

A “quick” overview of stated choice methods

Danny Campbell

Economics Division, University of Stirling

Northern Lights Economics
Tromsø, 6–8 December 2017

Talk overview

Background

- Stated choice experiments
- Trading-off between attributes
- Flexibility and some basic design considerations
- Utility maximisation

Probabilistic choice models

- Multinomial logit
- Model output
- Limitations of the multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model

Overview

Appendix: a practical demonstration

- Data generating process
- Multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model
- Appendix overview

Talk overview

Background

- Stated choice experiments
- Trading-off between attributes
- Flexibility and some basic design considerations
- Utility maximisation

Probabilistic choice models

- Multinomial logit
- Model output
- Limitations of the multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model

Overview

Appendix: a practical demonstration

- Data generating process
- Multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model
- Appendix overview

Stated choice experiments

Stated choice experiments are now widely used to estimate non-market values associated with environmental goods and market values for agri-food products.

Respondents are asked a series of choices and their answers are modelled to determine how they trade-off between the attributes.

Which policy do you prefer?

Policy A

Water quality



Recreation



€8

Policy B

Water quality



Recreation



€4

Stated choice experiments

Stated choice experiments are now widely used to estimate non-market values associated with environmental goods and market values for agri-food products.

Respondents are asked a series of choices and their answers are modelled to determine how they trade-off between the attributes.

Which policy do you prefer?

Policy A

Water quality



Recreation



€8

Policy B

Water quality



Recreation



€4

Stated choice experiments

Stated choice experiments are now widely used to estimate non-market values associated with environmental goods and market values for agri-food products.

Respondents are asked a series of choices and their answers are modelled to determine how they trade-off between the attributes.

Which policy do you prefer?

Policy A

Water quality



Recreation



€8

Policy B

Water quality



Recreation



€30

Stated choice experiments

Stated choice experiments are now widely used to estimate non-market values associated with environmental goods and market values for agri-food products.

Respondents are asked a series of choices and their answers are modelled to determine how they trade-off between the attributes.

Which policy do you prefer?

Policy A

Water quality



Recreation



€10

Policy B

Water quality



Recreation



€20

Stated choice experiments

Stated choice experiments are now widely used to estimate non-market values associated with environmental goods and market values for agri-food products.

Respondents are asked a series of choices and their answers are modelled to determine how they trade-off between the attributes.

Which policy do you prefer?

Policy A

Water quality



Recreation



€6

Policy B

Water quality



Recreation



€7

Alternatives

The **alternatives** are the options that people are asked to choose between.

They can be labelled (e.g., car, bus, train) or generic (e.g., policy A, policy B and policy C).

- ▶ The choice of whether to use label names or generic labels, depends on the topic under investigation.

In environmental case-studies it is typical to include a **do nothing** or **status quo** option.

- ▶ This option normally describes the situation that would be the case under a 'no policy' situation.
- ▶ It is important, to ensure the results are consistent with welfare economics.

Attributes and attribute levels

Each alternative describes a hypothetical environmental outcome, and is made up of different combinations of **attributes** and **levels**.

- ▶ In an environmental economics setting, the attributes may include such features as water quality, recreational opportunities, and cost.

The attribute levels describe the possible values, outcomes or interventions associated with each attribute.

Establishing preferences

By studying choices, we are able to say something about preferences. This is because we expect people to choose the alternatives that they most prefer.

- ▶ As economists we call this **utility**—people choose the options that give them the highest utility.

By looking further at the attributes and levels that describe the alternatives it is possible to determine the extent to which of them affects choice.

- ▶ Therefore, we get an idea into the preferences that people hold for each of the attribute levels.

Trading-off between attributes

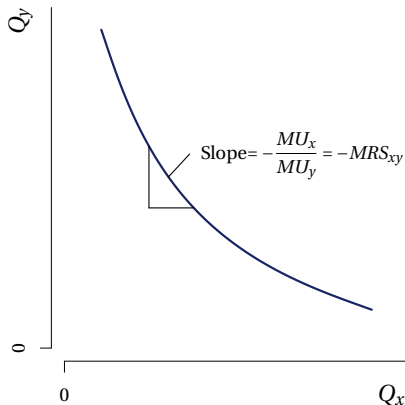
Because it is generally not possible to “have your cake and eat it”, when we study the choices made by people we can determine the extent to which they are willing to give up one thing in order to gain another.

The degree to which someone is willing to sacrifice one attribute in order to receive more of another attribute is called their **marginal rate of substitution**.

Marginal rate of substitution

The slope of an **indifference curve** indicates the rate at which an individual is willing to trade one attribute for another.

- ▶ **Marginal utility** (MU) is the additional utility gained from the last unit of an attribute or service consumed.
- ▶ The ratio of the marginal utilities of two attribute is called the **marginal rate of substitution** (MRS), and it represents the number of units of attribute y people are willing to give up to obtain one more unit of attribute x .



Marginal rate of substitution

One of the attributes used to describe the alternatives is its cost.

The inclusion of this is important (and indeed necessary) for valuation.

By including a cost attribute it is possible to determine how much money they are willing to forgo to obtain one more unit of another attribute (i.e., their **marginal willingness to pay (WTP)**).

Flexibility

The ability to uncover preferences and WTP has meant that choice experiments are a useful tool for resource economists.

- ▶ Not restricted to market goods—can accommodate non-market goods (very important for environmental economists).
- ▶ Can concurrently derive WTP estimates for a number of attributes and policy options.
- ▶ Can predict choice probabilities (e.g., how many people will visit a forest park, demand for marine ecosystem services etc.)

Widely used in a range of fields, environmental economics, agri-food marketing, transport economics and health economics.

However, the analysis (and design) requires some careful consideration.

Some basic design considerations

Stated choice experiments can place significant cognitive burden on respondents—important considerations include:

- ▶ Number of attributes.
- ▶ Number of alternatives.
- ▶ Rank *vis-à-vis* rate *vis-à-vis* choose.
- ▶ Labelled *vis-à-vis* generic labels.
- ▶ Status-quo (and/or opt-out) alternative.
- ▶ Number of choice tasks.
- ▶ Order and direction of alternatives.

The **experimental design** (the mapping of attributes and levels into choice tasks) also requires careful consideration.

Utility maximisation

As stated earlier, the basic expectation is that an individual chooses the alternative that provides them with the highest utility among those in the choice set:

For our notation, we write the utility respondent n derives from alternative i as U_{ni} :

$$U_{ni} = V_{ni} + \varepsilon_{ni},$$

where V_{ni} is the systematic (i.e. observable) part of utility:

$$V_{ni} = \beta x_{ni},$$

and ε_{ni} is the random utility component.

Note that the terms x are the attributes of the choice, and β are their coefficients (which can be interpreted, as the marginal effect the attribute on utility).

Utility maximisation

Individual n chooses alternative i over alternative j , if and only if:

$$U_{ni} > U_{nj}.$$

Due to the presence of the random components (i.e., ε_i), it is only possible to make statements about choice outcomes up to a probability of occurrence (i.e., the probability that respondent n will choose alternative i over alternative j):

$$\Pr(i_n) = \Pr(U_{ni} > U_{nj})$$

$$\Pr(V_{ni} + \varepsilon_{ni} > V_{nj} + \varepsilon_{nj})$$

$$\Pr(V_{ni} - V_{nj} > \varepsilon_{nj} - \varepsilon_{ni})$$

Assumptions need to be made regarding the distribution of the random components, ε_i and ε_j .

