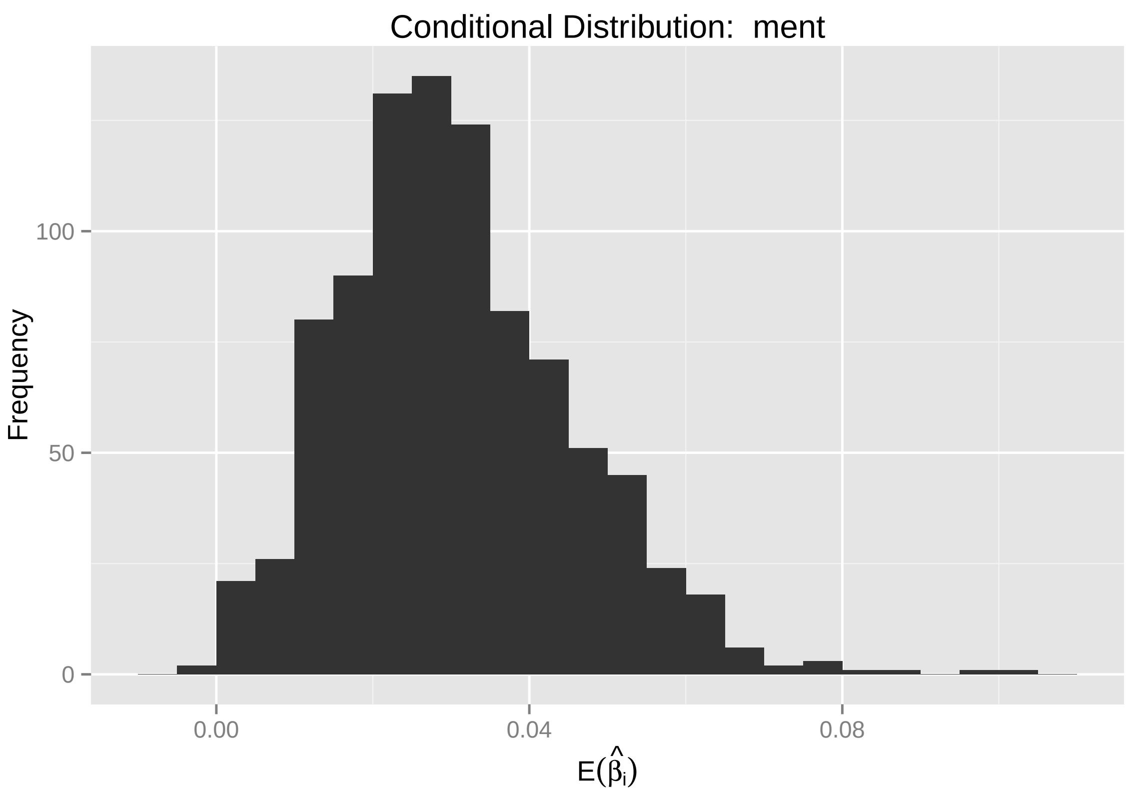
then computing the square root of the estimated variance,

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

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 $\boldsymbol{\beta}_i$ (Greene 2012).

