Week 13: Heterogeneity

November 18, 2026

THE POINT OF THIS WEEK

Every model so far has assumed **one set of β's for everyone**. That is rarely true. Some people love electric cars and some won't consider one; some barely notice price and some shop on nothing else. An average can describe nobody in the sample, and a design aimed at the average can miss every real segment.

There are two ways to let preferences differ:

Approach	Heterogeneity	How
Mixed logit	**Unobserved**: you don't know why	β's are distributions
Sub-groups	**Observed**: a variable you measured	Group interactions

Mixed logit tells you *how much* people disagree. Sub-group models tell you *who* disagrees, and by how much.

MIXED LOGIT

Instead of a single number, a coefficient gets a distribution across the population, and you estimate the distribution's parameters:

$$\tilde{u}_j = \beta_1 x_j + \varepsilon_j, \quad \beta_1 \sim \mathrm{N}(\mu_1, \sigma_1)$$

Parameter	Estimate	Meaning
μ_1	0.1	Average preference for x_j
σ_1	0.1	How spread out preferences are

Read the two together. With $\mu_1 = \sigma_1 = 0.1$, about 16% of people have a *negative* β_1: the mean says "people like this," the spread says "but a real minority dislikes it." A σ that is small and not significant means that attribute is effectively homogeneous.

Which distribution?

Distribution	Code	Use when
Normal	`"n"`	Preferences can be positive or negative (brand, powertrain)
Log-normal	`"ln"`	Preferences must be one sign (price)
Fixed	leave out	σ is tiny; preferences look homogeneous

A log-normal is strictly positive, so it only fits a "more is better" coefficient. Price is "more is worse," so model **negative price** as log-normal, not price itself.

Estimating it in {logitr}

Add `randPars`, a named vector of which parameters are random and how they're distributed (practice Q3, the built-in `yogurt` data):

```
yogurt <- fastDummies::dummy_cols(
  yogurt, "brand")

model_mxl <- logitr(
  data    = yogurt,
  outcome = "choice",
  obsID   = "obsID",
  pars = c("price", "feat", "brand_dannon",
           "brand_hiland", "brand_weight"),
  randPars = c(price = "n", feat = "n"))

summary(model_mxl)
```

The summary reports a mean and an `sd_` term for each random parameter. Every parameter not named in `randPars` stays fixed. The same gradient and eigenvalue checks from the WTP week apply. A mixed logit's log-likelihood is non-convex, so a few `numMultiStarts` are cheap insurance.

Preference space vs. WTP space

The two spaces are **not equivalent** once parameters are random. In preference space, WTP is a ratio of two distributions:

$$\omega = \frac{\beta}{(-\alpha)} = \frac{\mathrm{N}(\mu_2, \sigma_2)}{(-\ln \mathrm{N}(\mu_1, \sigma_1))}$$

That ratio can have extreme tails and is hard to summarize. In WTP space you assume a distribution for WTP *directly*:

$$\tilde{u}_j = \lambda(\omega_1 x_j - p_j) + \varepsilon_j, \quad \omega_1 \sim \mathrm{N}(\mu_1, \sigma_1)$$

If WTP is what you'll report, put the distribution on WTP. In {logitr} that's `scalePar` plus `randPars` on the WTP parameters (`code/8.1-model-mxl.R` in `logitr-cars` fits both versions):

```
model_mxl_wtp <- logitr(
  data     = data,
  outcome  = "choice",
  obsID    = "obsID",
  pars = c('fuelEconomy', 'accelTime',
           'powertrain_Electric',
           'powertrain_Hybrid'),
  scalePar = 'price',
  randPars = c(
    fuelEconomy = 'n',
    accelTime = 'n',
    powertrain_Electric = 'n',
    powertrain_Hybrid = 'n'))
```

SUB-GROUP MODELS

Pick a variable you measured (a demographic, an attitude question, a usage pattern) and split respondents into groups. Then **interact** each attribute with a dummy for one group. With two groups, A and B:

$$\tilde{u}_j = \beta_1 x_j + \beta_2 x_j \delta^{\mathrm{B}} + \varepsilon_j = (\beta_1 + \beta_2 \delta^{\mathrm{B}}) x_j + \varepsilon_j$$

Parameter	Meaning
β_1	Effect of x_j for group A (the reference)
β_2	*Difference* in the effect between B and A

So group B's effect is $\beta_1 + \beta_2$. A significant β_2 is evidence that the groups really differ on that attribute.

Why one model, not two?

Each logit model is only identified **up to scale**: its coefficients are divided by the (unknown) size of its error term. Two separately estimated models each get their own scale, so their coefficients can't be compared. Suppose:

Group	$\hat{\alpha}$ (price)	$\hat{\beta}$	WTP
A	−100	200	2
B	−1	2	2

The coefficients look 100× apart, yet the groups have identical preferences. Only the noise differs. Two safe options:

1. **One model with interactions**: both groups share one scale.
2. **Separate WTP-space models**: the scale is λ, estimated apart from ω, so WTPs compare across models.

> Never compare raw coefficients from two separately estimated models. Each model has its own scale, so the numbers live in different units. Compare **WTP** (where scale cancels) or estimate one model with group interactions.