Talk overview

Background

- Stated choice experiments
- Trading-off between attributes
- Flexibility and some basic design considerations
- Utility maximisation

Probabilistic choice models

- Multinomial logit
- Model output
- Limitations of the multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model

Overview

Appendix: a practical demonstration

- Data generating process
- Multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model
- Appendix overview

Multinomial logit

Multinomial logit (MNL) is the most commonly applied discrete choice model.

MNL models are the starting point of all discrete choice analysis.

The MNL model is based on the assumption that the distribution of the unobserved components associated with alternative i (ε_i) is **independently and identically distributed** (*iid*) to the unobserved components associated with alternative j (ε_j).

An extreme value distribution is also assumed for the unobserved components.

Multinomial logit specification

Given the assumption that the error terms are *iid* with extreme value, the probability that respondent n will choose alternative i from all J alternatives is:

$$\Pr(i_n | \beta, x_n) = \frac{\exp(\mu V_{ni})}{\sum_{j=1}^J \exp(\mu V_{nj})},$$

where μ is a strictly positive scale parameter, inversely proportional to the deviation of the error distribution:

$$\text{var}(\varepsilon_{ni}) = \frac{\pi^2}{6\mu^2}.$$

Note that in order to facilitate estimation μ is typically normalised to one, so that it drops out. But in some cases where you believe that different sub-samples have different variance, you may want to estimate a separate μ for each sub-sample (but where at least one of them is fixed to one).

Utility parameters

Recall: $V_{ni} = \beta x_{ni}$.

The rationale for estimating any discrete choice model is the derivation of the β 's.

The β 's can be interpreted as the marginal contribution of the attribute to utility (i.e., the marginal utility or weight that the attributes plays in the utility function).

The objective is to find the values of the β 's that best describe choice (i.e. the likelihood).

Maximum likelihood estimation

To find the values of the β 's that best describe choice, **maximum likelihood estimation** procedures are used.

Rather than maximise the likelihood, it is usually more convenient to **maximise the log of the likelihood**.

- ▶ The log-likelihood is always negative!
 - ▶ If we predict choices perfectly, the likelihood is 1 (i.e., log-likelihood is 0).
 - ▶ If we predict choices very poorly, the likelihood will approach 0 (i.e., log-likelihood approaches $-\infty$).

Behavioural outputs

The β 's indicate the influence of the attributes on utility.

But utility is a rather arbitrary concept.

A utility has no meaningful measurement.

Elasticities of choice are of greater interest and relevance—i.e., how does the change in one or more attribute change the probability of choosing a particular alternative.

Marginal willingness to pay

Usually of greater interest in environmental economics, is how much of one attribute respondents are willing to sacrifice in order to acquire more of another (i.e., the **marginal rate of substitution**).

Given that a monetary attribute is usually included, it is possible to determine how much money a respondent is willing to pay (or accept) to receive more (or less) of an environmental good.

Marginal WTP is calculated by dividing the coefficient for a non-cost attribute (β) by the negative of the coefficient for the price attribute ($\beta_{\$}$):

$$\text{Marginal WTP} = -\frac{\beta}{\beta_{\$}}$$

Limitations of the MNL model

The *iid* assumption of the MNL model gives rise to the **independence from irrelevant alternatives (IIA)** property.

- ▶ The probabilities of choosing one alternative over another is unaffected by the presence or absence of any additional alternatives.
- ▶ This assumption is often inappropriate when the alternatives are close substitutes.
 - ▶ For example, suppose the travel options are by car or red bus and that the probability is 50% for each. Now imagine a blue bus is introduced. Under the IIA property the relative probability between car and red bus should not change if another alternative is added. However, introducing a different colour of bus is unlikely to change the probability of car. It will probably stay at 50% and 25% will travel in each of the two buses. The IIA is unlikely to hold since the relative probability between car and red bus has changed.

Limitations of the MNL model

MNL models do not capture random taste variations (nor account correctly for repeated observations from the same respondent).

MNL models do not allow for correlation among alternatives.

The MNL model assumes rational decision-makers that adhere to the **random utility theory**.

A number of alternative models have been developed to accommodate the limitations of MNL models.

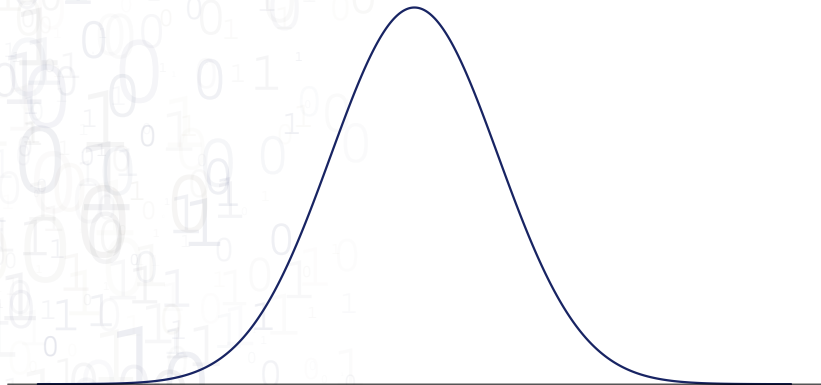
Preference heterogeneity

Should we assume respondents share the same preferences?



Preference heterogeneity

Preferences may follow a normal distribution.



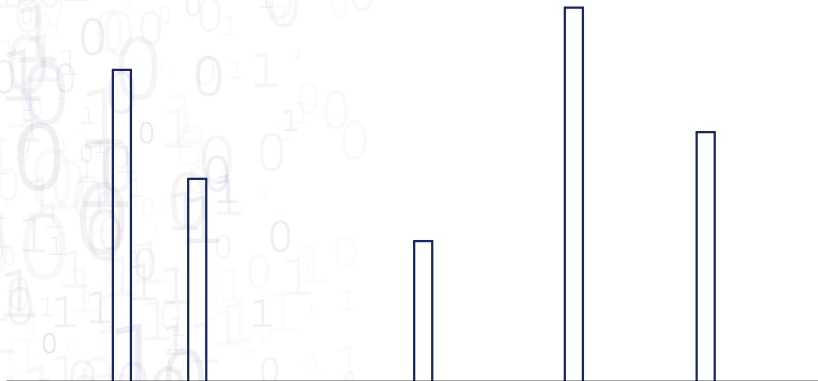
Preference heterogeneity

Preferences may follow a uniform distribution.



