Flexible demand systems in Stata

Chris Cheng, Ph.D.
StataCorp LLC

College Station, TX, USA
March 5th, 2024

Overview

- Motivation
- Data and sample
- Demand systems
  - ✓ Almost ideal demand system
  - ✓ Quadratic almost ideal demand system
- Postestimation
  - ✓ Expenditure elasticity
  - ✓ Price sensitivities
  - ✓ Presenting results
- Program your own demand systems

Motivation

- Often we need to assess the demand for goods and products, as well as understand the relationships involved.
- It is essential to gauge demand elasticity so that businesses and policymakers can ascertain the extent to which consumers react to fluctuations in price and income levels.

Motivation

- In Stata 18, we introduced a new command, called `demandsys` for eight flexible demand systems.
- You can choose from the Cobb–Douglas demand system, almost ideal demand system (AIDS), generalized AIDS, and more.
- Compute expenditure and price elasticities to evaluate sensitivity to expenditure and price changes.

demandsys syntax

- Here is the general syntax of the `demandsys` command

`demandsys` *model varlist_s*, `prices(`*varlist_p*`)` `expenditure(`*varname*`)`

The demandsys command and its postestimation

- demandsys models
  - ✓ cdouglas – Cobb–Douglas demand system
  - ✓ les – linear expenditure system
  - ✓ translog – basic translog demand system
  - ✓ gtranslog – generalized translog demand system
  - ✓ aids – almost ideal demand system (AIDS)*
  - ✓ gaids – generalized AIDS
  - ✓ quaids – quadratic AIDS*
  - ✓ gquaids – generalized quadratic AIDS
- Postestimation
  - ✓ estat elasticities – price and expenditure elasticities
  - ✓ estat parameters – estimated parameter vectors and matrices

Data and sample

- Nationwide Food Consumption Survey
  - **4,160** households
  - **Five** food categories:
    1. dairy products;
    2. proteins, including meats and fish;
    3. fruits and vegetables;
    4. flour-based products, including breads, pastas, and cereals;
    5. miscellaneous category
  - Budget/expenditure share, unit price, and total expenditure
  - Two socio-demographic factors
    1. Number of adults
    2. Number of kids

Data and sample

- Let's load up the data

```
. webuse food_consumption
(Food consumption)

. describe w* p* exp n_adults n_kids

Variable      Storage   Display    Value
    name         type    format    label      Variable label
---------------------------------------------------------------------------------------------------
w_dairy         float   %10.6f                Expenditure share on dairy
w_proteins      float   %10.6f                Expenditure share on meats and proteins
w_fruitveg      float   %10.6f                Expenditure share on fruits and vegetables
w_flours        float   %10.6f                Expenditure share on flours, breads, pasta, and cereals
w_misc          float   %10.6f                Expenditure share on misc. food items
p_dairy         float   %10.6f                Price of dairy
p_proteins      float   %10.6f                Price of meats and proteins
p_fruitveg      float   %10.6f                Price of fruits and vegetables
p_flours        float   %10.6f                Price of flours, breads, pasta, and cereals
p_misc          float   %10.6f                Price of misc. food items
expfd           float   %10.6f                Total expenditure on all food categories
n_adults        byte    %8.0g                 # adults in household
n_kids          byte    %8.0g                 # kids in household
```

Data and sample – budget shares

- Budget shares add up to 1

```
. egen w_total = rowtotal(w*)
. format %1.0f w_total
. list w* in 1/5, abb(10)
```

	w_dairy	w_proteins	w_fruitveg	w_flours	w_misc	w_total
1.	0.184035	0.377504	0.193469	0.158781	0.086212	1
2.	0.262640	0.221351	0.178916	0.161808	0.175284	1
3.	0.112694	0.398640	0.241256	0.034165	0.213245	1
4.	0.150052	0.353398	0.215074	0.160400	0.121076	1
5.	0.163283	0.240806	0.276587	0.152208	0.167116	1

- demandsys checks this and errors out if incorrect

Data and sample – prices

- Unit price

```
. list p* in 1/5, abb(10)
```

	p_dairy	p_proteins	p_fruitveg	p_flours	p_misc
1.	0.243285	1.487986	0.429723	1.294675	1.856250
2.	0.560245	1.319658	0.675812	1.199858	1.045610
3.	0.268002	1.146716	0.410808	1.098958	1.292444
4.	0.334101	1.534831	0.610078	1.371681	0.773980
5.	0.407801	1.570370	0.361947	2.504673	0.874443

