**Analysis of dcm13h.py (dcm10.py) Output**

**1. Key Improvements Over dcm13g.py**

* **Matrix Alignment Error Resolved:** 
  + **dcm13g.py:**

**text**

**CollapseWrap**

**Copy**

**Error occurred: shapes (2253,7) and (8,) not aligned: 7 (dim 1) != 8 (dim 0)**

* + **dcm13h.py: No error, and profile shares computed:**

**text**

**CollapseWrap**

**Copy**

**Profile Shares (%):**

**Baseline 20% Increase 20% Decrease Custom**

**0 11.07 10.97 11.16 11.54**

**...**

* + - **Reason: The fix in evaluate\_price\_scenario\_lp and plot\_price\_elasticity correctly selects utilities:**

**python**

**CollapseWrapRun**

**Copy**

**utilities = self.utilities[['const'] + X\_cols]**

**This ensures alignment by excluding Price\_Panel.**

* **Group-Specific Models:** 
  + **Successfully fitted models for Panel builder and Others:**

**text**

**CollapseWrap**

**Copy**

**Panel builder Pseudo R-squ.: 0.01496**

**Others Pseudo R-squ.: 0.07476**

* + - **Improvement: Provides group-specific insights, with Others showing a stronger fit (0.07476 vs. 0.04335 in dcm13d.py) and significant price coefficient (p=0.006).**
* **Profile Shares:** 
  + **Computed for combined and group-specific models, showing reasonable variation (4–21% across scenarios).**
* **Invalid chosen\_profile:** 
  + **Remains resolved:**

**text**

**CollapseWrap**

**Copy**

**Invalid chosen profiles not in profiles\_presented: 0**

**2. Comparison to dcm13g.py**

* **Data Processing:** 
  + **Identical:** 
    - **Dropped 432 choices (36.5% of 1,183) for profiles 14, 15.**
    - **choice\_data shape: (2,253, 12).**
    - **Group counts: Panel builder (1,500), Others (753).**
* **Model Fit:** 
  + **Combined model unchanged:**

**text**

**CollapseWrap**

**Copy**

**Pseudo R-squ.: 0.02441**

**Price: -0.0585 (p=0.419)**

**Price\_Panel: 0.0031 (p=0.971)**

* + **Group-specific models added:** 
    - **Panel builder: 0.01496, Price: 0.0226 (p=0.694, positive).**
    - **Others: 0.07476, Price: -0.2297 (p=0.006).**
* **SHAP Importance:** 
  + **Unchanged:**

**text**

**CollapseWrap**

**Copy**

**Price: 7.85%**

**Adv\_Feat\_Current: 38.94%**

**Adv\_Feat\_ElecLife: 31.38%**

**3. Persistent Issues**

* **Insignificant and Positive Price Coefficient for Panel Builders:** 
  + **Combined Model:**

**text**

**CollapseWrap**

**Copy**

**Price: -0.0585 (p=0.419)**

**Price\_Panel: 0.0031 (p=0.971)**

**Effective Panel builder price: -0.0585 + 0.0031 = -0.0554 (p=0.971)**

* + **Panel Builder Model:**

**text**

**CollapseWrap**

**Copy**

**Price: 0.0226 (p=0.694, positive)**

* + - **Issue: Positive price coefficient contradicts economic intuition (higher price increases utility). Insignificant (p=0.694) and worse than dcm13d.py (-0.0288, p=0.553).**
    - **Cause: Narrow price range ($112,200–$116,650, std = $1,594.67, ~4%) and reduced sample (1,500 rows) limit sensitivity. Dropping 432 choices may have biased the sample.**
* **Low Choice Set Counts:** 
  + **Unchanged:**

**text**

**CollapseWrap**

**Copy**

**Choice sets per respondent:**

**2 122**

**3 84**

**4 55**

**1 35**

**Warning: Maximum choice sets per respondent (4) is below expected (16).**

* + - **~4 sets vs. 16 expected, reducing statistical power.**
* **Higher Mean Price for Chosen Profiles:** 
  + **Unchanged:**

**text**

**CollapseWrap**

**Copy**

**0: 113469.432423**

**1: 113579.885153**

* + - **$110.45 gap contradicts negative price coefficient, likely due to narrow price range or sample bias.**
* **Missing Price-Adv\_Feat\_ModbusBasic Correlation:** 
  + **Unchanged:**

