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. Random Regret Minimization Models
2. Differences between RUM and RRM models.
3. Mixed Random Regret Minimization Models
4. Individual Level Parameters
5. Implementation
6. Conclusions
7. Bibliography

1 Outline

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

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

- From Utility to Regret.

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 for some of the attributes.

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 for some of the attributes.
- Individuals are assumed to minimize regret.

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 for some of the attributes.
- Individuals are assumed to minimize regret.
- Consider this hypothetical situation:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

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 for some of the attributes.
- Individuals are assumed to minimize regret.
- Consider this hypothetical situation:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

If we chose alternative 2:

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 for some of the attributes.
- Individuals are assumed to minimize regret.
- Consider this hypothetical situation:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

If we chose alternative 2:

- Alternative 1 is faster...

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 for some of the attributes.
- Individuals are assumed to minimize regret.
- Consider this hypothetical situation:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

If we chose alternative 2:

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

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 for some of the attributes.
- Individuals are assumed to minimize regret.
- Consider this hypothetical situation:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

If we chose alternative 2:

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

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

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).
- They decide among J alternatives (**altern**) (referred to by i or j).

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).
- They decide among J alternatives (**altern**) (referred to by i or j).
- Alternatives are described by M attributes referred to by m (**total_time** and **total_cost**).

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).
- They decide among J alternatives (**altern**) (referred to by i or j).
- Alternatives are described by M attributes referred to by m (**total_time** and **total_cost**).
- $x_{ins,m}$: value of attribute m of alternative i for individual n in choice situation s.

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).
- They decide among J alternatives (**altern**) (referred to by i or j).
- Alternatives are described by M attributes referred to by m (**total_time** and **total_cost**).
- $x_{ins,m}$: value of attribute m of alternative i for individual n in choice situation s. (yes, 4 sub-indexes;

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).
- They decide among J alternatives (**altern**) (referred to by i or j).
- Alternatives are described by M attributes referred to by m (**total_time** and **total_cost**).
- $x_{ins,m}$: value of attribute m of alternative i for individual n in choice situation s. (yes, 4 sub-indexes; I am sorry...)

1 Formalization of the previous example.

Some (unavoidable) notation for the rest of the presentation:

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

- Individuals (**id**) are referred to by n answer s choice situations (**cs**).
- They decide among J alternatives (**altern**) (referred to by i or j).
- Alternatives are described by M attributes referred to by m (**total_time** and **total_cost**).
- $x_{ins,m}$: value of attribute m of alternative i for individual n in choice situation s. (yes, 4 sub-indexes; I am sorry...)
- y_{ins}: response variable (**choice**). It takes the value of 1 when alternative i is chosen by individual n in choice situation s; 0 otherwise..

1 RUM vs RRM

- Random Utility Maximization (RUM)

1 RUM vs RRM

- Random Utility Maximization (RUM)

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$

1 RUM vs RRM

- Random Utility Maximization (RUM)

$$
\begin{aligned}
U_{ins} &= V_{ins} + \varepsilon_{ins} \\
&= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}
\end{aligned}
$$

1 RUM vs RRM

- Random Utility Maximization (RUM)

Utility ←

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$
$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

1 RUM vs RRM

- Random Utility Maximization (RUM)

Utility ← U_{ins} ; V_{ins} → Systematic Utility

$$\begin{aligned} U_{ins} &= V_{ins} + \varepsilon_{ins} \\ &= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins} \end{aligned}$$

1 RUM vs RRM

- Random Utility Maximization (RUM)

Systematic Utility

Utility $\quad U_{ins} = V_{ins} + \varepsilon_{ins}$ Error Term

$$
\begin{aligned}
U_{ins} &= V_{ins} + \varepsilon_{ins} \\
&= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}
\end{aligned}
$$

1 RUM vs RRM

- Random Utility Maximization (RUM)

$$\begin{aligned} U_{ins} &= V_{ins} + \varepsilon_{ins} \\ &= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins} \end{aligned}$$

Utility (U_{ins}) — Systematic Utility (V_{ins}) — Error Term (ε_{ins})