- demandsys verifies if all prices are positive

Data and sample – expenditure

- Total expenditure

```
. list exp in 1/5

     +-----------+
     |     expfd |
     |-----------|
  1. | 68.900002 |
  2. | 104.630005|
  3. | 61.760002 |
  4. | 57.980000 |
  5. | 70.430000 |
     +-----------+
```

- `demandsys` also verifies if expenditure is positive

Data and sample – demographics

- Household demographics

```
. list n* in 1/5

     ┌──────────────────┐
     │ n_adults  n_kids │
     ├──────────────────┤
  1. │        2       3 │
  2. │        2       3 │
  3. │        1       5 │
  4. │        1       0 │
  5. │        2       2 │
     └──────────────────┘
```

Data and sample – descriptive statistics

```
. qui dtable w* p* exp n_*, sample(,place(seplabels)) sformat("(N=%s)" frequency)
. collect style cell var[p_dairy expfd n_adults], border(top)
. collect preview
```

	Summary (N=4,160)
Expenditure share on dairy	0.151 (0.087)
Expenditure share on meats and proteins	0.399 (0.144)
Expenditure share on fruits and vegetables	0.241 (0.108)
Expenditure share on flours, breads, pasta, and cereals	0.103 (0.060)
Expenditure share on misc. food items	0.107 (0.073)
Price of dairy	0.439 (0.424)
Price of meats and proteins	1.729 (0.696)
Price of fruits and vegetables	0.547 (0.195)
Price of flours, breads, pasta, and cereals	1.494 (0.682)
Price of misc. food items	1.471 (0.859)
Total expenditure on all food categories	47.996 (29.722)
# adults in household	1.933 (0.731)
# kids in household	0.873 (1.184)

Two demand systems

- Utility function with parameters assumed
- We are going to see two of these demand systems, including

  AIDS: Deaton and Muellbauer (1980)

  QUAIDS: Banks, Blundell, and Lewbel (1997)

- Let's start with the AIDS model

Notations

- We are going to use the following notations going forward

$$
\begin{aligned}
i &- \text{index goods} \\
j &- \text{index goods; used in double summations} \\
h &- \text{index goods; used in double summations} \\
p_i &- \text{unit price of good } i \\
q_i &- \text{quantity of good } i \\
m &- \text{total expendutre} \\
w_i &- \text{budget/expenditure share} = p_i q_i / m
\end{aligned}
$$

Almost ideal demand system – model

- Deaton and Muellbaur (1980)
- One of the most commonly used demand systems
- Can be viewed as a first-order approximation to any demand system
- Can be fit to data aggregated across consumers
- Use linear estimation methods (SUR)
- Restrictions like homogeneity, symmetry, and adding-up

Almost ideal demand system – model

- Starting with the expenditure function and using Shephard's (1970) lemma, the budget-share functions:

$$w_i = \alpha_i + \sum_{j=1} \gamma_{ij} \ln p_j + \beta_i \ln \left[\frac{m}{a(\mathbf{p})} \right] \tag{1}$$

- translog price index:

  $\ln a(\mathbf{p}) = \alpha_0 + \sum_{i=1} \alpha_i \ln p_i + \frac{1}{2} \sum_{i=1} \sum_{j=1} \gamma_{ij} \ln p_i \ln p_j$

- adding-up:

  $\sum_{i=1} \alpha_i = 1; \quad \sum_{i=1} \beta_i = 0; \quad \sum_{i=1} \gamma_{ij} = 0$

- Slutsky symmetry: $\gamma_{ij} = \gamma_{ji}$

Almost ideal demand system – estimation

```
. demandsys aids w_dairy w_proteins w_fruitveg w_flours w_misc,      ///
            prices(p_dairy p_proteins p_fruitveg p_flours p_misc)   ///
            expenditure(expfd) nolog

Calculating NLS estimates ...
Calculating FGNLS estimates ...
FGNLS iteration 2 ...
FGNLS iteration 3 ...

AIDS model                                   Number of obs        = 4,160
Expenditure variable: expfd                  Number of goods      =     5
                                             Price index constant = 1.615

Centered R2 of model for
  Good 1 = 0.0254
  Good 2 = 0.1443
  Good 3 = 0.0731
  Good 4 = 0.1488
  Good 5 = 0.1575
```

