# ECN 594: Demographic Interactions and pyblp

Nicholas Vreugdenhil

January 4, 2026

# Plan for today

1. Recap: endogeneity and IVs
2. The demographic interaction model
3. Why demographics help with IIA

---

4. Introduction to pyblp
5. Worked example: car demand
6. Interpreting output

# Part 1: Demand with Demographic Interactions

# 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 $\beta$ and $\alpha$
- Problem: this is very restrictive
  - Rich and poor consumers have same price sensitivity?
  - Families and singles value car size the same?

# Extending the model: heterogeneous preferences

- **Basic logit:** Everyone has same $\beta$
- 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**

# Examples of demographic interactions

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

# 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 $\rightarrow$ Mercedes
  - Low-income BMW buyers $\rightarrow$ Civic
- Aggregate substitution is richer
- But IIA still holds *within* each consumer type
- Full discussion in Lecture 4

# Part 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

```python
import pyblp

# Load the BLP automobile data
product_data = pyblp.data.BLP_PRODUCTS

# Key columns:
# - market_ids: which market (year)
# - shares: market shares
# - prices: prices
# - hpwt, space, mpd: characteristics
# - demand_instruments0, ...: IVs
```

# Step 2: Define the formulation

```
# Basic logit: no random coefficients
formulation = pyblp.Formulation(
    '1 + hpwt + space + mpd + prices'
)

# With demographic interactions (income x price):
formulation = pyblp.Formulation(
    '1 + hpwt + space + mpd + prices',
    absorb='C(market_ids)'  # market fixed effects
)
```

# Step 3: Create the problem

```python
# Define the problem
problem = pyblp.Problem(
    formulation,
    product_data,
    agent_data=agent_data  # for demographics
)

# Check the problem
print(problem)
```

# Step 4: Solve

```
# Solve with 2SLS (IV estimation)
results = problem.solve()

# This gives us:
# - Coefficient estimates
# - Standard errors
# - Objective value
```

# Step 5: Extract results

```python
# Print coefficient estimates
print(results)

# Get elasticities
elasticities = results.compute_elasticities()

# Get markups (assuming Nash-Bertrand)
markups = results.compute_markups()
```

# 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 $\alpha$ mean?
  2. If you had used OLS instead of IV, would $\hat{\alpha}$ be more or less negative?

# Worked example: Interpreting coefficients (answers)

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

# 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

# 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 $\rightarrow$ Formulation $\rightarrow$ Problem $\rightarrow$ Solve $\rightarrow$ 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)