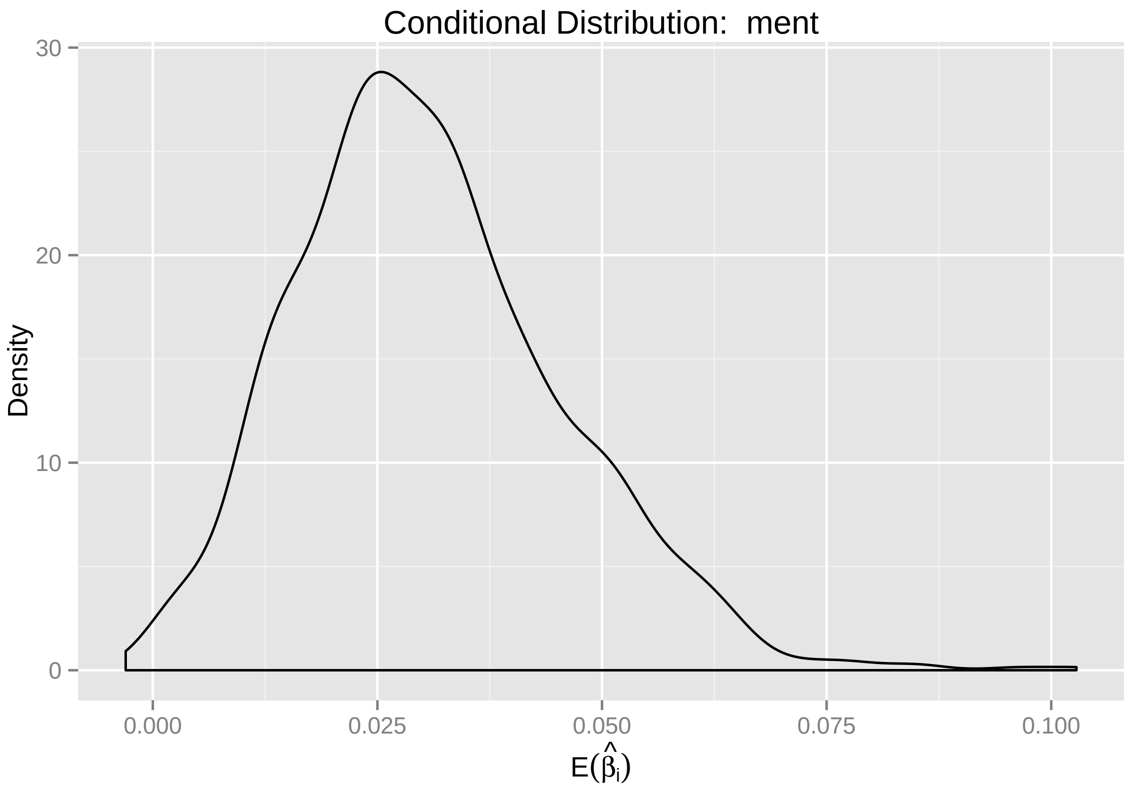
Rchoice allows to plot the histogram and kernel density of conditional means of random parameters using the function `plot`. For the histogram of the conditional mean of $\beta_{\texttt{ment},i}$, we can write:

```
plot(poissonH.ran, par = "ment", type = "histogram", bin = 0.005)
```



and for the kernel density:

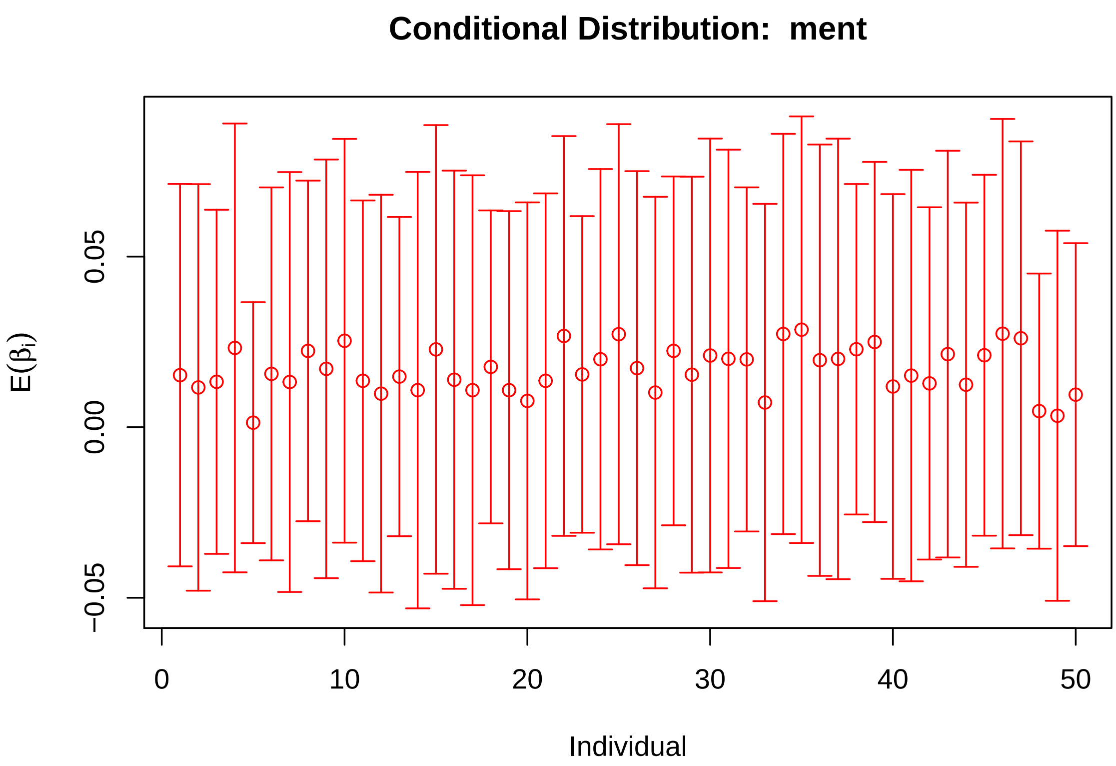
```
plot(poissonH.ran, par = "ment")
```



