blp

February 22, 2025

Henrique de Souza Passos

[452]:

```
get_ipython().magic('reset -sf')
```

```
/tmp/ipykernel_548659/3674724974.py:1: DeprecationWarning: `magic(…)` is
deprecated since IPython 0.13 (warning added in 8.1), use
run_line_magic(magic_name, parameter_s).
  get_ipython().magic('reset -sf')
```

[453]:

```
from IPython import get_ipython
import scipy.io
import numpy as np
import pandas as pd
import scipy.stats as stats
import pyblp
import statsmodels.api as sm
from itertools import product
from linearmodels.iv import IV2SLS
```

1 1. Setting

A number of national producers sell substitute products in regional markets. The government intends to bail out a struggling firm and and allow it to merge with one of its healthy competitors. What do you expect the welfare consequences to be?

No curto prazo isso pode evitar um aumento de preços devido a injeção de liquidez que o governo poderia fazer. No entanto, no longo prazo, poderia haver maior concentração de mercado aumentando os preços para os consumidores.

2 2. Data description

For the empirical exercise, we are giving you data on T = 10 markets. In these markets, 11 different firms sell a total of J = 247 products. All of the products are unique, so none of them are offered in multiple markets. The dataset is simulated, but you can still think of a product as a passenger vehicle with a set of characteristics if you like, although the units do not have an interpretation. The dataset contains the following pieces of data, where products are ordered by market (1-10): - “prodsMarket”: T-vector of the number of products in each market - “share”: J-vector of market shares - “f”: J-vector denoting the firm that sells the product - “ch”: J x 4-matrix of constant and three product characteristics - “pr”: J-vector of prices - “costShifters”: J x 2-matrix of cost shifters

3 3. Basic summary statistics

1. Prepare a table with the following pieces of information for each market: How many firms are active? How many products do they market in total? What fraction of agents bought one of the goods in the sample period?

[454]:
```
mat_data = scipy.io.loadmat("/home/hspassos/mestrado/industrial/demand_data.
↪mat")

prodsMarket = {'market': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
prodsMarket = pd.DataFrame(prodsMarket)
prodsMarket['prodsMarket'] = mat_data['prodsMarket'].flatten()

# Criar tabela com os dados por produto
data = pd.DataFrame({
    'market_ids': 1 + np.repeat(np.arange(len(prodsMarket)),␣
↪prodsMarket['prodsMarket']),
    'firm_ids': mat_data['f'].flatten(),
    'shares': mat_data['share'].flatten(),
    'prices': mat_data['pr'].flatten(),
    'const': mat_data['ch'][:, 0],
    'char1': mat_data['ch'][:, 1],
    'char2': mat_data['ch'][:, 2],
    'char3': mat_data['ch'][:, 3],
    'costsh1': mat_data['costShifters'][:, 0],
    'costsh2': mat_data['costShifters'][:, 1],
})

data
```

[454]:
```
     market_ids  firm_ids    shares    prices  const     char1     char2  \
0             1         1  0.002115  4.775915    1.0  2.750983  0.672526
1             1         9  0.032741  3.388449    1.0  1.022277  2.476778
2             1         9  0.009585  4.085734    1.0  2.667053  0.760962
3             1         7  0.024101  2.432229    1.0  1.068635  0.910503
4             1         2  0.012206  4.486977    1.0  2.138262  1.871875
..          …         …         …         …      …         …         …
242          10         2  0.006462  4.524368    1.0  3.006674  0.118876
243          10        10  0.002214  4.118809    1.0  1.125124  1.748192
244          10         3  0.103526  2.840987    1.0  2.284540  0.467589
245          10        11  0.003856  4.967610    1.0  2.869888  0.569319
246          10         3  0.002055  3.674037    1.0  0.933852  1.152601

        char3   costsh1   costsh2
0    2.139236  0.158296  0.264869
1    2.685004  0.479222  0.151009
2    1.980984  0.276290  0.002862
```

```
3    0.036386  0.109393  0.108614
4    3.655649  0.064100  0.468338
..        …         …         …
242  1.877790  0.480930  0.039359
243  1.067760  0.483401  0.003548
244  1.036729  0.018995  0.136472
245  2.678046  0.252503  0.184921
246  0.341722  0.326308  0.474412

[247 rows x 10 columns]
```

2. Prepare a table with summary statistics for market share, characteristics, price, and cost shifters. Please include mean, median, minimum, maximum, and standard deviation. You can inspect these statistics separately for each market, but in what you report, you may pool all markets.

[455]:
```python
statistics = pd.DataFrame({
    'mean': data.iloc[:, 2:].mean(),
    'median': data.iloc[:, 2:].median(),
    'minimum': data.iloc[:, 2:].min(),
    'maximum': data.iloc[:, 2:].max(),
    'standard deviation': data.iloc[:, 2:].std()
}).T

statistics
```

[455]:
```
                      shares    prices  const     char1     char2     char3  \
mean                0.024299  3.501501    1.0  1.473991  1.498017  1.472444
median              0.009474  3.520012    1.0  1.425105  1.451934  1.482724
minimum             0.000124  1.296677    1.0 -1.166353 -1.147205 -1.387708
maximum             0.268831  6.212918    1.0  4.362983  3.936271  4.482867
standard deviation  0.038427  0.828860    0.0  0.967242  0.913044  0.985834

                     costsh1   costsh2
mean                0.264297  0.238784
median              0.272270  0.236695
minimum             0.005698  0.002097
maximum             0.499834  0.499676
standard deviation  0.140623  0.139543
```

4 4. Pure logit model

1. Suppose agents have the following utility function, where i denotes the agent, and j denotes the product:

$$u_{ij} = \underbrace{\delta_j}_{x_j'\beta - \alpha_{pj} + \xi} + \epsilon_{ij}$$

where ϵ_{ij} is an iid error following a standard Type-I Extreme Value distribution with $F(\epsilon) = e^{-e^{-\epsilon}}$ ("logit" errors). Suppose further that the firms know ξ when setting prices but did not know ξ when setting characteristics.

1.a. What statistical assumptions can you make based on this? Which of your conditions, based on data provided to you, identify the parameter vector of interest, $\theta = (\alpha; \beta)$? In other words, what are valid (and relevant) instruments? Is the model over-identified?

Como ξ só é observado depois de definir os preços, ξ é endógeno em relação ao preço mas não em relação às características dos produtos. Como existe exogeneidade, um modelo OLS seria viesado, por isso é preciso usar instrumentos para estimar o preço.

1.b. Show how you can invert market shares to obtain the mean utility level δ_j for each product.

[456]:
```
data_s0 = (
    data.groupby('market_ids')['shares'].sum().reset_index()
    .assign(s0=lambda x: 1 - x['shares'])
    [['market_ids', 's0']]
)



data_delta = (
    data.merge(data_s0, on='market_ids', how='left')
    .assign(delta=lambda x: np.log(x['shares']) - np.log(x['s0']))
)
```

1.c. Estimate $\theta = (\alpha; \beta)$ and provide standard errors for your estimate. You can try different combinations of instruments, but please use all the different types of instruments that are included or can be constructed from the data (i.e., "BLP instruments").

[457]:
```
data_delta["log_shares"] = np.log(data_delta["shares"]) - np.
 ↪log(data_delta["s0"])
data_delta['firm_ids'] = data_delta['firm_ids'].astype('category')
data_delta['market_ids'] = data_delta['market_ids'].astype('category')
data_delta['char_sum'] = data_delta[['char1', 'char2', 'char3']].sum(axis=1)
market_sums = data_delta.groupby('market_ids')['char_sum'].transform('sum')
data_delta['blp_iv'] = (market_sums - data_delta['char_sum']) / (data_delta.
 ↪groupby('market_ids')['char_sum'].transform('count') - 1)
firm_sums = data_delta.groupby('firm_ids')['prices'].transform('sum')
data_delta['hausman_iv'] = (firm_sums - data_delta['prices']) / (data_delta.
 ↪groupby('market_ids')['prices'].transform('count') - 1)

formula = "log_shares ~ char1 + char2 + char3 + [prices ~ blp_iv + costsh1 +␣
 ↪costsh2 + hausman_iv]"

# Estimar o modelo IV-2SLS (IV-Logit)
logit = IV2SLS.from_formula(formula, data=data_delta).fit(cov_type="robust")

# Exibir os resultados
```

```python
print(logit.first_stage)
```

```
    First Stage Estimation Results
======================================
                                prices
--------------------------------------
R-squared                       0.9796
Partial R-squared               0.1166
Shea's R-squared                0.1166
Partial F-statistic             33.078
P-value (Partial F-stat)     1.151e-06
Partial F-stat Distn           chi2(4)
========================== ===========
char1                           1.0791
                              (8.4259)
char2                           0.8022
                              (5.7481)
char3                          -0.0259
                             (-0.2977)
blp_iv                          0.0575
                              (0.8559)
costsh1                         1.0038
                              (4.3869)
costsh2                         0.6137
                              (2.4982)
hausman_iv                      0.0235
                              (1.3949)
--------------------------------------

T-stats reported in parentheses
T-stats use same covariance type as original model
```

```
/tmp/ipykernel_548659/327344337.py:5: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  market_sums = data_delta.groupby('market_ids')['char_sum'].transform('sum')
/tmp/ipykernel_548659/327344337.py:6: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  data_delta['blp_iv'] = (market_sums - data_delta['char_sum']) /
(data_delta.groupby('market_ids')['char_sum'].transform('count') - 1)
/tmp/ipykernel_548659/327344337.py:7: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  firm_sums = data_delta.groupby('firm_ids')['prices'].transform('sum')
```

```
/tmp/ipykernel_548659/327344337.py:8: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  data_delta['hausman_iv'] = (firm_sums - data_delta['prices']) /
(data_delta.groupby('market_ids')['prices'].transform('count') - 1)
```

[458]:
```python
print(logit.summary)
```

```
                          IV-2SLS Estimation Summary
==============================================================================
Dep. Variable:             log_shares   R-squared:                      0.9868
Estimator:                    IV-2SLS   Adj. R-squared:                 0.9866
No. Observations:                 247   F-statistic:                 1.685e+04
Date:                Sat, Feb 22 2025   P-value (F-stat)                0.0000
Time:                        19:29:01   Distribution:                  chi2(4)
Cov. Estimator:                robust

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
char1          1.4340     0.2149     6.6718     0.0000      1.0128      1.8553
char2          1.0847     0.1800     6.0273     0.0000      0.7320      1.4374
char3          0.3763     0.0525     7.1737     0.0000      0.2735      0.4791
prices        -2.2472     0.1511    -14.875     0.0000     -2.5433     -1.9511
==============================================================================

Endogenous: prices
Instruments: blp_iv, costsh1, costsh2, hausman_iv
Robust Covariance (Heteroskedastic)
Debiased: False
```

