IO Problem Set 1 (BLP)

Chris Ackerman*

November 4, 2021

Problem 1

Estimate the Model using OLS, with price and promotion as characteristics

```
  #1.1: Estimate using OLS with price and promotion as product characteristics.
res_log1 = smf.ols('Y ~ prices + prom_', data=otc_dataDf).fit()
```

Dep. Variable:	Y	**R-squared:**	0.158
Model:	OLS	**Adj. R-squared:**	0.158
Method:	Least Squares	**F-statistic:**	3610.
Date:	Thu, 04 Nov 2021	**Prob (F-statistic):**	0.00
Time:	05:19:39	**Log-Likelihood:**	-56307.
No. Observations:	38544	**AIC:**	1.126e+05
Df Residuals:	38541	**BIC:**	1.126e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-7.6267	0.014	-532.839	0.000	-7.655	-7.599
prices	-0.2496	0.003	-84.768	0.000	-0.255	-0.244
prom_	-0.0311	0.019	-1.653	0.098	-0.068	0.006

Omnibus:	1648.591	**Durbin-Watson:**	0.434
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1461.418
Skew:	-0.415	**Prob(JB):**	0.00
Kurtosis:	2.529	**Cond. No.**	17.7

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

*I worked on this problem set with Luna Shen, David Kerns, and Benedikt Graf

Estimate the Model using OLS, with price and promotion as characteristics, and brand dummies

```
#1.2: Estimate using OLS with price and promotion as product characteristics and brand dummies.
res_log2 = smf.ols('Y ~ prices + prom_ + C(brand)', data=otc_dataDf).fit()
```

Dep. Variable:	Y	**R-squared:**	0.654
Model:	OLS	**Adj. R-squared:**	0.654
Method:	Least Squares	**F-statistic:**	6081.
Date:	Thu, 04 Nov 2021	**Prob (F-statistic):**	0.00
Time:	05:20:51	**Log-Likelihood:**	-39138.
No. Observations:	38544	**AIC:**	7.830e+04
Df Residuals:	38531	**BIC:**	7.841e+04
Df Model:	12		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-6.0745	0.036	-167.065	0.000	-6.146	-6.003
C(brand)[T.2]	-0.0048	0.022	-0.218	0.828	-0.048	0.039
C(brand)[T.3]	-0.4578	0.040	-11.502	0.000	-0.536	-0.380
C(brand)[T.4]	-0.4303	0.017	-25.951	0.000	-0.463	-0.398
C(brand)[T.5]	-0.8868	0.024	-37.403	0.000	-0.933	-0.840
C(brand)[T.6]	-1.3850	0.051	-27.408	0.000	-1.484	-1.286
C(brand)[T.7]	-1.6527	0.018	-94.139	0.000	-1.687	-1.618
C(brand)[T.8]	-2.2856	0.016	-141.034	0.000	-2.317	-2.254
C(brand)[T.9]	-1.9340	0.017	-111.950	0.000	-1.968	-1.900
C(brand)[T.10]	-1.8983	0.022	-87.306	0.000	-1.941	-1.856
C(brand)[T.11]	-2.1754	0.019	-113.355	0.000	-2.213	-2.138
prices	-0.3412	0.010	-33.864	0.000	-0.361	-0.321
prom_	0.3294	0.013	26.122	0.000	0.305	0.354

Omnibus:	2773.498	**Durbin-Watson:**	1.024
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	3867.305
Skew:	-0.618	**Prob(JB):**	0.00
Kurtosis:	3.938	**Cond. No.**	112.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Estimate the Model using OLS, with price and promotion as characteristics, and store-brand dummies

```
#1.3. Using OLS with price and promotion as product characteristics and store-brand
#(the interaction of brand and store) dummies.

res_log3 = smf.ols('Y ~ prices + prom_ + C(brand)*C(store)', data=otc_dataDf).fit()
```

Dep. Variable:	Y	**R-squared:**	0.722
Model:	OLS	**Adj. R-squared:**	0.716
Method:	Least Squares	**F-statistic:**	121.9
Date:	Thu, 04 Nov 2021	**Prob (F-statistic):**	0.00
Time:	05:21:32	**Log-Likelihood:**	-34946.
No. Observations:	38544	**AIC:**	7.150e+04
Df Residuals:	37739	**BIC:**	7.839e+04
Df Model:	804		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-6.1994	0.093	-66.365	0.000	-6.382	-6.016
prices	-0.3302	0.010	-34.349	0.000	-0.349	-0.311
prom_	0.3288	0.011	28.619	0.000	0.306	0.351

Omnibus:	4044.934	**Durbin-Watson:**	1.268
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	7222.052
Skew:	-0.721	**Prob(JB):**	0.00
Kurtosis:	4.555	**Cond. No.**	4.08e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.08e+03. This might indicate that there are strong multicollinearity or other numerical problems. Dummies omitted.