- Random Regret Minimization (RRM)

1 RUM vs RRM

- Random Utility Maximization (RUM)

Systematic Utility

Utility

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$

Error Term

$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

- Random Regret Minimization (RRM)

$$RR_{ins} = R_{ins} + \varepsilon_{ins}$$

1 RUM vs RRM

- Random Utility Maximization (RUM)

Utility ← U_{ins} ; V_{ins} → Systematic Utility ; ε_{ins} → Error Term

$$
\begin{aligned}
U_{ins} &= V_{ins} + \varepsilon_{ins} \\
&= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}
\end{aligned}
$$

- Random Regret Minimization (RRM)

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

1 RUM vs RRM

- Random Utility Maximization (RUM)

Utility ← $U_{ins} = V_{ins} + \varepsilon_{ins}$ → Systematic Utility (V_{ins}), Error Term (ε_{ins})

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$
$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

- Random Regret Minimization (RRM)

Regret ← RR_{ins}

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

1 RUM vs RRM

- Random Utility Maximization (RUM)

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$
$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

Utility (U_{ins}), Systematic Utility (V_{ins}), Error Term (ε_{ins})

- Random Regret Minimization (RRM)

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

Regret (RR_{ins}), Systematic Regret (R_{ins})

1 RUM vs RRM

- Random Utility Maximization (RUM)

Utility ← U_{ins}; V_{ins} → Systematic Utility; ε_{ins} → Error Term

$$
\begin{aligned}
U_{ins} &= V_{ins} + \varepsilon_{ins} \\
&= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}
\end{aligned}
$$

- Random Regret Minimization (RRM)

Regret ← RR_{ins}; R_{ins} → Systematic Regret; ε_{ins} → Error Term (the same btw)

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

1 RUM vs RRM

- Random Utility Maximization (RUM)

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$
$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

Utility (U_{ins}); Systematic Utility (V_{ins}); Error Term (ε_{ins})

- Random Regret Minimization (RRM)

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

Regret (RR_{ins}); Systematic Regret (R_{ins}); Error Term (the same btw) (ε_{ins})

- The notion of *regret* is characterized by the systematic regret R_{ins}.

1 RUM vs RRM

- Random Utility Maximization (RUM)

Utility ← U_{ins}; V_{ins} → Systematic Utility; ε_{ins} → Error Term

$$U_{ins} = V_{ins} + \varepsilon_{ins}$$
$$= \beta_{n,\boldsymbol{T}} \times x_{ins,\boldsymbol{T}} + \beta_{n,\boldsymbol{C}} \times x_{ins,\boldsymbol{C}} + \varepsilon_{ins}$$

- Random Regret Minimization (RRM)

Regret ← RR_{ins}; R_{ins} → Systematic Regret; ε_{ins} → Error Term (the same btw)

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

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

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

- Using the same example as before:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

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

- Using the same example as before:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

- $R_{i\leftrightarrow jns,m}$ corresponds to pairwise combination of regret for alternatives i and j for individual n on attribute m in choice situation s.

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

- Using the same example as before:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

- $R_{i\leftrightarrow jns,m}$ corresponds to pairwise combination of regret for alternatives i and j for individual n on attribute m in choice situation s.

$(x_{jns,m} - x_{ins,m})$	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

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

- Using the same example as before:

altern	total_time	total_cost
First	23	6
Second	27	4
Third	35	3

- $R_{i\leftrightarrow jns,m}$ corresponds to pairwise combination of regret for alternatives i and j for individual n on attribute m in choice situation s.

$(x_{jns,m} - x_{ins,m})$	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:

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

1 Classical RRM (Chorus, 2010)

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

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

- As we saw, $R_{i\leftrightarrow jns,m}$ compares alternative i with alternative j in attribute m.

1 Classical RRM (Chorus, 2010)

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

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

- As we saw, $R_{i\leftrightarrow jns,m}$ 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 utility model.

1 Classical RRM (Chorus, 2010)

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

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

- As we saw, $R_{i \leftrightarrow jns,m}$ 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 utility model.
- In both cases, $\beta_{n,m}$ is the taste parameter of attribute m of individual n.