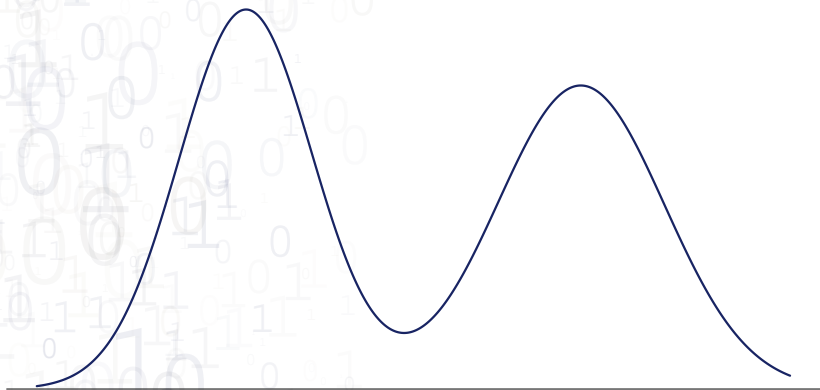
Preference heterogeneity

Preferences may follow a discrete distribution.



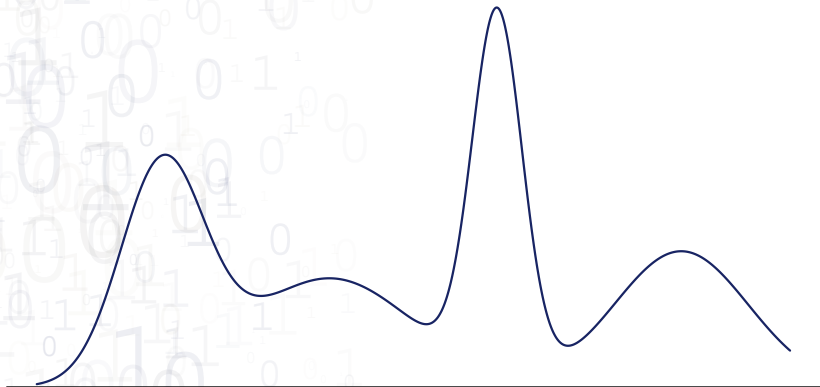
Preference heterogeneity

Preferences may follow a bi-modal distribution.



Preference heterogeneity

Preferences may follow a complex distribution.



Preference heterogeneity

While the MNL model directly uncovers estimates of preferences for each of the attributes, it does so in a manner that assumes that all respondents share the same preferences.

While this assumption may hold in some cases, for a variety of reasons one may postulate the hypothesis that it is more likely that the preferences will be heterogeneous across respondents

If this is the case, it is important to uncover the **preference heterogeneity**.

- ▶ A model that makes unrealistically simple assumptions can lead to considerable bias, poor prediction, and missed opportunities for insight.

Preference heterogeneity

While the MNL model directly uncovers estimates of preferences for each of the attributes, it does so in a manner that assumes that all respondents share the same preferences.

While this assumption may hold in some cases, for a variety of reasons one may postulate the hypothesis that it is more likely that the preferences will be heterogeneous across respondents

If this is the case, it is important to uncover the **preference heterogeneity**.

- ▶ A model that makes unrealistically simple assumptions can lead to considerable bias, poor prediction, and missed opportunities for insight.

Information processing and heuristics

Is it realistic to expect respondents to process all information?

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

Some may consider only Attributes 1, 3, 4 and Price.

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

Others may consider only Attributes 1, 3 and Price.

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

Worryingly, some may even ignore the Price attribute!

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

Some may be unwilling to change.

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

While others may dislike the current situation.

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

Some may exclude alternatives from their consideration set.

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

Some may use multiple processing strategies.

	Option A	Option B	Option C	Status-quo
Attribute 1	High	Low	Medium	Low
Attribute 2	Low	Medium	High	High
Attribute 3	Medium	High	Low	Medium
Attribute 4	High	Medium	Low	Low
Price	€10	€20	€5	€0
Choice?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Information processing and heuristics

While the random utility maximization model is widely used, it rests on the assumption of compensatory (indirect) utility functions.

However, individuals tend to fall back on simplifying heuristics and rules of thumb to better manage complex and difficult choice situations.

- ▶ Such behaviours represent deviations from random utility theory.
- ▶ Unless the models properly address the actual choice behaviour inferences about individuals' preferences will be misguided.

Instead, we might have to depart from this convenient assumption and allow for models that can capture **boundedly rational behaviour**.

Information processing and heuristics

While the random utility maximization model is widely used, it rests on the assumption of compensatory (indirect) utility functions.

However, individuals tend to fall back on simplifying heuristics and rules of thumb to better manage complex and difficult choice situations.

- ▶ Such behaviours represent deviations from random utility theory.
- ▶ Unless the models properly address the actual choice behaviour inferences about individuals' preferences will be misguided.

Instead, we might have to depart from this convenient assumption and allow for models that can capture **boundedly rational behaviour**.

Alternatives to the multinomial logit model

In this **quick** overview, three more flexible specifications are considered.

1. Latent class model.
2. Random parameters logit model.
3. Attribute non-attendance model.

While the first two specifications explain preference heterogeneity, the third specification allows for information processing strategies.

Note that there are a lot more specifications!

Latent class model

In a latent class model, the number of possible values for the parameter coefficients is finite.

Therefore, latent class models are especially suited for identifying and accommodating segments of respondents based on their underlying preferences.

This can be accommodated by retrieving Q class-specific parameters, β_q .

A respondent's true preferences cannot be known with certainty and, thus, remains latent.

To work around this, based on observed choice behaviour, the presence of each vector of parameters can be established up to a probability, with the full probability per respondent allocated across all Q classes.

Latent class model

The unconditional probability of observing β_q is denoted by π_q subject to $\sum_{q=1}^Q \pi_q = 1$ (i.e., the prior likelihood of competing marginal utilities being their actual marginal utilities).

With this, the probability of a sequence of choices made by respondent n , y_n , can then be rewritten as:

$$\Pr(y_n | \beta_q, x_n, Q) = \sum_{q=1}^Q \pi_q \prod_{t=1}^{T_n} \frac{\exp(\beta_q x_{nit})}{\sum_{j=1}^J \exp(\beta_q x_{njt})}.$$

Random parameters logit model

Heterogeneity across respondents (indexed by n) can be captured by treating $\tilde{\beta}_n$ as continuously distributed random terms entering the utility function.

Denote the joint density of $[\beta_{n1}, \beta_{n2}, \dots, \beta_{nK}]$ by $f(\Theta_n|\Omega)$, where Θ_n represents the vector comprised of the random parameters and Ω denotes the parameters of these distributions.

The unconditional choice probability is the integral of the MNL formula over all possible values of $\tilde{\beta}_n$:

$$\Pr(y_n|x_n, \Omega) = \int \prod_{t=1}^{T_n} \frac{\exp(\tilde{\beta}_n x_{nit})}{\sum_{j=1}^J \exp(\tilde{\beta}_n x_{njt})} f(\Theta_n|\Omega) d(\Theta_n).$$

Random parameters logit model

The choice probabilities in the random parameters model cannot be calculated exactly because the integrals do not have a closed form.

So, these are approximated by simulating the log-likelihood with R quasi-random draws:

$$\hat{\Pr}(y_n|x_n, \Omega) = \frac{1}{R} \sum_{r=1}^R \Pr(y_n|x_n, \beta_r),$$

where β_r is a vector drawn from $f(\Theta_n|\Omega)$.