[ ]:
```python
#usando o pyblp

logit_formulation = pyblp.Formulation('prices + blp_iv + hausman_iv + costsh1 +␣
 ↪costsh2', absorb='C(market_ids)')
problem = pyblp.Problem(logit_formulation, data_delta)

logit_results = problem.solve()
```

```
Initializing the problem …
Absorbing demand-side fixed effects …
Initialized the problem after 00:00:00.

Dimensions:
================================
 T    N    F    K1    MD    ED
---  ---  ---  ----  ----  ----
```

```
10   247  11    5     4     1
===============================

Formulations:
========================================================================
     Column Indices:          0       1         2          3        4
--------------------------  ------  ------  ----------  -------  -------
X1: Linear Characteristics  prices  blp_iv  hausman_iv  costsh1  costsh2
========================================================================
Solving the problem …
Updating the weighting matrix …
Computed results after 00:00:00.

Problem Results Summary:
=========================================================
GMM       Objective       Clipped   Weighting Matrix
Step        Value         Shares    Condition Number
----  --------------------  -------  --------------------
 1    +6.3923106496717E-29     0     +6.9542689411171E+01
=========================================================

Estimating standard errors …
Computed results after 00:00:00.

Problem Results Summary:
================================================================================
GMM       Objective       Clipped   Weighting Matrix     Covariance Matrix
Step        Value         Shares    Condition Number     Condition Number
----  --------------------  -------  --------------------  --------------------
 2    +3.9990103661963E-29     0     +6.4100425113730E+01  +1.3703000719903E+17
================================================================================

Cumulative Statistics:
========================
Computation   Objective
   Time      Evaluations
-----------  -----------
 00:00:00         2
========================

Beta Estimates (Robust SEs in Parentheses):
=================================================================================
======================================
        prices                  blp_iv                hausman_iv
costsh1                 costsh2
----------------------  ----------------------  ----------------------
----------------------  ----------------------
 -1.8674218370110E-01    -4.5342786153905E-01    -7.5194045607994E-02
```

```
-2.4337402093752E+00    -1.4264892232218E+00
(+1.5953975300724E-01)  (+1.2184238598572E-01)  (+1.2722296852473E-01)
(+6.3526567371086E-01)  (+6.6413984761463E-01)
================================================================================
======================================

The model may be under-identified. The total number of unfixed parameters is 5,
which is more than the total number of moments, 4. Consider checking whether
instruments were properly specified when initializing the problem, and whether
parameters were properly configured when solving the problem.
```

2. Estimate and present the matrix of cross- and own-price elasticities for market 10 based on your model and parameter estimates.

[460]:
```
elasticities = logit_results.compute_elasticities()
elasticities_market_10 = pd.DataFrame(elasticities[data.market_ids == 10])
elasticities_market_10 = elasticities_market_10.dropna(axis=1, how='all')

print(elasticities_market_10)
```

```
Computing elasticities with respect to prices …
Finished after 00:00:00.

          0         1         2         3         4         5         6  \
0  -0.551264  0.085959  0.011161  0.005541  0.035728  0.030416  0.005459
1   0.009565 -0.497555  0.011161  0.005541  0.035728  0.030416  0.005459
2   0.009565  0.085959 -0.759001  0.005541  0.035728  0.030416  0.005459
3   0.009565  0.085959  0.011161 -0.521117  0.035728  0.030416  0.005459
4   0.009565  0.085959  0.011161  0.005541 -0.644579  0.030416  0.005459
5   0.009565  0.085959  0.011161  0.005541  0.035728 -0.442167  0.005459
6   0.009565  0.085959  0.011161  0.005541  0.035728  0.030416 -0.839431
7   0.009565  0.085959  0.011161  0.005541  0.035728  0.030416  0.005459
8   0.009565  0.085959  0.011161  0.005541  0.035728  0.030416  0.005459
9   0.009565  0.085959  0.011161  0.005541  0.035728  0.030416  0.005459
10  0.009565  0.085959  0.011161  0.005541  0.035728  0.030416  0.005459

          7         8         9        10
0   0.001703  0.054924  0.003577  0.001410
1   0.001703  0.054924  0.003577  0.001410
2   0.001703  0.054924  0.003577  0.001410
3   0.001703  0.054924  0.003577  0.001410
4   0.001703  0.054924  0.003577  0.001410
5   0.001703  0.054924  0.003577  0.001410
6   0.001703  0.054924  0.003577  0.001410
7  -0.767453  0.054924  0.003577  0.001410
8   0.001703 -0.475608  0.003577  0.001410
9   0.001703  0.054924 -0.924085  0.001410
10  0.001703  0.054924  0.003577 -0.684688
```

3. In the next question, we are going to free up the substitution pattern by introducing random

coefficients as in BLP. Alternatively, we could think about implementing nested logit, the pure characteristics model, or multinomial probit. Would they be appealing in this setting? Why or why not?

Para usar o modelo de características puras seria necessário ter os dados da demanda que não estão disponíveis. O modelo Pure Logit supõe Independência de Alternativas Irrelevantes (IIA), o que significa que as elasticidades de preço cruzado são proporcionais às participações de mercado, o que pode ser irreal em muitos mercados. O BLP propõe heterogeneidade do consumidor ao permitir coeficientes aleatórios em preço e características, levando a padrões de substituição mais flexíveis.

5 5. Random Coefficient model

1. Suppose agents have the following utility function, where i denotes the agent, and j denotes the product:

$$u_{ij} = \underbrace{\delta_j}_{x_j' - \alpha p_j + \xi_j} + \sum_{k \in \{1,2\}} \sigma_k \nu_{ik} x_{jk} - \sigma_p \nu_{ip} p_j + \epsilon_{ij}$$

where $epsilon_{ij}$ is an iid error following a standard Type-I Extreme Value distribution, and $\nu_{i,\cdot} \overset{iid}{\sim} \mathcal{N}(0,1)$ is an iid standard normal error. To summarize: the model is as before, but with random coefficients on the constant, the first characteristic, and price. The orthogonality/exogeneity assumptions remain the same as before.

1.a. What is the contraction mapping used here for the inner loop? Is there a way to reduce the computational burden from the contraction mapping? (Hint: take a look at page 4 of the appendix to Nevo (2000).) In the following, make sure to set the "inner tolerance" level for the contraction mapping very tight, in your final run ideally on the order of 10^{-14}.

Contraction mapping é usado para encontrar a utilidade média δ_j correspondente ao marketshare s_j previsto pelo modelo. O modelo assume que as utilidades dos produtos convergirá média, assim podemos prever a utilidade média de um produto no próximo período dentro de uma margem de erro (que nesse caso será de $10^(- 14)$). A fórmula é:

$$\delta_j^{t+1} = \delta_j^t + ln(s_j^{obs}) - ln(s_j(\delta^t, \theta))$$

[246]:
```
X1_formulation = pyblp.Formulation('prices + char1 + char2 + char3 + costsh1 +
    ↪costsh2', absorb='C(market_ids)')
X2_formulation = pyblp.Formulation('prices + char1 + char2 + char3 + costsh1 +
    ↪costsh2')
product_formulations = (X1_formulation, X2_formulation)

mc_integration = pyblp.Integration('monte_carlo', size=247,
    ↪specification_options={'seed': 0})
mc_problem = pyblp.Problem(product_formulations, data,
    ↪integration=mc_integration)
bfgs = pyblp.Optimization('bfgs', {'gtol': 1e-14})

results1 = mc_problem.solve(sigma=np.ones((7, 7)), optimization=bfgs)
```

```
results1

#results2 = mc_problem.solve(sigma=np.eye(7), optimization=bfgs)
#results2
```

```
Initializing the problem ...
Absorbing demand-side fixed effects ...
Initialized the problem after 00:00:00.

Dimensions:
============================================
 T    N     F    I     K1    K2    MD    ED
---  ---   ---  ----  ----  ----  ----  ----
10   247   11   2470   6     7     5     1
============================================

Formulations:
================================================================================
======
       Column Indices:           0       1       2      3       4        5
6
-----------------------------  ------  ------  -----  -----  -------  -------
-------
 X1: Linear Characteristics    prices  char1   char2  char3  costsh1  costsh2
X2: Nonlinear Characteristics    1     prices  char1  char2   char3   costsh1
costsh2
================================================================================
======
Solving the problem ...

Nonlinear Coefficient Initial Values:
================================================================================
================================================================================
================================================================================
================================================================================
==============
Sigma:            1                  prices               char1
char2                char3                costsh1              costsh2
|  Sigma Squared:          1                  prices               char1
char2                char3                costsh1              costsh2
-------  --------------------  --------------------  --------------------
--------------------  --------------------  --------------------
--------------------  |  --------------  --------------------
--------------------  --------------------  --------------------
--------------------  --------------------  --------------------
   1     +1.000000000000E+00
|        1         +1.000000000000E+00  +1.000000000000E+00
+1.000000000000E+00  +1.000000000000E+00  +1.000000000000E+00
```

```
+1.0000000000000E+00  +1.0000000000000E+00
prices   +1.0000000000000E+00  +1.0000000000000E+00
|      prices      +1.0000000000000E+00  +2.0000000000000E+00
+2.0000000000000E+00  +2.0000000000000E+00  +2.0000000000000E+00
+2.0000000000000E+00  +2.0000000000000E+00
 char1   +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
|      char1       +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +3.0000000000000E+00  +3.0000000000000E+00
+3.0000000000000E+00  +3.0000000000000E+00
 char2   +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00
|      char2       +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +4.0000000000000E+00  +4.0000000000000E+00
+4.0000000000000E+00  +4.0000000000000E+00
 char3   +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00
|      char3       +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +4.0000000000000E+00  +5.0000000000000E+00
+5.0000000000000E+00  +5.0000000000000E+00
costsh1  +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
|     costsh1      +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +4.0000000000000E+00  +5.0000000000000E+00
+6.0000000000000E+00  +6.0000000000000E+00
costsh2  +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  |     costsh2      +1.0000000000000E+00
+2.0000000000000E+00  +3.0000000000000E+00  +4.0000000000000E+00
+5.0000000000000E+00  +6.0000000000000E+00  +7.0000000000000E+00
================================================================================
================================================================================
================================================================================
================================================================================
==============
Starting optimization …

The model may be under-identified. The total number of unfixed parameters is 34,
which is more than the total number of moments, 5. Consider checking whether
instruments were properly specified when initializing the problem, and whether
parameters were properly configured when solving the problem.

GMM   Computation  Optimization   Objective   Fixed Point  Contraction  Clipped
Objective            Objective               Gradient
Step     Time       Iterations   Evaluations  Iterations   Evaluations  Shares
Value              Improvement               Norm
Theta
----  -----------  ------------  -----------  -----------  -----------  -------
--------------------  --------------------  --------------------  --------------
```