Almost ideal demand system – estimation

	Estimate	Std. err.	z	P>|z|	[95% conf.	interval]
alpha						
Good						
1	.1948368	.0057939	33.63	0.000	.1834809	.2061927
2	.2475077	.0090901	27.23	0.000	.2296915	.2653238
3	.3553615	.0072053	49.32	0.000	.3412394	.3694836
4	.1097275	.003782	29.01	0.000	.102315	.11714
beta						
Good						
1	-.0056867	.0020128	-2.83	0.005	-.0096318	-.0017416
2	.0224212	.0031302	7.16	0.000	.0162861	.0285562
3	-.0234923	.0024448	-9.61	0.000	-.028284	-.0187007
4	-.0009142	.0013059	-0.70	0.484	-.0034738	.0016453
Gamma						
Good_g#Good_h						
1#1	.0236658	.0027836	8.50	0.000	.0182101	.0291215
1#2	-.0239425	.0030348	-7.89	0.000	-.0298906	-.0179944
1#3	-.0026764	.0026083	-1.03	0.305	-.0077886	.0024358
1#4	-.0001046	.0015118	-0.07	0.945	-.0030677	.0028585
2#2	.1369687	.0057856	23.67	0.000	.1256291	.1483083
2#3	-.0463497	.0039095	-11.86	0.000	-.0540122	-.0386872
2#4	-.0329757	.0022046	-14.96	0.000	-.0372967	-.0286548
3#3	.0602447	.0045134	13.35	0.000	.0513987	.0690908
3#4	-.0008966	.0019898	-0.45	0.652	-.0047965	.0030034
4#4	.0480769	.0017805	27.00	0.000	.0445873	.0515666

Note: alpha estimates are constant terms in expenditure-share equations and also appear in the price index.
Note: beta estimates measure sensitivity of expenditure shares to changes in deflated expenditure.
Note: Gamma estimates measure the effect of price on expenditure shares across goods.

Almost ideal demand system – estimation

Normalized parameters

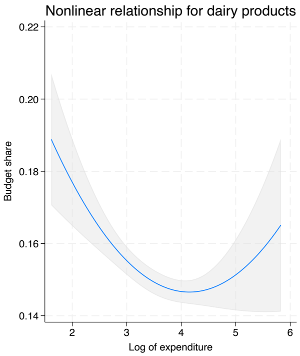
	Estimate	Std. err.	z	P>|z|	[95% conf.	interval]
alpha						
Good						
5	.0925665	.0044275	20.91	0.000	.0838888	.1012443
beta						
Good						
5	.0076721	.0015745	4.87	0.000	.0045862	.010758
Gamma						
Good_g#Good_h						
1#5	.0030578	.0016752	1.83	0.068	-.0002255	.0063412
2#5	-.0337007	.002441	-13.81	0.000	-.0384849	-.0289165
3#5	-.010322	.0021953	-4.70	0.000	-.0146247	-.0060194
4#5	-.0141	.0013383	-10.54	0.000	-.0167229	-.011477

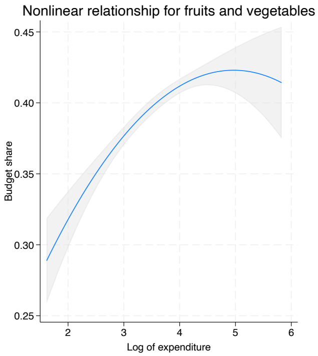
Note: alpha estimates sum to 1.
Note: beta estimates sum to 0.
Note: Gamma estimates sum to 0 over goods.