Attribute non-attendance model

Attribute non-attendance behaviour can be accommodated using a discrete mixtures approach.

This means that the number of possible values for the parameter coefficients is finite.

This can be achieved by specifying the vector of parameters relating to attribute k , β_k , as having $M = 2$ mass points.

The unconditional probability of observing β_k^m is denoted by π_k^m ,

subject to $\sum_{m=1}^M \pi_k^m = 1$.

Attribute non-attendance model

The number of possible segments with K attributes is $Q = \prod_{k=1}^K 2_k$ (i.e., 2^k).

Each segment, $q = \{1, 2, 3, \dots, Q\}$, implies a different combination of attribute marginal utilities for each of the K attributes.

The unconditional probability of observing combination q is the product of the unconditional probabilities of observing each π_k^m .

- ▶ This is denoted using ϕ_q .
- ▶ For example, the probability of observing β_1^1 , β_2^2 and β_3^1 is given by $\phi_q = \pi_1^1 \times \pi_2^2 \times \pi_3^1$.

Attribute non-attendance model

This means that the probability of a sequence of choices can then be rewritten as:

$$\Pr(y_n|x_n, \beta^q, Q) = \sum_{q=1}^Q \phi_q \prod_{t=1}^{T_n} \frac{\exp(\beta^q x_{nit})}{\sum_{j=1}^J \exp(\beta^q x_{njt})}.$$

To accommodate attribute non-attendance, it is necessary to set $\beta_k^1 = 0$ and allow β_k^2 to be freely estimateable.

Note that there is a confounding issue, since it is not possible to distinguish between preference heterogeneity and processing heterogeneity. However, this can be, somewhat, reduced by concurrently addressing preference heterogeneity. This is not done here.

Talk overview

Background

- Stated choice experiments
- Trading-off between attributes
- Flexibility and some basic design considerations
- Utility maximisation

Probabilistic choice models

- Multinomial logit
- Model output
- Limitations of the multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model

Overview

Appendix: a practical demonstration

- Data generating process
- Multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model
- Appendix overview

Overview

Stated choice experiments are a useful method for valuing market and non-market goods.

The design and analysis of this data requires some careful thought.

- ▶ While it is appropriate to begin with simple models, they are often too restrictive to fully explain preferences and the way that people make decisions.
- ▶ In particular, it is important to explain preference heterogeneity and information processing strategies.
- ▶ Not doing so leads to biases and missed opportunities for insight.

Danny Campbell

Economics Division
Stirling Management School
University of Stirling
Scotland

Email: danny.campbell@stir.ac.uk

www.dannycampbell.me

Twitter: @DCampbell_econ

Talk overview

Background

- Stated choice experiments
- Trading-off between attributes
- Flexibility and some basic design considerations
- Utility maximisation

Probabilistic choice models

- Multinomial logit
- Model output
- Limitations of the multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model

Overview

Appendix: a practical demonstration

- Data generating process
- Multinomial logit model
- Latent class model
- Random parameters logit model
- Attribute non-attendance model
- Appendix overview

A practical demonstration

In this appendix, some R code to estimate stated choice models on your own data is provided.*

Hopefully this will:

- ▶ Give you a better understanding of the practice of some econometric models used to analysis stated choice experiments.
- ▶ Improve your ability to critically assess the results obtained from stated choice models.

* R is a language and environment for statistical computing and graphics and is freely available from <https://cran.r-project.org/>.
Northern Lights Economics, 6–8 December 2017, Tromsø, Danny Campbell

Data generating process

For the purposes of illustration, an experimental design is obtained using the `Lma.design` and `make.design.matrix` functions in the package **support.CEs***.

```
## package for generating experimental design
library(support.CEs)
## number of non-status-quo alternatives
non.sq.alts <- 2
## number of experimental design blocks
blocks <- 2
## generate an experimental design using the LMA method
exp.design <- Lma.design(attribute.names = list(AttA = c("0", "1"), AttB = c("0",
"1"), AttC = c("0", "1"), Cost = c("2", "4", "6", "8")), nalternatives = non.sq.alts,
nblocks = blocks, row.renames = FALSE, seed = 1)
## number of choice tasks completed per respondent
n.tasks = exp.design$design.information$nquestions
## convert the design into a matrix for analysis
design.matrix <- make.design.matrix(choice.experiment.design = exp.design, optout = FALSE,
continuous.attributes = c("AttA", "AttB", "AttC", "Cost"), unlabeled = TRUE)
## specify number of respondents
N <- 250
## generate alternatives for each respondent
alternative.1 <- matrix(rep(t(as.matrix(design.matrix[design.matrix$ALT == 1, 5:8])),
N/blocks), ncol = 4, byrow = TRUE)
alternative.2 <- matrix(rep(t(as.matrix(design.matrix[design.matrix$ALT == 2, 5:8])),
N/blocks), ncol = 4, byrow = TRUE)
```

* <http://www.jstatsoft.org/v50/c02/>.

Data generating process

This produces a pedagogical choice experiment design containing 2000 individual choice sets with two non-status-quo alternatives per choice task.

The design is divided into two blocks leading to a panel of one thousand choice tasks per individual.

Next we replicate the choice observations for a synthetic sample of 250 individuals.

Data generating process

In generating the choices, the data generating process assumes that the utility, U , for alternative j in choice task t for individual n is given by:

$$U_{njt} = \beta_n x_{njt} + c_j - \delta_{jn} \infty + \varepsilon_{njt},$$

where:

- ▶ x is a vector of the attributes levels for the respective alternative;
- ▶ β_n is the associated vector of individual-specific marginal utilities;
- ▶ c is an alternative specific constant;
- ▶ δ_{jn} is an individual-specific dummy variable to signify if alternative j is excluded (denoted by δ_j^0) from or included (denoted by δ_j^1) in the individual's actual consideration set; and,
- ▶ ε_{njt} is random deviate from an *iid* type I extreme value (EV1) distributed error term, with constant variance equal to $\pi^2/6$.

Data generating process

The individual counterfactual choices are produced by identifying the alternative associated with the largest utility value.

In this data generating process, the true marginal utilities for the non-cost attributes follow a mixture of normal distributions, with probability density function:

$$\phi(x) = \sum_{k=1}^K \omega_k \phi_k(x),$$

where

- ▶ $\phi_1(x), \phi_2(x), \dots, \phi_K(x)$ are a finite set of probability density functions; and,
- ▶ $\omega_1(x), \omega_2(x), \dots, \omega_K(x)$ are weights, such that ω_k and $\sum_{k=1}^K \omega_k = 1$.

Data generating process

For all non-cost attributes, a mixture of three (i.e., $K = 3$) normal distributions are assumed.

The true marginal utilities are drawn using the parameters in this table.

	AttA	AttB	AttC
$\mu_{k=1}$	1.2	1.1	0.7
$\sigma_{k=1}$	0.3	0.2	0.1
$\omega_{k=1}$	0.5	0.6	0.4
$\mu_{k=2}$	0.6	1.0	1.3
$\sigma_{k=2}$	0.1	0.2	0.3
$\omega_{k=2}$	0.4	0.2	0.3
$\mu_{k=3}^*$	0.0	0.0	0.0
$\sigma_{k=3}^*$	1×10^{-10}	1×10^{-10}	1×10^{-10}
$\omega_{k=3}$	0.1	0.2	0.3

* Dirac delta function represented as a normal distribution with a tall spike at the origin.

