KU LEUVEN

# Fitting mixed random regret minimization models using `mixrandregret`.

**UK Stata Meeting - London, 2022.**
**Presenter: Álvaro A. Gutiérrez-Vargas**


- Ziyue Zhu (, )'
- Álvaro A. Gutiérrez-Vargas (, )'
- Martina Vandebroek

Research Centre for Operations Research and Statistics (ORSTAT)

# 1 Outline





KU LEUVEN

# 1 *What is Regret and how to use it for Choice Modeling?*

- From Utility to Regret.
- Regret: Situation where a non-chosen alternative ends up being more attractive than the chosen one
- Individuals are assumed to minimize regret.

Table: Hypothetical Choice Situation

| Attribute \ Route | 1 | 2 | 3 |
|---|---|---|---|
| Travel Time | 23 min. | 27 min. | 35 min. |
| Travel Cost | 6 euros | 4 euros | 3 euros |

If we chose alternative $2$:

- Alternative $1$ is faster...
- Alternative $3$ is cheaper...

$\Rightarrow$ RRM models will (formalize and) minimize this notion of regret!

# 1 Formalization of the later example.

| Attribute \ Route | 1 | 2 | 3 |
|---|---|---|---|
| Travel Time | 23 min. | 27 min. | 35 min. |
| Travel Cost | 6 euros | 4 euros | 3 euros |

Table: Hypothetical Choice Situation

We will denote in the following:

- Decision-makers (referred to $n$) answer $s$ choice situations
- They decide among $J$ alternatives (referred to $i$ or $j$ indistinctly).
- Each alternative is described by $M$ attributes (referred to $m$).
- $x_{ins,m}$ The value of attribute $m$ of alternative $i$ for individual $n$ in choice situation $s$ is denoted by .
- $y_{ins}$ is the response variable that takes the value of 1 when alternative $i$ is chosen by individual $n$ in choice situation $s$ and 0 otherwise.

# 1 RUM vs RRM

- Random Utility Maximization (RUM)

Systematic Utility

Utility

$$\underbrace{U_{ins}}_{\text{Utility}} = \overbrace{V_{ins}}^{\text{Systematic Utility}} + \varepsilon_{ins}$$
$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

- Random Regret Minimization (RRM)

Systematic Regret

Regret

$$RR_{ins} = R_{ins} + \varepsilon_{ins}$$
$$= \sum_{j \neq i}^{J} R_{i \leftrightarrow jns,\boldsymbol{T}} + \sum_{j \neq i}^{J} R_{i \leftrightarrow jns,\boldsymbol{T}} + \varepsilon_{ins}$$

  - The notion of *regret* is characterize by the systematic regret $R_{ins}$.
  - $R_{in}$ is described in terms of ***attribute level regret*** $(R_{i \leftrightarrow jns,m})$.

# 1 The Attribute level regret $R_{i \leftrightarrow jns,m}$

| Attribute \ Route | 1 | 2 | 3 |
|---|---|---|---|
| Travel Time | 23 min. | 27 min. | 35 min. |
| Travel Cost | 6 euros | 4 euros | 3 euros |

- $R_{i \leftrightarrow j,mn}$ describes the pairwise combinations of regret derived from alternatives.

| $(x_{jns,m} - x_{ins,m})$ | Attribute \ Route | $j=1$ | $j=2$ | $j=3$ |
|---|---|---|---|---|
| $(x_{jns,m} - x_{1ns,T})$ | Travel Time | 0 | 4 | 12 |
| $(x_{jns,m} - x_{1ns,C})$ | Travel Cost | 0 | -2 | -3 |
| $(x_{jns,m} - x_{2ns,T})$ | Travel Time | -4 | 0 | 8 |
| $(x_{jns,m} - x_{2ns,C})$ | Travel Cost | 2 | 0 | -1 |
| $(x_{jns,m} - x_{3ns,T})$ | Travel Time | -12 | -8 | 0 |
| $(x_{jns,m} - x_{3ns,C})$ | Travel Cost | 3 | 1 | 0 |

- Takeaway: We will define $R_{i \leftrightarrow jns,m}$ in terms of the attribute differences.

# 1 Classical RRM (Chorus, 2010)

- (Chorus, 2010) proposed the following attribute level regret:

$$R_{i\leftrightarrow jns,m} = \ln\left[1 + \exp\left\{\beta_{n,m} \cdot (x_{jns,m} - x_{ins,m})\right\}\right]$$

- $R_{i\leftrightarrow jns,m}$ compares alternative $i$ with alternative $j$ in attribute $m$.
- $\sum_{j\neq i} R_{i\leftrightarrow jns,m}$ is the equivalent to $x_{ins,m} \times \beta_{n,m}$ in an utilitarian model.
- $\beta_{n,m}$ is the taste parameter of attribute $m$ of individual $n$.

