Multinomial Logit Models with Continuous and Discrete Individual Heterogeneity in **R**: The gmnl Package

Mauricio Sarrias
Universidad Católica del Norte

Ricardo A. Daziano
Cornell University

Abstract

This paper introduces the package **gmnl** in R for estimation of multinomial logit models with unobserved heterogeneity across individuals for cross-sectional and panel (longitudinal) data. Unobserved heterogeneity is modeled by allowing the parameters to vary randomly over individuals according to a continuous, discrete, or discrete-continuous mixture distribution, which must be chosen a priori by the researcher. In particular, the models supported by **gmnl** are the multinomial or conditional logit, the mixed multinomial logit, the scale heterogeneity multinomial logit, the generalized multinomial logit, the latent class logit, and the mixed-mixed multinomial logit. These models are estimated using either the maximum likelihood estimator or the maximum simulated likelihood estimator. This article describes and illustrates with real databases all functionalities of **gmnl**, including the derivation of individual conditional estimates of both the random parameters and willingness-to-pay measures.

Keywords: latent class, mixed multinomial logit, random parameters, preference heterogeneity, R.

1. Introduction

Modeling individual choices has been a very important avenue of research in diverse fields such as marketing, transportation, political science, and environmental, health, and urban economics. In all these areas the most widely used method to model choice among mutually exclusive alternatives has been the conditional or multinomial logit model (MNL, McFadden 1974), which belongs to the family of random utility maximization (RUM) models. The main advantage of the MNL model has been its simplicity in terms of both estimation and interpretation of the resulting choice probabilities and elasticities. On the one hand, the MNL

has a closed-form choice probability and a likelihood function that is globally concave (for a complete overview of MNL and its properties, see Train 2009). MNL estimation is thus straightforward using the maximum likelihood estimator (MLE). On the other hand, it has been recognized that MNL not only imposes constant competition across alternatives – as a consequence of the independence of irrelevant alternatives (IIA) property – but also lacks the flexibility to allow for individual-specific preferences.

With the advent of more powerful computers and the improvement of simulation-aided inference in the last decades, researchers are no longer constrained to use models with closed-form solutions that may lead to unrealistic behavioral specifications. In fact, much of recent work in choice modeling focuses on extending MNL to allow for random-parameter models that accommodate unobserved preference heterogeneity.

The most popular MNL extension is the mixed logit model (MIXL). MIXL allows parameters to vary randomly over individuals by assuming some continuous heterogeneity distribution a priori while keeping the MNL assumption that the error term is independent and identically distributed (i.i.d) extreme value type 1 (McFadden and Train 2000; Train 2009; Hensher and Greene 2003). MIXL is a very flexible model (MIXL properties are discussed in McFadden and Train 2000) that does not exhibit pure IIA substitution (cf. Dotson, Brazell, Howell, Lenk, Otter, MacEachern, and Allenby 2015). Furthermore, using the parametric heterogeneity distribution that describes how preferences vary in the population it is possible to derive conditional estimates of the parameters at the individual-level (Train 2009).

Below we present a brief example to introduce the idea of unobserved heterogeneity across individuals, and how **gmnl** works. This example uses microdata about individual choice among four transportation modes: air, train, bus and car. Both the data itself and its correct formatting – which follows the **mlogit** data frame – as well as the **gmnl** syntax are explained further in the following sections.

```
R> library("gmnl")
R> data("TravelMode", package = "AER")
R> library("mlogit")
R> TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
+    alt.levels = c("air", "train", "bus", "car"))
R> mixl <- gmnl(choice ~ vcost + travel + wait | 1,
+    data = TM, model = "mixl", ranp  = c(travel = "n"), R = 50)
```

Estimating MIXL model

In the estimated model, `choice` is a discrete (multinomial response) dependent variable that indicates which of the four alternative modes was actually chosen by each individual. There are three explanatory variables that have values that are alternative-specific (`vcost`: in-vehicle cost, `travel`: in-vehicle time, and `wait`: waiting time). The model also includes alternative-specific constants (`| 1`). Random parameters are considered just for the in-vehicle time component, making this example an application of a mixed logit model (`model = "mixl"`). The 'marginal disutility' of travel time[1] is assumed to be *normally distributed* in the line

---

[1]In a discrete choice model, the parameters of interest are interpreted as marginal utilities, as choice is modeled as selecting the alternative that maximizes utility, which is a function of the explanatory variables. Some attributes cause a reduction in utility: travelers desire to reduce travel time, and hence its associated parameter is expected to be negative.

of code that identifies the random parameters (`ranp = c(travel = "n")`). The assumption of normality is an example of representing variation in preferences according to a parametric continuous distribution. All other parameters (the alternative-specific constants, `wait` and `vcost`) are assumed fixed. Finally, `R = 50` evaluation points are used to simulate the likelihood function.

```
R> summary(mixl)
```

```
Model estimated on: Thu Feb 18 10:49:43 2016


Call:
gmnl(formula = choice ~ vcost + travel + wait | 1, data = TM,
    model = "mixl", ranp = c(travel = "n"), R = 50, method = "bfgs")


Frequencies of categories:

  air train   bus   car
0.276 0.300 0.143 0.281


The estimation took: 0h:0m:10s


Coefficients:
                  Estimate Std. Error z-value Pr(>|z|)
train:(intercept)  0.36131    1.01014    0.36  0.72058
bus:(intercept)   -0.44685    1.03936   -0.43  0.66725
car:(intercept)   -4.90453    1.09319   -4.49  7.2e-06 ***
vcost             -0.02611    0.00954   -2.74  0.00621 **
wait              -0.11642    0.01484   -7.84  4.4e-15 ***
travel            -0.00808    0.00236   -3.42  0.00062 ***
sd.travel          0.00531    0.00204    2.60  0.00927 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Optimization of log-likelihood by BFGS maximization
Log Likelihood: -189
Number of observations: 210
Number of iterations: 156
Exit of MLE: successful convergence
Simulation based on 50 draws
```

The output of calling the `gmnl` function includes the point estimates, standard errors, z values, and p values of the parameters of each explanatory variable. Note that for the normally distributed parameter, both the population mean and standard deviation are estimated: `travel = -0.008` is the point estimate of the mean and `sd.travel = 0.005` is the point estimate of the standard deviation of the normal distribution that represents how preferences for in-vehicle travel time reductions are distributed in the population. Note that, we can infer the profile of individuals' preferences for travel time by looking at the shape of the parameter, which is distributed in this sample as $N(-0.008, 0.005^2)$.

In addition to MIXL, there are other models that can represent unobserved preference heterogeneity. Latent class (LC) discrete choice models offer an alternative to MIXL by replacing the continuous distribution assumption with a discrete distribution in which preference heterogeneity is captured by membership in distinct classes or segments (Boxall and Adamowicz 2002; Greene and Hensher 2003; Shen 2009). The standard LC specification with class-specific multinomial logit models of choice (LC-MNL) is useful if the assumption of preference homogeneity holds within segments. In effect, in an LC-MNL all individuals in a given class have the same parameters (fixed parameters within a class), but the parameters vary across classes (heterogeneity across classes).

Rossi, Allenby, and McCulloch (2005) in a Bayesian setting, and more recently Bujosa, Riera, and Hicks (2010) and Greene and Hensher (2013) in the frequentist context, have derived models with discrete-continuous mixing distributions of unobserved heterogeneity in the form of a (finite) mixture of normals. This model is also known as mixed-mixed logit (MM-MNL, Keane and Wasi 2013).

Other researchers have focused on MNL extensions that allow for a more flexible representation of heteroskedasticity. For example, Fiebig, Keane, Louviere, and Wasi (2010) proposed two new models, namely the scale heterogeneity (S-MNL) model and the generalized multinomial logit (G-MNL) model. S-MNL extends the MNL by letting the scale of errors vary across individuals (via a parametric specification of heteroskedasticity), whereas the G-MNL nests the S-MNL, MIXL, and MNL models. For a discussion of confounding effects between scale and preference heterogeneity, see Hess and Rose (2012) and Hess and Stathopoulos (2013).

In addition to **gmnl**, there exist different packages in R (R Core Team 2017) that estimate models with multinomial responses. Some packages for estimating multinomial logit models with fixed parameters are **mlogit** (Croissant 2013), **RSGHB** (Dumont, Keller, and Carpenter 2015), **mnlogit** (Hasan, Wang, and Mahani 2016), the function `multinom` function from the **nnet** package (Venables and Ripley 2002), **VGAM** (Yee 2010) **bayesm** (Rossi 2015), and **MCMCpack** (Martin, Quinn, and Park 2011). The multinomial probit (MNP) model is fitted in **bayesm**, **MNP** (Imai and Van Dyk 2005) and **mlogit**. Models with random parameters are supported by **mlogit**, **mclogit**[2] (Elff 2016), **bayesm**, **ChoiceModelR**[3] (Sermas 2012) and **RSGHB**. In terms of models with latent classes **RSGHB**, **flexmix** (Leisch 2004), and **poLCA** (Linzer and Lewis 2011) offer alternative estimation procedures. **gmnl** is the only package to date that handles G-MNL specifications, and the only package that implements the maximum simulated likelihood estimator for MM-MNL. MM-MNL models are also implemented using a Bayes estimator in **bayesm**. However, **gmnl** allows the researcher to specify covariates that explain assignment to classes, whereas **bayesm** assumes constant weights for the continuous components of the double mixture.[4]

Among all the packages mentioned above, **mlogit** is one of the most complete and user-friendly R packages for the estimation of models with multinomial responses. For this reason, we have adopted most of the **mlogit** syntax in **gmnl**. Table 1 presents a more complete overview of the models supported by each package and the estimation procedure used to estimate the parameters.

---

[2]Only random intercepts are allowed in the current version.

[3]Both **bayesm** and **ChoiceModelR** only allow normally distributed parameters.

[4]One of the advantages of the Bayes estimator and its implementation in **bayesm** is the possibility of letting the number of discrete components of the double mixture free. This is achieved by assuming a Dirichlet process prior for the heterogeneity distribution.

Model	Package	Estimation procedure
MNL	**gmnl**	Maximum likelihood
	mclogit	Maximum likelihood
	mlogit	Maximum likelihood
	mnlogit	Maximum likelihood
	`multinom` (**nnet**)	Maximum likelihood
	VGAM	Maximum likelihood
	bayesm	Bayesian inference
	MCMCpack	Bayesian inference
	RSGHB	Bayesian inference
MNP	**mlogit**	Maximum simulated likelihood
	bayesm	Bayesian inference
	MNP	Bayesian inference
MIXL	**gmnl**	Maximum simulated likelihood
	mlogit	Maximum simulated likelihood
	mclogit	Penalized quasi-likelihood
	bayesm	Bayesian inference
	ChoiceModelR	Bayesian inference
	RSGHB	Bayesian inference
G-MNL	**gmnl**	Maximum simulated likelihood
S-MNL	**gmnl**	Maximum simulated likelihood
LC-MNL	**gmnl**	Maximum likelihood
	flexmix	Expectation-Maximization
	poLCA	Expectation-Maximization
	RSGHB	Bayesian inference
MM-MNL	**gmnl**	Maximum simulated likelihood
	bayesm	Bayesian inference

Table 1: Packages available in R for models with multinomial response.

The **gmnl** package (Sarrias and Daziano 2017) is intended to consolidate in a single R package the whole range of discrete choice models with random parameters for the use of researchers and practitioners. It shares similar functionalities with **mlogit** and **mnlogit** in terms of the formula interface. Furthermore, since **gmnl** is able to estimate G-MNL models, it also allows the user to estimate models in willingness-to-pay space with a minimal extra reformulation. Our package also provides the ability of constructing the conditional estimates for the individual parameters and willingness-to-pay. **gmnl** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=gmnl`.

The paper is organized as follows: Section 2 presents a brief overview of the models supported by **gmnl**. Section 3 discusses the functionalities of the package. Section 4 explains some computational issues that may arise when estimating random parameter models using maximum simulated likelihood. Finally, Section 5 concludes the paper.

2. Models

2.1. Mixed and latent class logit models

MIXL generalizes the MNL model by allowing the preference or taste parameters to be different for each individual (McFadden and Train 2000; Train 2009). MIXL is basically a random parameter logit model with continuous heterogeneity distributions. The random utility of person i for alternative j and for choice occasion t is:

$$U_{ijt} = \mathbf{x}_{ijt}^{\top}\boldsymbol{\beta}_i + \epsilon_{ijt}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, J, \quad t = 1, \ldots, T_i, \tag{1}$$

where $\mathbf{x}_{ijt}$ is a $K \times 1$ vector of observed alternative attributes; ϵ_{ijt} is the idiosyncratic error term or taste shock, and is i.i.d. extreme value type 1; the parameter vector $\boldsymbol{\beta}_i$ is unobserved for each i and is assumed to vary in the population following the continuous density $f(\boldsymbol{\beta}_i|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of this distribution. This mixing distribution can in principle take any shape. For example, when assuming that the parameters are distributed multivariate normal, $\boldsymbol{\beta}_i \sim \mathrm{MVN}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, the vector $\boldsymbol{\beta}_i$ can be re-written as:

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \mathbf{L}\boldsymbol{\eta}_i,$$

where $\boldsymbol{\eta}_i \sim N(\mathbf{0}, \mathbf{I})$, and $\mathbf{L}$ is the lower-triangular Cholesky factor of $\boldsymbol{\Sigma}$ such that $\mathbf{L}\mathbf{L}^{\top} = \mathrm{VAR}(\boldsymbol{\beta}_i) = \boldsymbol{\Sigma}$. If the off-diagonal elements of $\mathbf{L}$ are zero, then the parameters are independently normally distributed. Observed heterogeneity (deterministic taste variations) can also be accommodated in the random parameters by including individual-specific covariates (see for example Greene 2012). Specifically, the vector of random coefficients is:

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \boldsymbol{\Pi}\mathbf{z}_i + \mathbf{L}\boldsymbol{\eta}_i, \tag{2}$$

where $\mathbf{z}_i$ is a set of M characteristics of individual i that influence the mean of the preference parameters; and $\boldsymbol{\Pi}$ is a $K \times M$ is a matrix of additional parameters.