Data generating process

The true marginal utilities for the cost attribute follow a negative log-normal distribution.

- ▶ The mean and variance of the underlying normal distribution used are -1.6 and 0.4 respectively.

The alternative specific constants for alternatives 1 and 2 are fixed to 0.2 and 0.1 respectively (i.e., where the status-quo alternative is set to be the baseline).

The actual consideration sets are defined assuming $\Pr(\delta_{j=1}^1) = 0.9$, $\Pr(\delta_{j=2}^1) = 0.8$ and $\Pr(\delta_{j=3}^1) = 0.4$.

Finally, for the purposes of this pedagogical demonstration, all parameter distributions and consideration set processing strategies are generated independently.

Data generating process

As a first step, specify the parameters for the data generating process:

```
## specify parameters for simulation
AttA.parameters <- c(1.2, 0.3, 0.5, 0.6, 0.1, 0.4, 0, 1e-10, 0.1)
AttB.parameters <- c(1.1, 0.2, 0.6, 1, 0.2, 0.2, 0, 1e-10, 0.2)
AttC.parameters <- c(0.7, 0.1, 0.4, 1.3, 0.3, 0.3, 0, 1e-10, 0.3)
Cost.parameters <- c(-1.6, 0.4)
ASC.parameters <- c(0.2, 0.1, 0)
delta.parameters <- c(0.9, 0.8, 0.4)
```

Simulate preferences for AttA (drawn from a mixture of three normal distributions):

```
set.seed(1)
rand.A <- runif(N)
set.seed(2)
AttA.i <- AttA.parameters[1] + AttA.parameters[2] * rnorm(N)
set.seed(3)
AttA.ii <- AttA.parameters[4] + AttA.parameters[5] * rnorm(N)
set.seed(4)
AttA.iii <- AttA.parameters[7] + AttA.parameters[8] * rnorm(N)
AttA.dist <- AttA.i * (rand.A <= AttA.parameters[3]) + AttA.ii * (rand.A > AttA.parameters[3] &
  rand.A <= (AttA.parameters[3] + AttA.parameters[6])) + AttA.iii * (rand.A > (1 -
  AttA.parameters[9]))
```

Data generating process

Simulate preferences for AttB (drawn from a mixture of three normal distributions):

```
set.seed(5)
rand.B <- runif(N)
set.seed(6)
AttB.i <- AttB.parameters[1] + AttB.parameters[2] * rnorm(N)
set.seed(7)
AttB.ii <- AttB.parameters[4] + AttB.parameters[5] * rnorm(N)
set.seed(8)
AttB.iii <- AttB.parameters[7] + AttB.parameters[8] * rnorm(N)
AttB.dist <- AttB.i * (rand.B <= AttB.parameters[3]) + AttB.ii * (rand.B > AttB.parameters[3] &
  rand.B <= (AttB.parameters[3] + AttB.parameters[6])) + AttB.iii * (rand.B > (1 -
  AttB.parameters[9]))
```

Simulate preferences for AttC (drawn from a mixture of three normal distributions):

```
set.seed(9)
rand.C <- runif(N)
set.seed(10)
AttC.i <- AttC.parameters[1] + AttC.parameters[2] * rnorm(N)
set.seed(11)
AttC.ii <- AttC.parameters[4] + AttC.parameters[5] * rnorm(N)
set.seed(12)
AttC.iii <- AttC.parameters[7] + AttC.parameters[8] * rnorm(N)
AttC.dist <- AttC.i * (rand.C <= AttC.parameters[3]) + AttC.ii * (rand.C > AttC.parameters[3] &
  rand.C <= (AttC.parameters[3] + AttC.parameters[6])) + AttC.iii * (rand.C > (1 -
  AttC.parameters[9]))
```

Data generating process

Simulate preferences for Cost (drawn from a negative log normal distribution):

```
set.seed(13)
Cost.dist <- -exp(Cost.parameters[1] + Cost.parameters[2] * rnorm(N))
```

Simulate the actual consideration sets:

```
set.seed(14)
consider.alt1 <- ifelse(runif(N) <= delta.parameters[1], 1, 0)
set.seed(15)
consider.alt2 <- ifelse(runif(N) <= delta.parameters[2], 1, 0)
set.seed(16)
consider.alt3 <- ifelse(runif(N) <= delta.parameters[3], 1, 0)
```

Data generating process

Using the simulated distributions, it is now possible to generate the individual counterfactual choices:

```
## generate observable component of utility
v1 <- rowSums(matrix(rep(cbind(AttA.dist, AttB.dist, AttC.dist, Cost.dist), each = n.tasks),
  ncol = 4) * alternative.1) + (1 - rep(consider.alt1, each = n.tasks)) * log(1e-100) +
  ASC.parameters[1]
v2 <- rowSums(matrix(rep(cbind(AttA.dist, AttB.dist, AttC.dist, Cost.dist), each = n.tasks),
  ncol = 4) * alternative.2) + (1 - rep(consider.alt2, each = n.tasks)) * log(1e-100) +
  ASC.parameters[2]
v3 <- (1 - rep(consider.alt3, each = n.tasks)) * log(1e-100) + ASC.parameters[3]
## package for generating generalized extreme value distributions
library(fExtremes)
## generate unobservable component of utility
set.seed(17)
error1 <- as.vector(rgev(N * n.tasks, xi = 0, mu = 0, beta = 1))
set.seed(18)
error2 <- as.vector(rgev(N * n.tasks, xi = 0, mu = 0, beta = 1))
set.seed(19)
error3 <- as.vector(rgev(N * n.tasks, xi = 0, mu = 0, beta = 1))
## utility
utility1 <- v1 + error1
utility2 <- v2 + error2
utility3 <- v3 + error3
## choice variable
choice <- as.vector(apply(rbind(utility1, utility2, utility3), 2, which.max))
```

Multinomial logit model

Recall the MNL choice probability:

$$\Pr(i_n | \beta, x_n) = \frac{\exp(\beta x_{ni})}{\sum_{j=1}^J \exp(\beta x_{nj})}$$

We can write the log-likelihood for this as a function in R, and label it `MNL.LL`:

```
MNL.LL <- function(coeff) {  
  util1 <- coeff[1:4] %*% t(alternative.1) + coeff[5]  
  util2 <- coeff[1:4] %*% t(alternative.2) + coeff[6]  
  choice.prob <- (exp(util1) * (choice == 1) + exp(util2) * (choice == 2) + 1 *  
    (choice == 3)) / (exp(util1) + exp(util2) + 1)  
  log(colProds(matrix(choice.prob, ncol = N)))  
}
```

Multinomial logit model

For estimating the choice model, we can use the `maxLik` functions in the package **maxLik***

```
## load necessary packages
library(maxLik)
library(sandwich)
library(lmtest)

## find the parameters that maximise the log-likelihood
MNL.result = maxLik(MNL.LL, start = c(0.4, 0.3, 0.3, -0.05, 0.2, 0.1), method = "BHHH")
## label the coefficients
names(MNL.result$estimate) <- c("AttA", "AttB", "AttC", "Cost", "ASC1", "ASC2")
```

*<http://dx.doi.org/10.1007/s00180-010-0217-1>.