As Greene (2012) points out, even if the analysis departs from normal marginal distributions $\boldsymbol{\beta}_i$, 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 50 individuals:

```
plot(poissonH.ran, par = "ment", ind = TRUE, id = seq(1, 50, 1))
```



6. Binary and Ordered Model Examples

In this section, we show how to estimate Binary and Ordered (Probit or Logit) models. Since the main characteristics of **Rchoice** package to estimate random parameter models were shown in the previous section, in this section we briefly show how to use **Rchoice** for Binary and Ordered models with random parameters.

MSL procedure is very complex. Sometimes, the model may not converge, or it may converge to a local maximum, if the initial values are not chosen well enough. To illustrate this, we will estimate a Probit model with random parameters using `"Workroz"` database. In order to see the maximization process, we use the argument `print.level`.

```
data("Workmroz")
```

```
probit.ran <- Rchoice(lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
                      ranp = c(k5 = "n", hc = "n"),
                      link = "probit",
                      print.level = 1,
                      data = Workmroz,
                      R = 100)
```

```
##
## Starting values Parameters:
## (Intercept)        k618         age          wc         lwg         inc
##     1.91842    -0.03859    -0.03782     0.48831     0.36563    -0.02053
##     mean.k5     mean.hc       sd.k5       sd.hc
##    -0.87471     0.05717     0.00000     0.00000
## initial  value 452.694964
## iter   1 value 452.694959
## final  value 452.694959
## converged
```

```
summary(probit.ran)
```

```
## Warning:  NaNs produced
```

```
##
## Model: probit
## Model estimated on: Mon Mar 24 12:04:44 2014
##
## Call:
## Rchoice(formula = lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
##     data = Workmroz, link = "probit", ranp = c(k5 = "n", hc = "n"),
##     R = 100, print.level = 1, method = "bfgs", iterlim = 2000)
##
##
## Frecuencies of categories:
## y
##     0     1
## 0.432 0.568
## The estimation took: 0h:0m:12s
##
## Coefficients:
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  1.92e+00   3.81e-01    5.04  4.7e-07 ***
## k618        -3.86e-02   4.05e-02   -0.95  0.34048
## age         -3.78e-02   7.61e-03   -4.97  6.7e-07 ***
## wc           4.88e-01   1.35e-01    3.60  0.00031 ***
## lwg          3.66e-01   8.78e-02    4.17  3.1e-05 ***
## inc         -2.05e-02   4.78e-03   -4.30  1.7e-05 ***
## mean.k5     -8.75e-01   1.14e-01   -7.70  1.3e-14 ***
## mean.hc      5.72e-02   1.24e-01    0.46  0.64476
## sd.k5        1.40e-05         NA      NA       NA
## sd.hc        4.16e-05         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -453
## Number of observations: 753
## Number of iterations: 7
## Exit of MLE: successful convergence
## Simulation based on 100 Halton draws
```