# 1 Classical RRM (Chorus, 2010)

- (Chorus, 2010) proposed the following systematic regret:

Attribute level regret.

$$R_{ins} = \sum_{j \neq i}^{J} \sum_{m=1}^{M} R_{i \leftrightarrow jns,m} = \sum_{j \neq i}^{J} \sum_{m=1}^{M} \ln \left[ 1 + \exp \left\{ \beta_{n,m} \cdot \left( x_{jns,m} - x_{ins,m} \right) \right\} \right] \tag{1}$$

Linear sum of all attribute level regret.

- From our example: $\mathcal{M} = \{T, C\}$, $J = 3$.
- Regret of alternative 1 ($R_{1ns}$) will be described by:

$$\begin{aligned} R_{1ns} &= \sum_{j \neq i}^{3} \sum_{m \in \mathcal{M}} \ln \left[ 1 + \exp \left\{ \beta_{n,m} ( x_{jns,m} - x_{ins,m} ) \right\} \right] \\ &= \ln \left[ 1 + \exp \left\{ \beta_{n,T} \left( x_{2ns,T} - x_{1ns,T} \right) \right\} \right] + \ln \left[ 1 + \exp \left\{ \beta_{n,c} \left( x_{2ns,C} - x_{1ns,C} \right) \right\} \right] \\ &+ \ln \left[ 1 + \exp \left\{ \beta_{n,T} \left( x_{3ns,T} - x_{1ns,T} \right) \right\} \right] + \ln \left[ 1 + \exp \left\{ \beta_{n,C} \left( x_{3ns,C} - x_{1ns,C} \right) \right\} \right] \end{aligned}$$

# 1 Classical RRM (Chorus, 2010): Towards the log-likelihood.

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where $\varepsilon_{ins}$ is a type I Extreme Value i.i.d. error.
2. Acknowledging that the minimization of the random regret is mathematically equivalent to maximizing the negative of the regret.
3. Hence, the probabilities may be derived using the Multinomial Logit:

$$P_{ins} = \frac{\exp\left(-R_{ins}\right)}{\sum_{j=1}^{J} \exp\left(-R_{jns}\right)} \qquad \text{for } i = 1, \ldots, J \tag{2}$$

4. Consequently, the log-likelihood will be described by:

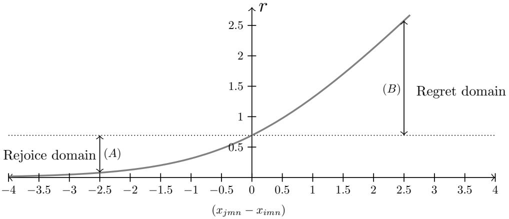
$$\ln L = \sum_{n=1}^{N} \sum_{s=1}^{S} \sum_{i=1}^{J} y_{in} \times \ln\left(P_{ins}\right)$$

# 2 Outline

# 2 Taste Parameter Interpretation in RRM models

- RUM: parameters are interpreted as the change in utility caused by an increase of a particular attribute level.
- RRM: parameters represent the *potential* change in regret associated with comparing a considered alternative with another alternative in terms of the attribute, caused by one unit change in a particular attribute level.
  - For instance: $\widehat{\beta}_m > 0$
    suggests that regret increases as the level of that attribute increases **in a non-chosen alternative**, in comparison to the level of the same attribute in the chosen alternative (e.g: Comfortable level).
  - For instance: $\widehat{\beta}_m < 0$
    suggests that regret decreases as the level of that attribute increases **in a non-chosen alternative**, in comparison to the level of the same attribute in the chosen alternative (e.g: Total Time).
- All in all. the parameters in RUM and RRM, are expected to have the same sign, even though their interpretation is dramatically different.

# 2 Semi-compensatory Behavior and the Compromise Effect




- ***Attribute level regret*** $R_{i\leftrightarrow j,mn}$ with $\beta_m = 1$.
- $(A)$ = *rejoice* and $(B)$ = *regret* on an equal difference of attribute level.
- For an equal difference of the attribute levels $\Rightarrow$ regret $>>>$ rejoice
- Linear RUM models $\Rightarrow$ fully-compensatory model.
- **Compromise Effect**: Alternatives with *"balanced"* performance in all attributes are more attractive than alternatives with a severe poor performance in one attribute.