**text**

**CollapseWrap**

**Copy**

**No Price-Adv\_Feat\_ModbusBasic correlations computed.**

* + - **Likely due to filtered choice sets lacking Adv\_Feat\_ModbusBasic or a bug in validate\_data.**
* **Narrow Price Range:** 
  + **Unchanged: $112,200–$116,650 (~4%, std = $1,594.67). Needs ~20–50% ($90,000–$140,000).**
* **Low Pseudo R-squared for Panel Builders:** 
  + **Combined: 0.02441 (improved but below ~0.06–0.1).**
  + **Panel builder: 0.01496 (below ~0.01–0.015).**
  + **Others: 0.07476 (meets ~0.06–0.1).**
* **Low Price Importance:** 
  + **SHAP: Price at 7.85% (below 15–25%), reflecting insignificant coefficients.**

**4. Evaluation Against Expectations**

* **Convergence: Achieved.**
* **Negative Price Coefficient:** 
  + **Combined: Achieved (-0.0585).**
  + **Panel builder: Not achieved (0.0226, positive).**
  + **Others: Achieved (-0.2297).**
* **Price Significance:** 
  + **Overall: p<0.05, missed (p=0.419).**
  + **Panel builder: p<0.05, not achieved (p=0.694).**
  + **Others: p<0.05, achieved (p=0.006).**
* **Pseudo R-squared:** 
  + **Overall: ~0.015–0.02, above (0.02441).**
  + **Panel builder: ~0.01–0.015, above (0.01496).**
  + **Others: ~0.06–0.1, met (0.07476).**
* **VIF: <3, achieved.**
* **SHAP Importance:** 
  + **Price: ~15–25%, missed (7.85%).**
  + **Size\_Perf\_High: Negligible (0.49%, consistent with p=0.946).**
* **Diagnostics:** 
  + **Uncentered price range: Achieved.**
  + **Invalid chosen\_profile: Resolved (0 invalid).**
  + **Price-Adv\_Feat\_ModbusBasic correlation: Not achieved.**
  + **Choice sets: ~4, warning triggered.**
* **Profile Shares: ~4–21%, reasonable but narrow due to low price sensitivity.**

**5. Is It Better Enough?**

* **Positives:** 
  + **Resolved matrix alignment error, enabling full execution.**
  + **Zero invalid chosen\_profile errors.**
  + **Improved Pseudo R-squared (0.02441 combined, 0.07476 for Others).**
  + **Significant price coefficient for Others (p=0.006).**
  + **Low VIFs (<3).**
  + **Profile shares computed, showing scenario impacts.**
* **Shortcomings:** 
  + **Positive and Insignificant Price for Panel Builders: Critical issue (0.0226, p=0.694).**
  + **Low Choice Set Counts: ~4 vs. 16.**
  + **Narrow Price Range: Limits sensitivity.**
  + **Missing Correlation: Diagnostic incomplete.**
  + **Reduced Sample Size: Dropping 36.5% of choices may bias results.**
  + **Low Price Importance: 7.85% vs. 15–25%.**
* **Conclusion: dcm13h.py is a substantial improvement (no errors, group-specific models, resolved chosen\_profile issue), but not better enough due to the positive and insignificant price coefficient for Panel builders, low choice set counts, narrow price range, and missing correlation. These issues, particularly the Panel builder price coefficient, undermine the model’s reliability for conjoint analysis.**

**Detailed Analysis**

**1. Data Issues**

* **Dropped Choices:** 
  + **432 choices (36.5%) dropped:**

**text**

**CollapseWrap**

**Copy**

**Dropping 432 choices with invalid profiles (chosen or presented): {14, 15}**

* + - **Impact: Reduced choice\_data to 2,253 rows, potentially biasing results if dropped choices were systematically different (e.g., Panel builders preferring profiles 14, 15).**
    - **Cause: Filtering removed choices with chosen\_profile or profiles\_presented containing 14 or 15 (AF = 'Terminal Temperature threshold monitoring').**
* **Narrow Price Range:** 
  + **$112,200–$116,650 (~4%, std = $1,594.67). Needs ~20–50%.**
* **Low Choice Set Counts:** 
  + **~4 sets:**

**text**

**CollapseWrap**

**Copy**

**2: 122**

**3: 84**

**4: 55**

**1: 35**

* + - **Likely incomplete survey data in CBC\_Data\_Final\_09Jun25.xlsx.**
* **Missing Correlation:** 
  + **No Price-Adv\_Feat\_ModbusBasic correlation, possibly due to filtered choice sets lacking Adv\_Feat\_ModbusBasic.**