Unlike the MIXL model, LC uses a discrete mixing distribution, where individual i belongs to class q with probability w_{iq}, i.e.,:

$$\boldsymbol{\beta}_i = \boldsymbol{\beta}_q \text{ with probability } w_{iq} \text{ for } q = 1, \ldots, Q,$$

where $\sum_q w_{iq} = 1$ and $w_{iq} > 0$. The discrete mixing distribution (or class assignment probability) is unknown to the analyst. The most widely used formulation for w_{iq} is the semi-parametric multinomial logit format (Greene and Hensher 2003; Shen 2009):

$$w_{iq} = \frac{\exp\left(\mathbf{h}_i^{\top}\boldsymbol{\gamma}_q\right)}{\sum_{q=1}^{Q}\exp\left(\mathbf{h}_i^{\top}\boldsymbol{\gamma}_q\right)}, \quad q = 1, \ldots, Q, \quad \boldsymbol{\gamma}_1 = \mathbf{0},$$

where $\mathbf{h}_i$ denotes a set of socio-economic characteristics that determine assignment to classes. The parameters of the first class are normalized to zero for identification of the model. Note that one could omit any socio-economic covariate as a determinant of the class assignment probability. Under this scenario, the class probabilities simply become:

$$w_{iq} = \frac{\exp\left(\gamma_q\right)}{\sum_{q=1}^{Q}\exp\left(\gamma_q\right)}, \quad q = 1, \ldots, Q, \quad \gamma_1 = 0,$$

where γ_q is a constant (Scarpa and Thiene 2005).

Let $y_{ijt} = 1$ if individual i chooses j on occasion t, and 0 otherwise. Then, the unconditional probabilities of the sequence of choices by individual i for MIXL and LC are respectively given by:

$$P_i(\boldsymbol{\theta}) = \int \left\{ \prod_t^T \prod_j^J \left[\frac{\exp\left(\mathbf{x}_{ijt}^\top \boldsymbol{\beta}_i\right)}{\sum_{j=1}^J \exp\left(\mathbf{x}_{ijt}^\top \boldsymbol{\beta}_i\right)} \right]^{y_{ijt}} \right\} f(\boldsymbol{\beta}_i) d\boldsymbol{\beta}_i,$$

$$P_i(\boldsymbol{\theta}) = \sum_q^Q w_{iq} \left\{ \prod_t^T \prod_j^J \left[\frac{\exp\left(\mathbf{x}_{ijt}^\top \boldsymbol{\beta}_q\right)}{\sum_{j=1}^J \exp\left(\mathbf{x}_{ijt}^\top \boldsymbol{\beta}_q\right)} \right]^{y_{ijt}} \right\}.$$

Both MIXL and LC are widely used in practice to accommodate preference heterogeneity across respondents. As discussed above, in the MIXL approach parameters are assumed to vary across the population according to some prespecified statistical distribution that continuously represents preferences. In the LC model a discrete number of separate classes or segments, each with different fixed parameters, recover preference heterogeneity. In addition to differentiation in terms of continuous versus discrete consumer segments, there exist further differences between MIXL and LC. For example, compared with the MIXL approach, the LC model has the advantage of being "relatively simple, reasonably plausible and statistically testable" (Shen 2009). In addition, because LC is a semiparametric specification that depends only on the prespecified number of classes, it avoids misspecification problems in the distribution of individual heterogeneity. In fact, the main disadvantage of MIXL is that the researcher has to choose the distribution of the random parameters a priori. Nevertheless, LC is less flexible than MIXL precisely because the parameters in each class are fixed. Another important difference between these two models is the estimation procedure. The MIXL requires the use of the maximum simulated likelihood estimator – which can be very costly in terms of computational time – but no simulation is required for LC.[5] **gmnl** implements the MLE for both LC and MIXL, using analytical expressions for the appropriate gradient.

2.2. Mixed-mixed logit model

To take advantage of the benefits of both MIXL and LC, recent empirical papers have derived a mixture of both models. This double-mixture is known as the 'mixed-mixed' logit model (MM-MNL, Keane and Wasi 2013).[6] Bujosa *et al.* (2010), and Greene and Hensher (2013) developed this MM-MNL model by extending the LC model to allow for random parameters within each class.

Consider the case where the heterogeneity distribution is generalized to a discrete mixture of multivariate normal distributions. In this case we have:

$$\boldsymbol{\beta}_i \sim N(\boldsymbol{\beta}_q, \boldsymbol{\Sigma}_q) \text{ with probability } w_{iq} \text{ for } q = 1, \ldots, Q. \quad (3)$$

The appeal of using a Gaussian mixture for the heterogeneity distribution is that any continuous distribution can be approximated by a discrete mixture of normal distributions (Train

---

[5]For an empirical comparison between these two models, see for example Greene and Hensher (2003), Shen (2009) and Hess, Ben-Akiva, Gopinath, and Walker (2011).

[6]Train (2008) refers to this model as 'discrete mixture of continuous distributions', whereas Greene and Hensher (2013) label it 'LC-MIXL'.

2008). Note that the MM-MNL with only one class is equivalent to the MIXL model. Furthermore, if $\boldsymbol{\Sigma}_q \to 0$ for all q, the model in Equation 3 becomes a LC-MNL model (Bujosa *et al.* 2010; Keane and Wasi 2013). Thus, MM-MNL nests both MIXL and LC.

The choice probabilities for the MM-MNL are given by:

$$P_i(\boldsymbol{\theta}) = \sum_q^Q w_{iq} \int \left\{ \prod_t^T \prod_j^J \left[\frac{\exp\left(\mathbf{x}_{ijt}^\top \boldsymbol{\beta}_i\right)}{\sum_{j=1}^J \exp\left(\mathbf{x}_{ijt}^\top \boldsymbol{\beta}_i\right)} \right]^{y_{ijt}} \right\} f_q(\boldsymbol{\beta}_i) d\boldsymbol{\beta}_i,$$

where $f_q(\boldsymbol{\beta}_i) = N(\boldsymbol{\beta}_q, \boldsymbol{\Sigma}_q)$. Due to the complex expression of the probability, **gmnl** implements the maximum likelihood estimator for the MM-MNL parameters with the Monte Carlo approximation of this choice probability and the analytical expression of the gradient.

2.3. Generalized multinomial logit model

Fiebig *et al.* (2010) proposed a general version of the MIXL model, which they called the G-MNL model, where the parameters vary across individuals according to:

$$\boldsymbol{\beta}_i = \sigma_i \boldsymbol{\beta} + [\gamma + \sigma_i(1 - \gamma)] \mathbf{L}\boldsymbol{\eta}_i, \tag{4}$$

where σ_i is the individual-specific scale of the idiosyncratic error term, and γ is a scalar parameter that controls how the variance of residual taste heterogeneity $\mathbf{L}\boldsymbol{\eta}_i$ varies with scale. To better understand this specification, it is useful to note that differing sub-models arise when some structural parameters in the G-MNL model are constrained:

- G-MNL-I: If $\gamma = 1$, then $\boldsymbol{\beta}_i = \sigma_i \boldsymbol{\beta} + \mathbf{L}\boldsymbol{\eta}_i$. In this model, the residual taste heterogeneity is independent of the scaling of $\boldsymbol{\beta}$.

- G-MNL-II: If $\gamma = 0$, then $\boldsymbol{\beta}_i = \sigma_i(\boldsymbol{\beta} + \mathbf{L}\boldsymbol{\eta}_i)$. In this model, the residual taste heterogeneity is proportional to σ_i.

- S-MNL: If $\mathsf{VAR}(\boldsymbol{\eta}_i) = \mathbf{0}$, then $\boldsymbol{\beta}_i = \sigma_i \boldsymbol{\beta}$. As pointed out by Fiebig *et al.* (2010), this model is observationally equivalent to the particular type of heterogeneity in which the parameters increase or decrease proportionally across individuals by the scaling factor σ_i. S-MNL provides a more parsimonious representation of continuous heterogeneity than MIXL, because $\boldsymbol{\beta}\sigma_i$ has fewer parameters than $\boldsymbol{\beta} + \mathbf{L}\boldsymbol{\eta}_i$ (Fiebig *et al.* 2010).

- MIXL: $\boldsymbol{\beta}_i = \boldsymbol{\beta} + \mathbf{L}\boldsymbol{\eta}_i$, if $\sigma_i = 1$.

- MNL: $\boldsymbol{\beta}_i = \boldsymbol{\beta}$, if $\sigma_i = 1$ and $\mathsf{VAR}(\boldsymbol{\eta}_i) = \mathbf{0}$.

Fiebig *et al.* (2010) note that some restrictions need to be considered to estimate the G-MNL model. First, the domain of σ_i should be the positive real line. A positive scale parameter is ensured by assuming that σ_i is distributed log-normal with standard deviation τ and mean $\bar{\sigma}$ Fiebig *et al.* (2010):

$$\sigma_i = \exp(\bar{\sigma} + \tau \upsilon_i),$$

where $\upsilon \sim N(0, 1)$. Fiebig *et al.* (2010) also note that when τ is too large, numerical problems arise for extreme draws of υ_i. To avoid this numerical issue, the authors suggest to use a truncated normal distribution for υ_i with truncation at ± 2, so that $\upsilon \sim TN[-2, +2]$. Greene

and Hensher (2010) found that constraining v_i at -1.96 and $+1.96$ maintains the smoothness of the estimator. Specifically, the authors used $v_{ir} = \Phi^{-1}(0.025 + 0.95u_{ir})$, where u_{ir} is a draw from the standard uniform distribution. **gmnl** allows the user to choose between these two ways of drawing from v_i, using the argument `typeR` (see Section 3.3).

Note that the parameters $\bar{\sigma}$, τ, and $\boldsymbol{\beta}$ are not separately identified. Fiebig *et al.* (2010) suggest that one can normalize the mean $\bar{\sigma}$ by setting:

$$\bar{\sigma} = -\log\left[\frac{1}{N}\sum_{i=1}^{N}\exp\left(\tau v_i\right)\right].$$

Another important issue in G-MNL is the domain of γ. Initially, Fiebig *et al.* (2010) imposed $\gamma \in [0, 1]$. To constrain γ in this interval, the authors used the logistic transformation:

$$\gamma = \frac{\exp(\gamma^*)}{1 + \exp(\gamma^*)},$$

and estimated γ^*. However, Keane and Wasi (2013) pointed out that both $\gamma < 0$ and $\gamma > 1$ still have meaningful behavioral interpretations. Thus, these authors estimate γ directly. **gmnl** allows to estimate γ using both procedures.

Finally, one can allow the mean of the scale to differ across individuals by including individual-specific characteristics. In this case the scale parameter can be written as:

$$\sigma_i = \exp(\bar{\sigma} + \boldsymbol{\delta}\mathbf{s}_i + \tau v_i),$$

where $\mathbf{s}_i$ is a vector of attributes of individual i.

In terms of computation, all models, except for the LC and the MNL model, are estimated in **gmnl** using the maximum simulated likelihood estimator (MSLE) and `maxLik` function from **maxLik** package (Henningsen and Toomet 2011). All models are estimated using the analytical gradient (instead of the numerical gradient). The MNL is estimated using also the analytical Hessian.

For a complete derivation of the asymptotic properties of the MSLE and a more comprehensive review of how to implement this estimator, see Train (2009), Lee (1992), Gourieroux and Monfort (1997) or Hajivassiliou and Ruud (1986).

3. The gmnl package

3.1. Format of data

The function `mlogit.data` from **mlogit** is very useful to handle multinomial data formats. **gmnl** thus uses the same class of data for estimation. If the user forgets to set the data in the `mlogit.data` format, **gmnl** will give an error message and the estimation process will stop.

For illustration purposes, we use the `TravelMode` data from the **AER** package (Kleiber and Zeileis 2008). As quickly presented in the introduction, the `TravelMode` data contains actual individual choices[7] among four transportation modes (`air`, `train`, `bus` and `car`) for travel between the cities of Sydney and Melbourne in Australia.

---

[7]In the choice modeling literature, microdata that represents real choices is known as 'revealed preference' (RP) data.

```
R> data("TravelMode", package = "AER")
R> with(TravelMode, prop.table(table(mode[choice == "yes"])))


  air train   bus   car
0.276 0.300 0.143 0.281
```

Each mode is characterized by four alternative-specific variables (`wait`, `vcost`, `travel`, `gcost`),[8] and two individual-specific variables (`income`, `size`). The observed shares of each mode are 27.62% (air), 30% (train), 14.29% (bus), and 28.1% (car). More details about and examples using this dataset can be found in Kleiber and Zeileis (2008).

```
R> head(TravelMode)


  individual  mode choice wait vcost travel gcost income size
1           1   air     no   69    59    100    70     35    1
2           1 train     no   34    31    372    71     35    1
3           1   bus     no   35    25    417    70     35    1
4           1   car    yes    0    10    180    30     35    1
5           2   air     no   64    58     68    68     30    2
6           2 train     no   44    31    354    84     30    2
```

As can be seen above, the data is in a "long" format (one row per available mode) and can be transformed into the structure needed by `gmnl` using the `mlogit.data` in the following way:

```
R> library("mlogit")
R> TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
+    alt.levels = c("air", "train", "bus", "car"))
```

The argument `choice` indicates the choice made by the individuals; `shape` specifies the original format of the data; and `alt.levels` is a character vector that contains the name of the alternatives. We show how to transform other kinds of data in the examples below. For a more complete treatment of the data using `mlogit.data` function see Croissant (2013).