# 3 Outline

# 3 Mixed Random Regret Minimization Models

New assumptions of the model.

- $\boldsymbol{\beta}_n$ follows a parametric distribution, denoted by $f(\boldsymbol{\beta}|\varphi)$.
- $\varphi$ are the parameters that describe the distribution (e.g., mean and variance of a Normal distribution).
- We define the conditional probability (CP) of the observed sequence of choices of individual $n$ (conditional on knowing $\boldsymbol{\beta}_n$) as:

$$P_n(\boldsymbol{\beta}_n) = \prod_{s=1}^{S} \prod_{i=1}^{J} \{P_{ins}\}^{y_{ins}} \tag{3}$$

- The unconditional probability of the observed sequence of choices is the CP integrated over the entire domain of the distribution of $\boldsymbol{\beta}$

$$\ln L(\boldsymbol{\beta}) = \sum_{n=1}^{N} \ln \left[ \int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi)\, d\boldsymbol{\beta} \right] \tag{4}$$

# 3 Mixed Random Regret Minimization Models

- Given that equation (4) has no closed form we will approximate it using simulations (Train, 2009).

- Hence, we will estimate the model using Maximum Simulate Likelihood where we will maximize the following simulated log-likelihood function:

$$SLL(\beta) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{R} \sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r) \right\} \tag{5}$$

- $R$ is the number of replications and $r$ is the r-th drawn from $f(\boldsymbol{\beta}|\varphi)$.

# 4 Outline

## 4 Individual Level Parameters

- We can obtain estimates of the individual level parameters $(\boldsymbol{\beta}_n)$:

$$\bar{\boldsymbol{\beta}}_n = \frac{\int_{\boldsymbol{\beta}} \boldsymbol{\beta} \times P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}}{\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}} \tag{6}$$

- We will approximate this individual level parameter by simulation using:

$$\check{\boldsymbol{\beta}}_n = \sum_{r=1}^{R} \left( \frac{\boldsymbol{\beta}^r \times P_n(\boldsymbol{\beta}^r)}{\sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r)} \right) \tag{7}$$

- For this estimation we will use the command `mixrbeta` after estimating the population parameters using `mixrandregret`.

# 5 Outline

1. Random Regret Minimization Models
2. Differences between RUM and RRM models.
3. Mixed Random Regret Minimization Models
4. Individual Level Parameters
5. Implementation
   - Syntax
   - Outputs
6. Conclusions

# 5 Syntax

`mixrandregret` is implemented as a Mata-based `gf-0 ml` evaluator. The command allows the inclusion of normally and log-normally distributed random parameters.

`mixrandregret` *depvar* [ *indepvars* ] [ *if* ] [ *in* ] *group(varname)* *alternative(varname)* *rand(varlist)* [ , `id`(*varname*) `basealternative`(*string*) `noconstant` `ln`(*string*) `nrep`(*string*) `burn`(*string*) `robust` `cluster`(*varname*) `level`(#) *maximize_options* ]

---

The command `mixrbeta` can be used after `mixrandregret` to calculate individual-level parameters corresponding to the variables in the specified varname using the method proposed by Train (2009).

`mixrbeta` *varlist* `saving`(filename) [ , `plot` `nrep`(#) `burn`(#) ]

# 5 The Data

- Data from van Cranenburgh (2018): Stated Choice (SC) experiment.

```
. rename (choice)  (choice_w)

. qui reshape long tt tc  , i(obs) j(altern)

. generate choice = 0

. replace  choice = 1 if  choice_w==altern
(1,060 real changes made)

. label define  alt_label 1 "First" 2 "Second" 3 "Third"

. label values  altern alt_label

. list obs altern choice id  tt tc   in 1/6, sepby(obs)
```

| | obs | altern | choice | id | tt | tc |
|---|---|---|---|---|---|---|
| 1. | 1 | First | 0 | 1 | 23 | 6 |
| 2. | 1 | Second | 0 | 1 | 27 | 4 |
| 3. | 1 | Third | 1 | 1 | 35 | 3 |
| 4. | 2 | First | 0 | 1 | 27 | 5 |
| 5. | 2 | Second | 1 | 1 | 35 | 4 |
| 6. | 2 | Third | 0 | 1 | 23 | 6 |

  - Three unlabeled route alternatives ($J = 3$).
  - Described by Travel Cost (***tc***) and Travel Time (***tt***) ($M = 2$).

- Each respondent (***id***) answered a total of 10 choice situations.