The output for the maximization shows that the model converges with 1 iteration, but some `NaNs` are produced when trying to get the standard errors. In order to avoid this problem, the user may provide different initial values using the argument `start`. [5] For example, if we set the initial values all zeros, we get:

```
probit2.ran <- update(probit.ran, start = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0))

##
## Starting values Parameters:
## (Intercept)        k618         age          wc         lwg         inc
##           0           0           0           0           0           0
##     mean.k5     mean.hc       sd.k5       sd.hc
##           0           0           0           0
## initial  value 521.939827
## iter   2 value 518.054688
## iter   3 value 515.033512
## iter   4 value 496.268521
## iter   5 value 484.357418
## iter   6 value 479.612746
## iter   7 value 471.905669
## iter   8 value 468.301470
## iter   9 value 465.997533
## iter  10 value 455.627889
## iter  11 value 453.348569
## iter  12 value 451.824880
## iter  13 value 450.731168
## iter  14 value 450.061913
## iter  15 value 449.684333
## iter  16 value 449.587915
## iter  17 value 449.223047
## iter  18 value 449.125297
## iter  19 value 449.077641
## iter  20 value 449.065350
## iter  21 value 449.062475
## iter  22 value 449.061079
## iter  23 value 449.061049
## iter  24 value 449.061007
## iter  25 value 449.060512
```

[5] As default, the initial values correspond to those for a Probit model.

```
## iter  26 value 449.060264
## iter  27 value 449.059889
## iter  28 value 449.059619
## iter  29 value 449.059456
## iter  30 value 449.059416
## iter  31 value 449.059366
## iter  32 value 449.059280
## iter  33 value 449.059129
## iter  34 value 449.058979
## iter  34 value 449.058976
## iter  34 value 449.058976
## final  value 449.058976
## converged
```

```
summary(probit2.ran)
```

```
##
## Model: probit
## Model estimated on: Mon Mar 24 12:05:38 2014
##
## Call:
## Rchoice(formula = lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
##     data = Workmroz, start = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
##     link = "probit", ranp = c(k5 = "n", hc = "n"), R = 100, print.level = 1,
##     method = "bfgs", iterlim = 2000)
##
##
## Frecuencies of categories:
## y
##     0     1
## 0.432 0.568
## The estimation took: 0h:0m:54s
##
## Coefficients:
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  2.66341    0.53722    4.96  7.1e-07 ***
## k618        -0.08032    0.05566   -1.44  0.14905
## age         -0.05407    0.01090   -4.96  7.1e-07 ***
## wc           0.68483    0.22517    3.04  0.00236 **
## lwg          0.51134    0.12408    4.12  3.8e-05 ***
## inc         -0.02688    0.00683   -3.94  8.2e-05 ***
## mean.k5     -1.58653    0.47126   -3.37  0.00076 ***
## mean.hc      0.31753    0.25560    1.24  0.21413
## sd.k5        1.59634    0.94812    1.68  0.09224 .
## sd.hc        1.51435    0.56342    2.69  0.00719 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -449
## Number of observations: 753
## Number of iterations: 115
## Exit of MLE: successful convergence
## Simulation based on 100 Halton draws
```

In order to analyze ordered models, we use the `"Health"` database:

```
data("Health")
```