Shares: w_dairy w_proteins w_fruitveg w_flours w_misc
Prices: p_dairy p_proteins p_fruitveg p_flours p_misc
Expenditure: expfd

Nonlinear relationship

- What if the relationship between budget share and log of expenditure (Engel curve) is not linear





See `dmdsys.do`

Quadratic almost ideal demand system – model

- Adding a quadratic term of the logarithm of total expenditures (Banks, Blundell, and Lewbel 1997)

$$w_i = \alpha_i + \sum_{j=1} \gamma_{ij} \ln p_j + \beta_i \ln\left[\frac{m}{a(\mathbf{p})}\right] + \frac{\lambda_i}{b(\mathbf{p})}\left\{\ln\left[\frac{m}{a(\mathbf{p})}\right]\right\}^2 \qquad (2)$$

- Additional price index: $b(\mathbf{p}) = \prod_{i=1} p_i^{\beta_i}$
- Additional adding-up: $\sum_{i=1} \lambda_i = 0$

Quadratic almost ideal demand system – estimation

```
. demandsys quaids w_dairy w_proteins w_fruitveg w_flours w_misc, ///
>         prices(p_dairy p_proteins p_fruitveg p_flours p_misc)   ///
>             expenditure(expfd) nolog

Calculating NLS estimates ...
Calculating FGNLS estimates ...
FGNLS iteration 2 ...
FGNLS iteration 3 ...

Quadratic AIDS model                                 Number of obs        = 4,160
Expenditure variable: expfd                          Number of goods      =     5
                                                     Price index constant = 1.615

Centered R2 of model for
  Good 1 = 0.0255
  Good 2 = 0.1455
  Good 3 = 0.0738
  Good 4 = 0.1513
  Good 5 = 0.1593
```

Quadratic almost ideal demand system – estimation

	Estimate	Std. err.	z	P>|z|	[95% conf.	interval]
alpha						
Good						
1	.2071544	.0102017	20.31	0.000	.1871595	.2271492
2	.2133004	.0159721	13.35	0.000	.1819957	.2446052
3	.3388197	.0125299	27.04	0.000	.3142616	.3633778
4	.128059	.0066406	19.28	0.000	.1150436	.1410743
beta						
Good						
1	-.0196036	.0096733	-2.03	0.043	-.0385629	-.0006442
2	.0606299	.0151515	4.00	0.000	.0309335	.0903262
3	-.0050897	.0115577	-0.44	0.660	-.0277423	.017563
4	-.0213393	.0062263	-3.43	0.001	-.0335426	-.009136
Gamma						
Good_g#Good_h						
1#1	.0229525	.0028426	8.07	0.000	.017381	.0285239
1#2	-.0223754	.0031765	-7.04	0.000	-.0286012	-.0161496
1#3	-.0023292	.0026307	-0.89	0.376	-.0074852	.0028268
1#4	-.0007712	.0015413	-0.50	0.617	-.0037922	.0022497
2#2	.1334479	.0061466	21.71	0.000	.1214008	.1454949
2#3	-.0467902	.0039915	-11.72	0.000	-.0546133	-.0389671
2#4	-.0316385	.0022979	-13.77	0.000	-.0361424	-.0271347
3#3	.0607485	.0045061	13.48	0.000	.0519168	.0695802
3#4	-.000997	.0020033	-0.50	0.619	-.0049235	.0029294
4#4	.0477099	.0017993	26.52	0.000	.0441834	.0512365

Quadratic almost ideal demand system – estimation

	Estimate	Std. err.	z	P>|z|	[95% conf. interval]	
lambda						
Good						
1	.003438	.0023414	1.47	0.142	-.0011511	.0080271
2	-.0094084	.0037056	-2.54	0.011	-.0166713	-.0021454
3	-.0045035	.0027509	-1.64	0.102	-.0098952	.0008881
4	.0050141	.001497	3.35	0.001	.0020801	.0079481

```
Note: alpha estimates are constant terms in expenditure-share equations and also appear in the price index.
Note: beta estimates measure sensitivity of expenditure shares to changes in deflated expenditure and also
      appear in the price aggregator function.
Note: Gamma estimates measure the effect of price on expenditures shares across goods.
Note: lambda estimates measure the sensitivity of expenditure shares to changes in deflated expenditure.
```

Normalized parameters

	Estimate	Std. err.	z	P>|z|	[95% conf. interval]	
alpha						
Good						
5	.1126665	.007932	14.20	0.000	.0971201	.1282129
beta						
Good						
5	-.0145973	.007465	-1.96	0.051	-.0292285	.0000339
Gamma						
Good_g#Good_h						
1#5	.0025234	.0016951	1.49	0.137	-.000799	.0058457
2#5	-.0326438	.0025155	-12.98	0.000	-.037574	-.0277135
3#5	-.0106321	.0021957	-4.84	0.000	-.0149355	-.0063287
4#5	-.0143032	.0013496	-10.60	0.000	-.0169484	-.0116579