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----------------------------------------
 1     00:00:00         0             1           289          875         0
+2.1095646627429E-26                          +1.4360154494112E-12
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0             2           289          874         0
+2.0244918422545E-26  +8.5072820488317E-28  +1.5408216236289E-12
+9.9999999999994E-01, +1.0000000000000E+00, +1.0000000000002E+00,
+9.9999999999911E-01, +9.9999999999938E-01, +1.0000000000002E+00,
+1.0000000000014E+00, +1.0000000000012E+00, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000014E+00, +1.0000000000014E+00,
+1.0000000000003E+00, +1.0000000000000E+00, +9.9999999999997E-01,
+9.9999999999985E-01, +9.9999999999988E-01, +1.0000000000001E+00,
+1.0000000000002E+00, +1.0000000000001E+00, +9.9999999999996E-01,
+9.9999999999983E-01, +9.9999999999984E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999995E-01,
+9.9999999999997E-01
```

```
ERROR:tornado.application:Exception in callback functools.partial(<bound method
OutStream._flush of <ipykernel.iostream.OutStream object at 0x7f1096bfb6a0>>)
Traceback (most recent call last):
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/tornado/ioloop.py", line 750, in _run_callback
    ret = callback()
          ^^^^^^^^^^

  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/ipykernel/iostream.py", line 639, in _flush
    msg = self.session.msg("stream", content, parent=parent)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 661, in msg
    header = self.msg_header(msg_type) if header is None else header
             ^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 644, in msg_header
    return msg_header(self.msg_id, msg_type, self.username, self.session)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 275, in msg_header
    date = utcnow()
           ^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 203, in utcnow
    return datetime.utcnow().replace(tzinfo=utc)
           ^^^^^^^^^^^^^^^^^
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for
removal in a future version. Use timezone-aware objects to represent datetimes
in UTC: datetime.datetime.now(datetime.UTC).

 1     00:00:00         1             4           287          871         0
+2.9563479070908E-26                          +1.8955266408674E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00         1             5           289          878         0
+1.9920805256139E-26                          +1.5614458852229E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00         1             6           287          871         0
+2.9563479070908E-26                          +1.8955266408674E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
```

```
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00        1            7            289          878          0
+2.9329679174586E-26                          +1.8833304272592E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:01        1            8            286          868          0
+4.2320586612017E-26                          +1.8842642122031E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00        1            9            289          878          0
+1.9920805256139E-26                          +1.5614458852229E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01

The optimization routine failed to converge. This problem can sometimes be
mitigated by choosing more reasonable initial parameter values, setting more
conservative bounds, or configuring other optimization settings.


Optimization failed after 00:00:04.
Computing the Hessian and and updating the weighting matrix …
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[246], line 9
      6 mc_problem = pyblp.Problem(product_formulations, data,␣
  ↪integration=mc_integration)
      7 bfgs = pyblp.Optimization('bfgs', {'gtol': 1e-14})
----> 9 results1 = mc_problem.solve(sigma=np.ones((7, 7)), optimization=bfgs)
     10 results1
     12 #results2 = mc_problem.solve(sigma=np.eye(7), optimization=bfgs)
     13 #results2

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/economies/
  ↪problem.py:715, in ProblemEconomy.solve(self, sigma, pi, rho, beta, gamma,␣
  ↪sigma_bounds, pi_bounds, rho_bounds, beta_bounds, gamma_bounds, delta, method␣
  ↪initial_update, optimization, scale_objective, check_optimality,␣
  ↪finite_differences, error_behavior, error_punishment, delta_behavior,␣
  ↪iteration, fp_type, shares_bounds, costs_bounds, W, center_moments, W_type,␣
  ↪se_type, covariance_moments_mean, micro_moments, micro_sample_covariances,␣
  ↪resample_agent_data)
    713 else:
    714     output("Estimating standard errors …")
--> 715 final_progress = compute_step_progress(
    716␣
  ↪    theta, progress, compute_gradient, compute_hessian, compute_micro_covariances,
    717     detect_micro_collinearity, compute_simulation_covariances
    718 )
    719 iteration_stats.append(final_progress.iteration_stats)
    720 optimization_stats.evaluations += 1

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/economies/
  ↪problem.py:1172, in ProblemEconomy._compute_progress(self, parameters,␣
  ↪moments, iv, W, scale_objective, error_behavior, error_punishment,␣
  ↪delta_behavior, iteration, fp_type, shares_bounds, costs_bounds,␣
  ↪finite_differences, covariance_moments_mean, resample_agent_data, theta,␣
  ↪progress, compute_gradient, compute_hessian, compute_micro_covariances,␣
  ↪detect_micro_collinearity, compute_simulation_covariances, agents_override)
   1169         return perturbed_progress.gradient
   1171     # compute the Hessian, enforcing shape and symmetry
-> 1172     hessian =␣
  ↪compute_finite_differences(compute_perturbed_gradient, theta)
   1173     hessian = np.c_[hessian + hessian.T] / 2
   1175 # optionally resample agents to compute simulation covariances

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/utilities/
  ↪basics.py:412, in compute_finite_differences(f, x, epsilon_scale)
    410     x1[index] += epsilon / 2
    411     x2[index] -= epsilon / 2
--> 412     arrays.append((f(x1) - f(x2)) / epsilon)
    414 if len(arrays[0].shape) == 1 or (len(arrays[0].shape) == 2 and arrays[0].
  ↪shape[1] == 1):
```

```
    415     return np.column_stack(arrays)

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/economies/problem.py:1162, in ProblemEconomy._compute_progress.<locals>.compute_perturbed_gradient(perturbed_theta)
   1160 def compute_perturbed_gradient(perturbed_theta: Array) -> Array:
   1161     """Evaluate the gradient at a perturbed parameter vector."""
-> 1162     perturbed_progress = self._compute_progress(
   1163         parameters, moments, iv, W, scale_objective, error_behavior, error_pu ishment, delt
   1164         iteration, fp_type, shares_bounds, costs_bounds, finite_differences, covariance_mom
   1165         resample_agent_data, perturbed_theta, progress, compute_gradient=True  compute_hess
   1166         compute_micro_covariances=False, detect_micro_collinearity=False,
   1167         compute_simulation_covariances=False,
   1168     )
   1169     return perturbed_progress.gradient

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/economies/problem.py:839, in ProblemEconomy._compute_progress(self, parameters, moments, iv, W, scale_objective, error_behavior, error_punishment, delta_behavior, iteration, fp_type, shares_bounds, costs_bounds, finite_differences, covariance_moments_mean, resample_agent_data, theta, progress, compute_gradient, compute_hessian, compute_micro_covariances, detect_micro_collinearity, compute_simulation_covariances, agents_override)
    837 parts_collinearity_candidate_values: Dict[Hashable, Dict[MicroDataset, Array]] = {}
    838 generator = generate_items(self.unique_market_ids, market_factory, ProblemMarket.solve)
--> 839 for t, generated_t in generator:
    840     (
    841         delta_t, xi_jacobian_t, parts_numerator_t, parts_denominator_t, parts numerator_jac
    842         parts_denominator_jacobian_t, parts_covariances_numerator_t, weights_ apping_t, va
    843         clipped_shares_t, iteration_stats_t, tilde_costs_t, omega_jacobian_t, clipped_costs
    844     ) = generated_t
    846     delta[self._product_market_indices[t]] = delta_t

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/utilities/basics.py:164, in <genexpr>(.0)
    159 """Generate (key, method(*factory(key))) tuples for each key. The first element returned by factory is an instance
    160 of the class to which method is attached. If a process pool has been initialized, use multiprocessing; otherwise,
    161 use serial processing.
    162 """
```

```
    163 if pool is None:
--> 164     return (generate_items_worker((k, factory(k), method)) for k in keys)
    165 return pool.imap_unordered(generate_items_worker, ((k, factory(k),␣
↪method) for k in keys))

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/utilities/
↪basics.py:173, in generate_items_worker(args)
    169 """Call the specified method of a class instance with any additional␣
↪arguments. Return the associated key along with
    170 the returned object.
    171 """
    172 key, (instance, *method_args), method = args
--> 173 return key, method(instance, *method_args)

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/utilities/
↪basics.py:665, in NumericalErrorHandler.__call__.<locals>.wrapper(*args,␣
↪**kwargs)
    663 with np.errstate(divide='call', over='call', under='ignore',␣
↪invalid='call'):
    664     np.seterrcall(detector)
--> 665     returned = decorated(*args, **kwargs)
    666 if detector.detected is not None:
    667     returned[-1].append(detector.detected)

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/markets/
↪problem_market.py:37, in ProblemMarket.solve(self, delta, last_delta,␣
↪last_tilde_costs, moments, iteration, fp_type, shares_bounds, costs_bounds,␣
↪compute_jacobians, compute_micro_covariances, keep_micro_mappings)
     34 errors: List[Error] = []
     36 # solve the contraction
---> 37 delta, clipped_shares, stats, delta_errors =␣
↪self.safely_compute_delta(delta, iteration, fp_type, shares_bounds)
     38 errors.extend(delta_errors)
     40 # replace invalid values in delta with their last computed values

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/utilities/
↪basics.py:665, in NumericalErrorHandler.__call__.<locals>.wrapper(*args,␣
↪**kwargs)
    663 with np.errstate(divide='call', over='call', under='ignore',␣
↪invalid='call'):
    664     np.seterrcall(detector)
--> 665     returned = decorated(*args, **kwargs)
    666 if detector.detected is not None:
    667     returned[-1].append(detector.detected)

