Week 12: Individual-Level Coefficients

ResEcon 703: Topics in Advanced Econometrics

Matt Woerman
University of Massachusetts Amherst

Agenda

Last week

- Simulation-based estimation

This week

- Conditional distributions of coefficients
- Derivation of conditional distributions
- Applications of conditional distributions
- Individual-level coefficients R example

This week's reading

- Train textbook, chapter 11

Conditional Distributions of Coefficients

Random Coefficients in Mixed Logit Model

The mixed logit model allows for unobserved variation in preferences throughout the population with the use of random coefficients

- The distribution of these coefficients in the population is $f(\beta \mid \theta)$
- We estimate the parameters, θ, that define these population distributions
- This population distribution and the parameters that define it tell us nothing about where any individual decision maker falls within that distribution of coefficients

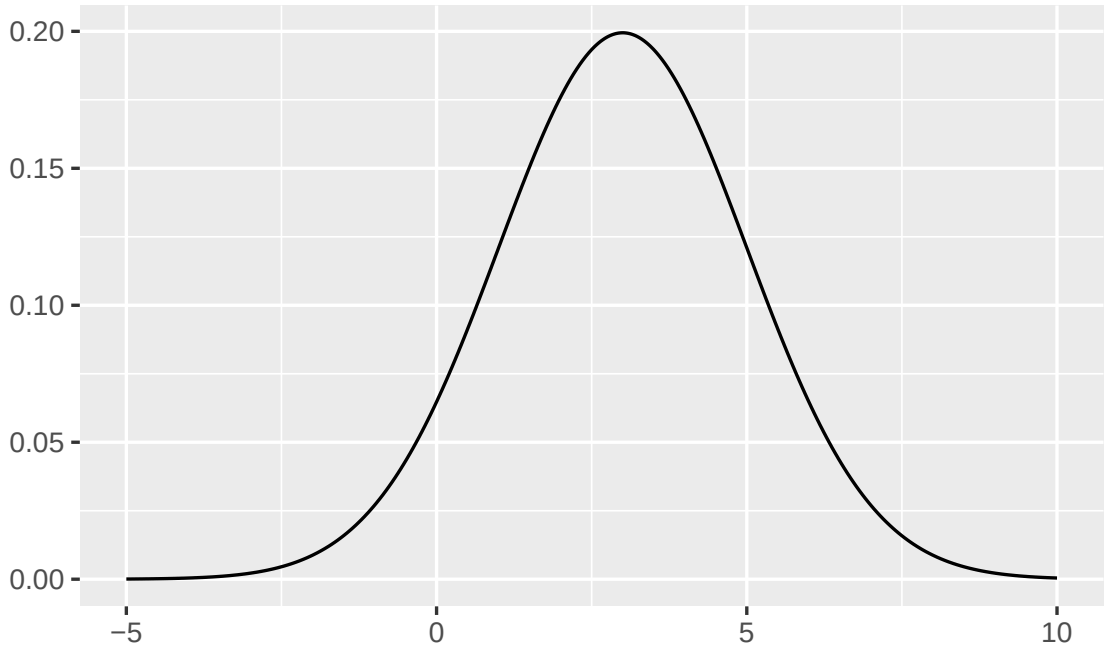
What if we want a better idea of an individual's coefficients?

- We can combine the unconditional (or population) distribution of coefficients and the choices made by the individual to define a conditional distribution of coefficients

Example of Conditional Distributions

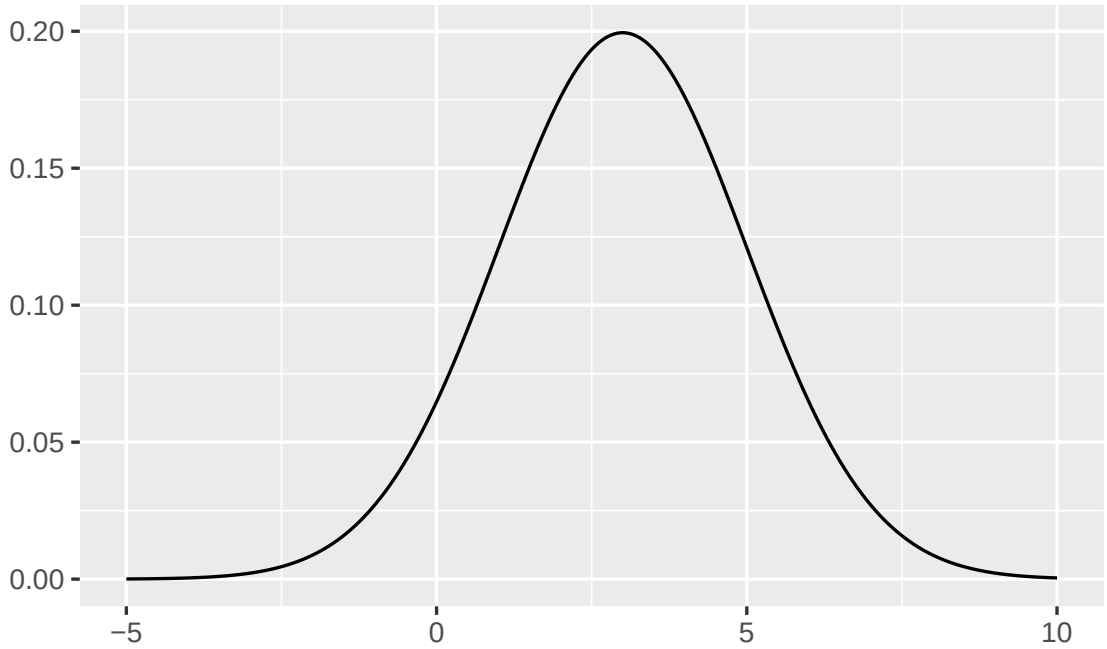
We are studying how commuters choose their travel mode