Before formal modeling, it is useful to summarize the (unconditional) relationship between the travel mode and the regressors. In the next example, we reshape the data from long to wide format:

```
R> wide_TM <- reshape(TravelMode, idvar = c("individual", "income", "size"),
+    timevar = "mode", direction = "wide")
R> wide_TM$chosen_mode[wide_TM$choice.air == "yes"] <- "air"
R> wide_TM$chosen_mode[wide_TM$choice.car == "yes"] <- "car"
R> wide_TM$chosen_mode[wide_TM$choice.train == "yes"] <- "train"
R> wide_TM$chosen_mode[wide_TM$choice.bus == "yes"] <- "bus"
```

For the case-specific `income` variable, we get the following summary

---

[8] `vcost` denotes travel cost, whereas `gcost` represents a generalized cost that combines both `vcost` and `travel` – which is in-vehicle travel time – using an exogenous value of time. `wait` is for waiting time.

```
R> library("plyr")
R> ddply(wide_TM, ~ chosen_mode, summarize, mean.income = mean(income))
```

```
  chosen_mode mean.income
1         air        41.7
2         bus        29.7
3         car        42.2
4       train        23.1
```

On average, those individuals choosing `train` have the lowest income and those choosing `car` have the highest. The relationship between the chosen travel mode and the alternative-specific regressor `vcost` is summarized as follows (a similar analysis can be done for the rest of the alternative-specific variables):

```
R> ddply(wide_TM, ~ chosen_mode, summarize, mean.air = mean(vcost.air),
+    mean.car = mean(vcost.car), mean.train = mean(vcost.train),
+    mean.bus = mean(vcost.bus))
```

```
  chosen_mode mean.air mean.car mean.train mean.bus
1         air     97.6     23.4       58.5     34.3
2         bus     89.3     26.8       62.3     33.7
3         car     76.3     15.6       53.6     33.8
4       train     80.4     21.1       37.5     32.3
```

Note that these figures show that the chosen mode is not determined solely by travel cost. The purpose of the choice model is precisely to determine the tradeoffs across attributes that help to explain choices.

3.2. Formula interface

The specification of multinomial logit models using **gmnl** is similar to that of **mlogit** and **mnlogit**. In particular, we use the R package **Formula** (Zeileis and Croissant 2010), which is able to handle multi-part formulae.

Consider the `TravelMode` data and suppose that we want to estimate a multinomial logit model where the variables `wait` and `vcost` are alternative-specific variables with a generic coefficient β; `income` is an individual-specific variable with an alternative specific coefficient γ_j; and the variable `travel` is alternative-specific variables with an alternative-specific coefficient δ_j. This is done using the following 3-part formula:

```
R> f1 <- choice ~ wait + vcost | income | travel
```

By default, the alternative-specific constants (ASC) for each alternative are included. They can be omitted by adding `+0` or `-1` in the second part of the formula. For example:

```
R> f2 <- choice ~ wait + vcost | income + 0 | travel
R> f2 <- choice ~ wait + vcost | income - 1 | travel
```

Some parts may be omitted when there is no ambiguity. For instance, a model with only individual specific variables can be specified as follows:

```
R> f3 <- choice ~ 0 | income + size | 0
R> f3 <- choice ~ 0 | income + size | 1
```

Similarly, a conditional logit model, that is, a model with alternative-specific variables with a generic coefficient β, can be specified using either of the following `formula` objects:

```
R> f4 <- choice ~ wait + vcost | 0
R> f4 <- choice ~ wait + vcost | 0 | 0
R> f4 <- choice ~ wait + vcost | -1 | 0
```

For other models, such as the MIXL, S-MNL, LC-MNL and MM-MNL model, we require to use the fourth and fifth part of the `formula`. As explained in Section 2.1, **gmnl** allows incorporating observed heterogeneity in the mean of the random parameters. This can be achieved by including individual-specific characteristics (`income` and `size`) in the fourth part of the `formula`:

```
R> f5 <- choice ~ wait + vcost | 0 | 0 | income + size - 1
```

and then use the `mvar` argument to indicate how these two variables modify the mean of the random parameters. For a more complete example see Section 3.4.

The fifth part of the `formula` is reserved for either models with heterogeneity in the scale parameter or models with latent classes. For example, an S-MNL or G-MNL model where the scale varies across individuals by individual-specific characteristics can be specified as follows:

```
R> f6 <- choice ~ wait + vcost | 1 | 0 | 0 | income + size - 1
```

The same formulation can be used if a model with latent classes is estimated and both `income` and `size` determine the class assignment.

3.3. Estimating S-MNL models

In this example, we estimate an S-MNL model using the `TravelMode` data where the ASCs are fixed and not scaled. Fiebig *et al.* (2010) found that in a model where all attributes are scaled – including the ASCs – the estimates often show a explosive behavior and the model actually produces a worse fit. The basic syntax for estimation is the following:

```
R> library("gmnl")
R> smnl.nh <- gmnl(choice ~ wait + vcost + travel | 1, data = TM,
+    model = "smnl", R = 30, notscale = c(1, 1, 1, rep(0, 3)))


The following variables are not scaled:
[1] "train:(intercept)" "bus:(intercept)"   "car:(intercept)"
Estimating SMNL model
```

Options for `model`	Model
`"mnl"`	Multinomial logit model
`"mixl"`	Mixed logit model
`"smnl"`	Scaled multinomial logit model
`"gmnl"`	Generalized multinomial logit model
`"lc"`	Latent class multinomial logit model
`"mm"`	Mixed-mixed multinomial logit model

Table 2: Models supported by **gmnl**.

The component | 1 in the `formula` means that the model is fitted using ASCs for the $J - 1$ alternatives. The main argument in the model is `model = "smnl"`, which indicates to the function that the user wants to estimate the S-MNL model (without random parameters). The rest of the models allowed by `gmnl` are given in Table 2. `R = 30` indicates that 30 Halton draws are used to simulate the probabilities. Another important argument in this example is `notscale`. This is a vector that indicates which variables will not be scaled (1 = not scaled and 0 = scaled). Since the ASCs are always the first variables entering in the model (if they are specified using | 1 in the second part of `formula`) and only $J - 1 = 3$ ASCs are created, `notscale = c(1, 1, 1, rep(0, 3))` implies that the constants will not be scaled.

```
R> summary(smnl.nh)
```

```
Model estimated on: Thu Feb 18 10:49:46 2016

Call:
gmnl(formula = choice ~ wait + vcost + travel | 1, data = TM,
    model = "smnl", R = 30, notscale = c(1, 1, 1, rep(0, 3)),
    method = "bfgs")

Frequencies of categories:

  air train   bus   car
0.276 0.300 0.143 0.281

The estimation took: 0h:0m:3s

Coefficients:
                 Estimate Std. Error z-value Pr(>|z|)
train:(intercept) -1.25946    0.52091   -2.42  0.01561 *
bus:(intercept)   -2.02267    0.64669   -3.13  0.00176 **
car:(intercept)   -7.19922    1.42002   -5.07  4.0e-07 ***
wait              -0.14085    0.02777   -5.07  3.9e-07 ***
vcost             -0.02198    0.01196   -1.84  0.06615 .
travel            -0.00481    0.00127   -3.78  0.00016 ***
tau                0.54412    0.16422    3.31  0.00092 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Optimization of log-likelihood by BFGS maximization
Log Likelihood: -188
Number of observations: 210
Number of iterations: 54
Exit of MLE: successful convergence
Simulation based on 30 draws
```

The results report the point estimates for each variable. Note that all preference parameters are statistically significant and negative,[9] meaning that cost and the time components create a 'disutility' to the traveler, so that people prefer modes that are both cheaper and faster. τ, which represents the standard deviation of σ_i, is also significant supporting the presence of heterogeneous preferences by means of different error variances for each individual. The output also gives additional information about estimation. The model is estimated using the BFGS procedure. Other optimization procedures such as the BHHH and Newton Raphson (NR) can be called using the argument `method` passed to the `maxLik` function.

Another important point is that the number of observations reported by `gmnl` corresponds to N/J if cross-sectional data is used, or $N \times T/J$ if panel data (repeated choice situations) is used. Finally, it is always important to check all the details in the estimation output. In our example, the output informs us that the convergence was achieved successfully. If convergence fails, the analyst needs to revise identification of the model, starting values, estimation procedure, and measurement scale of the attributes. For further details about potential convergence problems see Section 4.

In the next example, we allow the scale to differ across individuals according to their income. Basically, we assume that:

$$\sigma_i = \exp\left(\bar{\sigma} + \delta_{\texttt{income}} \texttt{income}_i + \tau \upsilon_i\right).$$

The syntax is very similar to our previous example, with minor changes in the `formula` argument:

```
R> smnl.het <- gmnl(choice ~ wait + vcost + travel | 1 | 0 | 0 | income - 1,
+    data = TM, model = "smnl", R = 30, notscale = c(1, 1, 1, 0, 0, 0),
+    typeR = FALSE)

The following variables are not scaled:
[1] "train:(intercept)" "bus:(intercept)"   "car:(intercept)"
Estimating SMNL model
```

The fifth part of the `formula` is reserved for individual-specific variables that affect scale. In this example, we specify that the variable `income` and no constant are included in σ_i.

```
R> summary(smnl.het)

Model estimated on: Thu Feb 18 10:49:50 2016
```

---

[9]`vcost` is significant at the 10% level.

```
Call:
gmnl(formula = choice ~ wait + vcost + travel | 1 | 0 | 0 | income -
    1, data = TM, model = "smnl", R = 30, notscale = c(1, 1,
    1, 0, 0, 0), typeR = FALSE, method = "bfgs")

Frequencies of categories:

  air train   bus   car
0.276 0.300 0.143 0.281

The estimation took: 0h:0m:4s

Coefficients:
                 Estimate Std. Error z-value Pr(>|z|)
train:(intercept) -0.99226    0.49290   -2.01  0.04410 *
bus:(intercept)   -1.69842    0.61673   -2.75  0.00589 **
car:(intercept)   -6.92476    1.22665   -5.65  1.6e-08 ***
wait              -0.11795    0.02253   -5.23  1.7e-07 ***
vcost             -0.01489    0.01019   -1.46  0.14412
travel            -0.00439    0.00111   -3.95  7.8e-05 ***
tau                0.54198    0.14234    3.81  0.00014 ***
het.income         0.00564    0.00339    1.66  0.09635 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization
Log Likelihood: -185
Number of observations: 210
Number of iterations: 66
Exit of MLE: successful convergence
Simulation based on 30 draws
```

The results are similar to those of the previous example. However, `vcost` is no longer statistically significant. All the parameters for the variables that enter in the scale are preceded by the string `het`. Thus, the coefficient `het.income` corresponds to δ_{income}, and is significant at the 10% level. This result indicates that the variability of the error term for each individual depends on their income. Finally, the argument `typeR` determines the type of draws used for the scale parameter. If `TRUE`, truncated normal draws are used for the scale parameter. In this case, the function `rtruncnorm` of **truncnorm** (Trautmann, Steuer, Mersmann, and Bornkamp 2014) is used. If `typeR = FALSE`, as in this example, the procedure suggested by Greene and Hensher (2010) is used. See Section 2.3 for more details.

Suppose now that we want to test the null hypothesis $H_0 : \delta_{income} = 0$. This test can be performed using the function `waldtest` or `lrtest` from the package **lmtest** (Zeileis and Hothorn 2002):

```
R> library("lmtest")
R> waldtest(smnl.nh, smnl.het)
```

```
Wald test

Model 1: choice ~ wait + vcost + travel | 1
Model 2: choice ~ wait + vcost + travel | 1 | 0 | 0 | income - 1
  Res.Df Df Chisq Pr(>Chisq)
1    203
2    202  1  2.76     0.096 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

R> lrtest(smnl.nh, smnl.het)

```
Likelihood ratio test

Model 1: choice ~ wait + vcost + travel | 1
Model 2: choice ~ wait + vcost + travel | 1 | 0 | 0 | income - 1
  #Df LogLik Df Chisq Pr(>Chisq)
1   7   -188
2   8   -185  1  5.21     0.022 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both tests are asymptotically equivalent, but they can provide different results for finite samples. In this case, both reject the null hypothesis, but at different significance levels.

In some cases, one will need to decide among non-nested models. In such cases the LR, Wald and Lagrange multiplier tests cannot be applied. However, one can use either the AIC or BIC criteria to measure the relative quality of models. In general, given a set of candidate models for the data, the preferred model is the one with the minimum AIC or BIC value. We can obtain the AIC and BIC criteria by typing:

R> AIC(smnl.nh)

```
[1] 390
```

R> AIC(smnl.het)

```
[1] 387
```

R> BIC(smnl.nh)

```
[1] 414
```

R> BIC(smnl.het)

```
[1] 414
```

The results show that AIC favors the `smnl.het` model, whereas the BIC is not able to discriminate between both models.

3.4. Estimating MIXL models

In the following examples we show how to estimate MIXL models using `gmnl`. The package **mlogit** is very efficient in estimating MIXL models. However, one advantage of using `gmnl` is the inclusion of individual-specific variables to explain the mean of the random parameters (see Equation 2). Other important expansions include the possibility of producing point and interval estimates at the individual level, and the consideration of Johnson S_b heterogeneity distributions.

If we assume that the coefficients of `travel` and `wait` vary across individuals according to:

$$\beta_{\texttt{travel},i} = \beta_1 + \pi_{11}\texttt{income} + \pi_{12}\texttt{size} + \sigma_1\eta_{1i},$$
$$\beta_{\texttt{wait},i} = \beta_2 + \pi_{21}\texttt{income} + \sigma_2\eta_{2i},$$

where η_{1i} is triangular and $\eta_{2i} \sim N(0,1)$, the corresponding MIXL model is estimated by typing:

```
R> mixl.hier <- gmnl(choice ~ vcost + travel + wait | 1 | 0 | income +
+    size - 1, data = TM, model = "mixl", ranp  = c(travel = "t",
+    wait = "n"), mvar = list(travel = c("income","size"),
+    wait = c("income")), R = 50,
+    haltons =  list("primes" = c(2, 17), "drop" = rep(19, 2)))
```

```
Estimating MIXL model
```

The argument `model = "mixl"` indicates that the MIXL model will be estimated. The distribution of the random coefficients are specified by the argument `ranp`. The distributions supported by **gmnl** are presented in Table 3.[10] Note also that the fourth part of the `formula` is reserved for all the variables that enter the mean of the random parameters. The argument `mvar` indicates which variables enter each specific random parameter. For example, `travel = c("income","size")` indicates that the mean of the `travel` coefficient varies according to `income` and `size`. Finally, `haltons` is relevant if `ranp` is not NULL. If `haltons = NULL`, pseudo-random draws are used instead of Halton sequences. If `haltons = NA`, the first K primes are used to generate the Halton draws, where K is the number of random parameters, and 15 of the initial sequence of elements are dropped. Otherwise, `haltons` should be a list with elements `prime` and `drop`. In this example we use the prime numbers 2 and 17, and we drop the first 19 elements for each series. For a further explanation of Halton draws see Train (2009).

```
R> summary(mixl.hier)
```

---

[10]It is worth mentioning that given how the random parameters of the G-MNL model are constructed (see Equation 4), the distributions allowed when `model = "gmnl"` are the normal, uniform, and triangular. Similarly, when the model is estimated with correlated random parameters, only the normal distribution and its transformations – log-normal and truncated normal – are allowed.

