# Code Review and Interpretation of Discrete Choice Analysis

This code implements a discrete choice modeling framework to analyze consumer preferences for circuit breaker products with different features and prices. Let me break down the key components and interpret the results.

# 1. Choice Set Generation

The `generate\_choice\_sets()` function creates balanced experimental designs where:

- Each of the 16 product profiles appears roughly equally often

- Each choice set contains 3 profiles

- There are 24 total choice sets

This ensures respondents see a diverse set of combinations while maintaining statistical balance.

# 2. Discrete Choice Analyzer Class

The main analysis is performed by the `DiscreteChoiceAnalyzer` class which:

# Key Methods:

1. prepare\_data():

- Scales price by dividing by 1000 (for numerical stability)

- One-hot encodes categorical features

- Validates and merges choice data

- Creates a long-format dataset suitable for modeling

2. fit\_model():

- Fits a conditional logit model using statsmodels

- Estimates utility coefficients for each feature level

3. calculate\_feature\_importance():

- Computes relative importance of features based on utility ranges

- Normalizes to sum to 1 (100%)

4. plot\_utilities() and plot\_feature\_importance():

- Visualize the model results

5. plot\_price\_elasticity():

- Calculates and plots how demand changes with price changes

6. predict\_market\_shares() and simulate\_price\_increases():

- Simulate market scenarios under different pricing strategies

# 3. Interpreting Results

# Utilities

Utilities represent how much each feature contributes to the product's attractiveness. Higher utilities mean more preferred features.

Example interpretation:

- If "High performance" has utility +1.2 and "Compact Frame" has +0.8, customers prefer high performance by 0.4 utility units

- If Price has utility -0.05 (negative since higher prices are less attractive)

# Feature Importance

This shows which attributes the most influence choices. For example, the output might show:

- Size\_Performance: 45%

- Advanced\_Feature: 35%

- Price: 20%

This means product size/performance is the most important factor, followed by advanced features, with price being least important (though still significant).

# Price Elasticity

Price elasticity measures how demand changes with price. The plot shows:

- X-axis: Price levels

- Y-axis: Elasticity values

Interpretation:

- Elasticity of -1.5 at $115,000 means a 1% price increase would reduce demand by ~1.5%

- The curve typically shows higher elasticity (more sensitivity) at higher prices

# Price Increase Simulation

The simulation predicts how market shares would change with different price increases (5%, 10%, etc.). For example:

```

profile price\_increase\_percent market\_share\_new market\_share\_original share\_change\_percent

0 0 5 0.0625 0.0625 0.0

1 0 10 0.0600 0.0625 -4.0

2 0 15 0.0575 0.0625 -8.0

```

This shows profile 0 would lose 4% market share with a 10% price increase.

# 4. Example Insights

1. Pricing Strategy: The maximum price increase analysis identifies how much each product can increase price before losing more than 5% of its market share.

2. Feature Optimization: If advanced features have high utilities but low current adoption, they may represent opportunities for product differentiation.

3. Market Segmentation: The group variable allows comparing utilities between Group A and Group B to tailor offerings.

# 5. Limitations

1. The simulated choices (random selection) don't reflect real consumer behavior - in practice you'd use real choice data.

2. The model assumes **independence of irrelevant alternatives (IIA),** which may not hold if some products are closer substitutes than others.

3. Price elasticity is calculated at the aggregate level - individual-level elasticities could provide richer insights.

This framework provides a powerful way to quantify tradeoffs between product features and pricing, enabling data-driven product and pricing decisions.