- β tells us the utility of driving relative to other commute modes
- We think there is heterogeneity in driving preferences, so we model β as a random coefficient, and we estimate $\beta \sim \mathcal{N}(3, 4)$



What is the individual-specific coefficient β_n for some specific individual?

Example of Conditional Distributions



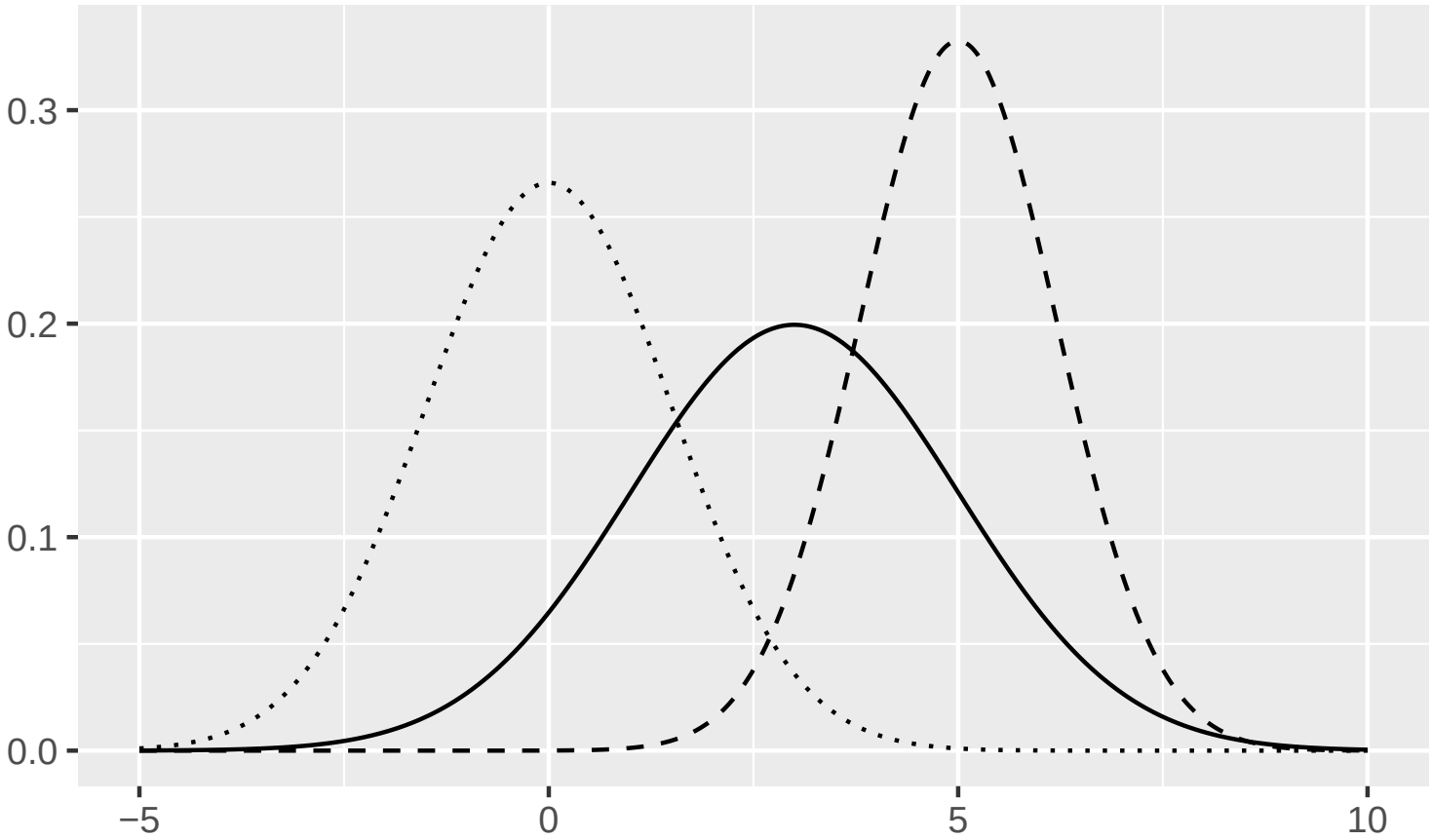
What is the individual-specific coefficient β_n for:

- A person drawn randomly from the population?
  - The unconditional distribution of β in the population, $\beta_n \sim \mathcal{N}(3, 4)$
- Someone who regularly drives to work?
  - Drivers are more likely to have relatively large values of β_n
- Someone who regularly does not drive to work?
  - Non-drivers are more likely to have relative low or negative values of β_n

Example of Conditional Distributions

We have described three different distributions for the coefficient β

- The unconditional distribution for the population (solid line)
- The conditional distribution for drivers (dashed line)
- The conditional distribution for non-drivers (dotted line)



Conditional Distribution of Coefficients

Suppose that a population of individuals:

- Faces an identical choice setting
  - The same choice set, $\{1, 2, \ldots, J\}$, and the same choice attributes, $\boldsymbol{x}$
- Has heterogeneous preferences denoted by the distribution of coefficients, $f(\beta \mid \boldsymbol{\theta})$
  - This is the unconditional (or population) distribution that we have previously defined

Consider everyone in the population who chooses alternative i

- This group is a non-random subset of the population
- These individuals also have a distribution of preferences, or β_n coefficients, but it is likely a different distribution from the population

The distribution of coefficients for this group is called a conditional distribution and is denoted by $h(\beta \mid i, \boldsymbol{x}, \boldsymbol{\theta})$

- Distribution of β among the group—from a population with an unconditional distribution defined by $\boldsymbol{\theta}$—who choose alternative i when faced with choice setting $\boldsymbol{x}$

Derivation of Conditional Distributions

Random Utility Model with Random Coefficients

The utility that decision maker n obtains from alternative j in choice situation t is

$$U_{njt} = \beta_n' \boldsymbol{x}_{njt} + \varepsilon_{njt}$$

- $\boldsymbol{x}_{njt}$: data about decision maker n and alternative j in situation t
- β_n: individual-specific coefficients with population density $f(\beta \mid \boldsymbol{\theta})$
- ε_{njt}: i.i.d. extreme value random utility term

To simplify notation

- $\boldsymbol{x}_n$: data collectively defined for all alternatives and choice situations
- $\boldsymbol{y}_n$: sequence of alternatives chosen by decision maker n

Choice Probabilities with Random Coefficients

If we knew a decision maker's coefficients, β_n, the probability of choosing sequence $\boldsymbol{y}_n$ when faced with choice settings $\boldsymbol{x}_n$ would be the product of conditional logit choice probabilities

$$P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta_n) = \prod_{t=1}^{T} L_{nt}(y_{nt} \mid \beta_n)$$

where $L_{nt}(y_{nt} \mid \beta_n)$ is the conditional logit choice probability

$$L_{nt}(y_{nt} \mid \beta_n) = \frac{e^{\beta_n' \boldsymbol{x}_{ny_{nt}t}}}{\sum_{j=1}^{J} e^{\beta_n' \boldsymbol{x}_{njt}}}$$

But we do not know each individual's coefficients, β_n, so we have to consider the unconditional distribution of coefficients in the population and intergrate over this density

$$P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = \int P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) f(\beta \mid \boldsymbol{\theta}) d\beta$$

Joint Density of Choices and Coefficients

The integrand of this choice probability is the joint density of $\boldsymbol{y}_n$ and β

$$P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) \times f(\beta \mid \boldsymbol{\theta})$$

- Conditional probability of $\boldsymbol{y}_n$ times the unconditional density of β

If we reverse the conditioning, we instead get

$$h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) \times P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$$

- Conditional density of β times the unconditional probability of $\boldsymbol{y}_n$

By Bayes' Rule, these two expressions are equal

$$h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) \times P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) \times f(\beta \mid \boldsymbol{\theta})$$

Conditional Distribution of Random Coefficients

The joint density of $\boldsymbol{y}_n$ and β is either side of the expression

$$h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) \times P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) \times f(\beta \mid \boldsymbol{\theta})$$

Rearranging terms gives an expression for the conditional distribution of β

$$h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) = \frac{P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) \times f(\beta \mid \boldsymbol{\theta})}{P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})}$$

- The numerator is the integrand of the mixed logit choice probability
- The denominator is the mixed logit choice probability

The conditional distribution, $h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta})$, is proportional to the product of

- The probability that an individual with coefficients β would choose $\boldsymbol{y}_n$
- The likelihood of observing β in the population

Applications of Conditional Distributions

Conditional Mean Coefficients

It is often easier to calculate a statistic derived from the conditional distribution, rather than the conditional distribution itself

- One example is the mean of the conditional distribution, or the conditional mean coefficients

The mean of $h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta})$, or the mean of β among the group—from a population with an unconditional distribution defined by $\boldsymbol{\theta}$—who choose sequence $\boldsymbol{y}_n$ when faced with choice setting $\boldsymbol{x}_n$, is

$$\begin{aligned}\bar{\beta}_n &= \int \beta h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) d\beta \\ &= \frac{\int \beta P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) f(\beta \mid \boldsymbol{\theta}) d\beta}{\int P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta) f(\beta \mid \boldsymbol{\theta}) d\beta}\end{aligned}$$