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/markets/
↪problem_market.py:145, in ProblemMarket.safely_compute_delta(self,␣
↪initial_delta, iteration, fp_type, shares_bounds)
    138 @NumericalErrorHandler(exceptions.DeltaNumericalError)
```

```
    139 def safely_compute_delta(
    140         self, initial_delta: Array, iteration: Iteration, fp_type: str, shares_bounds: Bounds) -> (
    141         Tuple[Array, Array, SolverStats, List[Error]]):
    142     """Compute the mean utility for this market that equates market shares to observed values by solving a fixed
    143     point problem, handling any numerical errors.
    144     """
--> 145     delta, clipped_shares, stats, errors = self.compute_delta(initial_delta, iteration, fp_type, shares_bounds)
    146     if not stats.converged:
    147         errors.append(exceptions.DeltaConvergenceError())

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/markets/market.py:568, in Market.compute_delta(self, initial_delta, iteration, fp_type, shares_bounds)
    565             return x, None, jacobian
    567     # solve the linear fixed point problem
--> 568     delta, stats = iteration._iterate(initial_delta, contraction)
    569 else:
    570     # solve for delta with a nonlinear fixed point
    571     assert 'nonlinear' in fp_type and self.epsilon_scale == 1

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/configurations/iteration.py:284, in Iteration._iterate(self, initial, contraction)
    281 raw_initial = initial.astype(np.float64, copy=False).flatten()
    283 # solve the problem and convert the raw final values to the same data type and shape as the initial values
--> 284 raw_final, converged = self._iterator(
    285     raw_initial, contraction_wrapper, iteration_callback, **self._method_options
    286 )
    287 final = np.asarray(raw_final).astype(initial.dtype, copy=False).reshape(initial.shape)
    288 stats = SolverStats(converged, iterations, evaluations)

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/configurations/iteration.py:442, in squarem_iterator(initial, contraction, iteration_callback, max_evaluations, atol, rtol, norm, scheme, step_min, step_max, step_factor)
    440 # acceleration step
    441 x2, x = x, x0 - 2 * alpha * r + alpha**2 * v
--> 442 x3, (x, weights) = x, contraction(x)[:2]
    443 if not all_finite(x, weights):
    444     x = x2
```

```
File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/
↪configurations/iteration.py:273, in Iteration._iterate.<locals>.
↪contraction_wrapper(raw_values)
    271     raw_values = np.asarray(raw_values)
    272 values = raw_values.reshape(initial.shape).astype(initial.dtype,␣
↪copy=False)
--> 273 values, weights, jacobian = contraction(values, iterations, evaluations
    274 return (
    275     values.astype(raw_values.dtype, copy=False).reshape(raw_values.
↪shape),
    276     None if weights is None else weights.astype(raw_values.dtype,␣
↪copy=False).reshape(raw_values.shape),
    277     None if jacobian is None else jacobian.astype(raw_values.dtype,␣
↪copy=False)
    278 )

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/markets/
↪market.py:540, in Market.compute_delta.<locals>.contraction(x, iterations,␣
↪evaluations)
    538 probabilities = compute_probabilities(x)[0]
    539 shares = probabilities @ self.agents.weights
--> 540 clip_shares(shares)
    541 x0, x = x, x + log_shares - np.log(shares)
    542 universal_display(x0, x, iterations, evaluations)

File ~/anaconda3/envs/blp_python/lib/python3.12/site-packages/pyblp/markets/
↪market.py:521, in Market.compute_delta.<locals>.clip_shares(shares)
    518         clipped_shares = small_shares | large_shares
    520 elif np.isfinite(shares_bounds[0]):
--> 521     def clip_shares(shares: Array) -> None:
    522         """Clip shares from below."""
    523         nonlocal clipped_shares

KeyboardInterrupt:
```

[161]:

```
results2 = mc_problem.solve(sigma=np.eye(7), optimization=bfgs)
results2
```

```
Solving the problem …

Nonlinear Coefficient Initial Values:
================================================================================
================================================================================
=
Sigma:             1                    prices                 char1
char2                    char3                    costsh1                  costsh2
-------  --------------------  --------------------  --------------------
--------------------  --------------------  --------------------
```

```
--------------------
   1     +1.0000000000000E+00
prices   +0.0000000000000E+00  +1.0000000000000E+00
 char1   +0.0000000000000E+00  +0.0000000000000E+00  +1.0000000000000E+00
 char2   +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+1.0000000000000E+00
 char3   +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +1.0000000000000E+00
costsh1  +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +0.0000000000000E+00  +1.0000000000000E+00
costsh2  +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+1.0000000000000E+00
================================================================================
================================================================================
=

Nonlinear Coefficient Lower Bounds:
================================================================================
================================================================================
=
Sigma:            1                   prices               char1
char2                 char3                costsh1              costsh2
-------  --------------------  --------------------  --------------------
--------------------  --------------------  --------------------
--------------------
   1            -INF
prices   +0.0000000000000E+00          -INF
 char1   +0.0000000000000E+00  +0.0000000000000E+00          -INF
 char2   +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
-INF
 char3   +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00          -INF
costsh1  +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +0.0000000000000E+00          -INF
costsh2  +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00          -INF
================================================================================
================================================================================
=

Nonlinear Coefficient Upper Bounds:
================================================================================
================================================================================
=
Sigma:            1                   prices               char1
char2                 char3                costsh1              costsh2
-------  --------------------  --------------------  --------------------
```

```
--------------------  --------------------  --------------------
--------------------
   1             +INF
prices   +0.0000000000000E+00          +INF
 char1   +0.0000000000000E+00  +0.0000000000000E+00          +INF
 char2   +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+INF
 char3   +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00          +INF
costsh1  +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +0.0000000000000E+00          +INF
costsh2  +0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00
+0.0000000000000E+00  +0.0000000000000E+00  +0.0000000000000E+00          +INF
================================================================================
================================================================================
=

Starting optimization …


The model may be under-identified. The total number of unfixed parameters is 13,
which is more than the total number of moments, 5. Consider checking whether
instruments were properly specified when initializing the problem, and whether
parameters were properly configured when solving the problem.

GMM   Computation  Optimization   Objective   Fixed Point  Contraction  Clipped
Objective           Objective               Gradient
Step     Time       Iterations   Evaluations  Iterations   Evaluations  Shares
Value              Improvement              Norm
Theta
----  -----------  ------------  -----------  -----------  -----------  -------
--------------------  --------------------  --------------------  --------------
--------------------------------------------------------------------------------
----------------------------------------------------------
 1     00:00:00         0             1           152          466         0
+8.9635197677287E-27                        +3.7197706976986E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0             2           153          466         0
+1.1365907480957E-26                        +2.4272527585076E-13
+9.9999999999999E-01, +9.9999999999963E-01, +9.9999999999973E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00         0             3           151          464         0
+1.0669137757789E-26                        +3.0739409171452E-13
+1.0000000000000E+00, +9.9999999999991E-01, +9.9999999999994E-01,
+1.0000000000001E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
```

```
 1     00:00:00         0              4            152           465          0
+1.2301786229545E-26                         +3.1210840751105E-13
+1.0000000000000E+00, +9.9999999999998E-01, +9.9999999999999E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              5            151           463          0
+1.0095763037280E-26                         +3.4684848159347E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              6            152           465          0
+9.3494129523740E-27                         +2.1150956344006E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              7            153           466          0
+1.1917740463648E-26                         +5.5019121699312E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              8            152           466          0
+8.9635197677287E-27                         +3.7197706976986E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              9            153           466          0
+1.1365907480957E-26                         +2.4272527585076E-13
+9.9999999999999E-01, +9.9999999999963E-01, +9.9999999999973E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00         0              10           151           465          0
+1.3217532034510E-26                         +5.0285435273880E-13
+9.9999999999999E-01, +9.9999999999982E-01, +9.9999999999987E-01,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+1.0000000000000E+00
 1     00:00:00         0              11           151           464          0
+1.3637691107537E-26                         +3.6957567372597E-13
+1.0000000000000E+00, +9.9999999999993E-01, +9.9999999999995E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              12           151           462          0
+1.0648897140064E-26                         +4.2098003357311E-13
+1.0000000000000E+00, +9.9999999999997E-01, +9.9999999999998E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00         0              13           152           466          0
+9.5215733570783E-27                         +2.2724792234986E-13
+1.0000000000000E+00, +9.9999999999999E-01, +9.9999999999999E-01,
```

```
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        0            14           152          464          0
+1.2693895229871E-26                        +3.6955743320319E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        0            15           153          466          0
+7.7899683121292E-27  +1.1735514555995E-27  +1.9191466875898E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            16           152          465          0
+1.0139815249664E-26                        +4.4672412659205E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            17           152          465          0
+8.1821715010909E-27                        +1.6223262548003E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            18           153          466          0
+7.7899683121292E-27                        +1.9191466875898E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            19           152          465          0
+1.0139815249664E-26                        +4.4672412659205E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            20           151          464          0
+9.7272583619471E-27                        +2.3974216159463E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            21           153          466          0
+1.0711903189629E-26                        +3.1753179573566E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 1     00:00:00        1            22           153          466          0
+7.7899683121292E-27                        +1.9191466875898E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
```

```
The optimization routine failed to converge. This problem can sometimes be
mitigated by choosing more reasonable initial parameter values, setting more
conservative bounds, or configuring other optimization settings.


Optimization failed after 00:00:03.
Computing the Hessian and and updating the weighting matrix …
Computed results after 00:00:02.