Estimate the models from parts 1–3 using wholesale cost as an instrument

```
  # OLS with price and promotion as product characteristics, using wholesale cost as instrument
wholeSale_IV1 = iv.IV2SLS.from_formula('Y ~ 1 + [prices ~ cost_] + prom_ ', data=otc_dataDf).fit()
```

Dep. Variable:	Y	**R-squared:**	0.1531
Estimator:	IV-2SLS	**Adj. R-squared:**	0.1531
No. Observations:	38544	**F-statistic:**	5255.4
Date:	Thu, Nov 04 2021	**P-value (F-stat)**	0.0000
Time:	05:25:20	**Distribution:**	chi2(2)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
Intercept	-7.8181	0.0150	-520.30	0.0000	-7.8476	-7.7887
prom_	-0.0068	0.0170	-0.4034	0.6866	-0.0401	0.0264
prices	-0.2066	0.0029	-71.884	0.0000	-0.2122	-0.2009

Endogenous: prices
Instruments: cost_
Robust Covariance (Heteroskedastic)
Debiased: False

```
    # OLS with price and promotion as product characteristics and brand dummies
# using wholesale cost as instrument

wholeSale_IV2 = iv.IV2SLS.from_formula('Y ~ 1 + [prices ~ cost_] + prom_ + C(brand)', data=otc_dataDf).:
```

Dep. Variable:	Y	**R-squared:**	0.6446
Estimator:	IV-2SLS	**Adj. R-squared:**	0.6445
No. Observations:	38544	**F-statistic:**	9.69e+04
Date:	Thu, Nov 04 2021	**P-value (F-stat)**	0.0000
Time:	05:26:08	**Distribution:**	chi2(12)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
Intercept	-7.2171	0.0652	-110.69	0.0000	-7.3449	-7.0893
C(brand)[T.2]	-0.5161	0.0309	-16.703	0.0000	-0.5766	-0.4555
C(brand)[T.3]	-1.6627	0.0698	-23.833	0.0000	-1.7994	-1.5260
C(brand)[T.4]	-0.2833	0.0147	-19.314	0.0000	-0.3121	-0.2546
C(brand)[T.5]	-1.4660	0.0358	-40.904	0.0000	-1.5363	-1.3958
C(brand)[T.6]	-2.9699	0.0914	-32.484	0.0000	-3.1491	-2.7907
C(brand)[T.7]	-1.4158	0.0181	-78.185	0.0000	-1.4513	-1.3803
C(brand)[T.8]	-2.3613	0.0137	-172.35	0.0000	-2.3882	-2.3344
C(brand)[T.9]	-2.1365	0.0169	-126.55	0.0000	-2.1696	-2.1034
C(brand)[T.10]	-1.4120	0.0321	-44.036	0.0000	-1.4749	-1.3492
C(brand)[T.11]	-2.5260	0.0283	-89.396	0.0000	-2.5814	-2.4707
prom_	0.4307	0.0145	29.801	0.0000	0.4024	0.4590
prices	-0.0081	0.0189	-0.4287	0.6682	-0.0452	0.0290

Endogenous: prices
Instruments: cost_
Robust Covariance (Heteroskedastic)
Debiased: False

```
# OLS with price and promotion as product characteristics and brand dummies
# using wholesale cost as instrument

wholeSale_IV3 = iv.IV2SLS.from_formula('Y ~ 1 + [prices ~ cost_] + prom_ + C(brand)*C(store)', data=otc_
```

