ECN 594: Demographic Interactions and pyblp

Nicholas Vreugdenhil

January 4, 2026

Plan

1. **Demand with Demographic Interactions**

2. Estimation with pyblp

Recap: The basic logit model

- Our utility model so far:

$$u_{ij} = x_j\beta - \alpha p_j + \xi_j + \varepsilon_{ij}$$

- Everyone has the same β and α
- Problem: this is very restrictive
    - Rich and poor consumers have same price sensitivity?
    - Families and singles value car size the same?

Why do we need heterogeneous preferences?

- **Policy questions require knowing WHO values WHAT:**
    - If a product is removed, who loses the most?
    - If prices rise, which consumers switch away?
    - Who benefits from new product variety?
- **Competitive questions need substitution patterns:**
    - When BMW's price rises, who switches to Mercedes vs Honda?
    - Basic logit: everyone substitutes proportionally (IIA)
    - Reality: depends on consumer demographics!

Extending the model: heterogeneous preferences

- **Basic logit:** Everyone has same β
- Problem: doesn't capture that different consumers value characteristics differently
- **Solution:** Let preferences vary with observed demographics

The demographic interaction model

- New utility specification:

$$u_{ij} = x_j\beta + (D_i \times x_j)\gamma - \alpha p_j + \xi_j + \varepsilon_{ij}$$

- D_i: observed consumer demographics (income, age, family size, etc.)
- $(D_i \times x_j)$: interactions between demographics and characteristics
- This creates **heterogeneous preferences**

Classic example: Nevo (2001) - Cereal demand

- “Measuring Market Power in the Ready-to-Eat Cereal Industry”
- Estimated demand for cereal using demographic interactions
- Key findings:
    - High-income households less price-sensitive
    - Families with children prefer kid-targeted cereals
    - Without demographics: wrong substitution patterns
- Result: Basic logit would underestimate markups!

Examples of demographic interactions

- **Income × price:**
  - High-income consumers less price-sensitive
  - $\gamma_{\text{inc}\times p} > 0$: price hurts less for rich consumers
- **Family size × car size:**
  - Families prefer larger vehicles
  - $\gamma_{\text{fam}\times\text{size}} > 0$
- **Age × fuel efficiency:**
  - Older consumers may care more about MPG

Worked example: Income × price interaction

- **Question:**

- Model: $u_{ij} = \beta x_j - \alpha p_j + \gamma(\text{income}_i \times p_j) + \varepsilon_{ij}$

- Estimated: $\alpha = 2.0$, $\gamma = 0.01$

- Consumer A has income \$50,000; Consumer B has income \$100,000

- What is the effective price coefficient for each consumer?

Take 2 minutes to solve this.

Worked example: Income × price interaction (solution)

Solution

Effective price coefficient $= -\alpha + \gamma \times$ income

Consumer A (income = \$50,000):

$$\text{Effect} = -2.0 + 0.01 \times 50 = -2.0 + 0.5 = -1.5$$

Consumer B (income = \$100,000):

$$\text{Effect} = -2.0 + 0.01 \times 100 = -2.0 + 1.0 = -1.0$$

Interpretation: Consumer B (richer) is less price-sensitive.
A \$1 price increase hurts B's utility by 1.0, but hurts A's utility by 1.5.

Where do demographics come from?

- Two scenarios:
  1. **Individual-level data:** Observe D_i for each consumer
     - Survey data, loyalty card data
  2. **Market-level data:** Know distribution of D_i in each market
     - Census data, Current Population Survey
     - This is more common in practice
- pyblp handles both cases

Why not random coefficients?

- Full BLP/mixed logit model adds unobserved heterogeneity:

$$\beta_i = \bar{\beta} + \Sigma \nu_i, \quad \nu_i \sim N(0, I)$$

- This is computationally harder (requires simulation)
- Demographic interactions capture a lot of the variation more simply
- **Mixed logit** is beyond our scope, but know it exists
- It fully relaxes IIA

Demographics and IIA (preview)

- Recall: basic logit has IIA problem
    - When BMW price rises, same fraction goes to Mercedes as to Civic
- With demographics, different consumer types substitute differently
    - High-income BMW buyers → Mercedes
    - Low-income BMW buyers → Civic
- Aggregate substitution is richer
- But IIA still holds *within* each consumer type
- Full discussion in Lecture 4

Limitations: Demographics vs Mixed Logit

- **Demographics model** (what we use):
  - Heterogeneity only from observed characteristics
  - IIA still holds within demographic groups
  - Simple and tractable
- **Mixed logit / random coefficients** (advanced):
  - Heterogeneity from unobserved factors too
  - Fully relaxes IIA
  - Computationally intensive (requires simulation)
- For this course: demographics are sufficient
- Know that mixed logit exists for research

Plan

1. Demand with Demographic Interactions
2. **Estimation with pyblp**

What is pyblp?