Problem Results Summary:
================================================================================
===========================================
GMM       Objective             Gradient              Hessian
Hessian       Clipped    Weighting Matrix
Step        Value                 Norm            Min Eigenvalue        Max
Eigenvalue     Shares     Condition Number
----  --------------------  --------------------  --------------------
--------------------  -------  --------------------
 1    +7.7899683121292E-27  +1.9191466875898E-13  -2.1941880127562E-05
+4.1381323863492E-05     0     +9.1579102736529E+01
================================================================================
===========================================

Starting optimization …

GMM   Computation  Optimization   Objective   Fixed Point  Contraction  Clipped
Objective             Objective              Gradient
Step     Time       Iterations   Evaluations  Iterations   Evaluations  Shares
Value              Improvement               Norm
Theta
----  -----------  ------------  -----------  -----------  -----------  -------
--------------------  --------------------  --------------------  --------------
--------------------------------------------------------------------------------
----------------------------------------------------------
 2     00:00:00         0             1            0           10           0
+8.3314662674042E-27                        +3.5399277824058E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00         0             2           19           72           0
+9.5832410829883E-27                        +6.0471461108433E-13
+9.9999999999999E-01, +1.0000000000004E+00, +9.9999999999996E-01,
+1.0000000000003E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00         0             3           11           39           0
+8.4415243908054E-27                        +8.3545398289872E-13
+1.0000000000000E+00, +1.0000000000001E+00, +9.9999999999999E-01,
+1.0000000000001E+00, +1.0000000000000E+00, +1.0000000000000E+00,
```

```
+1.0000000000000E+00
 2     00:00:00        0            4            1           21          0
+6.3573671462273E-27  +1.9740991211769E-27  +7.5026601248160E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            5            4           24          0
+8.3711178230008E-27                        +7.9593741277169E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            6            1           21          0
+5.7616852070022E-27  +5.9568193922502E-28  +5.3175733267908E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            7            3           23          0
+9.1658114707901E-27                        +9.6343570743282E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            8            1           21          0
+5.9363418430965E-27                        +5.6271220450056E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            9            1           21          0
+7.8824940785541E-27                        +6.1865822222269E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0           10            1           21          0
+5.7616852070022E-27                        +5.3175733267908E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0           11            1           21          0
+7.8824940785541E-27                        +6.1865822222269E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
```

```
ERROR:tornado.application:Exception in callback functools.partial(<bound method
OutStream._flush of <ipykernel.iostream.OutStream object at 0x7f1096bfb6a0>>)
Traceback (most recent call last):
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/tornado/ioloop.py", line 750, in _run_callback
    ret = callback()
```

```
          ^^^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/ipykernel/iostream.py", line 639, in _flush
    msg = self.session.msg("stream", content, parent=parent)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 661, in msg
    header = self.msg_header(msg_type) if header is None else header
             ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 644, in msg_header
    return msg_header(self.msg_id, msg_type, self.username, self.session)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 275, in msg_header
    date = utcnow()
           ^^^^^^^^
  File "/home/hspassos/anaconda3/envs/blp_python/lib/python3.12/site-
packages/jupyter_client/session.py", line 203, in utcnow
    return datetime.utcnow().replace(tzinfo=utc)
           ^^^^^^^^^^^^^^^^^
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for
removal in a future version. Use timezone-aware objects to represent datetimes
in UTC: datetime.datetime.now(datetime.UTC).
 2     00:00:00         0            18           1          21          0
+5.7616852070022E-27                       +5.3175733267908E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00         0            19           1          21          0
+7.8824940785541E-27                       +6.1865822222269E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00         0            20           1          21          0
+5.7616852070022E-27                       +5.3175733267908E-13
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00         0            21          19          72          0
+9.5832410829883E-27                       +6.0471461108433E-13
+9.9999999999999E-01, +1.0000000000004E+00, +9.9999999999996E-01,
+1.0000000000003E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00         0            22          16          61          0
+8.1764652982900E-27                       +6.5929271794309E-13
+9.9999999999999E-01, +1.0000000000002E+00, +9.9999999999998E-01,
```

```
+1.0000000000001E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            23           18           68            0
+6.5351327668380E-27                        +7.6696411209601E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
 2     00:00:00        0            24           19           71            0
+7.5589633217122E-27                        +5.3478663666325E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000003E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            25           19           70            0
+6.0631523025041E-27                        +6.5926697498519E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            26           19           71            0
+1.0030544328411E-26                        +6.9555238244526E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000003E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            27           19           72            0
+4.8864026296984E-27  +8.7528257730382E-28  +6.2486433780701E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            28           19           72            0
+8.8479763140511E-27                        +8.6748439712425E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000003E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            29           19           71            0
+1.1588915424696E-26                        +8.6759650561113E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            30           20           73            0
+6.0967049088638E-27                        +6.1533619431677E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
 2     00:00:00        0            31           18           69            0
+6.5375299705659E-27                        +6.6649400204766E-13
+9.9999999999999E-01, +1.0000000000003E+00, +9.9999999999997E-01,
+1.0000000000002E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999999E-01
```

```
The optimization routine failed to converge. This problem can sometimes be
mitigated by choosing more reasonable initial parameter values, setting more
conservative bounds, or configuring other optimization settings.


Optimization failed after 00:00:01.
Computing the Hessian and estimating standard errors …
Computed results after 00:00:01.

Problem Results Summary:
================================================================================
=================================================================
GMM       Objective            Gradient              Hessian
Hessian        Clipped    Weighting Matrix     Covariance Matrix
Step         Value                Norm            Min Eigenvalue        Max
Eigenvalue     Shares     Condition Number      Condition Number
----  --------------------  --------------------  --------------------
--------------------  -------  --------------------  --------------------
 2    +8.3314662674042E-27  +3.5399277824058E-13  -2.3998431332276E-05
+2.8380720415443E-05     0     +1.8856983557030E+02  +4.9371086509037E+18
================================================================================
=================================================================

Cumulative Statistics:
===========================================================================
Computation  Optimizer  Optimization   Objective   Fixed Point  Contraction
   Time      Converged   Iterations   Evaluations  Iterations   Evaluations
-----------  ---------  ------------  -----------  -----------  -----------
 00:00:07       No           1            55          3755         11960
===========================================================================

Nonlinear Coefficient Estimates (Robust SEs in Parentheses):
================================================================================
================================================================================
===============
Sigma:             1                     prices                  char1
char2                  char3                  costsh1                  costsh2
-------  ----------------------  ----------------------  ----------------------
----------------------  ----------------------  ----------------------
----------------------
   1      +1.0000000000000E+00
         (+6.9622432663470E-03)

prices    +0.0000000000000E+00    +1.0000000000000E+00
                                 (+1.1737609212766E-01)

 char1    +0.0000000000000E+00    +0.0000000000000E+00    +1.0000000000000E+00
                                                         (+9.0953194355645E-02)
```

```
 char2    +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+1.0000000000000E+00
 (+1.0909120619031E-01)

 char3    +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+0.0000000000000E+00    +1.0000000000000E+00
                         (+1.8631450401344E-01)

costsh1   +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+0.0000000000000E+00    +0.0000000000000E+00    +1.0000000000000E+00
                                                 (+1.5108293984020E-02)

costsh2   +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+1.0000000000000E+00
(+1.8371386259109E-02)
================================================================================
================================================================================
===============

Beta Estimates (Robust SEs in Parentheses):
================================================================================
==============================================================
        prices                  char1                   char2
char3                  costsh1                 costsh2
----------------------  ----------------------  ----------------------
----------------------  ----------------------  ----------------------
 -2.2313863244623E+00    -1.9431473549215E-02    -5.0087268870514E-02
+4.6325483590655E-01    -1.5282164349516E+00    -1.2621283788897E+00
(+8.4358996302135E-02)  (+1.4630974707531E-01)  (+1.8136509195552E-01)
(+1.3736290110478E-01)  (+2.7318420317347E-01)  (+3.1591550864354E-01)
================================================================================
==============================================================
```

[161]:

```
Problem Results Summary:
================================================================================
=================================================================
GMM        Objective            Gradient               Hessian
Hessian        Clipped    Weighting Matrix     Covariance Matrix
Step          Value                  Norm            Min Eigenvalue        Max
Eigenvalue     Shares     Condition Number      Condition Number
----   --------------------  --------------------  --------------------
--------------------  -------  --------------------  --------------------
 2    +8.3314662674042E-27  +3.5399277824058E-13  -2.3998431332276E-05
+2.8380720415443E-05     0     +1.8856983557030E+02  +4.9371086509037E+18
================================================================================
```

```
=================================================================

Cumulative Statistics:
===========================================================================
Computation  Optimizer  Optimization   Objective   Fixed Point  Contraction
   Time      Converged   Iterations   Evaluations  Iterations   Evaluations
-----------  ---------  ------------  -----------  -----------  -----------
 00:00:07       No           1             55          3755        11960
===========================================================================

Nonlinear Coefficient Estimates (Robust SEs in Parentheses):
================================================================================
================================================================================
===============
Sigma:             1                     prices                   char1
char2                   char3                  costsh1                  costsh2
-------  ----------------------  ----------------------  ----------------------
----------------------  ----------------------  ----------------------
----------------------
   1      +1.0000000000000E+00
         (+6.9622432663470E-03)

prices    +0.0000000000000E+00    +1.0000000000000E+00
                                 (+1.1737609212766E-01)

 char1    +0.0000000000000E+00    +0.0000000000000E+00    +1.0000000000000E+00
                                                         (+9.0953194355645E-02)

 char2    +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+1.0000000000000E+00
 (+1.0909120619031E-01)

 char3    +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+0.0000000000000E+00    +1.0000000000000E+00
                         (+1.8631450401344E-01)

costsh1   +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+0.0000000000000E+00    +0.0000000000000E+00    +1.0000000000000E+00
                                                 (+1.5108293984020E-02)

costsh2   +0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+0.0000000000000E+00    +0.0000000000000E+00    +0.0000000000000E+00
+1.0000000000000E+00
(+1.8371386259109E-02)
================================================================================
================================================================================
===============
```