Quadratic almost ideal demand system – estimation

```
─────────────┬────────────────────────────────────────────────────────────────
lambda       │
        Good │
           5 │   .0054598   .0017864     3.06   0.002     .0019585     .008961
─────────────┴────────────────────────────────────────────────────────────────
Note: alpha estimates sum to 1.
Note: beta estimates sum to 0.
Note: Gamma estimates sum to 0 over goods.
Note: lambda estimates sum to 0 over goods.

Shares: w_dairy w_proteins w_fruitveg w_flours w_misc
Prices: p_dairy p_proteins p_fruitveg p_flours p_misc
Expenditure: expfd
```

Testing for quadratic relationship

- We can perform a joint (simultaneous) test on the `lambda` parameters.

```
. test [lambda]1.Good [lambda]2.Good [lambda]3.Good [lambda]4.Good
 ( 1)  [lambda]1bn.Good = 0
 ( 2)  [lambda]2.Good = 0
 ( 3)  [lambda]3.Good = 0
 ( 4)  [lambda]4.Good = 0
           chi2(  4) =   21.18
         Prob > chi2 =    0.0003
```

- See `r(table)` if you do not know how to refer to these parameters

Controlling for demographic factors

- You might want to control for individual socio-demographics. In this household-level consumption data, we have two such variables available
- The `demographics()` option
- There are two types of methods for including these factors in
  - **scaling** (Pollak and Wales 1978)
  - **translating** (Ray 1983 and Poi 2002)
- They are supported for AIDS and QUAIDS models

Demographics – translating

- We can incorporate demographics by translating the consumer's available level of expenditures
- It allows constant terms that might be interpreted as subsistence or committed quantities
- The `demographics()` option assumes translating by default

```
. demandsys quaids w_dairy w_proteins w_fruitveg w_flours w_misc, ///
          prices(p_dairy p_proteins p_fruitveg p_flours p_misc)   ///
          expenditure(expfd) nolog demographics(n_kids n_adults, translating)
(output omitted)
```

Demographics – scaling

- Demographic variables have "scale" and "composition" effects on expenditures
- No subsistence quantity is allowed
- There is no straightforward interpretation

Scaling

```
. demandsys quaids w_dairy w_proteins w_fruitveg w_flours w_misc, ///
        prices(p_dairy p_proteins p_fruitveg p_flours p_misc)   ///
        expenditure(expfd) nolog demographics(n_kids n_adults, scaling)
Calculating NLS estimates ...
Calculating FGNLS estimates ...
FGNLS iteration 2 ...
FGNLS iteration 3 ...
FGNLS iteration 4 ...
Quadratic AIDS model                            Number of obs          = 4,160
Expenditure variable: expfd                     Number of goods        =     5
                                                Price index constant   = 1.615
Demographic method: Scaling                     Number of demographics =     2
Centered R2 of model for
  Good 1 = 0.0453
  Good 2 = 0.1530
  Good 3 = 0.0949
  Good 4 = 0.1813
  Good 5 = 0.1726
```

Scaling

	Estimate	Std. err.	z	P>|z|	[95% conf.	interval]
alpha						
Good						
1	.2003664	.005088	39.38	0.000	.190394	.2103388
2	.2710113	.0081223	33.37	0.000	.2550918	.2869308
3	.2803594	.0067721	41.40	0.000	.2670863	.2936325
4	.1276662	.0044845	28.47	0.000	.1188767	.1364557
beta						
Good						
1	-.0098312	.0058505	-1.68	0.093	-.021298	.0016356
2	-.0056425	.0093173	-0.61	0.545	-.0239041	.012619
3	.0741078	.0085536	8.66	0.000	.0573431	.0908726
4	-.0298682	.0045862	-6.51	0.000	-.038857	-.0208794
Gamma						
Good_g#Good_h						
1#1	.0245885	.0027487	8.95	0.000	.0192011	.0299758
1#2	-.0241353	.0030037	-8.04	0.000	-.0300224	-.0182481
1#3	-.0027598	.0025703	-1.07	0.283	-.0077974	.0022778
1#4	-.0004213	.0014989	-0.28	0.779	-.0033592	.0025165
2#2	.1356708	.0057562	23.57	0.000	.124389	.1469527
2#3	-.0477731	.0038913	-12.28	0.000	-.0553998	-.0401463
2#4	-.0313637	.0021814	-14.38	0.000	-.0356392	-.0270882
3#3	.061373	.0044453	13.81	0.000	.0526603	.0700856
3#4	-.0001086	.0019627	-0.06	0.956	-.0039555	.0037383
4#4	.046515	.0017694	26.29	0.000	.043047	.0499829