1 Classical RRM (Chorus, 2010)

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

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

- As we saw, $R_{i\leftrightarrow jns,m}$ 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 utility model.
- In both cases, $\beta_{n,m}$ is the taste parameter of attribute m of individual n.
- However, they have drastically different interpretation

1 Classical RRM (Chorus, 2010)

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

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

- As we saw, $R_{i\leftrightarrow jns,m}$ 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 utility model.
- In both cases, $\beta_{n,m}$ is the taste parameter of attribute m of individual n.
- However, they have drastically different interpretation(more on that later).

1 Classical RRM (Chorus, 2010)

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

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

1 Classical RRM (Chorus, 2010)

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

Sum over alternatives.

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

1 Classical RRM (Chorus, 2010)

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

Sum over alternatives.

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

Sum over attributes.

1 Classical RRM (Chorus, 2010)

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

Sum over alternatives.

Attribute level regret.

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

Sum over attributes.

1 Classical RRM (Chorus, 2010)

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

Sum over alternatives.

Attribute level regret.

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

Sum over attributes.

- In our example: $M = 2$ (Time and Cost) and $J = 3$.

1 Classical RRM (Chorus, 2010)

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

Sum over alternatives.

Attribute level regret.

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

Sum over attributes.

  - In our example: $M = 2$ (Time and Cost) and $J = 3$.
  - Regret of alternative 1 (R_{1ns}) will be given by:

1 Classical RRM (Chorus, 2010)

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

Sum over alternatives.

Attribute level regret.

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

Sum over attributes.

- In our example: $M = 2$ (Time and Cost) and $J = 3$.
- Regret of alternative 1 (R_{1ns}) will be given by:

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

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

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where ε_{ins} is a type I Extreme Value i.i.d. error.

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

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where ε_{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.

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

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where ε_{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 can be derived using the Multinomial Logit:

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

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where ε_{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 can be derived using the Multinomial Logit:

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

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

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where ε_{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 can be derived using the Multinomial Logit:

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

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

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

1. Defining $RR_{ins} = R_{ins} + \varepsilon_{ins}$, where ε_{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 can be derived using the Multinomial Logit:

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

4. Consequently, the log-likelihood will be given 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

1. Random Regret Minimization Models
2. Differences between RUM and RRM models.
   Taste Parameter Interpretation in RRM models
3. Mixed Random Regret Minimization Models
4. Individual Level Parameters
5. Implementation
6. Conclusions

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.

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$ 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

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$ 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).

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$ 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 if $\widehat{\beta}_{n,m} < 0$

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$ 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 if $\widehat{\beta}_{n,m} < 0$ 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

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$ 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 if $\widehat{\beta}_{n,m} < 0$ 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).

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 caused by one unit increase in a particular attribute level in one of the non-chosen alternatives.
  - For instance if $\widehat{\beta}_{n,m} > 0$ 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 if $\widehat{\beta}_{n,m} < 0$ 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 drastically different.

3 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
6. Conclusions
7. Bibliography

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

- $\boldsymbol{\beta}_n = (\beta_{n,1}, \ldots, \beta_{n,m})$ follows a parametric distribution: $f\left(\boldsymbol{\beta}|\varphi\right)$.

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

- $\boldsymbol{\beta}_n = (\beta_{n,1}, \ldots, \beta_{n,m})$ follows a parametric distribution: $f(\boldsymbol{\beta}|\varphi)$.
- φ are the parameters that describe the distribution (e.g., mean and variance of a Normal distribution)

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

- $\boldsymbol{\beta}_n = (\beta_{n,1}, \ldots, \beta_{n,m})$ follows a parametric distribution: $f(\boldsymbol{\beta}|\varphi)$.
- φ 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:

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

- $\boldsymbol{\beta}_n = (\beta_{n,1}, \ldots, \beta_{n,m})$ follows a parametric distribution: $f\left(\boldsymbol{\beta}|\varphi\right)$.
- φ 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:

$$P_n(\boldsymbol{\beta}_n) = \prod_{s=1}^{S} \prod_{i=1}^{J} \{P_{ins}\}^{y_{ins}} \tag{2}$$