Dep. Variable:	Y	**R-squared:**	0.7150
Estimator:	IV-2SLS	**Adj. R-squared:**	0.7090
No. Observations:	38544	**F-statistic:**	1.766e+05
Date:	Thu, Nov 04 2021	**P-value (F-stat)**	0.0000
Time:	05:27:37	**Distribution:**	chi2(804)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
Intercept	-7.2138	0.0775	-93.106	0.0000	-7.3657	-7.0620
prom_	0.4193	0.0132	31.774	0.0000	0.3934	0.4451
prices	-0.0346	0.0178	-1.9442	0.0519	-0.0695	0.0003

Table 1: IV-2SLS Estimation Summary, dummies suppressed

Endogenous: prices
Instruments: cost_
Robust Covariance (Heteroskedastic)
Debiased: False

Estimate the models from parts 1–3 using the Hausman instrument

Dep. Variable:	Y	**R-squared:**	0.1578
Estimator:	IV-2SLS	**Adj. R-squared:**	0.1577
No. Observations:	38544	**F-statistic:**	9465.0
Date:	Thu, Nov 04 2021	**P-value (F-stat)**	0.0000
Time:	05:40:33	**Distribution:**	chi2(2)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
Intercept	-7.6143	0.0135	-565.64	0.0000	-7.6407	-7.5879
prom_	-0.0327	0.0170	-1.9257	0.0541	-0.0660	0.0006
prices	-0.2524	0.0026	-97.062	0.0000	-0.2574	-0.2473

Endogenous: prices
Instruments: pricestore1, pricestore2, pricestore3, pricestore4, pricestore5, pricestore6, pricestore7, pricestore8, pricestore9, pricestore10, pricestore11, pricestore12, pricestore13, pricestore14, pricestore15, pricestore16, pricestore17, pricestore18, pricestore19, pricestore20, pricestore21, pricestore22, pricestore23, pricestore24, pricestore25, pricestore26, pricestore27, pricestore28, pricestore29, pricestore30
Robust Covariance (Heteroskedastic)
Debiased: False

Mean own-price elasticities from the estimates in models 1–3

These results make sense. As a rule of thumb, the own-price elasticities should be $\in (-2, -5)$. The IV estimates using the Hausman instrument are approximately in this range. The OLS estimates are not, indicating that endogeneity is a practical concern in this setting. The estimates with wholesale cost as an instrument are also "too small", indicating that wholesale cost may not be a viable instrument in this context.

Dep. Variable:	Y	**R-squared:**	0.6511
Estimator:	IV-2SLS	**Adj. R-squared:**	0.6510
No. Observations:	38544	**F-statistic:**	9.529e+04
Date:	Thu, Nov 04 2021	**P-value (F-stat)**	0.0000
Time:	05:41:01	**Distribution:**	chi2(12)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
Intercept	-5.4061	0.0539	-100.33	0.0000	-5.5117	-5.3005
C(product_ids)[T.2]	0.2942	0.0259	11.370	0.0000	0.2435	0.3449
C(product_ids)[T.3]	0.2471	0.0574	4.3071	0.0000	0.1347	0.3595
C(product_ids)[T.4]	-0.5162	0.0140	-36.812	0.0000	-0.5437	-0.4888
C(product_ids)[T.5]	-0.5479	0.0298	-18.367	0.0000	-0.6064	-0.4894
C(product_ids)[T.6]	-0.4578	0.0751	-6.0967	0.0000	-0.6050	-0.3107
C(product_ids)[T.7]	-1.7913	0.0169	-105.81	0.0000	-1.8245	-1.7581
C(product_ids)[T.8]	-2.2413	0.0143	-156.43	0.0000	-2.2694	-2.2132
C(product_ids)[T.9]	-1.8155	0.0156	-116.73	0.0000	-1.8460	-1.7850
C(product_ids)[T.10]	-2.1827	0.0280	-77.963	0.0000	-2.2376	-2.1279
C(product_ids)[T.11]	-1.9703	0.0231	-85.436	0.0000	-2.0155	-1.9251
prom_	0.2701	0.0143	18.947	0.0000	0.2421	0.2980
prices	-0.5361	0.0155	-34.507	0.0000	-0.5665	-0.5056

Endogenous: prices
Instruments: pricestore1, pricestore2, pricestore3, pricestore4, pricestore5, pricestore6, pricestore7, pricestore8, pricestore9, pricestore10, pricestore11, pricestore12, pricestore13, pricestore14, pricestore15, pricestore16, pricestore17, pricestore18, pricestore19, pricestore20, pricestore21, pricestore22, pricestore23, pricestore24, pricestore25, pricestore26, pricestore27, pricestore28, pricestore29, pricestore30
Robust Covariance (Heteroskedastic)
Debiased: False