```
Beta Estimates (Robust SEs in Parentheses):
================================================================================
==============================================================
        prices                  char1                  char2
char3                  costsh1                costsh2
----------------------  ----------------------  ----------------------
----------------------  ----------------------  ----------------------
 -2.2313863244623E+00    -1.9431473549215E-02    -5.0087268870514E-02
+4.6325483590655E-01    -1.5282164349516E+00    -1.2621283788897E+00
(+8.4358996302135E-02)  (+1.4630974707531E-01)  (+1.8136509195552E-01)
(+1.3736290110478E-01)  (+2.7318420317347E-01)  (+3.1591550864354E-01)
================================================================================
==============================================================
```

1.b. Write the parameter vector of interest as $\theta = (\theta_1, \theta_2)$, where θ_1 are the "linear" parameters, and " _2" are the "nonlinear" parameters. Which parameters are in θ_1 and which are in θ_2? What does this imply for estimation?

[137]:
```python
theta_1 = pd.DataFrame(logit_results.beta)
theta_2 = pd.DataFrame(results2.beta)
theta_1
theta_2
```

[137]:

	0
0	4.018726
1	-1.874091
2	-0.368675
3	-0.303791
4	0.452330
5	-1.865919
6	-1.473477
7	-0.789412
8	0.215525
9	-0.795686
10	-0.517398
11	-0.863983
12	0.769933
13	-0.301054
14	-2.337703
15	-1.704664

1.c. Bonus question: Explain how the variance terms σ are identified from variation in the choice set and prices.

1.d. Estimate the model using 2-step optimal GMM. In addition to your point estimates, please provide standard errors (Hint: take a look at page 6 of the appendix to Nevo (2000) for analytic standard errors, and/or use finite differences for a numerical approximation). If you try different starting values, do your estimates change?

[165]:
```python
results3 = mc_problem.solve(sigma=np.ones((7, 7)), method='2s',␣
    ↪optimization=bfgs)
results3

print("Parameter Estimates:")
print(results3.beta)
print(results3.sigma)
```

```
Solving the problem …

Nonlinear Coefficient Initial Values:
================================================================================
================================================================================
================================================================================
================================================================================
==============
Sigma:            1                   prices                char1
char2                  char3                 costsh1               costsh2
|  Sigma Squared:          1                   prices                char1
char2                  char3                 costsh1               costsh2
-------  --------------------  --------------------  --------------------
--------------------  --------------------  --------------------
--------------------  |  --------------  --------------------
--------------------  --------------------  --------------------
--------------------  --------------------  --------------------
   1     +1.0000000000000E+00
|        1          +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00
prices   +1.0000000000000E+00  +1.0000000000000E+00
|      prices       +1.0000000000000E+00  +2.0000000000000E+00
+2.0000000000000E+00  +2.0000000000000E+00  +2.0000000000000E+00
+2.0000000000000E+00  +2.0000000000000E+00
 char1   +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
|       char1       +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +3.0000000000000E+00  +3.0000000000000E+00
+3.0000000000000E+00  +3.0000000000000E+00
 char2   +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00
|       char2       +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +4.0000000000000E+00  +4.0000000000000E+00
+4.0000000000000E+00  +4.0000000000000E+00
 char3   +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00
|       char3       +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +4.0000000000000E+00  +5.0000000000000E+00
+5.0000000000000E+00  +5.0000000000000E+00
costsh1  +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
```

```
+1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
|     costsh1      +1.0000000000000E+00  +2.0000000000000E+00
+3.0000000000000E+00  +4.0000000000000E+00  +5.0000000000000E+00
+6.0000000000000E+00  +6.0000000000000E+00
costsh2  +1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  +1.0000000000000E+00  +1.0000000000000E+00
+1.0000000000000E+00  |     costsh2      +1.0000000000000E+00
+2.0000000000000E+00  +3.0000000000000E+00  +4.0000000000000E+00
+5.0000000000000E+00  +6.0000000000000E+00  +7.0000000000000E+00
================================================================================
================================================================================
================================================================================
================================================================================
==============
Starting optimization …


The model may be under-identified. The total number of unfixed parameters is 34,
which is more than the total number of moments, 5. Consider checking whether
instruments were properly specified when initializing the problem, and whether
parameters were properly configured when solving the problem.

GMM   Computation  Optimization   Objective   Fixed Point  Contraction  Clipped
Objective             Objective              Gradient
Step     Time       Iterations   Evaluations  Iterations   Evaluations  Shares
Value              Improvement               Norm
Theta
----  -----------  ------------  -----------  -----------  -----------  -------
--------------------  --------------------  --------------------  --------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----------------------------------------
 1     00:00:01         0             1            289          875         0
+2.1095646627429E-26                        +1.4360154494112E-12
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+1.0000000000000E+00
```

```
 1     00:00:00         0             2           289          874          0
+2.0244918422545E-26  +8.5072820488317E-28  +1.5408216236289E-12
+9.9999999999994E-01, +1.0000000000000E+00, +1.0000000000002E+00,
+9.9999999999911E-01, +9.9999999999938E-01, +1.0000000000002E+00,
+1.0000000000014E+00, +1.0000000000012E+00, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000014E+00, +1.0000000000014E+00,
+1.0000000000003E+00, +1.0000000000000E+00, +9.9999999999997E-01,
+9.9999999999985E-01, +9.9999999999988E-01, +1.0000000000001E+00,
+1.0000000000002E+00, +1.0000000000001E+00, +9.9999999999996E-01,
+9.9999999999983E-01, +9.9999999999984E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999995E-01,
+9.9999999999997E-01
 1     00:00:00         0             3           289          878          0
+1.9920805256139E-26  +3.2411316640615E-28  +1.5614458852229E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00         1             4           287          871          0
+2.9563479070908E-26                        +1.8955266408674E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00         1             5           289          878          0
+1.9920805256139E-26                        +1.5614458852229E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
```

```
 1     00:00:00        1             6            287          871          0
+2.9563479070908E-26                             +1.8955266408674E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00        1             7            289          878          0
+2.9329679174586E-26                             +1.8833304272592E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00        1             8            286          868          0
+4.2320586612017E-26                             +1.8842642122031E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 1     00:00:00        1             9            289          878          0
+1.9920805256139E-26                             +1.5614458852229E-12
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
```

```
The optimization routine failed to converge. This problem can sometimes be
mitigated by choosing more reasonable initial parameter values, setting more
conservative bounds, or configuring other optimization settings.


Optimization failed after 00:00:02.
Computing the Hessian and and updating the weighting matrix …
Computed results after 00:00:16.

Problem Results Summary:
================================================================================
===========================================
GMM       Objective              Gradient               Hessian
Hessian        Clipped    Weighting Matrix
Step        Value                  Norm             Min Eigenvalue        Max
Eigenvalue     Shares     Condition Number
----  --------------------  --------------------  --------------------
--------------------  -------  --------------------
 1    +1.9920805256139E-26  +1.5614458852229E-12  -2.9176296080531E-04
+3.1418709849730E-04     0     +9.1579102736529E+01
================================================================================
===========================================

Starting optimization …

GMM   Computation  Optimization   Objective   Fixed Point  Contraction  Clipped
Objective             Objective              Gradient
Step     Time       Iterations   Evaluations  Iterations   Evaluations  Shares
Value              Improvement               Norm
Theta
----  -----------  ------------  -----------  -----------  -----------  -------
--------------------  --------------------  --------------------  --------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----------------------------------------
 2     00:00:00         0            1            0           11           0
+6.0533391004769E-26                        +4.7124267143486E-13
+9.9999999999999E-01, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999981E-01, +9.9999999999986E-01, +1.0000000000000E+00,
+1.0000000000003E+00, +1.0000000000003E+00, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000003E+00, +1.0000000000003E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999999E-01,
```

```
+9.9999999999997E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999996E-01, +9.9999999999997E-01, +1.0000000000000E+00,
+1.0000000000000E+00, +1.0000000000000E+00, +9.9999999999999E-01,
+9.9999999999999E-01
 2     00:00:00         0             2            38           126          0
+3.1247090742030E-26  +2.9286300262740E-26  +4.1981634940432E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             3            38           127          0
+5.0124789450571E-26                        +4.9930396810756E-13
+9.9999999999998E-01, +9.9999999999960E-01, +9.9999999999953E-01,
+9.9999999999946E-01, +9.9999999999965E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999994E-01, +9.9999999999998E-01,
+9.9999999999962E-01, +1.0000000000005E+00, +1.0000000000006E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999989E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999994E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             4            37           122          0
+5.5345870726670E-26                        +6.2549288652871E-13
+9.9999999999998E-01, +9.9999999999962E-01, +9.9999999999956E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999964E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             5            35           118          0
+4.6923353827549E-26                        +4.0255780203410E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
```

```
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             6            38          126          0
+3.1247090742030E-26                          +4.1981634940432E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             7            38          127          0
+5.0124789450571E-26                          +4.9930396810756E-13
+9.9999999999998E-01, +9.9999999999960E-01, +9.9999999999953E-01,
+9.9999999999946E-01, +9.9999999999965E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999994E-01, +9.9999999999998E-01,
+9.9999999999962E-01, +1.0000000000005E+00, +1.0000000000006E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999989E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999994E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             8            37          123          0
+3.4308826922419E-26                          +3.2724554800249E-13
+9.9999999999998E-01, +9.9999999999962E-01, +9.9999999999956E-01,
+9.9999999999948E-01, +9.9999999999966E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999964E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1             9            35          119          0
+6.6824555505262E-26                          +4.7097457160500E-13
+9.9999999999998E-01, +9.9999999999962E-01, +9.9999999999956E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999964E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
```

```
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            10           35           118          0
+6.0963114828689E-26                       +4.3727056907543E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999964E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            11           36           120          0
+5.3255367199067E-26                       +4.7140564707289E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            12           37           124          0
+2.9012006319926E-26  +2.2350844221041E-27  +3.9014494604562E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            13           37           124          0
+6.2098836161097E-26                       +4.2781604140423E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
```

```
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            14            38            126           0
+4.7389390548980E-26                           +4.8196706324669E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            15            37            120           0
+4.5780218133173E-26                           +4.9812354105641E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            16            38            127           0
+3.6733960938338E-26                           +3.3692719887678E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01
 2     00:00:00         1            17            35            119           0
+6.4893051219192E-26                           +5.1985858289360E-13
+9.9999999999998E-01, +9.9999999999963E-01, +9.9999999999957E-01,
+9.9999999999949E-01, +9.9999999999967E-01, +1.0000000000001E+00,
+1.0000000000001E+00, +9.9999999999995E-01, +9.9999999999998E-01,
+9.9999999999965E-01, +1.0000000000005E+00, +1.0000000000005E+00,
+1.0000000000001E+00, +1.0000000000000E+00, +9.9999999999990E-01,
```

```
+1.0000000000001E+00, +1.0000000000001E+00, +9.9999999999997E-01,
+1.0000000000000E+00, +1.0000000000000E+00, +1.0000000000000E+00,
+9.9999999999995E-01, +9.9999999999995E-01, +1.0000000000001E+00,
+1.0000000000000E+00, +1.0000000000001E+00, +1.0000000000000E+00,
+9.9999999999998E-01

The optimization routine failed to converge. This problem can sometimes be
mitigated by choosing more reasonable initial parameter values, setting more
conservative bounds, or configuring other optimization settings.


Optimization failed after 00:00:01.
Computing the Hessian and estimating standard errors …
Computed results after 00:00:07.