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

- $\boldsymbol{\beta}_n = (\beta_{n,1}, \ldots, \beta_{n,m})$ follows a parametric distribution: $f(\boldsymbol{\beta}|\varphi)$.
- φ 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:

$$P_n(\boldsymbol{\beta}_n) = \prod_{s=1}^{S} \prod_{i=1}^{J} \{P_{ins}\}^{y_{ins}} \tag{2}$$

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

3 Mixed Random Regret Minimization Models

Additional assumptions of the mode: Individual Level Parameters.

- $\boldsymbol{\beta}_n = (\beta_{n,1}, \ldots, \beta_{n,m})$ follows a parametric distribution: $f\left(\boldsymbol{\beta}|\varphi\right)$.
- φ 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:

$$P_n(\boldsymbol{\beta}_n) = \prod_{s=1}^{S} \prod_{i=1}^{J} \{P_{ins}\}^{y_{ins}} \tag{2}$$

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

$$\ln L(\boldsymbol{\beta}) = \sum_{n=1}^{N} \ln \left[\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f\left(\boldsymbol{\beta}|\varphi\right) d\boldsymbol{\beta} \right] \tag{3}$$

3 Mixed Random Regret Minimization Models

- Given that equation (3) has no closed form we will approximate it using simulations (Train, 2009).

3 Mixed Random Regret Minimization Models

- Given that equation (3) has no closed form we will approximate it using simulations (Train, 2009).
- Hence, we will estimate the model using Maximum Simulated Likelihood where we will maximize the following simulated log-likelihood function:

3 Mixed Random Regret Minimization Models

- Given that equation (3) has no closed form we will approximate it using simulations (Train, 2009).
- Hence, we will estimate the model using Maximum Simulated Likelihood where we will maximize the following simulated log-likelihood function:

$$SLL(\beta) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{R} \sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r) \right\} \tag{4}$$

3 Mixed Random Regret Minimization Models

- Given that equation (3) has no closed form we will approximate it using simulations (Train, 2009).
- Hence, we will estimate the model using Maximum Simulated Likelihood where we will maximize the following simulated log-likelihood function:

$$SLL(\beta) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{R} \sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r) \right\} \tag{4}$$

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

4 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
6. Conclusions
7. Bibliography

4 Individual Level Parameters

- We can compute the mean $(\bar{\beta}_n)$ of the distribution of individuals that choose a given sequence of choices as:

4 Individual Level Parameters

- We can compute the mean $(\bar{\boldsymbol{\beta}}_n)$ of the distribution of individuals that choose a given sequence of choices as:

$$\bar{\boldsymbol{\beta}}_n = \frac{\int_{\boldsymbol{\beta}} \boldsymbol{\beta} \times P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}}{\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}} \tag{5}$$

4 Individual Level Parameters

- We can compute the mean ($\bar{\boldsymbol{\beta}}_n$) of the distribution of individuals that choose a given sequence of choices as:

$$\bar{\boldsymbol{\beta}}_n = \frac{\int_{\boldsymbol{\beta}} \boldsymbol{\beta} \times P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}}{\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}} \tag{5}$$

- Since the integrals of equation (5) do not have a close form solution. We will approximate $\bar{\boldsymbol{\beta}}_n$ by simulation using:

4 Individual Level Parameters

- We can compute the mean ($\bar{\boldsymbol{\beta}}_n$) of the distribution of individuals that choose a given sequence of choices as:

$$\bar{\boldsymbol{\beta}}_n = \frac{\int_{\boldsymbol{\beta}} \boldsymbol{\beta} \times P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}}{\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}} \tag{5}$$

- Since the integrals of equation (5) do not have a close form solution. We will approximate $\bar{\boldsymbol{\beta}}_n$ by simulation using:

$$\check{\boldsymbol{\beta}}_n = \sum_{r=1}^{R} \left(\frac{\boldsymbol{\beta}^r \times P_n(\boldsymbol{\beta}^r)}{\sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r)} \right) \tag{6}$$