# 5 Fixed Parameter RRM model

- First we estimate a fixed parameters regret model.
- As expected, both parameter estimates are negative.

```
. randregret choice  tc tt, gr(obs) alt(altern) rrmfn(classic) ///
> nocons cluster(id)  nolog
------------------------------------------------------------------------------
Fitting Classic RRM Model
------------------------------------------------------------------------------

RRM: Classic Random Regret Minimization Model
Case ID variable: obs                           Number of cases    =       1060
Alternative variable: altern                    Number of obs      =       3180
                                                Wald chi2(2)       =      40.41
Log likelihood = -1118.4784                     Prob > chi2        =     0.0000
                               (Std. Err. adjusted for   106 clusters in id)
------------------------------------------------------------------------------
             |               Robust
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
RRM          |
          tc |   -.417101    .068059    -6.13   0.000    -.5504943   -.2837078
          tt |   -.102813   .0182526    -5.63   0.000    -.1385874   -.0670386
------------------------------------------------------------------------------

. matrix b_rrm = e(b)
```

# 5 Mixed RRM model: Normal Distribution

- Mixed model with total time (tt) being Normally distributed.
- The mean of tt is negative, as expected, however...

```
. mixrandregret choice  tc , gr(obs) alt(altern) rand(tt) id(id) ///
> nocons cluster(id)  nrep(100) from(init_mix_rrm) tech(bhhh)  nolog

Case ID variable: obs                            Number of cases    =       1060
Alternative variable: altern
Random variable(s): tt

                                     (Std. Err. adjusted for  106 clusters in id)

Mixed random regret model                         Number of obs     =      3,180
                                                  Wald chi2(2)      =     605.56
Log likelihood = -770.80502                       Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 OPG
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
          tc |  -1.102093   .0449368   -24.53   0.000    -1.190167   -1.014018
          tt |  -.3528159   .0568895    -6.20   0.000    -.4643172   -.2413145
-------------+----------------------------------------------------------------
SD           |
          tt |   .5107968   .0419348    12.18   0.000     .4286061    .5929875
------------------------------------------------------------------------------

The sign of the estimated standard deviations is irrelevant: interpret them as
being positive
```

# 5 Mixed RRM model: Normal Distribution

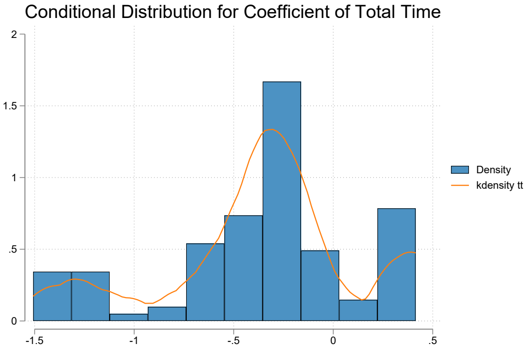
- `mixrbeta` allows us to compute the individual level parameters of Equation (7).
- It creates a new data set with one observation per individual (`id`) and its corresponding parameters estimates.

```
. preserve

. /* Computing Individual Level Parameters */
. qui mixrbeta tt , nrep(100)  replace saving("${graphs_route}\mixRRM_normal_idl")

. use "${graphs_route}\mixRRM_normal_idl" , replace

. list id  tt  in 1/3
```

| | id | tt |
|---|---|---|
| 1. | 1 | .34865652 |
| 2. | 2 | -.05720162 |
| 3. | 3 | .3747384 |

# 5 Mixed RRM model: Individual Level Parameters

- Individual Level Parameters for total time when we assume it as Normally distributed.




- We see some individuals with positive estimates.
- To prevent this from happening we can use a bounded distribution...

# 5 Mixed RRM model: Log-normal Distribution

- We can use a bounded distribution (negative Log-normal) to force that every individual has a negative parameter for total time.

```
. gen ntt = -1 * tt

. mixrandregret choice  tc , gr(obs) alt(altern) rand(ntt ) ln(1) id(id) ///
> nocons cluster(id)  nrep(100) tech(bhhh) from(b_mixrrm) nolog

Case ID variable: obs                               Number of cases     =       1060
Alternative variable: altern
Random variable(s): ntt

                                 (Std. Err. adjusted for  106 clusters in id)

Mixed random regret model                            Number of obs      =      3,180
                                                     Wald chi2(2)       =    1224.65
Log likelihood = -785.40416                          Prob > chi2        =     0.0000

------------------------------------------------------------------------------
             |                OPG
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
          tc |  -1.217628   .0443379   -27.46   0.000    -1.304529   -1.130727
         ntt |  -1.292678   .1549439    -8.34   0.000    -1.596362   -.9889936
-------------+----------------------------------------------------------------
SD           |
         ntt |   1.362446   .1179392    11.55   0.000      1.13129    1.593603
------------------------------------------------------------------------------

The sign of the estimated standard deviations is irrelevant: interpret them as
being positive
```