These integrals do not have closed-form expressions and must be simulated

Simulating Conditional Mean Coefficients

The steps to simulate conditional mean coefficients are similar to the steps we used to simulate mixed logit choice probabilities

1. Draw R random vectors from $f(\beta \mid \theta)$, denoted $\{\beta^1, \beta^2, \ldots, \beta^R\}$
2. For each random vector, β^r, calculate the conditional choice probability

$$P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r) = \prod_{t=1}^{T} \frac{e^{\beta^{r\prime} \boldsymbol{x}_{ny_{nt}t}}}{\sum_{j=1}^{J} e^{\beta^{r\prime} \boldsymbol{x}_{njt}}}$$

3. Simulate the conditional mean coefficients as the weighted average of the R random vectors

$$\check{\beta}_n = \sum_{r=1}^{R} w^r \beta^r$$

where the weight of each draw is proportional to $P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)$

$$w^r = \frac{P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)}{\sum_{r=1}^{R} P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)}$$

Individual-Specific Coefficients

As the number of observed choices (T) increases, the conditional mean coefficients for an individual, $\bar{\beta}_n$, converges to the individual-specific coefficients, β_n

- $\bar{\beta}_n$ is a consistent estimate of β_n

$$\bar{\beta}_n \xrightarrow{p} \beta_n$$

You must observe (and model) many choices for this convergence to become close

- Train conducts a Monte Carlo simulation exercise to find that even $T = 50$ yields a substantial difference between $\bar{\beta}_n$ and β_n
- See the Train textbook for more details on this point and the Monte Carlo simulation

Future Choice Probabilities

If we observe a decision maker's past choices, we can refine future choice probabilities by conditioning on those past choices

- We use the past choices to define a conditional distribution of coefficients for the decision maker
- We use this conditional distribution, instead of the unconditional distribution, to calculate mixed logit choice probabilities

The probability that decision maker n chooses alternative i in choice situation $T+1$ is

$$P(i \mid \boldsymbol{x}_{nT+1}, \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) = \int L_{nT+1}(i \mid \beta) h(\beta \mid \boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\theta}) d\beta$$

where $L_{nT+1}(i \mid \beta)$ is the conditional logit choice probability

$$L_{nT+1}(i \mid \beta) = \frac{e^{\beta' \boldsymbol{x}_{niT+1}}}{\sum_{j=1}^{J} e^{\beta' \boldsymbol{x}_{njT+1}}}$$

This is a mixed logit choice probability and must be simulated

Simulating Future Choice Probabilities

The steps to simulate future choice probabilities are similar to the steps we used to simulate mixed logit choice probabilities

1. Draw R random vectors from $f(\beta \mid \theta)$, denoted $\{\beta^1, \beta^2, \ldots, \beta^R\}$
2. For each random vector, β^r, calculate the conditional choice probability for the first T situations, $P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)$, and the conditional logit choice probabilities for situation $T+1$, $L_{nT+1}(i \mid \beta^r)$
3. Simulate the future mixed logit choice probabilities as the weighted average of $L_{nT+1}(i \mid \beta^r)$

$$\check{P}_{niT+1}(\boldsymbol{y}_n, \boldsymbol{x}_n, \theta) = \sum_{r=1}^{R} w^r L_{nT+1}(i \mid \beta^r)$$

where the weight of each draw is proportional to $P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)$

$$w^r = \frac{P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)}{\sum_{r=1}^{R} P(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \beta^r)}$$

Individual-Level Coefficients R Example

Maximum Simulated Likelihood Estimation Example

We are again studying how consumers make choices about expensive and highly energy-consuming systems in their homes

- We have (real) data on 250 households in California and the type of HVAC (heating, ventilation, and air conditioning) system in their home. Each household has the following choice set, and we observe the following data

Choice set

- `ec`: electric central
- `ecc`: electric central with AC
- `er`: electric room
- `erc`: electric room with AC
- `gc`: gas central
- `gcc`: gas central with AC
- `hpc`: heat pump with AC

Alternative-specific data

- `ich`: installation cost for heat
- `icca`: installation cost for AC
- `och`: operating cost for heat
- `occa`: operating cost for AC

Household demographic data

- `income`: annual income

Load Dataset

```
## Load tidyverse and mlogit
library(tidyverse)
library(mlogit)
## Load dataset from mlogit package
data('HC', package = 'mlogit')
```