Multinomial logit model

The MNL model produces a log-likelihood of -1,848.05:

```
## print the log-likelihood
MNL.result$maximum

## [1] -1848.051
```

Multinomial logit model

The parameters that maximise the MNL model log-likelihood:

```
## print the estimated parameters
coeftest(MNL.result, vcov = sandwich)

##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## AttA  0.628429  0.070780  8.8787 < 2.2e-16 ***
## AttB  0.551959  0.086444  6.3852 1.712e-10 ***
## AttC  0.376382  0.073431  5.1256 2.965e-07 ***
## Cost -0.139262  0.017429 -7.9901 1.348e-15 ***
## ASC1  1.212238  0.116594 10.3971 < 2.2e-16 ***
## ASC2  0.870327  0.116634  7.4621 8.518e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Multinomial logit model

We can use the estimated parameters to work out marginal WTP

(i.e., recall $-\frac{\beta}{\beta_s}$).

The Krinsky and Robb method is used to derive confidence intervals of marginal WTP:

```
## load necessary package
library(MASS)
set.seed(1)
multivariate.normals <- mvrnorm(10000, MNL.result$estimate, sandwich(MNL.result))
wtps <- -multivariate.normals[, 1:3]/matrix(rep(multivariate.normals[, 4], 3), ncol = 3)
MNL.wtps <- rbind(-MNL.result$estimate[1:3]/MNL.result$estimate[4], apply(-multivariate.normals[,
  1:3]/matrix(rep(multivariate.normals[, 4], 3), ncol = 3), 2, quantile, probs = c(0.025,
  0.975)))
row.names(MNL.wtps) <- c("Mean estimate", "Lower CI", "Upper CI")
MNL.wtps
```



```
##           AttA      AttB      AttC
## Mean estimate 4.512581 3.963471 2.702699
## Lower CI     3.253158 2.603471 1.637730
## Upper CI     6.358868 5.854673 4.134189
```

Multinomial logit model

So what does the MNL model tell us:

- ▶ This sample of respondents have **on average**, as expected, positive marginal utilities for AttA (i.e., $\hat{\beta}_{\text{AttA}} = 0.628$), AttB (i.e., $\hat{\beta}_{\text{AttB}} = 0.552$) and AttC (i.e., $\hat{\beta}_{\text{AttC}} = 0.376$).
- ▶ This sample of respondents have **on average**, as expected, prefer cheaper options (i.e., $\hat{\beta}_{\text{Cost}} = -0.139$).
- ▶ The alternative specific constants for the first and second alternatives are 1.212 and 0.87.
- ▶ This produces **average** marginal WTP estimates of €4.51 for AttA, €3.96 for AttB and €2.70 for AttC.

Latent class model

For a two class latent class model, we can define function LC2.LL:

```
LC2.LL <- function(coeff) {  
  ## class 1 class 1  
  util1.1 <- coeff[1:4] %*% t(alternative.1) + coeff[5]  
  util2.1 <- coeff[1:4] %*% t(alternative.2) + coeff[6]  
  choice.prob <- (exp(util1.1) * (choice == 1) + exp(util2.1) * (choice == 2) +  
    1 * (choice == 3))/(exp(util1.1) + exp(util2.1) + 1)  
  class.1.probs <- colProds(matrix(choice.prob, ncol = N))  
  ## class 2  
  util1.2 <- coeff[7:10] %*% t(alternative.1) + coeff[11]  
  util2.2 <- coeff[7:10] %*% t(alternative.2) + coeff[12]  
  choice.prob <- (exp(util1.2) * (choice == 1) + exp(util2.2) * (choice == 2) +  
    1 * (choice == 3))/(exp(util1.2) + exp(util2.2) + 1)  
  class.2.probs <- colProds(matrix(choice.prob, ncol = N))  
  ## unconditional class probabilities  
  class.probs = exp(c(coeff[13], 0))/sum(exp(c(coeff[13], 0)))  
  ## weighted log-likelihood  
  log(class.1.probs * class.probs[1] + class.2.probs * class.probs[2])  
}
```

Latent class model

For a three class latent class model, we can define function LC3.LL:

```
LC3.LL <- function(coeff) {  
  ## class 1 class 1  
  util1.1 <- coeff[1:4] %*% t(alternative.1) + coeff[5]  
  util2.1 <- coeff[1:4] %*% t(alternative.2) + coeff[6]  
  choice.prob <- (exp(util1.1) * (choice == 1) + exp(util2.1) * (choice == 2) +  
    1 * (choice == 3))/(exp(util1.1) + exp(util2.1) + 1)  
  class.1.probs <- colProds(matrix(choice.prob, ncol = N))  
  ## class 2  
  util1.2 <- coeff[7:10] %*% t(alternative.1) + coeff[11]  
  util2.2 <- coeff[7:10] %*% t(alternative.2) + coeff[12]  
  choice.prob <- (exp(util1.2) * (choice == 1) + exp(util2.2) * (choice == 2) +  
    1 * (choice == 3))/(exp(util1.2) + exp(util2.2) + 1)  
  class.2.probs <- colProds(matrix(choice.prob, ncol = N))  
  ## class 3  
  util1.3 <- coeff[13:16] %*% t(alternative.1) + coeff[17]  
  util2.3 <- coeff[13:16] %*% t(alternative.2) + coeff[18]  
  choice.prob <- (exp(util1.3) * (choice == 1) + exp(util2.3) * (choice == 2) +  
    1 * (choice == 3))/(exp(util1.3) + exp(util2.3) + 1)  
  class.3.probs <- colProds(matrix(choice.prob, ncol = N))  
  ## unconditional class probabilities  
  class.probs = exp(c(coeff[19], coeff[20], 0))/sum(exp(c(coeff[19], coeff[20],  
    0)))  
  ## weighted log-likelihood  
  log(class.1.probs * class.probs[1] + class.2.probs * class.probs[2] + class.3.probs *  
    class.probs[3])  
}
```

Latent class model

For estimating the LC2.LL and LC3.LL functions we can use:

```
## 2-class latent class model
LC2.result <- maxLik(LC2.LL, start = c(0.6, 0.4, 0.3, -0.15, 3.8, 3.5, 0.8, 0.9,
  0.6, -0.18, -0.5, -0.9, 0.8), method = "BHHH")
names(LC2.result$estimate) <- c("AttA.class1", "AttB.class1", "AttC.class1", "Cost.class1",
  "ASC1.class1", "ASC2.class1", "AttA.class2", "AttB.class2", "AttC.class2", "Cost.class2",
  "ASC1.class2", "ASC2.class2", "class1.constant")