Worked example: WTP by group

Chocolate bars, with $\hat{\beta} = [-0.7, 0.1, 0.2, 0.8]$ for price, cacao, price × B, and cacao × B (practice Q2):

Group	Price	Cacao	WTP for cacao
A	−0.7	0.1	0.1 / 0.7 = 0.14
B	−0.7 + 0.2	0.1 + 0.8	0.9 / 0.5 = 1.8

Get the intervals with draws: **add the interaction draws before dividing.**

```
draws <- as.data.frame(
  MASS::mvrnorm(10^5, beta, covariance))

draws_A <- draws %>%
  select(price = b1, cacao = b2)
draws_B <- draws %>%
  mutate(price = b1 + b3,
         cacao = b2 + b4) %>%
  select(price, cacao)

wtp_A <- draws_A / (-1 * draws_A$price)
wtp_B <- draws_B / (-1 * draws_B$price)
ci(wtp_A)
ci(wtp_B)
```

On real data

`code/8.2-mnl-groups.R` in `logitr-cars` does the whole workflow: dummy-code `group`, build one interaction column per attribute, and add them to `pars`.

```
data <- data %>%
  dummy_cols(c("powertrain", "group")) %>%
  mutate(
    price_B = price * group_B,
    fuelEconomy_B = fuelEconomy * group_B)
    # ...one per attribute
```

The same script also fits a separate WTP-space model per group, and runs `predict()` with each one to compare market shares for the same set of cars.

COMMON MISTAKES

Symptom	Usual cause
Log-normal price won't fit	Used price, not negative price
Comparing group coefficients	Separate pref-space models; scales differ
Group B's WTP looks wrong	Divided B's draws without adding the interaction
`price_B` not found	Interaction column never created
Wild WTP from a mixed logit	Ratio of distributions; estimate in WTP space
`sd_` term not significant	That attribute is homogeneous; fix it

USING THIS IN YOUR PROJECT

Picking groups

Split on something you have a reason to expect matters: a demographic, prior ownership, or an attitude question you asked in the survey. Keep groups big enough to estimate. Every interaction doubles the parameters for that attribute, so check that each group has plenty of respondents.

Reporting heterogeneity

- **Mixed logit:** report μ and σ together, then translate. For a normal parameter, the share of people with a negative preference is `pnorm(0, mu, sigma)`.
- **Sub-groups:** plot each group's WTP side by side, on the same axis, with 95% intervals. Intervals that don't overlap are strong evidence of a real difference.
- **Market shares by group:** run `predict()` on the same market with each group's model. A product that wins with one segment and loses with another is a targeting decision, not a failure.

SENSITIVITY ANALYSIS

A market simulation predicts **one** share for **one** market. A sensitivity analysis asks how much that share moves when you change your product: start from a **baseline** market, change one attribute of your product while holding everything else fixed, and run `predict()` again.

Analysis	What you change	Chart
Price	Sweep your product's price over a range	Share and revenue vs. price
Attributes	Each attribute to a "high" and "low" value (e.g. ±20%)	Tornado plot

Sweeping price

Stack one copy of the baseline market per price, then predict them all at once (`rep_df()` is from {jph}):

```
prices <- seq(10, 30)
n <- length(prices)
scenarios <- rep_df(baseline, n)
scenarios$obsID <- rep(seq(n), each = 3)
scenarios$price[scenarios$altID == 2] <- prices

sens_price <- predict(
  model_mnl,
  newdata    = scenarios,
  obsID      = "obsID",
  level      = 0.95,
  interval   = "confidence",
  returnData = TRUE) %>%
  filter(altID == 2)
```

Revenue is price × share × market size. Share is always highest at the lowest price, but revenue usually peaks somewhere in between, and that's the price worth reporting. Only trust prices inside the range you surveyed; beyond it you're extrapolating.

Reading a tornado plot

Each bar is your product's share when one attribute moves to its "low" or "high" value, with everything else at baseline. Sort the longest bars to the top: those are the attributes your share depends on most.

In `logitr-cars`, `code/9.1-compute-sensitivity.R` runs both analyses and `code/9.2-plot-sensitivity.R` makes the charts. All three charts belong in your Final Analysis report.

BEFORE NEXT CLASS

- HW 12: Exam Review is due **Nov 29**.
- With your team, re-estimate your pilot model as a mixed logit, try a sub-group model on a demographic in your pilot data, and run price and attribute sensitivity analyses on your market simulation. All of these feed the Final Analysis report, due **Dec 6** (DM me if you don't want yours on the showcase).
- No class **Nov 25** (Thanksgiving). The **final exam** is in class on **Dec 2**: 60 minutes, on paper, one cheat sheet.
- Final presentations are **Dec 9**, 2:30 to 5:00pm, in SEH B1270 (Lehman Auditorium).

You should be able to:

- Explain the difference between observed and unobserved heterogeneity.
- Estimate a mixed logit with `randPars` and interpret μ and σ.
- Choose between normal, log-normal, and fixed parameters.
- Say why mixed logits differ between preference and WTP space.
- Write a two-group model with interactions and state each group's effect.
- Explain why separate models' coefficients can't be compared, and compute WTP by group with confidence intervals.
- Run price and attribute sensitivity analyses with `predict()`, and read a tornado plot.