Dataset

```
## Look at dataset
tibble(HC)
## # A tibble: 250 x 18
##    depvar ich.gcc ich.ecc ich.erc ich.hpc ich.gc ich.ec ich.er  icca
##    <fct>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <dbl>
##  1 erc       9.7     7.86    8.79   11.4    24.1   24.5   7.37  27.3
##  2 hpc       8.77    8.69    7.09    9.37   28     32.7   9.33  26.5
##  3 gcc       7.43    8.86    6.94   11.7    25.7   31.7   8.14  22.6
##  4 gcc       9.18    8.93    7.22   12.1    29.7   26.7   8.04  25.3
##  5 gcc       8.05    7.02    8.44   10.5    23.9   28.4   7.15  25.4
##  6 gcc       9.32    8.03    6.22   12.6    27.0   21.4   8.6   19.9
##  7 gc        7.11    8.78    7.36   12.4    22.9   28.6   6.41  27.0
##  8 hpc       9.38    7.48    6.72    8.93   26.2   27.9   7.3   18.1
##  9 gcc       8.08    7.39    8.79   11.2    23.0   22.6   7.85  22.6
## 10 gcc       6.24    4.88    7.46    8.28   19.8   27.5   6.88  25.8
## # ... with 240 more rows, and 9 more variables: och.gcc <dbl>,
## #   och.ecc <dbl>, och.erc <dbl>, och.hpc <dbl>, och.gc <dbl>,
## #   och.ec <dbl>, och.er <dbl>, occa <dbl>, income <dbl>
```

Format Dataset in a Long Format

```
## Pivot into a long dataset
hvac_long <- HC %>%
  mutate(id = 1:n()) %>%
  pivot_longer(c(starts_with('ich.'), starts_with('och.')),
               names_to = c('cost', 'alt'), names_sep = '[.]',
               values_to = 'value') %>%
  pivot_wider(names_from = cost, values_from = value) %>%
  mutate(choice = (depvar == alt)) %>%
  select(-depvar)
```

Dataset in a Long Format

```
## Look at long dataset
tibble(hvac_long)
## # A tibble: 1,750 x 8
##     icca  occa income    id alt     ich   och choice
##    <dbl> <dbl>  <dbl> <int> <chr> <dbl> <dbl> <lgl>
##  1  27.3  2.95     20     1 gcc    9.7   2.26 FALSE
##  2  27.3  2.95     20     1 ecc    7.86  4.09 FALSE
##  3  27.3  2.95     20     1 erc    8.79  3.85 TRUE
##  4  27.3  2.95     20     1 hpc   11.4   1.73 FALSE
##  5  27.3  2.95     20     1 gc    24.1   2.26 FALSE
##  6  27.3  2.95     20     1 ec    24.5   4.09 FALSE
##  7  27.3  2.95     20     1 er     7.37  3.85 FALSE
##  8  26.5  1.63     50     2 gcc    8.77  2.3  FALSE
##  9  26.5  1.63     50     2 ecc    8.69  2.69 FALSE
## 10  26.5  1.63     50     2 erc    7.09  3.45 FALSE
## # ... with 1,740 more rows
```

Clean Dataset

```
## Combine heating and cooling costs into one variable
hvac_clean <- hvac_long %>%
  mutate(ac = 1 * (nchar(alt) == 3),
         ic = ich + ac * icca,
         oc = och + ac * occa) %>%
  select(id, alt, choice, ac, ic, oc, income) %>%
  arrange(id, alt)
```

Cleaned Dataset

```
## Look at cleaned dataset
tibble(hvac_clean)
## # A tibble: 1,750 x 7
##       id alt   choice    ac    ic    oc income
##    <int> <chr> <lgl>  <dbl> <dbl> <dbl>  <dbl>
##  1     1 ec    FALSE      0 24.5   4.09     20
##  2     1 ecc   FALSE      1 35.1   7.04     20
##  3     1 er    FALSE      0  7.37  3.85     20
##  4     1 erc   TRUE       1 36.1   6.8      20
##  5     1 gc    FALSE      0 24.1   2.26     20
##  6     1 gcc   FALSE      1 37.0   5.21     20
##  7     1 hpc   FALSE      1 38.6   4.68     20
##  8     2 ec    FALSE      0 32.7   2.69     50
##  9     2 ecc   FALSE      1 35.2   4.32     50
## 10     2 er    FALSE      0  9.33  3.45     50
## # ... with 1,740 more rows
```

Convert Dataset to dfidx Format

```
## Convert cleaned dataset to dfidx format
hvac_dfidx <- dfidx(hvac_clean, shape = 'long',
                    choice = 'choice', idx = c('id', 'alt'))
```

Dataset in `dfidx` Format

```
## Look at data in dfidx format
tibble(hvac_dfidx)
## # A tibble: 1,750 x 6
##    choice    ac    ic    oc income idx$id $alt
##    <lgl>  <dbl> <dbl> <dbl>  <dbl>  <int> <fct>
##  1 FALSE      0 24.5   4.09     20      1 ec
##  2 FALSE      1 35.1   7.04     20      1 ecc
##  3 FALSE      0  7.37  3.85     20      1 er
##  4 TRUE       1 36.1   6.8      20      1 erc
##  5 FALSE      0 24.1   2.26     20      1 gc
##  6 FALSE      1 37.0   5.21     20      1 gcc
##  7 FALSE      1 38.6   4.68     20      1 hpc
##  8 FALSE      0 32.7   2.69     50      2 ec
##  9 FALSE      1 35.2   4.32     50      2 ecc
## 10 FALSE      0  9.33  3.45     50      2 er
## # ... with 1,740 more rows
```