4 Individual Level Parameters

- We can compute the mean ($\bar{\boldsymbol{\beta}}_n$) of the distribution of individuals that choose a given sequence of choices as:

$$\bar{\boldsymbol{\beta}}_n = \frac{\int_{\boldsymbol{\beta}} \boldsymbol{\beta} \times P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}}{\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\varphi) d\boldsymbol{\beta}} \tag{5}$$

- Since the integrals of equation (5) do not have a close form solution. We will approximate $\bar{\boldsymbol{\beta}}_n$ by simulation using:

$$\check{\boldsymbol{\beta}}_n = \sum_{r=1}^{R} \left(\frac{\boldsymbol{\beta}^r \times P_n(\boldsymbol{\beta}^r)}{\sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r)} \right) \tag{6}$$

- For this estimation we will use the command `mixrbeta` after estimating the population parameters using `mixrandregret` (Zhu, 2022).

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` (Zhu, 2022) 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*]

---

5 Syntax

mixrandregret (Zhu, 2022) 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 *varlist* using equation (6).

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

5 The Data

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

5 The Data

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

```
. list id cs altern total_time total_cost choice in 1/6, sepby(cs) ab(10) noo
```

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

5 The Data

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

```
. list id cs altern total_time total_cost choice in 1/6, sepby(cs) ab(10) noo
```

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

  - Three unlabeled route alternatives ($J = 3$).

5 The Data

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

```
. list id cs altern total_time total_cost choice in 1/6, sepby(cs) ab(10) noo
```

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

  - Three unlabeled route alternatives ($J = 3$).
  - Described by **total_time** and **total_cost** ($M = 2$).

5 The Data

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

```
. list id cs altern total_time total_cost choice in 1/6, sepby(cs) ab(10) noo
```

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

  - Three unlabeled route alternatives ($J = 3$).
  - Described by **total_time** and **total_cost** ($M = 2$).

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

5 The Data

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

```
. list id cs altern total_time total_cost choice in 1/6, sepby(cs) ab(10) noo
```

id	cs	altern	total_time	total_cost	choice
1	1	First	23	6	0
1	1	Second	27	4	0
1	1	Third	35	3	1
1	2	First	27	5	0
1	2	Second	35	4	1
1	2	Third	23	6	0

  - Three unlabeled route alternatives ($J = 3$).
  - Described by **total_time** and **total_cost** ($M = 2$).
- Each respondent (**id**) answered a total of 10 choice situations.
- Variables **choice** and **altern** allows us to identify each choice.

5 Fixed Parameter RRM model

- First we estimate a fixed parameters RRM model.

5 Fixed Parameter RRM model

- First we estimate a fixed parameters RRM model.

```
. randregret choice total_time total_cost , gr(cs) alt(altern) rrmfn(classic) ///
> nocons cluster(id)  nolog
-----------------------------------------------------------------------------
Fitting Classic RRM Model
-----------------------------------------------------------------------------

RRM: Classic Random Regret Minimization Model

Case ID variable: cs                           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          |
  total_time |   -.102813   .0182526    -5.63   0.000    -.1385874   -.0670386
  total_cost |   -.417101    .068059    -6.13   0.000    -.5504943   -.2837078
-----------------------------------------------------------------------------

. matrix b_rrm = e(b)
```

5 Fixed Parameter RRM model

- First we estimate a fixed parameters RRM model.

```
. randregret choice total_time total_cost , gr(cs) alt(altern) rrmfn(classic) ///
> nocons cluster(id)  nolog
--------------------------------------------------------------------------------
Fitting Classic RRM Model
--------------------------------------------------------------------------------

RRM: Classic Random Regret Minimization Model

Case ID variable: cs                          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          |
  total_time |   -.102813   .0182526    -5.63   0.000    -.1385874   -.0670386
  total_cost |   -.417101    .068059    -6.13   0.000    -.5504943   -.2837078
--------------------------------------------------------------------------------

. matrix b_rrm = e(b)
```

- As expected, both parameter estimates are negative.