## 3-class latent class model
LC3.result <- maxLik(LC3.LL, start = c(0.7, 0.5, 0.4, -0.15, 3.4, 3.1, 0.8, 0.9,
  0.6, -0.18, -0.4, -0.9, 1.4, 0.7, 0.4, -0.22, 1.3, 3.6, 0.6, 0.1), method = "BHHH")
names(LC3.result$estimate) <- c(names(LC2.result$estimate)[1:12], "AttA.class3",
  "AttB.class3", "AttC.class3", "Cost.class3", "ASC1.class3", "ASC2.class3", "class1.constant",
  "class2.constant")
```

Note that these models (and all following models) are vulnerable to local maxima of the sample-likelihood function and, thus, there may be no guarantee that the maximum likelihood is reached. For this reason, the candidate models should be evaluated from a variety of random starting points. This is not done here.

Latent class model

The two-class latent class model produces a log-likelihood of -1,697.12:

```
LC2.result$maximum

## [1] -1697.123

## class sizes
exp(c(LC2.result$estimate[13], 0))/sum(exp(c(LC2.result$estimate[13], 0)))

## class1.constant
##      0.6742249      0.3257751
```

The three-class latent class model produces a log-likelihood of -1,632.39:

```
LC3.result$maximum

## [1] -1632.387

## class sizes
exp(c(LC3.result$estimate[19:20], 0))/sum(exp(c(LC3.result$estimate[19:20], 0)))

## class1.constant class2.constant
##      0.59508751      0.30729041      0.09762208
```

Latent class model

Both latent class models (with log-likelihoods of -1,697.12 and -1,632.39) outperform the MNL model (which had a log-likelihood of -1,848.05).

- ▶ This gives strong evidence that there is preference heterogeneity.
- ▶ Therefore, the MNL model is inappropriate.

Latent class model

Class 1 parameters for the two-class model:

```
coefest(LC2.result, vcov = sandwich)[1:6, ]

##           Estimate Std. Error   z value   Pr(>|z|)
## AttA.class1  0.6289303 0.08935002  7.038950 1.936944e-12
## AttB.class1  0.4588667 0.11205727  4.094930 4.222955e-05
## AttC.class1  0.2871681 0.09351376  3.070864 2.134402e-03
## Cost.class1 -0.1321514 0.02187078 -6.042372 1.518644e-09
## ASC1.class1  3.8875627 0.39608790  9.814899 9.713540e-23
## ASC2.class1  3.5630949 0.39663958  8.983206 2.629905e-19

## size of class
exp(LC2.result$estimate[13])/sum(exp(c(LC2.result$estimate[13], 0)))

## class1.constant
##           0.6742249
```


Latent class model

Class 2 parameters for the two-class model:

```
coefest(LC2.result, vcov = sandwich)[7:12, ]

##           Estimate Std. Error   z value   Pr(>|z|)
## AttA.class2  0.8012691 0.14287692  5.608107 2.045515e-08
## AttB.class2  0.8982187 0.16837125  5.334751 9.567579e-08
## AttC.class2  0.6887123 0.14236975  4.837491 1.314886e-06
## Cost.class2 -0.1791393 0.03689133 -4.855865 1.198624e-06
## ASC1.class2 -0.4837117 0.22849903 -2.116909 3.426756e-02
## ASC2.class2 -0.8925802 0.22583552 -3.952346 7.738862e-05

## size of class
1/sum(exp(c(LC2.result$estimate[13], 0)))

## [1] 0.3257751
```

Latent class model

Class 1 parameters for the three-class model:

```
coefest(LC3.result, vcov = sandwich)[1:6, ]

##           Estimate Std. Error   z value   Pr(>|z|)
## AttA.class1  0.6396289 0.11033453  5.797178 6.744004e-09
## AttB.class1  0.5697156 0.12261748  4.646284 3.379681e-06
## AttC.class1  0.3761145 0.09857503  3.815515 1.358993e-04
## Cost.class1 -0.1476660 0.02350780 -6.281575 3.351606e-10
## ASC1.class1  3.9095878 0.40079641  9.754548 1.763847e-22
## ASC2.class1  3.2563774 0.40242975  8.091791 5.879376e-16

## size of class
exp(LC3.result$estimate[19])/sum(exp(c(LC3.result$estimate[19:20], 0)))

## class1.constant
##           0.5950875
```

Latent class model

Class 2 parameters for the three-class model:

```
coefest(LC3.result, vcov = sandwich)[7:12, ]

##              Estimate Std. Error   z value   Pr(>|z|)
## AttA.class2  0.7059793 0.15586762  4.529352 5.916499e-06
## AttB.class2  0.8751943 0.17043058  5.135195 2.818515e-07
## AttC.class2  0.6942816 0.15033564  4.618210 3.870638e-06
## Cost.class2 -0.1880468 0.03818361 -4.924804 8.444474e-07
## ASC1.class2 -0.3857379 0.23879055 -1.615382 1.062281e-01
## ASC2.class2 -0.8883858 0.22893715 -3.880479 1.042510e-04

## size of class
exp(LC3.result$estimate[20])/sum(exp(c(LC3.result$estimate[19:20], 0)))

## class2.constant
##           0.3072904
```

Latent class model

Class 3 parameters for the three-class model:

```
coefest(LC3.result, vcov = sandwich)[13:18, ]
```

##		Estimate	Std. Error	z value	Pr(> z)
##	AttA.class3	4.4863902	1.7350496	2.5857417	0.009716971
##	AttB.class3	0.7422920	0.8262222	0.8984169	0.368963319
##	AttC.class3	-0.4728362	0.7466731	-0.6332573	0.526565644
##	Cost.class3	-0.1978455	0.1735827	-1.1397762	0.254379548
##	ASC1.class3	-1.8393123	1.5300296	-1.2021417	0.229308631
##	ASC2.class3	3.0133208	0.9227207	3.2656913	0.001091972

```
## size of class  
1/sum(exp(c(LC3.result$estimate[19:20], 0)))
```

```
## [1] 0.09762208
```

Random parameters logit model

For a Random parameters logit model, where we have normally distributed non-cost attributes and a log-normal cost distribution, we can use RPL.LL:

```
RPL.LL <- function(coeff) {  
  rand.draws <- cbind(coeff[1] + coeff[2] * qnorm(sobol.draws[, 1]), coeff[3] +  
    coeff[4] * qnorm(sobol.draws[, 2]), coeff[5] + coeff[6] * qnorm(sobol.draws[,  
    3]), -exp(coeff[7] + coeff[8] * qnorm(sobol.draws[, 4])))  
  for (i in 1:R) {  
    util1 <- rowSums(rand.draws[(1 + (N * n.tasks) * (i - 1)):(N * n.tasks) *  
      (i)], ] * (alternative.1)) + coeff[9]  
    util2 <- rowSums(rand.draws[(1 + (N * n.tasks) * (i - 1)):(N * n.tasks) *  
      (i)], ] * (alternative.2)) + coeff[10]  
    choice.prob <- (exp(util1) * (choice == 1) + exp(util2) * (choice == 2) +  
      1 * (choice == 3)) / (exp(util1) + exp(util2) + 1)  
    choice.probs[, i] <- colProds(matrix(choice.prob, ncol = N))  
  }  
  log(rowMeans(choice.probs))  
}
```

Random parameters logit model

For estimating the RPL . LL function we can use:

```
## define number of random draws
R <- 10
choice.probs <- matrix(0, nrow = N, ncol = R)
## load necessary package
library(randtoolbox)
## Generate random draws (Sobol)
sobol.draws <- matrix(rep(sobol(R * N, dim = 4, scrambling = 3), each = n.tasks),
  ncol = 4)
RPL.result <- maxLik(RPL.LL, start = c(0.6, 0.2, 0.5, 0.1, 0.7, 0.2, -1.8, 0.2, 0.4,
  0.2), method = "BHHH")
names(RPL.result$estimate) <- c("AttA.mean", "AttA.sd", "AttB.mean", "AttB.sd", "AttC.mean",
  "AttC.sd", "Cost.mean", "Cost.sd", "ASC1", "ASC2")
```

Note that this random parameter uses only 10 random draws. In reality many more (at least 1,000) are needed.

Random parameters logit model

This random parameters logit model (albeit with 10 random draws) produces a log-likelihood of -1,797.74:

```
RPL.result$maximum
```

```
## [1] -1797.736
```

This also outperforms the MNL model (which had a log-likelihood of -1,848.05), which further highlights the weaknesses of the MNL model.

Random parameters logit model

The parameters that maximise the RPL model:

```
coeftest(RPL.result, vcov = sandwich)
```

```
##  
## z test of coefficients:  
##  
##           Estimate Std. Error  z value  Pr(>|z|)  
## AttA.mean  0.6711606  0.0733779   9.1466 < 2.2e-16 ***  
## AttA.sd    -0.1274667  0.0985611  -1.2933  0.1959  
## AttB.mean  0.5641114  0.0895685   6.2981 3.013e-10 ***  
## AttB.sd    0.1374916  0.1694567   0.8114  0.4172  
## AttC.mean  0.3760493  0.0730619   5.1470 2.647e-07 ***  
## AttC.sd    0.0078446  0.1513881   0.0518  0.9587  
## Cost.mean -2.1277188  0.1743400 -12.2044 < 2.2e-16 ***  
## Cost.sd    1.2160319  0.1073238  11.3305 < 2.2e-16 ***  
## ASC1      1.5422187  0.1163467  13.2554 < 2.2e-16 ***  
## ASC2      1.1912172  0.1167784  10.2007 < 2.2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Attribute non-attendance model

For this type of model we can define the function ANA.LL:

```
ANA.LL <- function(coeff) {
  consider.AttA <- exp(coeff[7])/(exp(coeff[7]) + 1)
  consider.AttB <- exp(coeff[8])/(exp(coeff[8]) + 1)
  consider.AttC <- exp(coeff[9])/(exp(coeff[9]) + 1)
  consider.Cost <- exp(coeff[10])/(exp(coeff[10]) + 1)
  for (a in 0:1) {
    for (b in 0:1) {
      for (c in 0:1) {
        for (d in 0:1) {
          util1 <- (matrix(rep(c(a, b, c, d), each = N * n.tasks), ncol = 4) *
            coeff[1:4]) %*% t(alternative.1) + coeff[5]
          util2 <- (matrix(rep(c(a, b, c, d), each = N * n.tasks), ncol = 4) *
            coeff[1:4]) %*% t(alternative.2) + coeff[6]
          p.choice <- (exp(util1) * (choice == 1) + exp(util2) * (choice ==
            2) + 1 * (choice == 3))/(exp(util1) + exp(util2) + 1)
          class.probs[, (1 + 8 * a + 4 * b + 2 * c + 1 * d)] <- colProds(matrix(p.choice,
            ncol = N))
          class.size[(1 + 8 * a + 4 * b + 2 * c + 1 * d)] <- ifelse(a ==
            0, (1 - consider.AttA), consider.AttA) * ifelse(b == 0, (1 -
            consider.AttB), consider.AttB) * ifelse(c == 0, (1 - consider.AttC),
            consider.AttC) * ifelse(d == 0, (1 - consider.Cost), consider.Cost)
        }
      }
    }
  }
  ## weighted log-likelihood
  log(rowSums(matrix(rep(class.size, each = N), ncol = 16) * class.probs))
}
```

Attribute non-attendance model

For estimating the ANA.LL function we can use:

```
class.probs <- matrix(0, ncol = 16, nrow = N)
class.size <- rep(0, 16)
ANA.result <- maxLik(ANA.LL, start = c(0.6, 0.5, 0.7, -0.1, 0.2, 0.1, 0.3, 0.4, 0.2,
  0.4), method = "BHHH", iterlim = 3)
names(ANA.result$estimate) <- c("AttA", "AttB", "AttC", "Cost", "ASC1", "ASC2", "c.consидAttA",
  "c.consидAttB", "c.consидAttC", "c.consидCost")
```

Attribute non-attendance model

To explore attribute non-attendance for two attributes, define function `ANA.AttAB.LL` as:

```
ANA.AttAB.LL <- function(coeff) {
  consider.AttA <- exp(coeff[7])/(exp(coeff[7]) + 1)
  consider.AttB <- exp(coeff[8])/(exp(coeff[8]) + 1)
  for (a in 0:1) {
    for (b in 0:1) {
      util1 <- (matrix(rep(c(a, b, 1, 1), each = N * n.tasks), ncol = 4) *
        coeff[1:4]) %*% t(alternative.1) + coeff[5]
      util2 <- (matrix(rep(c(a, b, 1, 1), each = N * n.tasks), ncol = 4) *
        coeff[1:4]) %*% t(alternative.2) + coeff[6]
      p.choice <- (exp(util1) * (choice == 1) + exp(util2) * (choice == 2) +
        1 * (choice == 3))/(exp(util1) + exp(util2) + 1)
      class.probs[, (1 + 2 * a + 1 * b)] <- colProds(matrix(p.choice, ncol = N))
      class.size[(1 + 2 * a + 1 * b)] <- ifelse(a == 0, (1 - consider.AttA),
        consider.AttA) * ifelse(b == 0, (1 - consider.AttB), consider.AttB)
    }
  }
  ## weighted log-likelihood
  log(rowSums(matrix(rep(class.size, each = N), ncol = 4) * class.probs))
}
```

Attribute non-attendance model

For estimating the ANA.AttAB.LL function we can use:

```
class.probs <- matrix(0, ncol = 4, nrow = N)
class.size <- rep(0, 4)
ANA.AttAB.result <- maxLik(ANA.AttAB.LL, start = c(0.6, 0.5, 0.7, -0.1, 0.2, 0.1,
  0.3, 0.4), method = "BHHH", iterlim = 3)
names(ANA.result$estimate) <- c("AttA", "AttB", "AttC", "Cost", "ASC1", "ASC2", "c.consிடAttA",
  "c.consிடAttB")
```

Appendix overview

Hopefully this has given you a flavour of choice modelling.

But more importantly, hopefully this demonstrates the advantages in writing your own log-likelihood functions.

- ▶ Canned routines in many software programmes are “black boxes”.
- ▶ Learning to code empowers you to estimate models that you wouldn't otherwise be able to do.
- ▶ Moreover, this really helps you better understand the econometric model.
- ▶ There's a steep learning curve—but it's well worth it.