Mixed Logit Model of HVAC System Choice

We previously estimated a mixed logit model with representative utility

$$V_{nj} = \beta_{1n} AC_j + \beta_{2n} IC_{nj} + \beta_{3n} OC_{nj}$$

where the random coefficients are normally distributed

$$\beta_{1n} \sim \mathcal{N}(\mu_1, \sigma_1^2)$$
$$\beta_{2n} \sim \mathcal{N}(\mu_2, \sigma_2^2)$$
$$\beta_{3n} \sim \mathcal{N}(\mu_3, \sigma_3^2)$$

Conditional Mean Coefficients for HVAC System Choice

The public utility commission is considering a subsidy on the installation cost of heat pump systems to incentivize households to switch to this most efficient HVAC system

- The PUC would like to target this subsidy and its marketing to households with certain HVAC system preferences
- We can use information about the HVAC system that a household currently has to generate a conditional distribution of coefficients that better describes that household's preferences

For each alternative, what are the mean β_n coefficients for the households with that HVAC system?

1. Estimate the mixed logit model
2. Simulate $\check{\beta}_n$ for each household
3. Average $\check{\beta}_n$ for the households with each HVAC system

Simulating Conditional Mean Coefficients

Two ways to simulate conditional mean coefficients for each household

- `mlogit` package
- Code the simulation by hand

The `fitted()` function with `type = 'parameters'` simulates the conditional mean coefficients for every individual

- This function returns the $N \times K$ matrix of conditional mean coefficients

We can instead code the simulation by hand

- We may want to simulate additional objects that are not part of the `mlogit` functionality

Mixed Logit Model Using `mlogit`

```
## Model choice using ac, ic, and oc with normal random coefficients
model_1 <- mlogit(formula = choice ~ ac + ic + oc | 0 | 0,
                  data = hvac_dfidx,
                  reflevel = 'hpc',
                  rpar = c(ac = 'n', ic = 'n', oc = 'n'),
                  R = 1000, seed = 703)
```

Mixed Logit Model Results Using `mlogit`

```
## Summarize model results
summary(model_1)
##
## Call:
## mlogit(formula = choice ~ ac + ic + oc | 0 | 0, data = hvac_dfidx,
##     reflevel = "hpc", rpar = c(ac = "n", ic = "n", oc = "n"),
##     R = 1000, seed = 703)
##
## Frequencies of alternatives:choice
##   hpc    ec   ecc    er   erc    gc   gcc
## 0.104 0.004 0.016 0.032 0.004 0.096 0.744
##
## bfgs method
## 22 iterations, 0h:0m:29s
## g'(-H)^-1g = 7.26E-07
## gradient close to zero
##
## Coefficients :
##         Estimate Std. Error z-value  Pr(>|z|)
## ac    10.8829832  3.8264644  2.8441  0.004453 **
## ic    -0.2150971  0.0349316 -6.1577 7.382e-10 ***
## oc    -1.1233808  0.1865858 -6.0207 1.736e-09 ***
## sd.ac  4.4597527  3.6209211  1.2317  0.218075
## sd.ic  0.0010176  0.3371230  0.0030  0.997592
## sd.oc  0.0110849  1.7285017  0.0064  0.994883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -327.22
##
## random coefficients
##    Min.     1st Qu.      Median       Mean    3rd Qu. Max.
## ac -Inf  7.8749258 10.8829832 10.8829832 13.8910407  Inf
## ic -Inf -0.2157834 -0.2150971 -0.2150971 -0.2144107  Inf
```

Conditional Mean Coefficients Using `mlogit`

```
## Calculate mean coefficient for each household
coefs_1 <- model_1 %>%
  fitted(type = 'parameters') %>%
  as_tibble() %>%
  rename(ac_coef = ac, ic_coef = ic, oc_coef = oc)
coefs_1
## # A tibble: 250 x 3
##    ac_coef ic_coef oc_coef
##      <dbl>   <dbl>   <dbl>
##  1   12.4   -0.215   -1.12
##  2   11.8   -0.215   -1.12
##  3   11.8   -0.215   -1.12
##  4   12.0   -0.215   -1.12
##  5   12.5   -0.215   -1.12
##  6   11.6   -0.215   -1.12
##  7    4.44  -0.215   -1.12
##  8   11.6   -0.215   -1.12
##  9   11.8   -0.215   -1.12
## 10   11.7   -0.215   -1.12
## # ... with 240 more rows
```