Shorthands	Distributions
`"n"`	Normal distribution
`"ln"`	Log-normal distribution
`"cn"`	Truncated (at zero) normal distribution
`"t"`	Triangular distribution
`"u"`	Uniform distribution
`"sb"`	Johnson S_b distribution

Table 3: Continuous distributions supported by **gmnl**.

```
Model estimated on: Thu Feb 18 10:51:09 2016

Call:
gmnl(formula = choice ~ vcost + travel + wait | 1 | 0 | income +
    size - 1, data = TM, model = "mixl", ranp = c(travel = "t",
    wait = "n"), R = 50, haltons = list(primes = c(2, 17), drop = rep(19,
    2)), mvar = list(travel = c("income", "size"), wait = c("income")),
    method = "bfgs")

Frequencies of categories:

  air train   bus   car
0.276 0.300 0.143 0.281

The estimation took: 0h:1m:19s

Coefficients:
                 Estimate Std. Error z-value Pr(>|z|)
train:(intercept) -1.70e-01   1.13e+00   -0.15  0.88005
bus:(intercept)   -1.14e+00   1.23e+00   -0.93  0.35120
car:(intercept)   -9.00e+00   2.52e+00   -3.57  0.00035 ***
vcost             -3.07e-02   1.52e-02   -2.02  0.04288 *
travel            -7.75e-03   3.29e-03   -2.35  0.01861 *
wait              -1.53e-01   4.48e-02   -3.42  0.00062 ***
travel.income     -1.59e-04   6.42e-05   -2.48  0.01324 *
travel.size        3.51e-03   1.37e-03    2.56  0.01057 *
wait.income       -1.19e-03   6.48e-04   -1.84  0.06568 .
sd.travel          5.48e-03   4.02e-03    1.36  0.17288
sd.wait            8.41e-02   3.48e-02    2.41  0.01582 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization
Log Likelihood: -165
Number of observations: 210
Number of iterations: 357
```

```
Exit of MLE: successful convergence
Simulation based on 50 draws
```

The output shows the estimates in the following order: fixed parameters, mean of the random parameters, effect of the variables that affect the mean of the random parameters, and finally the standard deviation/spread of the random parameters. Note that `travel.income` corresponds to π_{11}, `travel.size` corresponds to π_{12}, and `wait.income` corresponds to π_{21}. The parameters have the expected sign. Note that individuals with higher income are more sensitive to in-vehicle time (due to the significant and negative `travel.income`). Whereas both components of time were assumed random, only waiting time appears as having significant variation in how it is perceived across travelers. This result may be due to `income` and `size` explaining taste variations (observed heterogeneity).

We now estimate a correlated random parameter model. For this example, we will use the `Electricity` data from the **mlogit** package, which is a panel dataset. There are 4,308 observations in this microdata set, but only 361 individuals. The analyst designed 12 hypothetical choice scenarios according to a discrete choice experiment,[11] where four hypothetical electricity suppliers were described in terms of price – which could be fixed (`pf`), time-of-day rate (`tod`), or seasonal rate (`seas`); length of contract (`cl`); and being local (`loc`) or 'well-known' (`wk`). The experimental design considered unlabeled alternatives, which means that alternative-specific constants can be set at zero. This microdata is in a "wide format" (one row describes all alternatives in a given choice situation). Given time compilation restrictions, in this example we will use just a subsample of this database (`subset = 1:3000`). The user may want to use the whole sample to reproduce this case study.

```
R> data("Electricity", package = "mlogit")
R> Electr <- mlogit.data(Electricity, id.var = "id", choice = "choice",
+     varying = 3:26, shape = "wide", sep = "")
```

In this example, two arguments are especially relevant in the `gmnl` function. First, `panel = TRUE` indicates that the data is a panel. When using panel data, the user needs to specify a variable in the `id.var` argument of the `mlogit.data` function that identifies the individual. Second, to estimate correlated random parameters `correlation = TRUE` needs to be indicated in the `gmnl` function. The syntax is the following:

```
R> Elec.cor <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0,
+     data = Electr, subset = 1:3000, model = 'mixl', R = 50, panel = TRUE,
+     ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n"),
+     correlation =  TRUE)

Estimating MIXL model


R> summary(Elec.cor)


Model estimated on: Thu Feb 18 10:51:26 2016
```

---

[11] Discrete choice experiments collect 'stated preference' (SP) data, where choices reflect intended behavior. In this example, some respondents did not provide their choices for all 12 choice situations.

```
Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0,
    data = Electr, subset = 1:3000, model = "mixl", ranp = c(cl = "n",
        loc = "n", wk = "n", tod = "n", seas = "n"), R = 50,
    correlation = TRUE, panel = TRUE, method = "bfgs")
```

Frequencies of categories:

```
    1     2     3     4
0.215 0.303 0.217 0.265
```

The estimation took: 0h:0m:16s

Coefficients:

	Estimate	Std. Error	z-value	Pr(>|z|)	
pf	-0.8702	0.0786	-11.07	< 2e-16	***
cl	-0.1765	0.0430	-4.11	4.0e-05	***
loc	2.3822	0.3053	7.80	6.0e-15	***
wk	1.9447	0.2493	7.80	6.2e-15	***
tod	-8.5026	0.7423	-11.45	< 2e-16	***
seas	-8.6456	0.7803	-11.08	< 2e-16	***
sd.cl.cl	0.3919	0.0420	9.33	< 2e-16	***
sd.cl.loc	0.4921	0.1983	2.48	0.01311	*
sd.cl.wk	0.5514	0.2131	2.59	0.00966	**
sd.cl.tod	-0.9834	0.2802	-3.51	0.00045	***
sd.cl.seas	-0.1470	0.2297	-0.64	0.52206	
sd.loc.loc	2.5925	0.4226	6.14	8.5e-10	***
sd.loc.wk	1.9311	0.3610	5.35	8.8e-08	***
sd.loc.tod	1.0198	0.5651	1.80	0.07114	.
sd.loc.seas	0.0941	0.4579	0.21	0.83723	
sd.wk.wk	-0.3330	0.2212	-1.51	0.13226	
sd.wk.tod	1.9341	0.3208	6.03	1.7e-09	***
sd.wk.seas	0.7349	0.3030	2.43	0.01529	*
sd.tod.tod	2.0635	0.3301	6.25	4.1e-10	***
sd.tod.seas	1.1689	0.2539	4.60	4.2e-06	***
sd.seas.seas	1.7034	0.2533	6.72	1.8e-11	***

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

```
Optimization of log-likelihood by BFGS maximization
Log Likelihood: -692
Number of observations: 750
Number of iterations: 97
Exit of MLE: successful convergence
Simulation based on 50 draws
```

Note that the estimates for the population means indicate that on average individuals prefer

contracts that are cheaper and shorter, and companies that are local and well-known. But, there is great variation in preferences. The estimates from `sd.cl.cl` to `sd.seas.seas` are the elements of the lower triangular matrix **L**. If the user is interested in the standard errors of the variance-covariance matrix of the random parameters $\mathbf{LL}^\top = \boldsymbol{\Sigma}$ or the standard deviations, the `S3` function `vcov` can be used for finding these elements. The syntax for both cases is the following:[12]

```
R> vcov(Elec.cor, what = "ranp", type = "cov", se = "true")
```

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

```
            Estimate Std. Error z-value Pr(>|z|)
v.cl.cl       0.1536     0.0329    4.67  3.1e-06 ***
v.cl.loc      0.1928     0.0816    2.36   0.0181 *
v.cl.wk       0.2161     0.0917    2.36   0.0185 *
v.cl.tod     -0.3854     0.1290   -2.99   0.0028 **
v.cl.seas    -0.0576     0.0906   -0.64   0.5249
v.loc.loc     6.9630     2.2065    3.16   0.0016 **
v.loc.wk      5.2776     1.6637    3.17   0.0015 **
v.loc.tod     2.1599     1.3323    1.62   0.1050
v.loc.seas    0.1715     1.1222    0.15   0.8785
v.wk.wk       4.1440     1.3293    3.12   0.0018 **
v.wk.tod      0.7832     0.8530    0.92   0.3585
v.wk.seas    -0.1441     0.7525   -0.19   0.8481
v.tod.tod    10.0058     3.4763    2.88   0.0040 **
v.tod.seas    4.0739     1.5217    2.68   0.0074 **
v.seas.seas   4.8384     1.1851    4.08  4.5e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
R> vcov(Elec.cor, what = "ranp", type = "sd", se = "true")
```

```
Standard deviations of the random parameters
```

```
     Estimate Std. Error z-value Pr(>|z|)
cl      0.392      0.042    9.33  < 2e-16 ***
loc     2.639      0.418    6.31  2.8e-10 ***
wk      2.036      0.327    6.23  4.5e-10 ***
tod     3.163      0.549    5.76  8.6e-09 ***
seas    2.200      0.269    8.17  2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The correlation matrix of the random parameters can be recovered using the following syntax:

---

[12]To compute the standard errors, **gmnl** uses the `deltamethod` function from the **msm** package (Jackson 2011).

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


         cl     loc      wk     tod    seas
cl    1.0000  0.1865  0.2708  -0.311  -0.0668
loc   0.1865  1.0000  0.9825   0.259   0.0295
wk    0.2708  0.9825  1.0000   0.122  -0.0322
tod  -0.3109  0.2588  0.1216   1.000   0.5855
seas -0.0668  0.0295 -0.0322   0.586   1.0000
```

3.5. Estimating G-MNL models

In the following examples we show how to estimate G-MNL models in **gmnl**. Although we do not need to specify constants in an unlabeled experiment (alternatives are fully defined by the experimental attributes), just for illustrative purposes we will assume that the ASCs are random. Using the `formula` to create the ASCs produces problems in the `ranp` argument due to the way the constants are labeled. So, we first create the ASCs *by hand*:

```
R> Electr$asc2 <- as.numeric(Electr$alt == 2)
R> Electr$asc3 <- as.numeric(Electr$alt == 3)
R> Electr$asc4 <- as.numeric(Electr$alt == 4)
```

The G-MNL model is estimated using `model = "gmnl"`:

```
R> Elec.gmnl <- gmnl(choice ~ pf + cl + loc + wk + tod + seas + asc2 +
+    asc3 + asc4 | 0, data = Electr, subset = 1:3000, model = 'gmnl',
+    R = 50, panel = TRUE, notscale = c(rep(0, 6), 1, 1, 1),
+    ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n",
+      asc2 = "n", asc3 = "n", asc4 = "n"))


The following variables are not scaled:
[1] "asc2" "asc3" "asc4"
Estimating GMNL model


R> summary(Elec.gmnl)


Model estimated on: Thu Feb 18 10:51:48 2016


Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas + asc2 +
    asc3 + asc4 | 0, data = Electr, subset = 1:3000, model = "gmnl",
    ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n",
        asc2 = "n", asc3 = "n", asc4 = "n"), R = 50, panel = TRUE,
    notscale = c(rep(0, 6), 1, 1, 1), method = "bfgs")


Frequencies of categories:
```

```
     1     2     3     4
0.215 0.303 0.217 0.265
```

```
The estimation took: 0h:0m:22s
```

```
Coefficients:
        Estimate Std. Error z-value Pr(>|z|)
pf       -0.8733     0.1066   -8.19  2.2e-16 ***
cl       -0.1718     0.0422   -4.07  4.7e-05 ***
loc       1.8081     0.2289    7.90  2.9e-15 ***
wk        1.7543     0.2222    7.89  2.9e-15 ***
tod      -8.5960     0.9865   -8.71  < 2e-16 ***
seas     -8.8653     1.0128   -8.75  < 2e-16 ***
asc2      0.3044     0.1539    1.98   0.0479 *
asc3      0.1563     0.1598    0.98   0.3279
asc4      0.1133     0.1568    0.72   0.4698
sd.cl     0.3643     0.0442    8.25  2.2e-16 ***
sd.loc    1.1015     0.2738    4.02  5.7e-05 ***
sd.wk     1.2053     0.2493    4.84  1.3e-06 ***
sd.tod    1.4655     0.2336    6.28  3.5e-10 ***
sd.seas   1.8110     0.2958    6.12  9.3e-10 ***
sd.asc2   0.5264     0.1791    2.94   0.0033 **
sd.asc3   0.0849     0.2246    0.38   0.7055
sd.asc4   0.2407     0.1881    1.28   0.2007
tau       0.6777     0.1490    4.55  5.4e-06 ***
gamma     0.3625     0.1747    2.08   0.0380 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Optimization of log-likelihood by BFGS maximization
Log Likelihood: -729
Number of observations: 750
Number of iterations: 98
Exit of MLE: successful convergence
Simulation based on 50 draws
```

Since we are including the ASCs as additional variables, the second part of the formula does not include the ASCs (| 0). Note also that even though the ASCs are random, they are not scaled: `notscale = c(rep(0, 6), 1, 1, 1)` indicates that the last three variables in the first part of the `formula` (asc2, asc3, and asc4) are not scaled.

Another important issue is that `gmnl` estimates γ directly by default as suggested by Keane and Wasi (2013). However, one can estimate γ^*, where $\gamma = \exp(\gamma^*)/(1+\exp(\gamma^*))$ as suggested by Fiebig *et al.* (2010), by specifying `hgamma = "indirect"`. Thus, `hgamma = "direct"` is the default setting.