5 Mixed RRM model: Normal Distribution

- `total_time` assumed to be normally distributed: $\beta_T \sim \mathcal{N}(\mu_T, \sigma_T)$

5 Mixed RRM model: Normal Distribution

- `total_time` assumed to be normally distributed: $\beta_T \sim \mathcal{N}(\mu_T, \sigma_T)$
- We estimate the two parameters of a normal distribution: μ_T and σ_T

5 Mixed RRM model: Normal Distribution

- **total_time** assumed to be normally distributed: $\beta_T \sim \mathcal{N}(\mu_T, \sigma_T)$
- We estimate the two parameters of a normal distribution: μ_T and σ_T

```
. mixrandregret choice  total_cost , gr(cs) alt(altern) rand(total_time) id(id) ///
> nocons cluster(id)  nrep(500) from(init_mix_rrm) tech(bhhh)  nolog

Case ID variable: cs                              Number of cases    =       1060
Alternative variable: altern
Random variable(s): total_time

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

Mixed random regret model                          Number of obs      =      3,180
                                                   Wald chi2(2)       =     606.11
Log likelihood = -771.05731                        Prob > chi2        =     0.0000

------------------------------------------------------------------------------
             |                 OPG
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
  total_cost |  -1.102136   .0449727   -24.51   0.000    -1.190281   -1.013991
  total_time |  -.3580736   .0581449    -6.16   0.000    -.4720355   -.2441117
-------------+----------------------------------------------------------------
SD           |
  total_time |   .5068268    .041366    12.25   0.000      .425751    .5879027
------------------------------------------------------------------------------

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

. matrix b_mixrrm = e(b)
```

5 Mixed RRM model: Normal Distribution

- **total_time** assumed to be normally distributed: $\beta_T \sim \mathcal{N}(\mu_T, \sigma_T)$
- We estimate the two parameters of a normal distribution: μ_T and σ_T

```
. mixrandregret choice  total_cost , gr(cs) alt(altern) rand(total_time) id(id) ///
> nocons cluster(id)  nrep(500) from(init_mix_rrm) tech(bhhh)  nolog

Case ID variable: cs                              Number of cases    =       1060
Alternative variable: altern
Random variable(s): total_time

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

Mixed random regret model                          Number of obs      =      3,180
                                                   Wald chi2(2)       =     606.11
Log likelihood = -771.05731                        Prob > chi2        =     0.0000

------------------------------------------------------------------------------
             |                 OPG
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
  total_cost |  -1.102136   .0449727   -24.51   0.000    -1.190281   -1.013991
  total_time |  -.3580736   .0581449    -6.16   0.000    -.4720355   -.2441117
-------------+----------------------------------------------------------------
SD           |
  total_time |   .5068268    .041366    12.25   0.000      .425751    .5879027
------------------------------------------------------------------------------

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

. matrix b_mixrrm = e(b)
```

- The mean of **total_time** is negative, as expected.

5 Mixed RRM model: Normal Distribution

- We can compute the individual level parameters of Equation (6) using `mixrbeta`.

5 Mixed RRM model: Normal Distribution

- We can compute the individual level parameters of Equation (6) using `mixrbeta`.
- `mixrbeta` creates a new data set with one observation per individual (**`id`**) and its corresponding parameters estimates.

5 Mixed RRM model: Normal Distribution

- We can compute the individual level parameters of Equation (6) using `mixrbeta`.
- `mixrbeta` creates a new data set with one observation per individual (**id**) and its corresponding parameters estimates.

```
. preserve

. /* Computing Individual Level Parameters */
. qui mixrbeta total_time , nrep(500)  replace saving("${graphs_route}\mixRRM_normal_idl")

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

. list id  total_time  in 1/5

     +-----------------+
     | id   total_time |
     |-----------------|
  1. |  1    .37640482 |
  2. |  2   -.05517462 |
  3. |  3    .37672848 |
  4. |  4    .38495822 |
  5. |  5    .37607978 |
     +-----------------+
```