Problem Results Summary:
=================================================================================
==================================================================
GMM        Objective             Gradient              Hessian
Hessian        Clipped    Weighting Matrix     Covariance Matrix
Step        Value                 Norm             Min Eigenvalue        Max
Eigenvalue     Shares     Condition Number      Condition Number
----  --------------------  --------------------  --------------------
--------------------  -------  --------------------  --------------------
 2    +3.1247090742030E-26  +4.1981634940432E-13  -6.4693617698989E-05
+7.1284129245656E-05     0     +6.2384331654490E+02  +8.5572903008789E+19
=================================================================================
==================================================================

Cumulative Statistics:
============================================================================
Computation  Optimizer  Optimization   Objective   Fixed Point  Contraction
   Time      Converged   Iterations   Evaluations  Iterations   Evaluations
-----------  ---------  ------------  -----------  -----------  -----------
 00:00:27       No           2            28          3510         10852
============================================================================

Nonlinear Coefficient Estimates (Robust SEs in Parentheses):
=================================================================================
=================================================================================
=================================================================================
=================================================================================
===========================================
Sigma:             1                     prices                 char1
char2                  char3                  costsh1                costsh2
|  Sigma Squared:          1                     prices                 char1
char2                  char3                  costsh1                costsh2
-------  ----------------------  ----------------------  ----------------------
```

```
---------------------  ---------------------  ---------------------
---------------------  |  -------------  ---------------------
---------------------  ---------------------  ---------------------
---------------------  ---------------------  ---------------------
   1      +9.9999999999998E-01
|       1          +9.9999999999997E-01    +9.9999999999961E-01
+9.9999999999947E-01    +1.0000000000001E+00    +1.0000000000005E+00
+1.0000000000001E+00    +9.9999999999993E-01
         (+2.5646592826311E-03)
|                 (+5.1293185652622E-03)  (+1.3207545868449E-01)
(+1.5058832748724E-01)  (+8.9404092204642E-02)  (+9.3373440200829E-02)
(+1.7107045772261E-01)  (+2.6340767802889E-01)
                 |
prices    +9.9999999999963E-01    +9.9999999999957E-01
|      prices       +9.9999999999961E-01    +1.9999999999984E+00
+1.9999999999984E+00    +1.9999999999993E+00    +2.0000000000003E+00
+1.9999999999994E+00    +1.9999999999991E+00
         (+1.3184999792281E-01)  (+1.3991843509955E-01)
|                 (+1.3207545868449E-01)  (+5.3590031997448E-01)
(+3.8885552172465E-01)  (+4.3424917372389E-01)  (+2.1839917494496E-01)
(+5.1392529984977E-01)  (+4.8697033740639E-01)
                 |
 char1    +9.9999999999949E-01    +9.9999999999967E-01    +1.0000000000001E+00
|      char1        +9.9999999999947E-01    +1.9999999999984E+00
+2.9999999999985E+00    +2.9999999999993E+00    +3.0000000000004E+00
+2.9999999999994E+00    +2.9999999999992E+00
         (+1.4880232509789E-01)  (+8.0273493684577E-02)  (+9.4414289766014E-02)
|                 (+1.5058832748724E-01)  (+3.8885552172465E-01)
(+3.6269074507471E-01)  (+3.8084105778425E-01)  (+2.8790374275702E-01)
(+2.3897977261083E-01)  (+5.0303961256615E-01)
                 |
 char2    +1.0000000000001E+00    +9.9999999999995E-01    +9.9999999999998E-01
+9.9999999999965E-01
|      char2        +1.0000000000001E+00    +1.9999999999993E+00
+2.9999999999993E+00    +3.9999999999994E+00    +4.0000000000008E+00
+3.9999999999998E+00    +3.9999999999997E+00
         (+8.7941082407489E-02)  (+1.4071118102061E-01)  (+4.2451382594656E-02)
(+1.3039574930044E-01)
|                 (+8.9404092204642E-02)  (+4.3424917372389E-01)
(+3.8084105778425E-01)  (+7.3633613361793E-01)  (+2.4854032239524E-01)
(+4.8473258216525E-01)  (+4.4218389501217E-01)
                 |
 char3    +1.0000000000005E+00    +1.0000000000005E+00    +1.0000000000001E+00
+1.0000000000000E+00    +9.9999999999990E-01
|      char3        +1.0000000000005E+00    +2.0000000000003E+00
+3.0000000000004E+00    +4.0000000000008E+00    +5.0000000000021E+00
+5.0000000000012E+00    +5.0000000000011E+00
         (+9.5426665448296E-02)  (+1.1099111604828E-01)  (+1.6537543482697E-01)
```

```
(+1.3255442181497E-01)  (+7.5283353952592E-02)
|                    (+9.3373440200829E-02)  (+2.1839917494496E-01)
(+2.8790374275702E-01)  (+2.4854032239524E-01)  (+9.7494536173366E-01)
(+5.5271441003316E-01)  (+9.4982228556229E-01)
                 |
costsh1   +1.0000000000001E+00    +1.0000000000001E+00    +9.9999999999997E-01
+1.0000000000000E+00    +1.0000000000000E+00    +1.0000000000000E+00
|     costsh1       +1.0000000000001E+00    +1.9999999999994E+00
+2.9999999999994E+00    +3.9999999999998E+00    +5.0000000000012E+00
+6.0000000000003E+00    +6.0000000000002E+00
         (+1.7209551147330E-01)  (+1.6301684701002E-01)  (+2.0104049049979E-02)
(+3.1957990853817E-02)  (+1.7559969886542E-02)  (+1.3298527238005E-02)
|                    (+1.7107045772261E-01)  (+5.1392529984977E-01)
(+2.3897977261083E-01)  (+4.8473258216525E-01)  (+5.5271441003316E-01)
(+6.2362815988359E-01)  (+4.5250014100917E-01)
                 |
costsh2   +9.9999999999995E-01    +9.9999999999995E-01    +1.0000000000001E+00
+1.0000000000000E+00    +1.0000000000001E+00    +1.0000000000000E+00
+9.9999999999998E-01   |     costsh2       +9.9999999999993E-01
+1.9999999999991E+00    +2.9999999999992E+00    +3.9999999999997E+00
+5.0000000000011E+00    +6.0000000000002E+00    +7.0000000000000E+00
         (+2.6398477796815E-01)  (+2.4699729656152E-01)  (+1.2703129848035E-02)
(+1.6743589140495E-02)  (+2.4658442430366E-02)  (+1.1503218110289E-02)
(+2.3565435220409E-02)  |                    (+2.6340767802889E-01)
(+4.8697033740639E-01)  (+5.0303961256615E-01)  (+4.4218389501217E-01)
(+9.4982228556229E-01)  (+4.5250014100917E-01)  (+1.0570647756308E+00)
================================================================================
================================================================================
================================================================================
================================================================================
==========================================

Beta Estimates (Robust SEs in Parentheses):
================================================================================
==============================================================
        prices                  char1                   char2
char3                   costsh1                 costsh2
----------------------  ----------------------  ----------------------
----------------------  ----------------------  ----------------------
 -4.1707432781090E+00    -3.2391633241385E-02    -5.5085591957456E-01
-5.6303480514095E-01    -2.1522577797152E+00    -2.8679942066731E+00
(+1.4491747153455E-01)  (+1.7680611626277E-01)  (+1.8504052281318E-01)
(+1.5484102168733E-01)  (+2.4464903720885E-01)  (+3.8605561201587E-01)
================================================================================
==============================================================
Parameter Estimates:
[[-4.17074328]
 [-0.03239163]
```

```
 [-0.55085592]
 [-0.56303481]
 [-2.15225778]
 [-2.86799421]]
[[1. 0. 0. 0. 0. 0. 0.]
 [1. 1. 0. 0. 0. 0. 0.]
 [1. 1. 1. 0. 0. 0. 0.]
 [1. 1. 1. 1. 0. 0. 0.]
 [1. 1. 1. 1. 1. 0. 0.]
 [1. 1. 1. 1. 1. 1. 0.]
 [1. 1. 1. 1. 1. 1. 1.]]
```

1.e. Provide an explicit expression for the variance-covariance matrix of your estimates, and discuss how simulation error affects it.

[183]:
```python
results3_cov = pd.DataFrame(results3.parameter_covariances)
print(results3_cov)
```