The G-MNL estimation code is also very convenient when one wants to estimate S-MNL models with random effects (Keane and Wasi 2013). In this case, the user can fix γ and use `model = "gmnl"`.

```
R> Elec.smnl.re <- gmnl(choice ~ pf + cl + loc + wk + tod + seas + asc2 +
+    asc3 + asc4 | 0, data = Electr, subset = 1:3000, model = 'gmnl', R = 50,
+    panel = TRUE, print.init =  TRUE, notscale = c(rep(0, 6), 1, 1, 1),
+    ranp = c(asc2 = "n", asc3 = "n", asc4 = "n"), init.gamma = 0,
+    fixed = c(rep(FALSE, 16), TRUE), correlation =  TRUE)

The following variables are not scaled:
[1] "asc2" "asc3" "asc4"

Starting Values:
          pf            cl           loc            wk           tod
      -0.6018       -0.1350        1.2223        1.0387       -5.3686
         seas          asc2          asc3          asc4 sd.asc2.asc2
      -5.5623        0.2097        0.0811        0.1065        0.1000
sd.asc2.asc3 sd.asc2.asc4 sd.asc3.asc3 sd.asc3.asc4 sd.asc4.asc4
       0.1000        0.1000        0.1000        0.1000        0.1000
          tau         gamma
       0.1000        0.0000
Estimating GMNL model


R> summary(Elec.smnl.re)

Model estimated on: Thu Feb 18 10:52:04 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas + asc2 +
    asc3 + asc4 | 0, data = Electr, subset = 1:3000, model = "gmnl",
    ranp = c(asc2 = "n", asc3 = "n", asc4 = "n"), R = 50, correlation = TRUE,
    panel = TRUE, init.gamma = 0, notscale = c(rep(0, 6), 1,
        1, 1), print.init = TRUE, fixed = c(rep(FALSE, 16), TRUE),
    method = "bfgs")

Frequencies of categories:

    1     2     3     4
0.215 0.303 0.217 0.265

The estimation took: 0h:0m:16s

Coefficients:
          Estimate Std. Error z-value Pr(>|z|)
pf          -0.6299     0.1148   -5.49 4.1e-08 ***
cl          -0.1377     0.0328   -4.20 2.7e-05 ***
loc          1.2749     0.2200    5.80 6.8e-09 ***
wk           1.1119     0.1929    5.76 8.2e-09 ***
tod         -6.2008     1.0450   -5.93 3.0e-09 ***
seas        -6.3681     1.0802   -5.90 3.7e-09 ***
```

```
asc2            0.2124    0.1442    1.47    0.1409
asc3            0.2295    0.1351    1.70    0.0894 .
asc4            0.1536    0.1310    1.17    0.2410
sd.asc2.asc2    0.5694    0.2184    2.61    0.0091 **
sd.asc2.asc3    0.3066    0.1813    1.69    0.0908 .
sd.asc2.asc4    0.1508    0.1995    0.76    0.4497
sd.asc3.asc3    0.0445    0.2192    0.20    0.8390
sd.asc3.asc4    0.0893    0.2188    0.41    0.6832
sd.asc4.asc4   -0.0406    0.2034   -0.20    0.8419
tau             1.1009    0.1852    5.94   2.8e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization
Log Likelihood: -841
Number of observations: 750
Number of iterations: 76
Exit of MLE: successful convergence
Simulation based on 50 draws
```

The argument `init.gamma` indicates the initial value for γ. In this case we set it at zero. The next step is to set the parameters that are fixed by using the argument `fixed`, which is passed to the `maxLik` function. Note that the user needs to be careful with the order of the parameters. We encourage the user to estimate first a model where all the parameters are freely estimated with the argument `print.init = TRUE`. This argument will display the initial values and the order used by `gmnl`. Generally, γ is the last parameter that enters the likelihood specification. So, by typing `fixed = c(rep(FALSE, 16), TRUE)` we are only holding γ fixed at zero, and the rest of the coefficients are freely estimated.

By default, the initial values for the mean of the random parameters come from an MNL, and the standard deviations or spread are set at 0.1. However, the starting values from an MNL model may not be the best guess, since the G-MNL model is not globally concave. The best starting values for a G-MNL model with correlated parameters might be: 1) G-MNL with uncorrelated parameters, 2) MIXL with correlated parameters, or 3) GMNL with correlated parameters with γ fixed at 0. One can first get these initial parameters and then use the `start` argument of `gmnl` to indicate the vector of appropriate starting values (see Section 3.7 for an example of how to use the `start` argument).

3.6. Estimating LC and MM-MNL models

The next example shows how an LC model with two classes can be estimated:

```
R> Elec.lc <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0 | 0 | 0 | 1,
+     data = Electr, subset = 1:3000, model = 'lc', panel = TRUE, Q = 2)

Estimating LC model
```

Note that for the LC model, one needs to specify at least a constant in the fifth part of the `formula`. If the class assignment w_{iq} is also determined by socio-economic characteristics,

those covariates can also be included in the fifth part. The LC model is estimated by typing `model = "lc"`, and the prespecified number of classes is indicated with the argument `Q`.

```
R> summary(Elec.lc)
```

```
Model estimated on: Thu Feb 18 10:52:05 2016
```

```
Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0 |
    0 | 0 | 1, data = Electr, subset = 1:3000, model = "lc",
    Q = 2, panel = TRUE, method = "bfgs")
```

```
Frequencies of categories:
```

```
    1     2     3     4
0.215 0.303 0.217 0.265
```

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

```
Coefficients:
             Estimate Std. Error z-value Pr(>|z|)
class.1.pf    -0.4458     0.0876   -5.09  3.6e-07 ***
class.1.cl    -0.1847     0.0301   -6.14  8.3e-10 ***
class.1.loc    1.2144     0.1618    7.50  6.2e-14 ***
class.1.wk     0.9641     0.1429    6.75  1.5e-11 ***
class.1.tod   -3.2184     0.6880   -4.68  2.9e-06 ***
class.1.seas  -3.4865     0.6929   -5.03  4.9e-07 ***
class.2.pf    -0.8431     0.0968   -8.71  < 2e-16 ***
class.2.cl    -0.1242     0.0453   -2.74   0.0061 **
class.2.loc    1.6445     0.2689    6.12  9.6e-10 ***
class.2.wk     1.4139     0.2120    6.67  2.6e-11 ***
class.2.tod   -9.3732     0.8676  -10.80  < 2e-16 ***
class.2.seas  -9.2647     0.8847  -10.47  < 2e-16 ***
(class)2      -0.2200     0.0788   -2.79   0.0052 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Optimization of log-likelihood by BFGS maximization
Log Likelihood: -793
Number of observations: 750
Number of iterations: 77
Exit of MLE: successful convergence
```

Note that the underlying assumption in this example is that there are two types of customers. For both types, the parameters have expected signs and are statistically significant. Probability of class assignment is assumed constant in this case, but in a dataset with sociodemographics, assignment to classes can vary for each individual using those sodiodemographics as explanatory variables of the class assignment probability.

The following example estimates an MM-MNL with a mixture of two normal distributions:

```
R> Elec.mm <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0 | 0 | 0 | 1,
+     data = Electr, subset = 1:3000, model = 'mm', R = 50, panel = TRUE,
+     ranp = c(pf = "n", cl = "n", loc = "n", wk = "n", tod = "n",
+       seas = "n"), Q = 2, iterlim = 500)

Estimating MM-MNL model

R> summary(Elec.mm)

Model estimated on: Thu Feb 18 10:57:27 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0 |
    0 | 0 | 1, data = Electr, subset = 1:3000, model = "mm",
    ranp = c(pf = "n", cl = "n", loc = "n", wk = "n", tod = "n",
        seas = "n"), R = 50, Q = 2, panel = TRUE, iterlim = 500,
    method = "bfgs")

Frequencies of categories:

    1     2     3     4
0.215 0.303 0.217 0.265

The estimation took: 0h:5m:22s

Coefficients:
                Estimate Std. Error z-value Pr(>|z|)
class.1.pf      -1.28036    0.12279  -10.43  < 2e-16 ***
class.1.cl      -0.49715    0.08070   -6.16  7.3e-10 ***
class.1.loc      0.64445    0.24097    2.67  0.00749 **
class.1.wk       0.71241    0.21823    3.26  0.00110 **
class.1.tod    -11.62474    1.02748  -11.31  < 2e-16 ***
class.1.seas   -12.65698    1.13957  -11.11  < 2e-16 ***
class.2.pf      -0.43575    0.11588   -3.76  0.00017 ***
class.2.cl       0.08464    0.08288    1.02  0.30713
class.2.loc      3.38611    0.38188    8.87  < 2e-16 ***
class.2.wk       2.72095    0.32688    8.32  < 2e-16 ***
class.2.tod     -4.71637    1.05414   -4.47  7.7e-06 ***
class.2.seas    -4.64753    0.96000   -4.84  1.3e-06 ***
class.1.sd.pf    0.10526    0.03938    2.67  0.00752 **
class.1.sd.cl    0.26906    0.05835    4.61  4.0e-06 ***
class.1.sd.loc   0.00764    0.26552    0.03  0.97704
class.1.sd.wk    0.11414    0.58053    0.20  0.84413
class.1.sd.tod   2.20533    0.45035    4.90  9.7e-07 ***
class.1.sd.seas  2.32243    0.45203    5.14  2.8e-07 ***
```

```
class.2.sd.pf     0.19696     0.03371     5.84   5.2e-09 ***
class.2.sd.cl     0.35523     0.07491     4.74   2.1e-06 ***
class.2.sd.loc    0.58717     0.28699     2.05   0.04076 *
class.2.sd.wk     1.10025     0.30323     3.63   0.00029 ***
class.2.sd.tod    1.40225     0.51875     2.70   0.00687 **
class.2.sd.seas   0.07309     0.25655     0.28   0.77572
(class)2         -0.13714     0.07844    -1.75   0.08039 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization
Log Likelihood: -672
Number of observations: 750
Number of iterations: 125
Exit of MLE: successful convergence
Simulation based on 50 draws
```

The specification is similar to that of the LC model, but we now allow the parameters in each
class to be normally distributed using the argument `ranp`. It is worth mentioning that the
number of iterations required for this model is greater than that for previous models.[13] For
that reason we have set the maximum of iterations at 500 using the argument `iterlim`. The
example below adds the consideration of correlated parameters, which is the model typically
used in Bayesian treatments.

```
R> Elec.mm.c <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0 | 0 | 0 |
+    1, data = Electr, subset = 1:3000, model = 'mm', R = 50, panel = TRUE,
+    ranp = c(pf = "n", cl = "n", loc = "n", wk = "n", tod = "n",
       seas = "n"), Q = 2, iterlim = 500, correlation = TRUE)

Estimating MM-MNL model

R> summary(Elec.mm.c)

Model estimated on: Thu Feb 18 10:59:08 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0 |
    0 | 0 | 1, data = Electr, subset = 1:3000, model = "mm",
    ranp = c(pf = "n", cl = "n", loc = "n", wk = "n", tod = "n",
        seas = "n"), R = 50, Q = 2, correlation = TRUE, panel = TRUE,
    iterlim = 500, method = "bfgs")
```

---

[13]Another important issue is weak identification. Large magnitudes of the coefficients or standard errors,
or even models with slow convergence, might be a sign of weak identification, especially in complex models
such as the MM-logit model (Ruud 2007). Users should remain suspicious and further investigate such cases
by analyzing the Hessian matrix of second-order partial derivatives. If the Hessian matrix is positive definite
(e.g., all its eigenvalues are positive), the model is said to be locally identified (see Wedel and Kamakura 2012,
p. 91). The Hessian can be obtained by `my_model$logLik$hessian` after estimating a model of class `gmnl`.

```
Frequencies of categories:

     1     2     3     4
0.215 0.303 0.217 0.265

The estimation took: 0h:1m:41s

Coefficients:
                     Estimate Std. Error z-value Pr(>|z|)
class.1.pf            -0.9996     0.1311   -7.63  2.4e-14 ***
class.1.cl            -0.2168     0.0545   -3.98  7.0e-05 ***
class.1.loc            2.7749     0.4689    5.92  3.3e-09 ***
class.1.wk             2.3225     0.3165    7.34  2.2e-13 ***
class.1.tod           -9.8537     1.1690   -8.43  < 2e-16 ***
class.1.seas          -9.5464     1.0975   -8.70  < 2e-16 ***
class.2.pf            -1.9182     0.4702   -4.08  4.5e-05 ***
class.2.cl            -0.8815     0.1673   -5.27  1.4e-07 ***
class.2.loc            3.7753     1.0432    3.62  0.00030 ***
class.2.wk             3.0521     0.8006    3.81  0.00014 ***
class.2.tod          -13.4974     3.6628   -3.68  0.00023 ***
class.2.seas         -15.0237     3.6817   -4.08  4.5e-05 ***
class.1.sd.pf.pf       0.6324     0.1245    5.08  3.8e-07 ***
class.1.sd.pf.cl       0.2018     0.0522    3.87  0.00011 ***
class.1.sd.pf.loc      1.4214     0.3932    3.62  0.00030 ***
class.1.sd.pf.wk       1.0639     0.2577    4.13  3.6e-05 ***
class.1.sd.pf.tod      5.5303     1.0575    5.23  1.7e-07 ***
class.1.sd.pf.seas     4.3941     1.0112    4.35  1.4e-05 ***
class.1.sd.cl.cl       0.2116     0.0638    3.32  0.00091 ***
class.1.sd.cl.loc     -1.0122     0.3917   -2.58  0.00976 **
class.1.sd.cl.wk      -0.6071     0.2838   -2.14  0.03239 *
class.1.sd.cl.tod      2.0922     0.4038    5.18  2.2e-07 ***
class.1.sd.cl.seas     1.5087     0.3366    4.48  7.4e-06 ***
class.1.sd.loc.loc     1.3945     0.5937    2.35  0.01883 *
class.1.sd.loc.wk      0.8431     0.3562    2.37  0.01793 *
class.1.sd.loc.tod     0.5368     0.3483    1.54  0.12325
class.1.sd.loc.seas    0.6143     0.4411    1.39  0.16371
class.1.sd.wk.wk       0.5915     0.1638    3.61  0.00030 ***
class.1.sd.wk.tod      0.2274     0.2398    0.95  0.34301
class.1.sd.wk.seas     0.7723     0.2370    3.26  0.00112 **
class.1.sd.tod.tod     1.6883     0.5447    3.10  0.00194 **
class.1.sd.tod.seas    1.9974     0.4498    4.44  9.0e-06 ***
class.1.sd.seas.seas   0.0275     0.4979    0.06  0.95597
class.2.sd.pf.pf      -1.0278     0.2735   -3.76  0.00017 ***
class.2.sd.pf.cl       0.3755     0.1017    3.69  0.00022 ***
class.2.sd.pf.loc      1.5795     0.5169    3.06  0.00224 **
class.2.sd.pf.wk       1.9678     0.5497    3.58  0.00034 ***
```

```
class.2.sd.pf.tod     -7.1137    2.0080   -3.54  0.00040 ***
class.2.sd.pf.seas    -6.0019    2.0065   -2.99  0.00278 **
class.2.sd.cl.cl       1.2556    0.2070    6.07  1.3e-09 ***
class.2.sd.cl.loc      0.0671    0.5375    0.12  0.90058
class.2.sd.cl.wk      -0.5454    0.4364   -1.25  0.21141
class.2.sd.cl.tod     -1.0686    0.5221   -2.05  0.04068 *
class.2.sd.cl.seas     4.0996    0.8738    4.69  2.7e-06 ***
class.2.sd.loc.loc     3.8107    0.9585    3.98  7.0e-05 ***
class.2.sd.loc.wk      3.3262    0.7860    4.23  2.3e-05 ***
class.2.sd.loc.tod     0.3255    0.6085    0.53  0.59279
class.2.sd.loc.seas    2.8477    0.9473    3.01  0.00265 **
class.2.sd.wk.wk      -0.7910    0.3234   -2.45  0.01446 *
class.2.sd.wk.tod      0.7507    0.3089    2.43  0.01508 *
class.2.sd.wk.seas    -1.6942    0.6594   -2.57  0.01019 *
class.2.sd.tod.tod    -2.6490    0.6178   -4.29  1.8e-05 ***
class.2.sd.tod.seas   -0.4250    0.6364   -0.67  0.50425
class.2.sd.seas.seas   1.4412    0.4171    3.46  0.00055 ***
(class)2              -1.0232    0.0983  -10.40  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Optimization of log-likelihood by BFGS maximization
Log Likelihood: -640
Number of observations: 750
Number of iterations: 292
Exit of MLE: successful convergence
Simulation based on 50 draws
```