We estimate a Random Parameter Ordered Probit model for the variable `newhsat` for year 1988 (see `help(Health)`)

```
oprobit.ran<-Rchoice(newhsat ~ age + educ + hhinc + married + hhkids,
                      data = Health, link = "ordered probit",
                      subset = year == 1988,
                      ranp = c(age = "n", hhinc = "n"), print.level=1,
                      start = rep(0,11))

##
## Starting values Parameters:
##     kappa.1     kappa.2     kappa.3 (Intercept)        educ     married
##           0           0           0           0           0           0
##      hhkids    mean.age  mean.hhinc      sd.age    sd.hhinc
##           0           0           0           0           0
## initial  value 10456.167937
## iter   2 value 10323.220929
## iter   3 value 8133.920402
## iter   4 value 7925.069572
## iter   5 value 7811.151177
## iter   6 value 7668.311141
## iter   7 value 6510.547046
## iter   8 value 6446.957898
## iter   9 value 6428.178848
## iter  10 value 6410.795476
## iter  11 value 6356.885833
## iter  12 value 6326.116873
## iter  13 value 6088.976699
## iter  14 value 5963.933089
## iter  15 value 5958.387238
## iter  16 value 5896.992361
## iter  17 value 5884.078367
## iter  18 value 5874.994362
```

```
## iter  19 value 5868.967744
## iter  20 value 5861.629593
## iter  21 value 5856.574575
## iter  22 value 5852.521990
## iter  23 value 5849.886503
## iter  24 value 5844.727724
## iter  25 value 5842.625367
## iter  26 value 5842.137855
## iter  27 value 5840.192156
## iter  28 value 5840.149051
## iter  29 value 5825.140189
## iter  30 value 5824.344202
## iter  31 value 5821.540175
## iter  32 value 5820.300356
## iter  33 value 5819.638570
## iter  34 value 5817.554181
## iter  35 value 5816.434034
## iter  36 value 5792.657668
## iter  37 value 5767.920917
## iter  38 value 5752.499507
## iter  39 value 5749.729388
## iter  40 value 5748.830598
## iter  41 value 5748.418433
## iter  42 value 5748.193528
## iter  43 value 5748.033235
## iter  44 value 5748.021204
## iter  45 value 5747.998675
## iter  46 value 5747.997704
## iter  47 value 5747.997120
## iter  47 value 5747.997118
## final  value 5747.997118
## converged

summary(oprobit.ran)

##
## Model: ordered probit
## Model estimated on: Mon Mar 24 12:12:41 2014
##
## Call:
## Rchoice(formula = newhsat ~ age + educ + hhinc + married + hhkids,
##     data = Health, subset = year == 1988, start = rep(0, 11),
##     link = "ordered probit", ranp = c(age = "n", hhinc = "n"),
##     print.level = 1, method = "bfgs", iterlim = 2000)
##
##
```

```
## Frecuencies of categories:
## y
##      0      1      2      3      4
## 0.0513 0.2483 0.4965 0.1115 0.0923
## The estimation took: 0h:7m:3s
##
## Coefficients:
##              Estimate Std. Error t-value Pr(>|t|)
## kappa.1      1.27e+00   5.53e-02   22.90  < 2e-16 ***
## kappa.2      2.78e+00   9.46e-02   29.43  < 2e-16 ***
## kappa.3      3.33e+00   1.09e-01   30.55  < 2e-16 ***
## (Intercept)  2.17e+00   1.47e-01   14.78  < 2e-16 ***
## educ         3.92e-02   7.82e-03    5.00  5.6e-07 ***
## married     -3.01e-02   4.57e-02   -0.66    0.510
## hhkids       6.43e-02   4.13e-02    1.56    0.119
## mean.age    -2.00e-02   1.90e-03  -10.50  < 2e-16 ***
## mean.hhinc   2.74e-05   1.13e-05    2.42    0.015 *
## sd.age       9.61e-03   1.86e-03    5.17  2.4e-07 ***
## sd.hhinc     4.11e-06   2.23e-05    0.18    0.854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Optimization of log-likelihood by BFGS maximisation
## Log Likelihood: -5750
## Number of observations: 4483
## Number of iterations: 214
## Exit of MLE: successful convergence
## Simulation based on 40 Halton draws
```

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.

Cameron AC, Trivedi PK (2005). *Microeconometrics: methods and applications*. Cambridge university press.

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.

Greene WH (2002). *LIMDEP: Version 8.0: Econometric Modeling Guide*. Econometric Software.

Greene WH (2012). *Econometric analysis*. 7 edition. Prentice Hall. ISBN 0131395386.

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.

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.

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
Cornell University
Universidad Catolica del Norte
E-mail: msarrias86@gmail.com