- Python package for demand estimation
- Conlon & Gortmaker (2020), “Best Practices for Differentiated Products Demand Estimation with PyBLP”
- Handles:
  - Basic logit and logit with demographics
  - Instrumental variables
  - Standard errors
  - Post-estimation (elasticities, markups, etc.)
- Why use a package?
  - Correct standard errors
  - Well-tested code

pyblp workflow

1. **Set up data:** products, markets, shares, characteristics
2. **Define formulation:** which variables, which IVs
3. **Create problem:** combine data and formulation
4. **Solve:** estimate the model
5. **Extract results:** coefficients, standard errors, elasticities

Let's walk through each step with car data

Step 1: Load and inspect data

- Load the BLP automobile data
- Key columns: market_ids, shares, prices, hpwt, air, mpd, space, demand_instruments0, ...

```
import pyblp
import pandas as pd

product_data = pd.read_csv(pyblp.data.BLP_PRODUCTS_LOCATION)
```

What data do we need?

- **Product data** (for each product j in market t):
  - Market share s_{jt}
  - Price p_{jt}
  - Characteristics x_{jt} (size, HP, fuel efficiency, etc.)
  - Instruments z_{jt} (BLP IVs, cost shifters)
- **Demographic data** (optional, for interactions):
  - Census data on income, family size, age by market
  - Or micro-level survey data

The BLP automobile data

market_id	firm_id	shares	prices	hpwt	air	mpd	space
1971	15	0.0012	4.935	0.524	0	1.888	0.917
1971	15	0.0011	5.516	0.494	0	1.935	0.920
1971	15	0.0006	7.108	0.467	0	1.716	1.074
1971	26	0.0038	4.296	0.426	0	2.449	0.853
1971	26	0.0018	4.080	0.521	0	2.398	0.772
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1990	19	0.0008	12.75	0.569	1	2.012	1.145

- 2,217 products across 20 years (1971–1990)
- Each row: one car model in one year

Step 2: Define the formulation

- Specify which variables enter the utility function

- Variables: `1` (constant), `hpwt` (HP/weight), `air` (A/C), `mpd` (MPG), `space`, `prices`

```
formulation = pyblp.Formulation('1 + hpwt + air + mpd + space + prices
    ')
```

Step 3: Create the problem

- Combine formulation and data into a Problem object
- pyblp automatically detects: demand_instruments, market_ids, firm_ids

```
problem = pyblp.Problem(formulation, product_data)
```

Step 4: Solve

- Estimate the model (uses IV if instruments present)

- Returns: coefficient estimates, robust standard errors, GMM objective

```
results = problem.solve()
```

Step 5: View results

- Print coefficient estimates and standard errors
- Compute elasticity matrix (own and cross-price)
- Compute markups (assuming Nash-Bertrand pricing)

```
print(results)
elasticities = results.compute_elasticities()
markups = results.compute_markups()
```

Adding demographic interactions

- Load demographic data (income draws for each market)
- Add second formulation: demographics interact with price

```
agent_data = pd.read_csv(pyblp.data.BLP_AGENTS_LOCATION)

formulation = pyblp.Formulation(
    ’1 + hpwt + air + mpd + space + prices’,
    ’0 + prices’,
    agent_formulation=pyblp.Formulation(’0 + income’)
)
```

Computing aggregate elasticities

- `elasticities[j,k]` $=$ % change in share of j per % change in price of k

- Diagonal $=$ own-price elasticities

```python
elasticities = results.compute_elasticities()
own_elasticities = np.diag(elasticities)
print(f"Mean own-price elasticity: {own_elasticities.mean():.2f}")
```

Estimation output

Variable	Coefficient	Std. Error
Constant	−10.215	(0.253)
HP/Weight	2.893	(0.367)
Air conditioning	1.521	(0.104)
Miles per dollar	0.158	(0.043)
Space	2.384	(0.125)
Price	−0.142	(0.012)

GMM objective: 142.35

N = 2,217 products, T = 20 markets

- Price coefficient is **negative** (as expected)

- All characteristics positive: consumers prefer more HP, air conditioning, fuel efficiency, space

Interpreting pyblp output

- **Coefficients:**
  - $\hat{\alpha}$ (price): should be negative
  - $\hat{\beta}$ (characteristics): interpret as marginal utility
- **Standard errors:**
  - Check statistical significance
  - pyblp computes robust SEs by default
- **First-stage F-statistic:**
  - Check that IVs are relevant
  - Rule of thumb: $F > 10$

Worked example: Interpreting coefficients

- Suppose you estimate:
    - $\hat{\alpha} = -0.8$ (price coefficient)
    - $\hat{\beta}_{HP} = 0.3$ (horsepower coefficient)
- **Questions:**
    1. Interpret $\hat{\alpha}$. What does a more negative α mean?
    2. If you had used OLS instead of IV, would $\hat{\alpha}$ be more or less negative?