Dep. Variable:	Y	**R-squared:**	0.7183
Estimator:	IV-2SLS	**Adj. R-squared:**	0.7123
No. Observations:	38544	**F-statistic:**	1.711e+05
Date:	Thu, Nov 04 2021	**P-value (F-stat)**	0.0000
Time:	05:42:30	**Distribution:**	chi2(804)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
Intercept	-5.4606	0.0662	-82.482	0.0000	-5.5903	-5.3308
prom_	0.2630	0.0130	20.297	0.0000	0.2376	0.2884
prices	-0.5454	0.0138	-39.560	0.0000	-0.5725	-0.5184

Endogenous: prices
Instruments: pricestore1, pricestore2, pricestore3, pricestore4, pricestore5, pricestore6, pricestore7, pricestore8, pricestore9, pricestore10, pricestore11, pricestore12, pricestore13, pricestore14, pricestore15, pricestore16, pricestore17, pricestore18, pricestore19, pricestore20, pricestore21, pricestore22, pricestore23, pricestore24, pricestore25, pricestore26, pricestore27, pricestore28, pricestore29, pricestore30
Robust Covariance (Heteroskedastic) Dummies omitted.
Debiased: False

brand	OLS1	OLS2	OLS3	IV4.1	IV4.2	IV4.3	IV5.1	IV5.2	IV5.3
1	-0.852929	-1.166183	-1.128481	-0.706043	-0.027733	-0.118239	-0.862496	-1.832176	-1.864240
2	-1.232714	-1.685451	-1.630960	-1.020423	-0.040082	-0.170887	-1.246541	-2.647991	-2.694332
3	-1.750607	-2.393550	-2.316167	-1.449128	-0.056921	-0.242681	-1.770243	-3.760477	-3.826287
4	-0.739114	-1.010568	-0.977896	-0.611828	-0.024032	-0.102461	-0.747405	-1.587691	-1.615476
5	-1.283685	-1.755141	-1.698398	-1.062616	-0.041739	-0.177953	-1.298083	-2.757481	-2.805738
6	-2.036190	-2.784018	-2.694011	-1.685529	-0.066207	-0.282271	-2.059029	-4.373936	-4.450483
7	-0.666923	-0.911862	-0.882382	-0.552069	-0.021685	-0.092453	-0.674403	-1.432616	-1.457688
8	-0.900107	-1.230688	-1.190900	-0.745096	-0.029267	-0.124779	-0.910203	-1.933518	-1.967356
9	-0.989782	-1.353297	-1.309545	-0.819327	-0.032183	-0.137210	-1.000884	-2.126149	-2.163357
10	-0.481135	-0.657841	-0.636573	-0.398277	-0.015644	-0.066698	-0.486532	-1.033526	-1.051613
11	-1.109697	-1.517254	-1.468201	-0.918591	-0.036082	-0.153834	-1.122144	-2.383739	-2.425456

Problem 2

Our results for this section don't make sense, and there are a few indications that we may have errors in our code. Our GMM function reached the maximum number of iterations. It's not clear if we set the tolerance level too low or if it simply failed to converge; if it didn't converge then the estimation procedure is incorrect and our results *shouldn't* make sense.

Estimate the parameter values using BLP

α	β	σ_{ib}	σ_I	σ_I^2
1.67671625	$\begin{bmatrix} 40.5539681 \\ -37.20820459 \\ 22.10159763 \\ -9.87790967 \\ -18.56618279 \\ 17.5039916 \\ -32.63598696 \\ 34.20049778 \\ -12.49605257 \\ 0.95162148 \\ -53.64393846 \end{bmatrix}$	$\begin{bmatrix} 0.35372042 \\ 0.4527695 \\ 0.2383683 \end{bmatrix}$	0.005161	-0.121263

Note that we have one more element of β than we should. When we constructed the dummies, we neglected to exclude one dummy variable, but we were unable to re-run our code before submitting our results. Given more time, we would re-run this step, and we suspect that the coefficients would then make more sense.

What are the elasticities for store 9 in week 10?