The standard deviations of the random parameters and their standard errors for each class can be obtained using the vcov function in the following way

```
R> vcov(Elec.mm.c, what = "ranp", Q = 1, type = "sd", se = TRUE)


Standard deviations of the random parameters


      Estimate Std. Error z-value Pr(>|z|)
pf      0.6324     0.1245    5.08  3.8e-07 ***
cl      0.2924     0.0514    5.69  1.2e-08 ***
loc     2.2337     0.4054    5.51  3.6e-08 ***
wk      1.6004     0.3159    5.07  4.1e-07 ***
tod     6.1767     0.9809    6.30  3.0e-10 ***
seas    5.1525     0.9761    5.28  1.3e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


R> vcov(Elec.mm.c, what = "ranp", Q = 2, type = "sd", se = TRUE)
```

```
Standard deviations of the random parameters

     Estimate Std. Error z-value Pr(>|z|)
pf      1.028      0.273    3.76  0.00017 ***
cl      1.311      0.209    6.26  3.8e-10 ***
loc     4.126      1.017    4.06  4.9e-05 ***
wk      3.982      0.885    4.50  6.8e-06 ***
tod     7.709      1.960    3.93  8.4e-05 ***
seas    8.128      1.853    4.39  1.2e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.7. Willingness-to-pay space

Willingness-to-pay space models reparameterize the parameter space in such a way that the marginal WTP for each attribute (and the parameters of its heterogeneity distribution for a random parameter model) is directly estimated rather than the marginal utility (preference parameters). To motivate the WTP space model, consider the following latent utility

$$U_{ijt} = -\alpha p_{ijt} + \mathbf{x}_{ijt}^\top \boldsymbol{\beta} + \epsilon_{ijt}, \tag{5}$$

where p_{ijt} is the price coefficient. This model is known as the model in preference space. The utility in WTP-space is obtained by dividing the attribute's coefficients by the price coefficient in the following way

$$\begin{aligned} U_{ijt} &= -\alpha p_{ijt} + \mathbf{x}_{ijt}^\top \left(-\alpha \frac{\boldsymbol{\beta}}{\alpha}\right) + \epsilon_{ijt} \\ &= -\alpha p_{ijt} + \mathbf{x}_{ijt}^\top \left(-\alpha \boldsymbol{\gamma}\right) + \epsilon_{ijt}, \end{aligned}$$

where $\boldsymbol{\gamma}$ is the WTP parameter vector, and α is fixed and equal to 1. Although both, preference and WTP-space are behaviorally equivalent, the latter approach is useful when allowing for random heterogeneity in $\boldsymbol{\gamma}$.

In effect, the WTP-space approach is very appealing because it allows the analyst to specify and estimate the distributions of WTP directly, rather than deriving them indirectly from distributions of coefficients in preference space model (Scarpa, Thiene, and Train 2008). In the preference space model, the distribution of WTP is derived from the distribution of the ratio of both α and $\boldsymbol{\beta}$. However this ratio may not result in a well-specified distribution. For example, if both α and $\boldsymbol{\beta}$ are normally distributed, then the ratio produces a Cauchy distribution with no finite moments (Daly, Hess, and Train 2011). Motivated by this problem, Train and Weeks (2005) and Sonnier, Ainslie, and Otter (2007) extended the WTP-space approach by allowing $\boldsymbol{\gamma}$ to follow any distribution and thus to avoid the problem of non-finite moments for the distribution of WTP.

To illustrate the concept of WTP-space, and how it can be estimated using **gmnl**, we will first show the case without random parameters. The standard procedure to derive willingness-to-pay measures is to start with a model in preference space, and then make inference on the appropriate ratio that represents the marginal rate of substitution between a given attribute and price. For example, consider the simple conditional logit model,

```
R> clogit <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0,
+    data = Electr, subset = 1:3000)
R> summary(clogit)
```

```
Model estimated on: Thu Feb 18 10:59:09 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0,
    data = Electr, subset = 1:3000, method = "nr")

Frequencies of categories:

    1     2     3     4
0.215 0.303 0.217 0.265

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

Coefficients:
     Estimate Std. Error z-value Pr(>|z|)
pf    -0.6113     0.0548  -11.15  < 2e-16 ***
cl    -0.1398     0.0204   -6.85  7.2e-12 ***
loc    1.1986     0.1197   10.01  < 2e-16 ***
wk     1.0304     0.1063    9.69  < 2e-16 ***
tod   -5.4540     0.4341  -12.56  < 2e-16 ***
seas  -5.6648     0.4419  -12.82  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by Newton-Raphson maximisation
Log Likelihood: -870
Number of observations: 750
Number of iterations: 4
Exit of MLE: gradient close to zero
```

To estimate the willingness to pay for each attribute, one needs to divide each attribute parameter by that of price `pf`. This ratio can be easily retrieved using the function `wtp.gmnl`:

```
R> wtp.gmnl(clogit, wrt = "pf")
```

```
Willigness-to-pay respect to:  pf

     Estimate Std. Error t-value Pr(>|t|)
cl     0.2287     0.0358    6.38  1.8e-10 ***
loc   -1.9610     0.2304   -8.51  < 2e-16 ***
wk    -1.6858     0.1949   -8.65  < 2e-16 ***
tod    8.9226     0.2025   44.07  < 2e-16 ***
seas   9.2675     0.2164   42.83  < 2e-16 ***
```

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

The argument `wrt = "pf"` indicates that all the parameters should be divided by the parameter of the attribute `pf`.[14] Using the estimated ratios, we can say, for example, that a individual with average price and contract lenght (`cl`) is willing to pay ≈ 0.23, or almost one-fifth a cent per kWh extra, to have a contract that is one year shorter.

Another way to estimate the same WTP coefficients is to use the S-MNL model to derive a specification in WTP-space. To do so, we need first to compute the negative of the price attribute using the `mlogit.data` function:

```
R> ElectrO <- mlogit.data(Electricity, id = "id", choice = "choice",
+    varying = 3:26, shape = "wide", sep = "", opposite = c("pf"))
```

Next, we need to set the values for the price parameter and τ at 1 and 0, respectively. The `fixed` argument is used to set these values.

```
R> start <- c(1, 0, 0, 0, 0, 0, 0, 0)
R> wtps <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0 | 0 | 0 | 1,
+    data = ElectrO, model = "smnl", subset = 1:3000, R = 1,
+    fixed = c(TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FALSE),
+    panel = TRUE, start = start, method = "bhhh", iterlim = 500)

Estimating SMNL model
```

Note that we fitted the S-MNL model with a constant in the scale. This constant, after a proper transformation, will represent the price parameter. Since we are working with a fixed parameter model, the number of draws is set equal to 1.

```
R> summary(wtps)

Model estimated on: Thu Feb 18 10:59:09 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0 |
    0 | 0 | 1, data = ElectrO, subset = 1:3000, model = "smnl",
    start = start, R = 1, panel = TRUE, fixed = c(TRUE, FALSE,
        FALSE, FALSE, FALSE, FALSE, TRUE, FALSE), method = "bhhh",
    iterlim = 500)

Frequencies of categories:

    1     2     3     4
0.215 0.303 0.217 0.265
```

---

[14]In the current version of **gmnl**, the standard error of the WTP estimates are calculated using the delta method, which only works well when there are no problems of weak identification in the ratio.

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

Coefficients:
                Estimate Std. Error z-value Pr(>|z|)
cl               -0.2287     0.0361   -6.34  2.4e-10 ***
loc               1.9609     0.2284    8.59  < 2e-16 ***
wk                1.6857     0.1915    8.80  < 2e-16 ***
tod              -8.9226     0.2025  -44.06  < 2e-16 ***
seas             -9.2675     0.2166  -42.79  < 2e-16 ***
het.(Intercept)  -0.4922     0.0917   -5.37  7.9e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BHHH maximisation
Log Likelihood: -870
Number of observations: 750
Number of iterations: 14
Exit of MLE: successive function values within tolerance limit
Simulation based on 1 draws
```

Each value in the output represents the WTP estimates for each respective attribute. Note that these WTP estimates are the same as those obtained using the `wtp.gmnl` function. The price coefficient can be obtained using the following transformation:

```
R> -exp(coef(wtps)["het.(Intercept)"])


het.(Intercept)
       -0.611
```

If one requires the standard error for the price coefficient the `deltamethod` function from the **msm** (Jackson 2011) package can be used in the following way:

```
R> library("msm")
R> estmean <- coef(wtps)
R> estvar <- vcov(wtps)
R> se <- deltamethod(~ -exp(x6), estmean, estvar, ses = TRUE)
R> se


[1] 0.056
```

Using the same idea, one can let the WTP vary across individuals. To do so, we can estimate a G-MNL where the parameter of price and γ are fixed as in the previous example:

```
R> start2 <- c(1, coef(wtps), rep(0.1, 5), 0.1, 0)
R> wtps2 <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0 | 0 | 0 | 1,
+    data = Electr0, subset = 1:3000, model = "gmnl", R = 50,
+    fixed = c(TRUE, rep(FALSE, 12), TRUE), panel = TRUE, start = start2,
+    ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n"))
```

```
Estimating GMNL model

R> summary(wtps2)

Model estimated on: Thu Feb 18 10:59:37 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0 |
    0 | 0 | 1, data = ElectrO, subset = 1:3000, model = "gmnl",
    start = start2, ranp = c(cl = "n", loc = "n", wk = "n", tod = "n",
        seas = "n"), R = 50, panel = TRUE, fixed = c(TRUE, rep(FALSE,
        12), TRUE), method = "bfgs")

Frequencies of categories:

    1     2     3     4
0.215 0.303 0.217 0.265

The estimation took: 0h:0m:28s

Coefficients:
                Estimate Std. Error z-value Pr(>|z|)
cl               -0.2727     0.0518   -5.26  1.4e-07 ***
loc               2.1631     0.2452    8.82  < 2e-16 ***
wk                1.9424     0.1935   10.04  < 2e-16 ***
tod              -9.6782     0.2933  -32.99  < 2e-16 ***
seas             -9.8866     0.2772  -35.67  < 2e-16 ***
het.(Intercept)   0.1142     0.1401    0.82     0.41
sd.cl             0.4115     0.0520    7.91  2.7e-15 ***
sd.loc            1.7850     0.2515    7.10  1.3e-12 ***
sd.wk             1.2865     0.2213    5.81  6.1e-09 ***
sd.tod            1.7174     0.2478    6.93  4.2e-12 ***
sd.seas           2.2146     0.3723    5.95  2.7e-09 ***
tau               0.6904     0.1394    4.95  7.3e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization
Log Likelihood: -737
Number of observations: 750
Number of iterations: 143
Exit of MLE: successful convergence
Simulation based on 50 draws
```

Note that the model recast in WTP-space that is implemented in the G-MNL specification above allows the researcher to specify directly the heterogeneity distribution of WTP measures (Sonnier *et al.* 2007). When working in preference space, and then deriving WTP as

a parameter ratio, normally distributed WTP measures can only be derived if the price parameter is fixed (i.e., no unobserved heterogeneity in the marginal utility of income, which is a strong assumption) and the rest of the attributes are assumed to be normally distributed. The problem with all normals in preference space is that the ratio of two normally distributed parameters has a distribution with very long tails and without moments, which leads to unexpected individual-level predictions.