Worked example: Interpreting coefficients (answers)

Answers

1. $\hat{\alpha} = -0.8$: A \$1 price increase reduces mean utility by 0.8 utils

- More negative α = more price-sensitive consumers

2. OLS would give $\hat{\alpha}$ biased toward zero (less negative)

- Because $\text{Cov}(p, \xi) > 0$
- OLS thinks high prices don't hurt demand much
- So OLS $\hat{\alpha}$ might be -0.3 instead of -0.8

Post-estimation: Elasticities

- pyblp computes elasticities automatically:
    - Own-price elasticity for each product
    - Cross-price elasticity matrix
- Check if elasticities are reasonable:
    - Own-price should be negative
    - Magnitude: typically -2 to -10 for durable goods
    - Cross-price: positive for substitutes

Elasticity matrix example

	Prod 1	**Prod 2**	**Prod 3**	**Prod 4**	**Prod 5**
Prod 1	-0.89	0.002	0.003	0.001	0.002
Prod 2	0.002	-1.24	0.003	0.002	0.003
Prod 3	0.001	0.002	-0.73	0.001	0.002
Prod 4	0.002	0.003	0.002	-1.08	0.002
Prod 5	0.001	0.002	0.001	0.001	-0.95

- **Diagonal (red):** Own-price elasticities ≈ -0.7 to -1.3

- **Off-diagonal:** Cross-price elasticities ≈ 0.001 to 0.003 (small but positive)

Post-estimation: Markups

- Recall: Lerner index $L = (p - MC)/p = 1/|\varepsilon|$

- Can recover markups from elasticities:

$$\text{markup}_j = \frac{p_j - mc_j}{p_j} = \frac{1}{|\eta_{jj}|}$$

- pyblp assumes Nash-Bertrand pricing

- Multi-product firms internalize substitution between own products

Worked example: Interpret this output

- pyblp gives you this output:

Variable	**Estimate**	**Std. Error**
Constant	3.5	0.8
HP/weight	1.2	0.4
Air cond.	0.9	0.3
Price	0.05	0.02

- **Question:** Something is wrong. What is it?

Take 1 minute to identify the problem.

Worked example: Interpret this output (solution)

Solution

Problem: Price coefficient is positive $(+0.05)$!
This means: higher prices → higher utility. That's economically nonsensical.

Why might this happen?

1. **Weak instruments:** IVs don't predict prices well
2. **Wrong signs:** Maybe you forgot a negative sign somewhere
3. **Endogeneity still present:** IVs are correlated with ξ

Fix: Check first-stage F-stat, try different IVs, verify data

Common pyblp errors and how to fix them

- **“Singular matrix” error:**
  - Likely: collinear variables in formulation
  - Fix: Remove redundant variables
- **Price coefficient positive or near zero:**
  - Likely: weak instruments
  - Fix: Check first-stage F-stat; try different IVs
- **Huge standard errors:**
  - Likely: weak identification or too few observations
  - Fix: Aggregate to fewer markets; add more IVs

How to check your results make sense

1. **Price coefficient negative?** (Required!)
2. **Elasticities reasonable?**
   - Own-price: typically -2 to -10 for durables
   - Cross-price: positive but small
3. **Markups reasonable?**
   - Cars: typically 15-30%
   - Cereal: typically 30-50%
4. **First-stage F-stat > 10?** (IV relevance)
5. **Coefficients stable across specifications?**

This prepares you for HW1

- HW1 asks you to:
    1. Load car data
    2. Estimate demand with pyblp
    3. Compute elasticities
    4. Interpret your results
    5. Discuss IV choice
- Today's worked example is a template for HW1
- Start early!

Key Points

1. **Demographic interaction model**: $u_{ij} = x_j\beta + (D_i \times x_j)\gamma - \alpha p_j + \xi_j + \varepsilon_{ij}$
2. Demographics allow **heterogeneous preferences** without random coefficients
3. Demographics **partially help** with IIA (different types substitute differently)
4. **pyblp workflow**: Data → Formulation → Problem → Solve → Interpret
5. **Key outputs**: Coefficients (check signs), standard errors, elasticities, markups
6. Price coefficient should be **negative**; OLS biases it toward zero
7. **Check IV relevance**: First-stage F-statistic > 10
8. This is the foundation for HW1

Next time

- **Lecture 4:** Consumer Surplus, IIA, and Price Discrimination
    - Log-sum formula for consumer surplus
    - Red bus/blue bus: the IIA problem in detail
    - Selection by indicators (group pricing)