```
           0         1         2         3         4         5         6  \
0   0.001625  0.006537  0.025125  0.065225  0.030401 -0.029274  0.031231
1   0.006537  4.293952  4.302209  0.314232  0.113451 -1.159126  1.101348
2   0.025125  4.302209  4.835561  1.956313  0.986260 -1.732861  1.941210
3   0.065225  0.314232  1.956313  5.469107  2.906676 -2.341999  2.428853
4   0.030401  0.113451  0.986260  2.906676  1.591627 -1.134491  1.223724
5  -0.029274 -1.159126 -1.732861 -2.341999 -1.134491  2.201772 -0.888219
6   0.031231  1.101348  1.941210  2.428853  1.223724 -0.888219  1.910208
7   0.057660  1.833390  3.264304  4.553202  2.302630 -2.288523  2.881109
8   0.000791  1.169556  1.126886  0.065212 -0.010042 -0.652926  0.148581
9   0.056135  2.397238  3.580147  3.990872  2.006543 -2.315535  2.476680
10 -0.048687 -1.922865 -2.368904 -1.759663 -0.743116  1.462130 -1.205421
11 -0.064725 -1.463398 -2.322486 -3.016845 -1.406388  1.774001 -1.765258
12 -0.044718 -0.675702 -1.792690 -4.408653 -2.222433  3.693224 -1.279845
13 -0.045929  0.589425 -0.681120 -4.395705 -2.290715  2.425964 -1.512774
14 -0.009695  0.528634  0.133831 -1.604920 -0.838303  1.364510 -0.141741
15 -0.044255  3.156325  1.913687 -3.133666 -1.750429 -1.008606 -1.638262
16 -0.040716  2.972820  1.846746 -2.777120 -1.557138 -1.097925 -1.504849
17  0.007546  0.272980  0.467612  0.665468  0.342117 -0.400281  0.352863
18  0.009163  0.037604  0.338548  1.043295  0.546932 -0.631044  0.402680
19  0.004235 -0.242256 -0.102985  0.351293  0.198142  0.089797  0.213553
20 -0.003252 -0.043419 -0.151378 -0.395354 -0.206164  0.281848 -0.135488
21 -0.038401 -2.446844 -3.626070 -5.294880 -2.655337  5.859947 -1.301978
22 -0.034888 -2.341269 -3.410529 -4.855273 -2.429463  5.474742 -1.175007
23 -0.004949  0.003678 -0.057309 -0.141531 -0.067825 -0.083358 -0.153364
24 -0.001956 -0.405537 -0.390187 -0.058574 -0.009977  0.285681 -0.025659
25 -0.005055  0.151606  0.089393 -0.053596 -0.029830 -0.338742 -0.185900
26 -0.003472  0.035177 -0.081011 -0.390399 -0.207578  0.199150 -0.148903
27 -0.001639 -0.159239 -0.235506 -0.383542 -0.193347  0.492592 -0.045827
28  0.013635  4.634001  4.934335  1.185445  0.594246 -1.429526  1.683661
29  0.072356  0.986000  2.844016  6.401705  3.414889 -3.155520  2.730571
```

30	0.060550	2.360282	4.205826	5.854666	2.993545	-3.145988	3.669605
31	-0.081578	-1.208150	-1.786713	-1.761148	-0.659836	0.246470	-1.713496
32	-0.065609	4.370627	2.543022	-4.674718	-2.599999	-1.282539	-2.441129
33	-0.060148	-3.549730	-5.377535	-8.041879	-4.043283	8.610663	-2.081699

	7	8	9	…	24	25	26	27 \
0	0.057660	0.000791	0.056135	…	-0.001956	-0.005055	-0.003472	-0.001639
1	1.833390	1.169556	2.397238	…	-0.405537	0.151606	0.035177	-0.159239
2	3.264304	1.126886	3.580147	…	-0.390187	0.089393	-0.081011	-0.235506
3	4.553202	0.065212	3.990872	…	-0.058574	-0.053596	-0.390399	-0.383542
4	2.302630	-0.010042	2.006543	…	-0.009977	-0.029830	-0.207578	-0.193347
5	-2.288523	-0.652926	-2.315535	…	0.285681	-0.338742	0.199150	0.492592
6	2.881109	0.148581	2.476680	…	-0.025659	-0.185900	-0.148903	-0.045827
7	4.890510	0.464468	4.385462	…	-0.163979	-0.073260	-0.308801	-0.317627
8	0.464468	0.445124	0.661046	…	-0.168965	0.165019	-0.006892	-0.153018
9	4.385462	0.661046	4.199754	…	-0.255128	0.000437	-0.254887	-0.334154
10	-2.265049	-0.585183	-2.485594	…	0.260699	0.014439	0.078719	0.161280
11	-3.207016	-0.417269	-3.176785	…	0.204011	0.097503	0.170447	0.191032
12	-3.659117	-0.796811	-3.473419	…	0.374819	-0.613231	0.402446	0.891077
13	-3.369697	-0.100447	-2.827373	…	0.085807	-0.208868	0.371262	0.532934
14	-0.965147	-0.160943	-0.810933	…	0.095682	-0.306758	0.174394	0.390548
15	-1.845625	1.447917	-0.719684	…	-0.557595	0.779719	0.187301	-0.429655
16	-1.607376	1.391905	-0.560559	…	-0.538172	0.765104	0.158660	-0.444743
17	0.667629	0.093925	0.620327	…	-0.037033	0.021758	-0.048606	-0.072839
18	0.873108	0.078986	0.753338	…	-0.036547	0.069206	-0.089207	-0.142420
19	0.251349	-0.132236	0.132957	…	0.053746	-0.080715	-0.020547	0.042947
20	-0.330159	-0.051181	-0.296815	…	0.022987	-0.042000	0.035314	0.067555
21	-4.747676	-1.797835	-4.874613	…	0.772879	-1.261742	0.531638	1.517863
22	-4.375954	-1.703983	-4.518373	…	0.731358	-1.190591	0.490366	1.421377
23	-0.159202	0.056768	-0.127487	…	-0.022106	0.072348	-0.000615	-0.048964
24	-0.163979	-0.168965	-0.255128	…	0.069246	-0.068253	0.004862	0.066288
25	-0.073260	0.165019	0.000437	…	-0.068253	0.150186	-0.015139	-0.123607
26	-0.308801	-0.006892	-0.254887	…	0.004862	-0.015139	0.032684	0.043616
27	-0.317627	-0.153018	-0.334154	…	0.066288	-0.123607	0.043616	0.137166
28	2.735584	1.200271	3.171575	…	-0.404945	0.105406	-0.020954	-0.184703
29	5.390968	0.347227	4.901551	…	-0.175681	0.092422	-0.465018	-0.577466
30	6.324197	0.687308	5.583490	…	-0.227217	0.042736	-0.427522	-0.518958
31	-2.322752	0.029236	-2.402519	…	0.038799	0.608749	-0.005710	-0.327040
32	-2.848033	2.015741	-1.241121	…	-0.775266	1.096274	0.282225	-0.582249
33	-7.204460	-2.592059	-7.332985	…	1.115226	-1.808972	0.795536	2.212190

	28	29	30	31	32	33
0	0.013635	0.072356	0.060550	-0.081578	-0.065609	-0.060148
1	4.634001	0.986000	2.360282	-1.208150	4.370627	-3.549730
2	4.934335	2.844016	4.205826	-1.786713	2.543022	-5.377535
3	1.185445	6.401705	5.854666	-1.761148	-4.674718	-8.041879
4	0.594246	3.414889	2.993545	-0.659836	-2.599999	-4.043283
5	-1.429526	-3.155520	-3.145988	0.246470	-1.282539	8.610663

```
6   1.683661   2.730571   3.669605 -1.713496  -2.441129  -2.081699
7   2.735584   5.390968   6.324197 -2.322752  -2.848033  -7.204460
8   1.200271   0.347227   0.687308  0.029236   2.015741  -2.592059
9   3.171575   4.901551   5.583490 -2.402519  -1.241121  -7.332985
10 -2.182505  -2.237446  -2.554063  2.733128  -0.572666   3.847288
11 -1.911046  -3.573607  -3.733153  3.225490   1.415680   4.727002
12 -1.125759  -5.767854  -5.207991 -0.556063  -0.808843  15.200772
13  0.045620  -5.247973  -4.606672  0.163070   2.931613   9.586299
14  0.437732  -2.094585  -1.559621 -1.042414  -0.087876   6.267674
15  2.590839  -2.594269  -1.935009  2.310893  10.395048  -5.739507
16  2.457645  -2.212195  -1.619857  2.269410   9.827562  -6.065444
17  0.391004   0.823860   0.885637 -0.200549  -0.202089  -1.430222
18  0.185777   1.281073   1.222963  0.052191  -0.456763  -2.543752
19 -0.166722   0.308779   0.287236 -0.236494  -1.044304   0.543970
20 -0.094833  -0.503648  -0.472981 -0.064596   0.045866   1.169754
21 -2.957286  -7.588814  -7.182399 -2.382362  -5.556556  25.164128
22 -2.802350  -6.999528  -6.637352 -2.258583  -5.395718  23.534621
23 -0.034504  -0.097785  -0.124754  0.412807   0.544900  -0.638996
24 -0.404945  -0.175681  -0.227217  0.038799  -0.775266   1.115226
25  0.105406   0.092422   0.042736  0.608749   1.096274  -1.808972
26 -0.020954  -0.465018  -0.427522 -0.005710   0.282225   0.795536
27 -0.184703  -0.577466  -0.518958 -0.327040  -0.582249   2.212190
28  5.187265   1.997688   3.550095 -1.498578   3.529799  -4.355901
29  1.997688   7.721319   7.040875 -1.608728  -3.968287 -11.409915
30  3.550095   7.040875   8.457279 -1.880133  -3.047115 -10.819213
31 -1.498578  -1.608728  -1.880133  5.922008   3.357589  -3.289193
32  3.529799  -3.968287  -3.047115  3.357589  14.783728  -7.613216
33 -4.355901 -11.409915 -10.819213 -3.289193  -7.613216  36.812617

[34 rows x 34 columns]
```

2. Compare the cross and own-price elasticities for market 10 for the RC logit and pure logit model.

[185]:
```python
elasticities2 = results3.compute_elasticities()
elasticities2_market_10 = pd.DataFrame(elasticities2[data.market_ids == 10])
elasticities2_market_10 = elasticities2_market_10.dropna(axis=1, how='all')

print(elasticities_market_10)
```

```
Computing elasticities with respect to prices …
Finished after 00:00:00.

          0         1         2         3         4         5         6  \
0 -3.791656  0.591234  0.076763  0.038112  0.245743  0.209206  0.037551
1  0.065788 -3.422239  0.076763  0.038112  0.245743  0.209206  0.037551
2  0.065788  0.591234 -5.220499  0.038112  0.245743  0.209206  0.037551
3  0.065788  0.591234  0.076763 -3.584304  0.245743  0.209206  0.037551
4  0.065788  0.591234  0.076763  0.038112 -4.433488  0.209206  0.037551
```

5	0.065788	0.591234	0.076763	0.038112	0.245743	-3.041279	0.037551
6	0.065788	0.591234	0.076763	0.038112	0.245743	0.209206	-5.773704
7	0.065788	0.591234	0.076763	0.038112	0.245743	0.209206	0.037551
8	0.065788	0.591234	0.076763	0.038112	0.245743	0.209206	0.037551
9	0.065788	0.591234	0.076763	0.038112	0.245743	0.209206	0.037551
10	0.065788	0.591234	0.076763	0.038112	0.245743	0.209206	0.037551

	7	8	9	10
0	0.011711	0.377774	0.024602	0.009697
1	0.011711	0.377774	0.024602	0.009697
2	0.011711	0.377774	0.024602	0.009697
3	0.011711	0.377774	0.024602	0.009697
4	0.011711	0.377774	0.024602	0.009697
5	0.011711	0.377774	0.024602	0.009697
6	0.011711	0.377774	0.024602	0.009697
7	-5.278629	0.377774	0.024602	0.009697
8	0.011711	-3.271287	0.024602	0.009697
9	0.011711	0.377774	-6.355967	0.009697
10	0.011711	0.377774	0.024602	-4.709363

3. We are assuming here that demand in all markets is identical. With data on the distribution of income within each market, how could you let the distribution of α_i (the random variable coeffcient on price) vary systematically across markets?

Uma possibilidade seria estimar uma variável γ que captura como a renda afeta a distribuição de α_i dos consumidores em cada mercado, ou seja, a sensibilidade dos consumidores ao preço.

$$\alpha_i = \alpha + \gamma Renda + \varepsilon$$