Finally, a WTP-space model with correlated random parameters can be estimated in the following way:

```
R> n_ran <- 5
R> start3 <- c(1, coef(wtps), rep(0.1, .5 * n_ran * (n_ran + 1)), 0.1, 0)
R> wtps3 <- gmnl(choice ~ pf + cl + loc + wk + tod + seas | 0 | 0 | 0 | 1,
+    data = Electr0, subset = 1:3000, model = "gmnl", R = 50,
+    fixed = c(TRUE, rep(FALSE, 22), TRUE), panel = TRUE, start = start3,
+    ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n"),
+    correlation = TRUE)

Estimating GMNL model

R> summary(wtps3)

Model estimated on: Thu Feb 18 11:00:19 2016

Call:
gmnl(formula = choice ~ pf + cl + loc + wk + tod + seas | 0 |
    0 | 0 | 1, data = Electr0, subset = 1:3000, model = "gmnl",
    start = start3, ranp = c(cl = "n", loc = "n", wk = "n", tod = "n",
        seas = "n"), R = 50, correlation = TRUE, panel = TRUE,
    fixed = c(TRUE, rep(FALSE, 22), TRUE), method = "bfgs")

Frequencies of categories:

    1     2     3     4
0.215 0.303 0.217 0.265

The estimation took: 0h:0m:41s

Coefficients:
                Estimate Std. Error z-value Pr(>|z|)
cl               -0.1968     0.0588   -3.35  0.00082 ***
loc               2.1682     0.3228    6.72  1.9e-11 ***
wk                1.8273     0.2523    7.24  4.4e-13 ***
tod              -9.9662     0.3736  -26.68  < 2e-16 ***
seas             -9.9928     0.3535  -28.27  < 2e-16 ***
het.(Intercept)  -0.1385     0.0945   -1.47  0.14268
sd.cl.cl          0.4741     0.0625    7.59  3.2e-14 ***
sd.cl.loc         0.3903     0.2560    1.52  0.12730
```

```
sd.cl.wk            0.6118     0.2444    2.50   0.01231 *
sd.cl.tod          -1.3139     0.2727   -4.82   1.5e-06 ***
sd.cl.seas         -0.2438     0.2655   -0.92   0.35840
sd.loc.loc          2.5819     0.4937    5.23   1.7e-07 ***
sd.loc.wk           1.6391     0.4104    3.99   6.5e-05 ***
sd.loc.tod          1.9078     0.4725    4.04   5.4e-05 ***
sd.loc.seas         0.9679     0.3765    2.57   0.01014 *
sd.wk.wk           -0.7259     0.2745   -2.64   0.00818 **
sd.wk.tod           2.2373     0.3745    5.97   2.3e-09 ***
sd.wk.seas          1.5985     0.3521    4.54   5.6e-06 ***
sd.tod.tod          2.5361     0.4169    6.08   1.2e-09 ***
sd.tod.seas         1.7418     0.3636    4.79   1.7e-06 ***
sd.seas.seas        2.0034     0.2872    6.98   3.0e-12 ***
tau                -0.2557     0.1219   -2.10   0.03595 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization
Log Likelihood: -691
Number of observations: 750
Number of iterations: 174
Exit of MLE: successful convergence
Simulation based on 50 draws
```

Note that `n_ran` is the number of random coefficients, which is used to compute the number of initial values for the **L** matrix in the `start3` vector of initial parameters: let this number be K_a, then the number of elements is equal to $(1/2) \cdot K_a \cdot (K_a + 1)$.

3.8. Individual parameters

Similarly to the **Rchoice** package (Sarrias 2016), **gmnl** also allows the analyst to get the conditional estimates for each individual in the sample (see for example Train 2009; Greene 2012). Using Bayes' theorem we obtain

$$f(\boldsymbol{\beta}_i|\mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}) = \frac{f(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\beta}_i)g(\boldsymbol{\beta}_i|\boldsymbol{\theta})}{\int_{\boldsymbol{\beta}_i} f(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\beta}_i)g(\boldsymbol{\beta}_i|\boldsymbol{\theta})d\boldsymbol{\beta}_i},$$

where $f(\boldsymbol{\beta}_i|\mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta})$ is the distribution of the individual parameters $\boldsymbol{\beta}_i$ conditional on the observed sequence of choices, and $g(\boldsymbol{\beta}_i|\boldsymbol{\theta})$ is the unconditional distribution. The conditional expectation of $\boldsymbol{\beta}_i$ is thus given by:

$$\mathsf{E}\left[\boldsymbol{\beta}_i|\mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}\right] = \frac{\int_{\boldsymbol{\beta}_i} \boldsymbol{\beta}_i f(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\beta}_i)g(\boldsymbol{\beta}_i|\boldsymbol{\theta})d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i} f(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\beta}_i)g(\boldsymbol{\beta}_i|\boldsymbol{\theta})d\boldsymbol{\beta}_i}. \tag{6}$$

The expectation in Equation 6 gives us the conditional mean of the distribution of the random parameters, which can also be interpreted as the posterior distribution of the individual parameters. Simulators for this conditional expectation are presented below for the continuous,

discrete and mixture cases, respectively:

$$
\begin{aligned}
\widehat{\bar{\boldsymbol{\beta}}}_i = \widehat{\mathsf{E}}\left[\boldsymbol{\beta}_i | \mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}\right] &= \frac{\frac{1}{R}\sum_{r=1}^{R}\widehat{\boldsymbol{\beta}}_{ir}\prod_t f(y_{it}|\mathbf{x}_{it},\widehat{\boldsymbol{\beta}}_{ir},\widehat{\boldsymbol{\theta}})}{\frac{1}{R}\sum_{r=1}^{R}\prod_t f(y_{it}|\mathbf{x}_{it},\widehat{\boldsymbol{\beta}}_{ir},\widehat{\boldsymbol{\theta}})}, \\
\widehat{\bar{\boldsymbol{\beta}}}_i = \widehat{\mathsf{E}}\left[\boldsymbol{\beta}_i | \mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}_q\right] &= \frac{\sum_{q=1}^{Q}\widehat{\boldsymbol{\beta}}_q\widehat{w}_{iq}\prod_t f(y_{it}|\mathbf{x}_{it},\widehat{\boldsymbol{\beta}}_{iq},\widehat{\boldsymbol{\theta}}_q)}{\sum_{q=1}^{Q}\widehat{w}_{iq}\prod_t f(y_{it}|\mathbf{x}_{it},\widehat{\boldsymbol{\beta}}_{iq},\widehat{\boldsymbol{\theta}}_q)}, \\
\widehat{\bar{\boldsymbol{\beta}}}_i = \widehat{\mathsf{E}}\left[\boldsymbol{\beta}_i | \mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}_q\right] &= \frac{\sum_{q=1}^{Q}\widehat{w}_{iq}\frac{1}{R}\sum_{r=1}^{R}\widehat{\boldsymbol{\beta}}_{iqr}\prod_t f(y_{it}|\mathbf{x}_{it},\widehat{\boldsymbol{\beta}}_{irq},\widehat{\boldsymbol{\theta}}_q)}{\sum_{q=1}^{Q}\widehat{w}_{iq}\frac{1}{R}\sum_{r=1}^{R}\prod_t f(y_{it}|\mathbf{x}_{it},\widehat{\boldsymbol{\beta}}_{iqr},\widehat{\boldsymbol{\theta}}_q)}.
\end{aligned}
$$

In order to construct the confidence interval for $\widehat{\bar{\boldsymbol{\beta}}}_i$, we can derive an estimator of the conditional variance from the point estimates as follows (Greene 2012, chap. 15):

$$
\widehat{V}_i = \widehat{\mathsf{E}}\left[\boldsymbol{\beta}_i^2 | \mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}\right] - \widehat{\mathsf{E}}\left[\boldsymbol{\beta}_i | \mathbf{y}_i, \mathbf{X}_i, \boldsymbol{\theta}\right]^2. \tag{7}
$$

An approximate normal-based 95% confidence interval can be then constructed as $\widehat{\bar{\boldsymbol{\beta}}}_i \pm 1.96 \times \widehat{V}_i^{1/2}$. The **gmnl** package uses these formulae to compute the individual parameters along with their 95% confidence interval. However, it is worth mentioning that there are two shortcomings with the procedure describe above for computing the conditional variance of $\boldsymbol{\beta}_i$. First, the estimator in Equation 7 is an estimator of the variance of the conditional distribution of $\bar{\boldsymbol{\beta}}_i$, and not an estimator of the sampling variance of the estimator of the expected value. The estimated conditional variance will approach the estimated variance in the population as the number of choice situations faced by each person increases without bound (Hensher, Greene, and Rose 2006). Second, it does not take into account the sampling variability of the parameter estimates.[15]

As an illustration, we can plot the kernel density of the individuals' conditional mean for the `loc` parameter using `Elec.cor` model by typing the following:

```
R> plot(Elec.cor, par = "loc", effect = "ce", type = "density", col = "grey")
```

Figure 1 displays the distribution of the individuals' conditional mean for the parameter of `loc`. The gray area gives us the proportion of individuals with a positive conditional mean. The 95% confidence interval of the conditional mean for the first 30 individuals is shown in Figure 2, which was plotted using the following syntax:[16]

```
R> plot(Elec.cor, par = "loc", effect = "ce", ind = TRUE, id = 1:30)
```

Another important function in **gmnl** is `effect.gmnl`. This function allows the users to get the individuals' conditional mean of both the preference parameters and the willingness-to-pay measures.

For example, one can get the individual conditional mean and standard errors plotted in Figure 2 by typing:

---

[15]Under the Bayesian framework, the estimation of the individual-level estimates fully accounts for uncertainty in the population-level parameters in the estimation routine (see for example Daziano and Achtnicht 2014).

[16]**gmnl** uses **plotrix** package (Lemon 2006) to create the confidence interval graph.
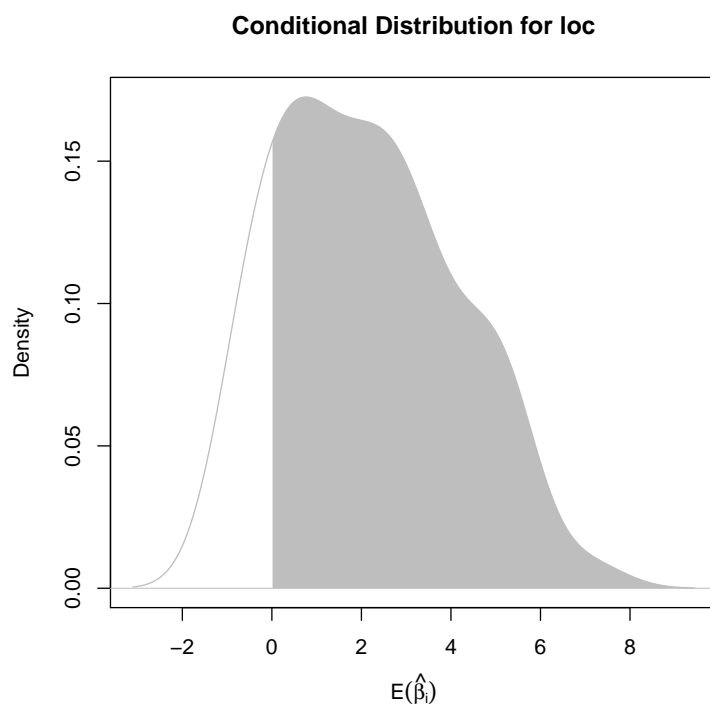
Conditional Distribution for loc

Figure 1: Kernel density of the individuals' conditional mean.

95% Probability Intervals for loc
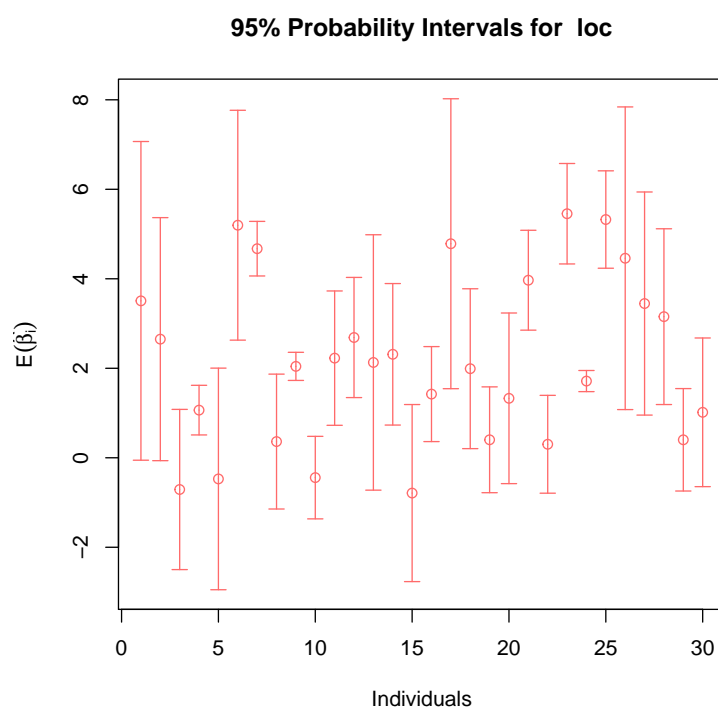

Figure 2: 95% confidence interval for the conditional means.

```
R> bi.loc <- effect.gmnl(Elec.cor, par = "loc", effect = "ce")
R> summary(bi.loc$mean)

  Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
 -0.79    0.42    2.04   2.12    3.46    7.13


R> summary(bi.loc$sd.est)

  Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
 0.113   0.564   0.795  0.866   1.130   1.860
```

The conditional mean of the willingness to pay for "`loc`" (wtp $= \beta_{i,\texttt{loc}}/\beta_{pf}$) for all individuals in the sample can be obtained using:

```
R> wtp.loc <- effect.gmnl(Elec.cor, par = "loc", effect = "wtp", wrt = "pf")
```

Note that the argument `par` is the variable whose parameter goes in the numerator, and the argument `wrt` is a string indicating which parameter goes in the denominator.

```
R> summary(wtp.loc$mean)

  Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
 -8.19   -3.98   -2.35  -2.44   -0.48    0.91


R> summary(wtp.loc$sd.est)

  Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
 0.130   0.648   0.914  0.996   1.300   2.130
```

4. Computational issues

There are some issues about computation and convergence of maximum likelihood worth mentioning before concluding this paper. Regarding models estimated by maximum simulated likelihood, there are at least four factors influencing the estimation of the parameters. First, if the draws used in the estimation are pseudo-random draws, instead of Halton draws, then the parameters might change if the seed is changed. Second, the number of draws used in the simulations is very important in order to have a good approximation of the likelihood. In this paper, we used just a few draws due to time restrictions. Nevertheless, in applied work researchers must use a greater number of draws, especially if pseudo-random draws are used. The 'standard rule' is to increase the number of draws in each run until the estimates stabilize. For a comprehensive review of the impact of the number of draws see for example Bhat (2001) and Sándor and Train (2004).