5 Mixed RRM model: Normal Distribution

- We can compute the individual level parameters of Equation (6) using `mixrbeta`.
- `mixrbeta` creates a new data set with one observation per individual (`id`) and its corresponding parameters estimates.

```
. preserve
. /* Computing Individual Level Parameters */
. qui mixrbeta total_time , nrep(500)  replace saving("${graphs_route}\mixRRM_normal_idl")
. use "${graphs_route}\mixRRM_normal_idl" , replace
. list id  total_time  in 1/5
```

	id	total_time
1.	1	.37640482
2.	2	-.05517462
3.	3	.37672848
4.	4	.38495822
5.	5	.37607978

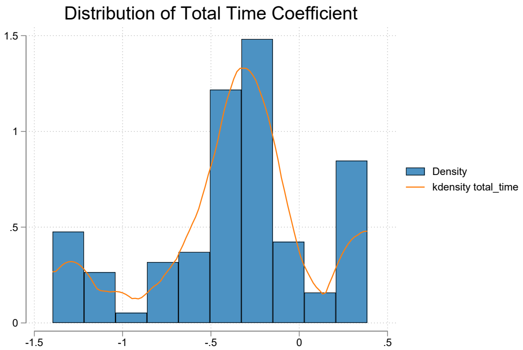
- We observe that some of the individuals has a positive coefficient for Total Time (`total_time`).

5 Mixed RRM model: Individual Level Parameters

- We can plot the individual level parameters for `total_time` when we assume it as Normally distributed.

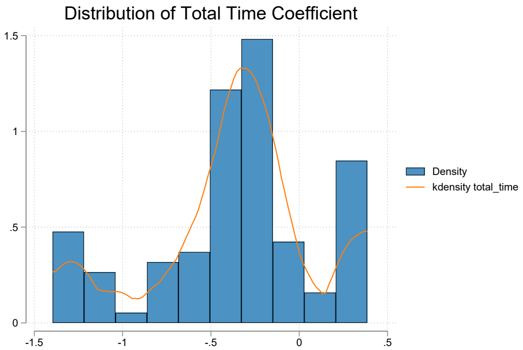
5 Mixed RRM model: Individual Level Parameters

- We can plot the individual level parameters for `total_time` when we assume it as Normally distributed.



5 Mixed RRM model: Individual Level Parameters

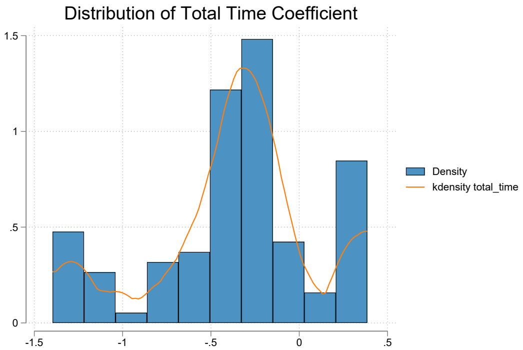
- We can plot the individual level parameters for **`total_time`** when we assume it as Normally distributed.



- We see some individuals with positive estimates.

5 Mixed RRM model: Individual Level Parameters

- We can plot the 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

- **total_time** assumed Log-normal: $\beta_T \sim -1 \times \exp\left(\mathcal{N}\left(\mu_T, \sigma_T\right)\right)$

5 Mixed RRM model: Log-normal Distribution

- **`total_time`** assumed Log-normal: $\beta_T \sim -1 \times \exp\left(\mathcal{N}\left(\mu_T, \sigma_T\right)\right)$
- Given that **`total_time`** is expected to be negative, we created (**`ntt=-total_time`**), since the log-normal distribution implies that the coefficient is positive.