Scaling

lambda						
Good						
1	-.0053093	.0023054	-2.30	0.021	-.0098278	-.0007908
2	.0126616	.0036869	3.43	0.001	.0054353	.0198878
3	-.0165185	.003475	-4.75	0.000	-.0233294	-.0097077
4	.003624	.0017086	2.12	0.034	.0002752	.0069728
rho						
n_kids	2.980711	.4686357	6.36	0.000	2.062202	3.89922
n_adults	.212198	.1373207	1.55	0.122	-.0569456	.4813416
Eta						
Good#c.n_kids						
1	-.0013818	.0021237	-0.65	0.515	-.0055441	.0027805
2	.010457	.0037207	2.81	0.005	.0031646	.0177495
3	-.0171813	.0031025	-5.54	0.000	-.0232621	-.0111005
4	.0004587	.0014143	0.32	0.746	-.0023133	.0032306
Good#c.n_adults						
1	.0019131	.0014235	1.34	0.179	-.0008769	.0047031
2	.0014948	.0025882	0.58	0.564	-.003578	.0065676
3	-.0094714	.0021673	-4.37	0.000	-.0137192	-.0052235
4	.0030685	.0008435	3.64	0.000	.0014153	.0047216

Note: alpha estimates are constant terms in expenditure-share equations and also appear in the price index.
Note: beta estimates measure sensitivity of expenditure shares to changes in deflated expenditure and also appear in the price aggregator function.
Note: Gamma estimates measure the effect of price on expenditures shares across goods.
Note: lambda estimates measure the sensitivity of expenditure shares to changes in deflated expenditure.
Note: rho estimates modify the price index to account for demographic variables.
Note: Eta estimates measure the effect of demographic variables on expenditure elasticities.

Scaling

Normalized parameters

	Estimate	Std. err.	z	P>|z|	[95% conf.	interval]
alpha						
Good						
5	.1205967	.0038554	31.28	0.000	.1130401	.1281532
beta						
Good						
5	-.0287659	.0049614	-5.80	0.000	-.0384901	-.0190417
Gamma						
Good_g#Good_h						
1#5	.0027279	.0016658	1.64	0.102	-.000537	.0059929
2#5	-.0323988	.0024273	-13.35	0.000	-.0371562	-.0276414
3#5	-.0107315	.0021697	-4.95	0.000	-.014984	-.006479
4#5	-.0146213	.0013254	-11.03	0.000	-.017219	-.0120237
lambda						
Good						
5	.0055422	.0020234	2.74	0.006	.0015765	.009508
Eta						
Good#c.n_kids						
5	.0076474	.0017373	4.40	0.000	.0042424	.0110524
Good#c.n_adults						
5	.002995	.0010839	2.76	0.006	.0008707	.0051194

Note: alpha estimates sum to 1.
Note: beta estimates sum to 0.
Note: Gamma estimates sum to 0 over goods.
Note: lambda estimates sum to 0 over goods.
Note: Eta estimates sum to 0.

Shares: w_dairy w_proteins w_fruitveg w_flours w_misc
Prices: p_dairy p_proteins p_fruitveg p_flours p_misc
Expenditure: expfd
Demographics: n_kids n_adults

Other options

- `piconstant(#)`

  By default, it is the logarithm of the minimum expenditure observed

- `vce(`*vcetype*`)`

  vcetype: `gnr`, `robust`, `cluster` *clustervar*, `bootstrap`, or `jackknife`

- `elasticities(`*e_type*`)`

  e_type: `compensated`, `uncompensated`, or `expenditure`

- `labels(`*string*`)`

Expenditure and price sensitivities

- We just need to differentiate the budget share questions (such as eq (2)) concerning prices and expenditure
- This is easily done by the `estat elasticities` command
- Here is its syntax