# 5 Mixed RRM model: Log-normal Distribution

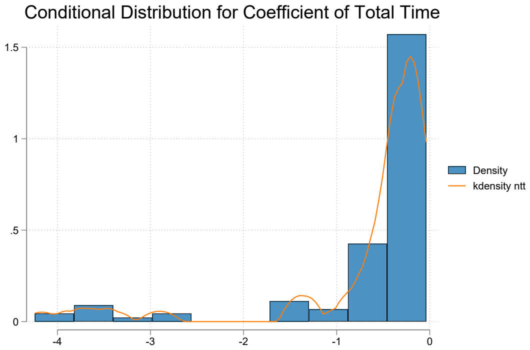
- Similarly, we can compute the individual level parameters for the log-normally distributed variable `tt`.

```
. /* Computing Individual Level Parameters */
. qui mixrbeta ntt , nrep(100)  replace saving("${graphs_route}\mixRRM_ln_idl")
. use "${graphs_route}\mixRRM_ln_idl" , replace
. replace ntt = -1 * ntt /*reverse sign for graph*/
(106 real changes made)
. list id  ntt  in 1/3
```

| | id | ntt |
|---|---|---|
| 1. | 1 | -.04005948 |
| 2. | 2 | -.08146095 |
| 3. | 3 | -.04124789 |

# 5 Mixed RRM model: Log-normal Distribution

- Individual Level Parameters when total time is assumed to be Log-normally distributed.




- Now we observe that the individual level parameters are all negative.

# 5 Mixed RRM model: Log-normal Distribution

- The parameters we estimated are the mean ($\beta_{tt}$) and standard deviation ($s_{tt}$) of the natural logarithm of the total time coefficient.
- Hence, the mean, median and variance of log-normal distributed parameter are equal to $\exp(\beta_{tt})$, $\exp(\beta_{tt} + s_{tt}/2)$ and $\exp(\beta_{tt} + s_{tt}/2) \times \sqrt{\exp(s_{tt}^2) - 1}$, respectively.
- Finally, we can compute them using nlcom.

```
. nlcom (mean_time: -1*exp([Mean]_b[ntt ]+0.5*[SD]_b[ntt ]^2)) ///
> (med_time: -1*exp([Mean]_b[ntt ])) ///
> (sd_time: exp([Mean]_b[ntt ]+0.5*[SD]_b[ntt ]^2) * sqrt(exp([SD]_b[ntt ]^2)-1))

   mean_time:  -1*exp([Mean]_b[ntt ]+0.5*[SD]_b[ntt ]^2)
    med_time:  -1*exp([Mean]_b[ntt ])
     sd_time:  exp([Mean]_b[ntt ]+0.5*[SD]_b[ntt ]^2) * sqrt(exp([SD]_b[ntt ]^2)-1)
```

| choice | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| mean_time | -.6945106 | .1603422 | -4.33 | 0.000 | -1.008776 | -.3802456 |
| med_time | -.2745346 | .0425375 | -6.45 | 0.000 | -.3579065 | -.1911627 |
| sd_time | 1.613861 | .6351531 | 2.54 | 0.011 | .3689834 | 2.858738 |

# 6 Outline

# 6 Conclusions

- The `mixrandregret` command extends its predecessor `randregret` (Gutiérrez-Vargas et al., 2021) by allowing the inclusion of random coefficients in the regret functions.

- The parameters are estimated by Simulated Maximum Likelihood.

- The random parameters can follow either a Normal or Log-normal distribution.

- Additionally, we can compute the individual level parameters using the `mixrbeta` command.

- The programs can be downloaded from Ziyue's Github account.
- The example code used in this presentation is available here.

# 7 Outline

# 8 Bibliography

Chorus, C. G. (2010). A new model of random regret minimization. *European Journal of Transport and Infrastructure Research*, 10(2):181–196.

Gutiérrez-Vargas, I. A., Meulders, M., and Vandebroek, M. (2021). randregret: A command for fitting random regret minimization models using stata. *The Stata Journal*, 21(3):626–658.

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

van Cranenburgh, S. (2018). Small value-of-time experiment, netherlands. 4TU.Centre for Research Data, Dataset https://doi.org/10.4121/uuid:1ccca375-68ca-4cb6-8fc0-926712f50404.