We are going to use the approximation

$$\eta_j^k \approx \frac{p_k}{s_j} \sum_i (\alpha + \sigma_I I_i)(-s_{ij}s_{ik} + \mathbb{1}_{k=j}s_{ik}).$$

For the purposes of this question i is a singleton and there is no joint ownership.

	0	1	2	3	4	5	6	7	8
0	-39.302792	-0.019898	-0.006467	-0.015919	-0.006964	-0.000497	-0.011939	-0.004975	-0.001492
1	-0.029454	-58.177689	-0.009573	-0.023563	-0.010309	-0.000736	-0.017672	-0.007364	-0.002209
2	-0.038587	-0.038587	-76.190310	-0.030869	-0.013505	-0.000965	-0.023152	-0.009647	-0.002894
3	-0.017116	-0.017116	-0.005563	-33.804145	-0.005991	-0.000428	-0.010270	-0.004279	-0.001284
4	-0.031994	-0.031994	-0.010398	-0.025595	-63.174270	-0.000800	-0.019197	-0.007999	-0.002400
5	-0.050743	-0.050743	-0.016492	-0.040595	-0.017760	-100.178617	-0.030446	-0.012686	-0.003806
6	-0.016390	-0.016390	-0.005327	-0.013112	-0.005737	-0.000410	-32.367476	-0.004098	-0.001229
7	-0.020201	-0.020201	-0.006565	-0.016160	-0.007070	-0.000505	-0.012120	-39.884948	-0.001515
8	-0.024011	-0.024011	-0.007804	-0.019209	-0.008404	-0.000600	-0.014406	-0.006003	-47.403955
9	-0.010221	-0.010221	-0.003322	-0.008177	-0.003577	-0.000256	-0.006133	-0.002555	-0.000767
10	-0.027156	-0.027156	-0.008826	-0.021725	-0.009505	-0.000679	-0.016293	-0.006789	-0.002037

These elasticities are not believable. The important/theoretical difference from the logit elasticities is that elasticities are no longer simply a function of shares, and are now a function of product characteristics as well (so for instance consumers are more likely to switch from one branded product to another than from a branded product to a generic, even if the branded product and generic have similar market shares). The striking problems are the massive magnitudes for own-price elasticities and the uniform negativity of the cross-price elasticities. This is most likely due to an error in our parameter estimates from the previous section.

Back out the marginal costs for store 9 in week 10. How are they different from wholesale costs?

We have code that does this (for part 3) but we were unable to run it before turning in the HW. It should be almost exactly the same procedure as in part 3, but we need to use results from the random coefficients model, and we are running into bugs in the code when we do that.

Problem 3

Predict the post-merger prices using the logit model, but only for store 9 in week 10

	store	week	brand	prices	new_prices	price_change
33	9	10	1	3.29	3.235631	-0.054369
34	9	10	2	4.87	4.815631	-0.054369
35	9	10	3	6.38	6.325631	-0.054369
36	9	10	4	2.83	2.797369	-0.032631
37	9	10	5	5.29	5.257369	-0.032631
38	9	10	6	8.39	8.357369	-0.032631
39	9	10	7	2.71	2.694294	-0.015706
40	9	10	8	3.34	3.324294	-0.015706
41	9	10	9	3.97	3.954294	-0.015706
42	9	10	10	1.69	1.671601	-0.018399
43	9	10	11	4.49	4.471601	-0.018399

How to predict the change in prices after the merger using the random coefficients model?

The procedure for predicting the effects of a merger using estimates from the random coefficients model is exactly the same. When we estimate the logit model, we end up with unrealistic elasticities, and hence unrealistic substitution patterns. The random coefficients model gives us a more reasonable matrix of elasticities, but we use it in the same way. When we want to assess the change of moving from two single-product firms to a multi-product firm, we switch from looking at firms that take FOCs with respect to a single price to a single firm that takes FOCs with respect to two prices (the price of each good that it produces). What does this mean? With a more realistic estimate of substitution patterns, we have a better idea of how consumers will respond to changes in prices. Pre-merger, a firm would be "unwilling" to raise its price because it would lose customers. Now, however, the merged firm *may* recognize that increasing the price of one of it's products will cause consumers to switch to its other product, so its profit maximizing price may be higher in the merged case than in the pre-merger case. The random coefficients model is giving us a better insight into these actual substitution patterns.