  `estat elasticities, compensated | uncompensated | expenditure`

Price sensitivities

```
. estat elasticities, compensated
Compensated (Hicksian) price elasticities               Number of obs = 4,160
(output omitted)

. estat elasticities, uncompensated
Uncompensated (Marshallian) price elasticities          Number of obs = 4,160
```

Price	Elasticity	Std. err.	z	P>|z|	[95% conf.	interval]
Good 1						
Good						
1	-.811714	.0188288	-43.11	0.000	-.8486178	-.7748102
2	-.0766982	.0078948	-9.72	0.000	-.0921716	-.0612247
3	-.013281	.0111339	-1.19	0.233	-.035103	.008541
4	.0212294	.0158979	1.34	0.182	-.00993	.0523887
5	.0340141	.0171309	1.99	0.047	.0004381	.06759
Good 2						
Good						
1	-.1108084	.0216528	-5.12	0.000	-.1532471	-.0683697
2	-.6860889	.0154335	-44.45	0.000	-.7163381	-.6558397
3	-.2040042	.0175709	-11.61	0.000	-.2384425	-.1695659
4	-.2705123	.0240964	-11.23	0.000	-.3177405	-.2232842
5	-.3129865	.0260175	-12.03	0.000	-.3639799	-.261993

Price sensitivities

Good 3						
Good						
1	.0147703	.0178819	0.83	0.409	-.0202775	.0498181
2	-.1449055	.0102831	-14.09	0.000	-.1650599	-.124751
3	-.7347995	.0190742	-38.52	0.000	-.7721842	-.6974147
4	.0272859	.0208954	1.31	0.192	-.0136683	.0682401
5	-.1059105	.0223895	-4.73	0.000	-.1497931	-.0620279
Good 4						
Good						
1	.0122822	.0101837	1.21	0.228	-.0076775	.0322419
2	-.0901695	.0056552	-15.94	0.000	-.1012535	-.0790856
3	-.004098	.0083854	-0.49	0.625	-.0205331	.0123371
4	-.4955827	.0183324	-27.03	0.000	-.5315136	-.4596518
5	-.1399958	.0134104	-10.44	0.000	-.1662798	-.1137118
Good 5						
Good						
1	.0331797	.0113969	2.91	0.004	.0108422	.0555171
2	-.0927011	.0063303	-14.64	0.000	-.1051083	-.0802939
3	-.048519	.0093377	-5.20	0.000	-.0668205	-.0302174
4	-.1326723	.0139445	-9.51	0.000	-.160003	-.1053416
5	-.4403348	.0202901	-21.70	0.000	-.4801025	-.400567

Expenditure sensitivities

```
. estat elasticities if n_kids == 2, expenditure atmeans
```

Expenditure elasticities Number of obs = 751

Expenditure	Elasticity	Std. err.	z	P>|z|	[95% conf.	interval]
Good						
1	.9060931	.0258604	35.04	0.000	.8554076	.9567787
2	1.08634	.0169256	64.18	0.000	1.053166	1.119514
3	1.007483	.0230983	43.62	0.000	.9622111	1.052755
4	.8360825	.023701	35.28	0.000	.7896295	.8825356
5	.9912156	.0274	36.18	0.000	.9375127	1.044919

Note: Elasticities are calculated at prices’, demographic variables’, and expenditure means.

Presenting price sensitivities

Table 1. Own- and cross-price elasticities

	Dairy	Proteins	Fruits	Flours	Misc
Dairy	-0.812***	-0.077***	-0.013	0.021	0.034**
Proteins	-0.111***	-0.686***	-0.204***	-0.271***	-0.313***
Fruits	0.015	-0.145***	-0.735***	0.027	-0.106***
Flours	0.012	-0.090***	-0.004	-0.496***	-0.140***
Misc	0.033***	-0.093***	-0.049***	-0.133***	-0.440***

*** p<.01, ** p<.05, * p<.1
See dmdsys.do for how to create this table using the -collect- system

Summarizing demand interrelationship

Table 2. Demand interrelationship

	Dairy	Proteins	Fruits	Flours	Misc
Dairy	--	Comp ***	Comp	Subs	Subs **
Proteins	Comp ***	--	Comp ***	Comp ***	Comp ***
Fruits	Subs	Comp ***	--	Subs	Comp ***
Flours	Subs	Comp ***	Comp	--	Comp ***
Misc	Subs ***	Comp ***	Comp ***	Comp ***	--

*** p<.01, ** p<.05, * p<.1
Notes: Subs -- substitutes; Comp -- complements
See dmdsys.do for how to create this table using the -collect- system

Program your own demand system

- Apart from the eight flexible demand systems available through the `demandsys` command, there is an option to program custom demand models.
- This approach provides complete control over functional forms, allowing for the incorporation of unique specifications.
- You have the flexibility to introduce personalized variations to your demand systems.
- Achieving this customization is possible through the utilization of the `nlsur` command.