**2. Model Performance**

* **Combined Model:**

**text**

**CollapseWrap**

**Copy**

**Pseudo R-squ.: 0.02441**

**Price: -0.0585 (p=0.419)**

**Price\_Panel: 0.0031 (p=0.971)**

**Size\_Perf\_High: -0.0065 (p=0.946)**

**Adv\_Feat\_ElecLife: 0.9155 (p=0.000)**

**Adv\_Feat\_Current: 0.9291 (p=0.000)**

**Adv\_Feat\_Safety: 0.2180 (p=0.151)**

**Adv\_Feat\_Health: 0.4124 (p=0.007)**

* + **Weak price sensitivity, dominated by advanced features.**
* **Panel Builder Model:**

**text**

**CollapseWrap**

**Copy**

**Pseudo R-squ.: 0.01496**

**Price: 0.0226 (p=0.694, positive)**

**Size\_Perf\_High: -0.1742 (p=0.138)**

**Adv\_Feat\_ElecLife: 0.4243 (p=0.020)**

**Adv\_Feat\_Current: 0.7308 (p=0.000)**

* + **Poor fit, positive price coefficient is problematic.**
* **Others Model:**

**text**

**CollapseWrap**

**Copy**

**Pseudo R-squ.: 0.07476**

**Price: -0.2297 (p=0.006)**

**Size\_Perf\_High: 0.3675 (p=0.033)**

**Adv\_Feat\_ElecLife: 1.9628 (p=0.000)**

**Adv\_Feat\_Current: 1.3969 (p=0.000)**

* + **Strong fit, significant price sensitivity.**
* **SHAP Importance:** 
  + **Price: 7.85% (too low).**
  + **Adv\_Feat\_Current: 38.94% (dominant).**

**3. Diagnostics**

* **Mean Price by Chosen Status:** 
  + **$110.45 gap, contradicting negative price coefficient.**
* **Correlations:** 
  + **Price and Price\_Panel: 0.810913, VIF ~3.**
* **Choice Sets:** 
  + **~4 sets, limiting power.**
* **Profile Shares:** 
  + **Combined: 4–11%, narrow shifts.**
  + **Panel builder: 4–12%, minimal price response.**
  + **Others: 2–21%, stronger price response.**

**Resolutions and Recommendations**

1. **Positive and Insignificant Price for Panel Builders:** 
   * **Bootstrap to Assess Variability: Add to fit\_model:**

**python**

**CollapseWrapRun**

**Copy**

**from sklearn.utils import resample**

**if group == 'Panel builder':**

**coefs = []**

**panel\_data = data.copy()**

**for \_ in range(1000):**

**sample = resample(panel\_data, stratify=panel\_data['chosen'])**

**X\_sample = sample[X\_cols]**

**y\_sample = sample['chosen']**

**X\_sample = sm.add\_constant(X\_sample)**

**model = sm.Logit(y\_sample, X\_sample).fit(disp=0, maxiter=2000)**

**coefs.append(model.params['Price'])**

**print("Bootstrap Price Coefficient for Panel builder:\n", pd.Series(coefs).describe())**

* + **Wider Price Range: Update profiles.xlsx to $90,000–$140,000.**
  + **Inspect Dropped Choices:**

**python**

**CollapseWrapRun**

**Copy**

**dropped = choices[~choices['chosen\_profile'].isin(valid\_profiles) | ~choices['parsed\_profiles'].apply(lambda x: all(p in valid\_profiles for p in x))]**

**print("Dropped choices by group:\n", dropped.merge(groups, on='respondent\_id')['group'].value\_counts())**

1. **Low Choice Set Counts:** 
   * **Verify:**

**python**

**CollapseWrapRun**

**Copy**

**print(choices['choice\_set'].nunique())**

**print(choices.groupby('respondent\_id')['choice\_set'].nunique().describe())**

* + **Request complete data (16 sets per respondent).**

1. **Missing Price-Adv\_Feat\_ModbusBasic Correlation:** 
   * **Debug validate\_data:**