Average Conditional Mean Coefficients Using `mlogit`

```
## Average coefficient over all households with each HVAC system
hvac_clean %>%
  filter(choice == 1) %>%
  cbind(coefs_1) %>%
  group_by(alt) %>%
  summarize(ac_coef = mean(ac_coef),
            ic_coef = mean(ic_coef),
            oc_coef = mean(oc_coef),
            .groups = 'drop')
## # A tibble: 7 x 4
##   alt   ac_coef ic_coef oc_coef
##   <chr>   <dbl>   <dbl>   <dbl>
## 1 ec       4.40  -0.215   -1.12
## 2 ecc     11.9   -0.215   -1.12
## 3 er       4.35  -0.215   -1.12
## 4 erc     12.4   -0.215   -1.12
## 5 gc       4.30  -0.215   -1.12
## 6 gcc     11.9   -0.215   -1.12
## 7 hpc     11.9   -0.215   -1.12
```

Steps for Simulating Conditional Mean Coefficients

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid \boldsymbol{x}_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid \boldsymbol{x}_n, \beta^r)}$$

1. Draw $K \times N \times R$ standard normal random variables
   - K random coefficients for each of
   - N different decision makers for each of
   - R different simulation draws
2. Find the MSL estimator, $\widehat{\boldsymbol{\theta}}$
   - See slides from last week on MSL estimation
3. Simulate conditional mean coefficients using the MSL estimator, $\widehat{\boldsymbol{\theta}}$
   1. Transform each set of K standard normals using $\widehat{\boldsymbol{\theta}}$ to get β^r
   2. Calculate the conditional logit choice probability of the chosen alternative, $P(y_n \mid \boldsymbol{x}_n, \beta^r)$, for each household and random draw
   3. For each household, take a weighted average of β^r, with weights proportional to $P(y_n \mid \boldsymbol{x}_n, \beta^r)$, to get $\check{\beta}_n$

Step 1: Draw Random Variables and Organize Data

```
## Set seed for replication
set.seed(703)
## Draw standard normal random variables for each household
draws_hh <- map(1:250, ~ tibble(ac_draw = rnorm(100),
                                ic_draw = rnorm(100),
                                oc_draw = rnorm(100)))
## Split data into list by household
data_hh <- hvac_clean %>%
  group_by(id) %>%
  group_split()
```

Step 2a: Simulate Choice Probabilities for One Household

```
## Function to simulate choice probabilities for an individual household
sim_probs_ind <- function(params, draws_ind, data_ind){
  ## Select relevant variables and convert into a matrix [J x K]
  data_matrix <- data_ind %>%
    select(ac, ic, oc) %>%
    as.matrix()
  ## Transform random coefficients based on parameters [R x K]
  coef_matrix <- draws_ind %>%
    mutate(ac_coef = params[1] + params[4] * ac_draw,
           ic_coef = params[2] + params[5] * ic_draw,
           oc_coef = params[3] + params[6] * oc_draw) %>%
    select(ac_coef, ic_coef, oc_coef) %>%
    as.matrix()
  ## Calculate representative utility for each alternative in each draw [R x J]
  utility <- (coef_matrix %*% t(data_matrix)) %>%
    pmin(700) %>%
    pmax(-700)
  ## Sum the exponential of utility over alternatives [R x 1]
  prob_denom <- utility %>%
    exp() %>%
    rowSums()
  ## Calculate the conditional probability for each alternative and draw [R x J]
  cond_prob <- exp(utility) / prob_denom
  ## Calculate simulated choice probabilities as means over all draws [1 x J]
  sim_prob <- colMeans(cond_prob)
  ## Add simulated probability to initial dataset
  data_ind_out <- data_ind %>%
    mutate(prob = sim_prob)
  ## Return initial dataset with simulated probability variable
  return(data_ind_out)
}
```

Step 2b: Calculate Simulated Log-Likelihood

```r
## Function to calculate simulated log-likelihood
sim_ll_fn <- function(params, draws_list, data_list){
  ## Simulate probabilities for each individual household
  data_sim_ind <- map2(.x = draws_list, .y = data_list,
                       .f = ~ sim_probs_ind(params = params,
                                            draws_ind = .x,
                                            data_ind = .y))
  ## Combine individual datasets into one
  data_sim <- data_sim_ind %>%
    bind_rows()
  ## Calculate log of simulated probability for the chosen alternative
  data_sim <- data_sim %>%
    filter(choice == TRUE) %>%
    mutate(log_prob = log(prob))
  ## Calculate the simulated log-likelihood
  sim_ll <- sum(data_sim$log_prob)
  ## Return the negative of simulated log-likelihood
  return(-sim_ll)
}
```

Step 2c: Maximize Simulated Log-Likelihood

```
## Maximize the log-likelihood function
model_2 <- optim(par = c(6.53, -0.17, -1.04, 0, 0, 0), fn = sim_ll_fn,
                 draws_list = draws_hh, data_list = data_hh,
                 method = 'BFGS', hessian = TRUE,
                 control = list(trace = 1, REPORT = 5))
## initial  value 330.051626
## iter   5 value 329.842440
## iter  10 value 329.559384
## iter  15 value 326.652634
## iter  20 value 325.961965
## iter  25 value 325.960937
## final  value 325.959722
## converged
```