Another important factor affecting convergence and estimates is the starting values. It is important to stress that the likelihood of the models reviewed in this article are complex and are not globally concave. Thus, poor choice of initial values might lead to local maxima instead

of global maxima, or getting stuck in a flat region of the loglikelihood due to numeric overruns in the Hessian. If this is the case, the user of `gmnl` will receive the following message: `Warning message: In sqrt(diag(vcov(object))): NaNs produced`. We encourage users to try different initial values using the argument `start`.

Finally, the algorithm used for optimization of the MSL is another important factor that users should consider. `gmnl` uses the function `maxLik` to maximize the log-likelihood function, which implements the Newton-Raphson (NR), BGFS, and Berndt-Hall-Hall-Hausman (BHHH) procedures. As default, all models using simulation are estimated using the BFGS algorithm. But if the estimation does not converge, users should try a different algorithm. As a caution note, `gmnl` uses the numerical Hessian if the NR algorithm is used. Thus, it can be very slow compared to the other methods. BHHH is generally faster, but it might fail if the variables have very different scale. The larger the ratio between the largest standard deviation and the smallest standard deviation of the variables, the more problems the user will have with the estimation procedure. Therefore, users should check the variables and re-scale them if necessary, and always look at the output message regarding convergence. It is good practice to use the argument `print.level = 2` to trace the optimization procedure in real time. For more information about the arguments for optimization, type `help(maxLik)`.

5. Conclusions

The package **gmnl** implements the maximum likelihood estimator of random parameter logit models with heterogeneity distributions that can be continuous, discrete, or discrete-continuous mixtures. In this paper we have shown how **gmnl** can fit several extensions to the standard multinomial logit model, including the recently derived mixed-mixed multinomial logit (MM-MNL). To our knowledge there is no other widely available statistical package that has implemented the maximum simulated likelihood estimator of MM-MNL, and we want to highlight that **gmnl** makes use of analytical expressions of the gradient. **gmnl** is also the first implementation in R of the estimator of the scale heterogeneity multinomial logit (S-MNL), the generalized multinomial logit (G-MNL), and the latent class logit (LC). Whereas there are other packages in R for the estimation of MIXL, **gmnl** allows for the inclusion of individual-specific variables to explain the mean of the random parameters for a mixture of deterministic taste variations and unobserved preference heterogeneity. In addition, **gmnl** also implements Johnson S_b heterogeneity distributions.

Another key post-estimation functionality of **gmnl** that we have illustrated in this paper is the derivation of conditional point and interval estimates of either the random parameters or willingness-to-pay measures at the individual level. Random parameter models can be used to make inference on the preference parameters of each individual in the sample, but most packages that estimate MIXL models lack a command to produce individual-level estimates. **gmnl** is able to compute individual parameters for all generalized logit models that are implemented in the package, including G-MNL, MIXL, and LC.

Additional functionalities that we expect to incorporate in the future are the consideration of different choice sets for each individual and the implementation of different methods for the construction of confidence intervals of willingness-to-pay measures.

Acknowledgments

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. `doi:10.1016/s0191-2615(00)00014-x`.

Boxall PC, Adamowicz WL (2002). "Understanding Heterogeneous Preferences in Random Utility Models: A Latent Class Approach." *Environmental and Resource Economics*, **23**(4), 421–446. `doi:10.1023/a:1021351721619`.

Bujosa A, Riera A, Hicks RL (2010). "Combining Discrete and Continuous Representations of Preference Heterogeneity: A Latent Class Approach." *Environmental and Resource Economics*, **47**(4), 477–493. `doi:10.1007/s10640-010-9389-y`.

Croissant Y (2013). **mlogit**: *Multinomial Logit Model*. R package version 0.2-4, URL `https://CRAN.R-project.org/package=mlogit`.

Daly A, Hess S, Train K (2011). "Assuring Finite Moments for Willingness to Pay in Random Coefficient Models." *Transportation*, **39**(1), 19–31. `doi:10.1007/s11116-011-9331-3`.

Daziano RA, Achtnicht M (2014). "Accounting for Uncertainty in Willingness to Pay for Environmental Benefits." *Energy Economics*, **44**, 166–177. `doi:10.1016/j.eneco.2014.03.023`.

Dotson J, Brazell JD, Howell JR, Lenk P, Otter T, MacEachern SN, Allenby GA (2015). "A Probit Model with Structured Covariance for Similarity Effects and Source of Volume Calculations." `doi:10.2139/ssrn.1396232`.

Dumont J, Keller J, Carpenter C (2015). **RSGHB**: *Functions for Hierarchical Bayesian Estimation: A Flexible Approach*. R package version 1.1.2, URL `https://CRAN.R-project.org/package=RSGHB`.

Elff M (2016). **mclogit**: *Mixed Conditional Logit Models*. R package version 0.4.4, URL `https://CRAN.R-project.org/package=mclogit`.

Fiebig DG, Keane MP, Louviere J, Wasi N (2010). "The Generalized Multinomial Logit Model: Accounting for Scale and Coefficient Heterogeneity." *Marketing Science*, **29**(3), 393–421. `doi:10.1287/mksc.1090.0508`.

Gourieroux C, Monfort A (1997). *Simulation-Based Econometric Methods*. Oxford University Press.

Greene WH (2012). *Econometric Analysis*. 7th edition. Prentice Hall.

Greene WH, Hensher DA (2003). "A Latent Class Model for Discrete Choice Analysis: Contrasts With Mixed Logit." *Transportation Research Part B: Methodological*, **37**(8), 681–698. doi:10.1016/s0191-2615(02)00046-2.

Greene WH, Hensher DA (2010). "Does Scale Heterogeneity across Individuals Matter? An Empirical Assessment of Alternative Logit Models." *Transportation*, **37**(3), 413–428. doi:10.1007/s11116-010-9259-z.

Greene WH, Hensher DA (2013). "Revealing Additional Dimensions of Preference Heterogeneity in a Latent Class Mixed Multinomial Logit Model." *Applied Economics*, **45**(14), 1897–1902. doi:10.1080/00036846.2011.650325.

Hajivassiliou VA, Ruud PA (1986). "Classical Estimation Methods for LDV Models Using Simulation." In RF Engle, D McFadden (eds.), *Handbook of Econometrics*, volume 4, chapter 40, pp. 2383–2441. Elsevier.

Hasan A, Wang Z, Mahani AS (2016). "Fast Estimation of Multinomial Logit Models: R Package **mnlogit**." *Journal of Statistical Software*, **75**(3), 1–24. doi:10.18637/jss.v075.i03.

Henningsen A, Toomet O (2011). "**maxLik**: A Package for Maximum Likelihood Estimation in R." *Computational Statistics*, **26**(3), 443–458. doi:10.1007/s00180-010-0217-1.

Hensher DA, Greene WH (2003). "The Mixed Logit Model: The State of Practice." *Transportation*, **30**(2), 133–176. doi:10.1023/a:1022558715350.

Hensher DA, Greene WH, Rose JM (2006). "Deriving Willingness-to-Pay Estimates of Travel-Time Savings from Individual-Based Parameters." *Environment and Planning A*, **38**(12), 2365. doi:10.1068/a37395.

Hess S, Ben-Akiva M, Gopinath D, Walker J (2011). "Advantages of Latent Class over Continuous Mixture of Logit Models." *Working paper*, Institute for Transport Studies, University of Leeds.

Hess S, Rose JM (2012). "Can Scale and Coefficient Heterogeneity Be Separated in Random Coefficients Models?" *Transportation*, **39**(6), 1225–1239. doi:10.1007/s11116-012-9394-9.

Hess S, Stathopoulos A (2013). "Linking Response Quality to Survey Engagement: A Combined Random Scale and Latent Variable Approach." *Journal of Choice Modelling*, **7**, 1–12. doi:10.1016/j.jocm.2013.03.005.

Imai K, Van Dyk DA (2005). "**MNP**: R Package for Fitting the Multinomial Probit Model." *Journal of Statistical Software*, **14**(3), 1–32. doi:10.18637/jss.v014.i03.

Jackson CH (2011). "Multi-State Models for Panel Data: The **msm** Package for R." *Journal of Statistical Software*, **38**(8), 1–29. doi:10.18637/jss.v038.i08.

Keane M, Wasi N (2013). "Comparing Alternative Models of Heterogeneity in Consumer Choice Behavior." *Journal of Applied Econometrics*, **28**(6), 1018–1045. doi:10.1002/jae.2304.

Kleiber C, Zeileis A (2008). *Applied Econometrics with* R. Springer-Verlag, New York. `doi:10.1007/978-0-387-77318-6`.

Lee LF (1992). "On Efficiency of Methods of Simulated Moments and Maximum Simulated Likelihood Estimation of Discrete Response Models." *Econometric Theory*, **8**(4), 518–552.

Leisch F (2004). "**FlexMix**: A General Framework for Finite Mixture Models and Latent Class Regression in R." *Journal of Statistical Software*, **11**(8), 1–18. `doi:10.18637/jss.v011.i08`.

Lemon J (2006). "**plotrix**: A Package in the Red Light District of R." R *News*, **6**(4), 8–12.

Linzer DA, Lewis JB (2011). "**poLCA**: An R Package for Polytomous Variable Latent Class Analysis." *Journal of Statistical Software*, **42**(10), 1–29. `doi:10.18637/jss.v042.i10`.

Martin AD, Quinn KM, Park JH (2011). "**MCMCpack**: Markov Chain Monte Carlo in R." *Journal of Statistical Software*, **42**(9), 22. `doi:10.18637/jss.v042.i09`.

McFadden D (1974). "Conditional Logit Analysis of Qualitative Choice Behavior." In P Zarembka (ed.), *Frontiers in Econometrics*, pp. 105–142. Academic Press, New York.

McFadden D, Train K (2000). "Mixed MNL Models for Discrete Response." *Journal of Applied Econometrics*, **15**(5), 447–470. `doi:10.1002/1099-1255(200009/10)15:5<447::aid-jae570>3.0.co;2-1`.

R Core Team (2017). R*: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Rossi P (2015). **bayesm***: Bayesian Inference for Marketing/Micro-Econometrics*. R package version 3.0-2, URL `https://CRAN.R-project.org/package=bayesm`.

Rossi PE, Allenby GA, McCulloch R (2005). *Bayesian Statistics and Marketing*. John Wiley & Sons, Hoboken. `doi:10.1002/0470863692`.

Ruud PA (2007). "Estimating Mixtures of Discrete Choice Model."

Sándor Z, Train K (2004). "Quasi-Random Simulation of Discrete Choice Models." *Transportation Research Part B: Methodological*, **38**(4), 313–327. `doi:10.1016/s0191-2615(03)00014-6`.

Sarrias M (2016). "Discrete Choice Models with Random Parameters in R: The **Rchoice** Package." *Journal of Statistical Software*, **74**(10), 1–31. `doi:10.18637/jss.v074.i10`.

Sarrias M, Daziano R (2017). **gmnl***: Multinomial Logit Models with Random Parameters*. R package version 1.1-3, URL `https://CRAN.R-project.org/package=gmnl`.

Scarpa R, Thiene M (2005). "Destination Choice Models for Rock Climbing in the Northeastern Alps: A Latent-Class Approach Based on Intensity of Preferences." *Land Economics*, **81**(3), 426–444. `doi:10.3368/le.81.3.426`.

Scarpa R, Thiene M, Train K (2008). "Utility in Willingness to Pay Space: A Tool to Address Confounding Random Scale Effects in Destination Choice to the Alps." *American Journal of Agricultural Economics*, **90**(4), 994–1010. `doi:10.1111/j.1467-8276.2008.01155.x`.

Sermas R (2012). **ChoiceModelR**: *Choice Modeling in R*. R package version 1.2, URL https://CRAN.R-project.org/package=ChoiceModelR.

Shen J (2009). "Latent Class Model or Mixed Logit Model? A Comparison by Transport Mode Choice Data." *Applied Economics*, **41**(22), 2915–2924. doi:10.1080/00036840801964633.

Sonnier G, Ainslie A, Otter T (2007). "Heterogeneity Distributions of Willingness-to-Pay in Choice Models." *Quantitative Marketing and Economics*, **5**(3), 313–331. doi:10.1007/s11129-007-9024-6.

Train K (2009). *Discrete Choice Methods with Simulation.* 2nd edition. Cambridge University Press.

Train K, Weeks M (2005). "Discrete Choice Models in Preference Space and Willingness-to-Pay Space." In R Scarpa, A Alberini (eds.), *Applications of Simulation Methods in Environmental and Resource Economics*, volume 6 of *The Economics of Non-Market Goods and Resources*, pp. 1–16. Springer-Verlag. doi:10.1007/1-4020-3684-1_1.

Train KE (2008). "EM Algorithms for Nonparametric Estimation of Mixing Distributions." *Journal of Choice Modelling*, **1**(1), 40–69. doi:10.1016/s1755-5345(13)70022-8.

Trautmann H, Steuer D, Mersmann O, Bornkamp B (2014). **truncnorm**: *Truncated Normal Distribution.* R package version 1.0-7, URL https://CRAN.R-project.org/package=truncnorm.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S.* 4th edition. Springer-Verlag, New York.

Wedel M, Kamakura WA (2012). *Market Segmentation: Conceptual and Methodological Foundations.* Springer-Verlag.

Yee TW (2010). "The **VGAM** Package for Categorical Data Analysis." *Journal of Statistical Software*, **32**(10), 1–34. doi:10.18637/jss.v032.i10.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. doi:10.18637/jss.v034.i01.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10.

Affiliation:

Mauricio Sarrias
Department of Economics
Universidad Católica del Norte
0610 Avenida Angamos, Antofagasta, Chile
E-mail: msarrias86@gmail.com
URL: https://msarrias.weebly.com/

Ricardo A. Daziano
School of Civil and Environmental Engineering
Cornell University
305 Hollister Hall
Ithaca, NY 14850, United States of America
E-mail : daziano@cornell.edu
URL: http://www.cee.cornell.edu/research/groups/daziano/