5 Mixed RRM model: Log-normal Distribution

- **total_time** assumed Log-normal: $\beta_T \sim -1 \times \exp\left(\mathcal{N}\left(\mu_T, \sigma_T\right)\right)$
- Given that **total_time** is expected to be negative, we created (**ntt=-total_time**), since the log-normal distribution implies that the coefficient is positive.

```
. gen ntt = -1 * total_time

. mixrandregret choice  total_cost , gr(cs) alt(altern) rand(ntt ) ln(1) id(id) ///
> nocons cluster(id)  nrep(500) tech(bhhh) from(b_mixrrm) nolog

Case ID variable: cs                             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)      =    1230.55
Log likelihood = -785.27671                       Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 OPG
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
  total_cost |  -1.217682   .0442047   -27.55   0.000    -1.304321   -1.131042
         ntt |  -1.312285   .1562202    -8.40   0.000    -1.618471   -1.006099
-------------+----------------------------------------------------------------
SD           |
         ntt |   1.363632   .1185994    11.50   0.000     1.131181    1.596082
------------------------------------------------------------------------------

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` using mixrbeta.

5 Mixed RRM model: Log-normal Distribution

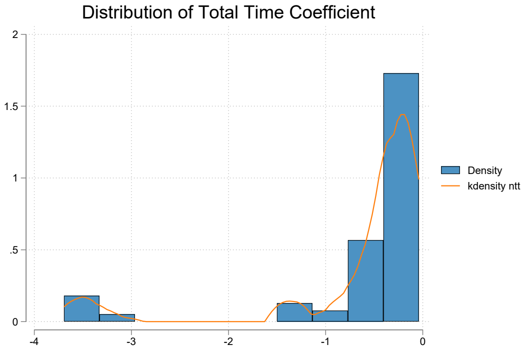
- Similarly, we can compute the individual level parameters for the log-normally distributed variable **tt** using <u>mixrbeta</u>.

```
. /* Computing Individual Level Parameters */
. qui mixrbeta ntt , nrep(500)  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 1/5
```

	id	ntt
1.	1	-.04032598
2.	2	-.08142616
3.	3	-.04047817
4.	4	-.04110615
5.	5	-.04025335

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 (β_T) and standard deviation (σ_T) of the natural logarithm of the total time coefficient.

5 Mixed RRM model: Log-normal Distribution

- The parameters we estimated are the mean (β_T) and standard deviation (σ_T) of the natural logarithm of the total time coefficient.
- Hence, the mean, median and variance of log-normal distributed parameter are equal to $\exp(\beta_T)$, $\exp(\beta_T + \sigma_T/2)$ and $\exp(\beta_T + \sigma_T/2) \times \sqrt{\exp(\sigma_T^2) - 1}$, respectively.

5 Mixed RRM model: Log-normal Distribution

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

5 Mixed RRM model: Log-normal Distribution

- The parameters we estimated are the mean (β_T) and standard deviation (σ_T) of the natural logarithm of the total time coefficient.
- Hence, the mean, median and variance of log-normal distributed parameter are equal to $\exp(\beta_T)$, $\exp(\beta_T + \sigma_T/2)$ and $\exp(\beta_T + \sigma_T/2) \times \sqrt{\exp(\sigma_T^2) - 1}$, 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	-.682127	.1587961	-4.30	0.000	-.9933616	-.3708923
med_time	-.2692041	.0420551	-6.40	0.000	-.3516307	-.1867776
sd_time	1.588122	.6295756	2.52	0.012	.3541763	2.822067

6 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
6. Conclusions
7. Bibliography

6 Conclusions

- The `mixrandregret` (Zhu, 2022) command extends its predecessor `randregret` (Gutiérrez-Vargas et al., 2021) by allowing the inclusion of random coefficients in the regret functions.

6 Conclusions

- The `mixrandregret` (Zhu, 2022) 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 Maximum Simulated Likelihood.

6 Conclusions

- The `mixrandregret` (Zhu, 2022) 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 Maximum Simulated Likelihood.
- The random parameters can follow either a Normal or Log-normal distribution.

6 Conclusions

- The `mixrandregret` (Zhu, 2022) 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 Maximum Simulated 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.

6 Conclusions

- The `mixrandregret` (Zhu, 2022) 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 Maximum Simulated 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.

6 Conclusions

- The mixrandregret (Zhu, 2022) 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 Maximum Simulated 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

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

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, A. 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.

Zhu, Z. (2022). *mixrandregret: A command for fitting mixed random regret minimization models using Stata*. Master's thesis, KU Leuven.

GitHub with Slides + Example code here:

Thanks ✌