Step 2d: Report MSLE Results

```
## Report model results
model_2
## $par
## [1] 11.0917037025 -0.2164020694 -1.1278947803  4.5934515313  0.0002048493  0.0057092340
##
## $value
## [1] 325.9597
##
## $counts
## function gradient
##       94       28
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## [1,]   4.0451882   95.551233    3.536574  -4.1442779    3.285546   0.2552387
## [2,]  95.5512329 4130.546546 -215.794416 -92.6246226  122.374461   9.2682826
## [3,]   3.5365739 -215.794416  144.136869  -3.9213534   -5.159820   1.3656793
## [4,]  -4.1442779  -92.624623   -3.921353   4.4151970   -7.345430  -0.2337583
## [5,]   3.2855456  122.374461   -5.159820  -7.3454298 2686.692261 -15.9361930
## [6,]   0.2552387    9.268283    1.365679  -0.2337583  -15.936193  42.8194368
```

Step 3a: Simulate Coefficients for One Household

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid \boldsymbol{x}_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid \boldsymbol{x}_n, \beta^r)}$$

```
## Function to simulate individual coefficients for one individual
calc_mean_coefs <- function(params, draws_ind, data_ind){
  ## Select relevant variables and convert into a matrix [J x K]
  data_matrix <- data_ind %>%
    select(ac, ic, oc) %>%
    as.matrix()
  ## Transform random draws into coefficients based on parameters
  coef <- draws_ind %>%
    mutate(ac_coef = params[1] + params[4] * ac_draw,
           ic_coef = params[2] + params[5] * ic_draw,
           oc_coef = params[3] + params[6] * oc_draw) %>%
    select(ac_coef, ic_coef, oc_coef)
  ## Convert coefficients tibble to a matrix [R x K]
  coef_matrix <- as.matrix(coef)
  ## Calculate representative utility for each alternative in each draw [R x J]
  utility <- (coef_matrix %*% t(data_matrix)) %>%
    pmin(700) %>%
    pmax(-700)
  ## Sum the exponential of utility over alternatives [R x 1]
  prob_denom <- utility %>%
    exp() %>%
    rowSums()
  ## Calculate the conditional probability for each alternative and draw [R x J]
  cond_prob <- exp(utility) / prob_denom
  ## Calculate the numerator of the draw weights as prob of chosen alt [R x 1]
  weights_num <- c(cond_prob %*% data_ind$choice)
  ## Calculate the draw weights [R x 1]
  weights <- weights_num / sum(weights_num)
```

Step 3a: Simulate Coefficients for One Household

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid \boldsymbol{x}_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid \boldsymbol{x}_n, \beta^r)}$$

```
  ## Sum the exponential of utility over alternatives [R x 1]
  prob_denom <- utility %>%
    exp() %>%
    rowSums()
  ## Calculate the conditional probability for each alternative and draw [R x J]
  cond_prob <- exp(utility) / prob_denom
  ## Calculate the numerator of the draw weights as prob of chosen alt [R x 1]
  weights_num <- c(cond_prob %*% data_ind$choice)
  ## Calculate the draw weights [R x 1]
  weights <- weights_num / sum(weights_num)
  ## Add draw weights to dataset of coefficients
  coef <- coef %>%
    mutate(weight = weights)
  ## Calculate weighted mean for each coefficient
  coef_means <- coef %>%
    summarize(ac_coef_mean = sum(ac_coef * weight),
              ic_coef_mean = sum(ic_coef * weight),
              oc_coef_mean = sum(oc_coef * weight))
  ## Add individual coefficient means to initial dataset
  data_ind_out <- data_ind %>%
    bind_cols(coef_means)
  ## Return initial dataset with simulated probability variable
  return(data_ind_out)
}
```

Step 3b: Simulate Coefficients for All Households

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid \boldsymbol{x}_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid \boldsymbol{x}_n, \beta^r)}$$

```
## Calculate mean coefficients for each individual
data_2_ind <- map2(.x = draws_hh, .y = data_hh,
                   .f = ~ calc_mean_coefs(params = model_2$par,
                                          draws_ind = .x,
                                          data_ind = .y))
## Combine list of data into one tibble
data_2 <- data_2_ind %>%
  bind_rows()
```

Conditional Mean Coefficients for Each HVAC System

```
## Calculate mean coefficients by chosen alternative
data_2 %>%
  filter(choice == 1) %>%
  group_by(alt) %>%
  summarize(ac_coef = mean(ac_coef_mean),
            ic_coef = mean(ic_coef_mean),
            oc_coef = mean(oc_coef_mean),
            .groups = 'drop')
## # A tibble: 7 x 4
##   alt   ac_coef ic_coef oc_coef
##   <chr>   <dbl>   <dbl>   <dbl>
## 1 ec       4.05  -0.216   -1.13
## 2 ecc     12.3   -0.216   -1.13
## 3 er       4.39  -0.216   -1.13
## 4 erc     12.1   -0.216   -1.13
## 5 gc       4.20  -0.216   -1.13
## 6 gcc     12.2   -0.216   -1.13
## 7 hpc     12.2   -0.216   -1.13
```