**python**

**CollapseWrapRun**

**Copy**

**print("Choice sets with Adv\_Feat\_ModbusBasic:", (self.choice\_data['Adv\_Feat\_ModbusBasic'] == 1).sum())**

* + **Ensure profiles with Adv\_Feat\_ModbusBasic are in choice sets.**

1. **Narrow Price Range:** 
   * **Simulate wider range in price\_scenarios:**

**python**

**CollapseWrapRun**

**Copy**

**price\_scenarios = {**

**'Baseline': profiles['Price'].to\_dict(),**

**'50% Increase': (profiles['Price'] \* 1.5).to\_dict(),**

**'50% Decrease': (profiles['Price'] \* 0.5).to\_dict(),**

**'Custom': {i: profiles['Price'].mean() \* 1.2 for i in profiles.index}**

**}**

1. **No Script Update Needed:** 
   * **dcm13h.py resolved the error and completed execution. Issues stem from data limitations, not code.**

**Next Steps for Mixed Logit**

1. **Use xlogit:** 
   * **Install:**

**bash**

**CollapseWrapRun**

**Copy**

**pip install xlogit**

* + **Example:**

**python**

**CollapseWrapRun**

**Copy**

**from xlogit import MixedLogit**

**model = MixedLogit()**

**model.fit(**

**X=self.choice\_data[['Price', 'Size\_Perf\_High', 'Adv\_Feat\_ElecLife', 'Adv\_Feat\_Current', 'Adv\_Feat\_Safety', 'Adv\_Feat\_Health']],**

**y=self.choice\_data['chosen'],**

**ids=self.choice\_data['respondent\_id'],**

**panels=self.choice\_data['respondent\_id'],**

**randvars={'Price': 'n'},**

**num\_draws=500**

**)**

**print(model.summary())**

* + **Add after resolving data issues.**

1. **Fix pylogit:** 
   * **Edit choice\_tools.py:**

**python**

**CollapseWrapRun**

**Copy**

**from collections.abc import Iterable**

1. **Python 3.9:** 
   * **bash**

**CollapseWrapRun**

**Copy**

**conda create -n py39 python=3.9**

**conda activate py39**

**pip install pandas numpy statsmodels matplotlib seaborn shap pylogit**

**Testing Instructions**

1. **Rerun dcm13h.py with Diagnostics:** 
   * **Add:**

**python**

**CollapseWrapRun**

**Copy**

**print("Unique choice sets:", choices['choice\_set'].nunique())**

**print("Choices with profiles 14 or 15:\n", choices[choices['profiles\_presented'].str.contains('14|15')][['chosen\_profile', 'profiles\_presented']].head(10))**

**print("Dropped choices by group:\n", choices[~choices['chosen\_profile'].isin(valid\_profiles) | ~choices['parsed\_profiles'].apply(lambda x: all(p in valid\_profiles for p in x))].merge(groups, on='respondent\_id')['group'].value\_counts())**

**print("Choice sets with Adv\_Feat\_ModbusBasic:", (choice\_data['Adv\_Feat\_ModbusBasic'] == 1).sum())**

* + **Check:** 
    - **Invalid choices: Expect 0.**
    - **Choice sets: Expect >4.**
    - **Adv\_Feat\_ModbusBasic presence.**
    - **Profile shares: Expect wider shifts with modified price\_scenarios.**

1. **Share Results:** 
   * **Diagnostics (choice sets, dropped choices, correlations).**
   * **Model summaries (p-values, Pseudo R-squared).**
   * **VIF, SHAP, profile shares.**
   * **Bootstrap results if added.**
2. **Data Correction:** 
   * **Request CBC\_Data\_Final\_09Jun25.xlsx with 16 choice sets.**
   * **Update profiles.xlsx with wider price range.**

**Summary**

* **Output Review: dcm13h.py resolves the matrix alignment error, computes profile shares, and fits group-specific models. The chosen\_profile issue is fixed (0 invalid).**
* **Is It Better Enough?: No, due to positive and insignificant price coefficient for Panel builders (0.0226, p=0.694), low choice set counts (~4), narrow price range (~4%), and missing correlation. Others model performs well (p=0.006, Pseudo R-squ.: 0.07476).**
* **Improvements: Error-free execution, group-specific insights.**
* **Recommendations:** 
  + **Rerun dcm13h.py with added diagnostics.**
  + **Inspect dropped choices for bias.**
  + **Request corrected data (16 sets, wider price range).**
  + **Prepare for xlogit after data fixes.**

**Please rerun dcm13h.py with the additional diagnostics and share the output. Let me know if you need help with data correction, bootstrap code, or xlogit setup!**