Program your own demand system – AIDS

- We can write a function evaluator program and feed it into `nlsur` for estimation
- Let's see how we can implement this for the AIDS model
- The structure of this evaluator program will look like:

```
program define program_name
syntax ...
define your parameters
write up functions where parameters are evaluated
end
```

Program your own demand system – AIDS

Let's see a detailed example for AIDS (see `dmdsys.do`)

$$w_i = \alpha_i + \sum_{j=1} \gamma_{ij} \ln p_j + \beta_i \ln \left[\frac{m}{a(\mathbf{p})} \right]$$

```
capture program drop nlsurmyaids
program nlsurmyaids
version 18.0
syntax varlist(min=10 max=10) if, at(name)
tokenize `varlist'
args w1 w2 w3 w4 p1 p2 p3 p4 p5 exp
//defining alphas and adding-up
tempname a1 a2 a3 a4 a5
scalar `a1' = `at'[1,1]
scalar `a2' = `at'[1,2]
scalar `a3' = `at'[1,3]
scalar `a4' = `at'[1,4]
scalar `a5' = 1-`a1'-`a2'-`a3'-`a4'
//defining betas and adding-up
tempname b1 b2 b3 b4
scalar `b1' = `at'[1,5]
scalar `b2' = `at'[1,6]
scalar `b3' = `at'[1,7]
scalar `b4' = `at'[1,8]
```

Program your own demand system – AIDS

We have seen so far?

- We defined a `nlsur` program called `myaids`
- We wrote the `syntax` statement to accept 10 variables: four budget shares, five price variables, and one for expenditure.
- The tokenize command assigns to macros ‘1’, ‘2’,. . . , variables stored in ‘varlist’, and the `args` command transfers those numbered macros to macros ‘w1’, ‘w2’, . . . , etc.
- We defined the alpha and beta parameters in equation (3)
- Let’s continue

Program your own demand system – AIDS

```
//defining gammas, adding-up, and symmetry
tempname g11 g12 g13 g14 g15
tempname g21 g22 g23 g24 g25
tempname g31 g32 g33 g34 g35
tempname g41 g42 g43 g44 g45
tempname g51 g52 g53 g54 g55
scalar `g11' = `at'[1,9]
scalar `g12' = `at'[1,10]
scalar `g13' = `at'[1,11]
scalar `g14' = `at'[1,12]
scalar `g15' = -`g11'-`g12'-`g13'-`g14'
scalar `g21' = `g12'
scalar `g22' = `at'[1,13]
scalar `g23' = `at'[1,14]
scalar `g24' = `at'[1,15]
scalar `g25' = -`g21'-`g22'-`g23'-`g24'
scalar `g31' = `g13'
scalar `g32' = `g23'
scalar `g33' = `at'[1,16]
scalar `g34' = `at'[1,17]
scalar `g35' = -`g31'-`g32'-`g33'-`g34'
scalar `g41' = `g14'
scalar `g42' = `g24'
scalar `g43' = `g34'
scalar `g44' = `at'[1,18]
scalar `g45' = -`g41'-`g42'-`g43'-`g44'
scalar `g51' = `g15'
scalar `g52' = `g25'
scalar `g53' = `g35'
scalar `g54' = `g45'
scalar `g55' = -`g51'-`g52'-`g53'-`g54'
```

Program your own demand system – AIDS

```
//price indexes/aggregators and budget share functions
quietly {
tempvar lnpindex
generate double ‘lnpindex’ = 0 + ‘a1’*ln(‘p1’) + ‘a2’*ln(‘p2’) + ///
‘a3’*ln(‘p3’) + ‘a4’*ln(‘p4’) + ‘a5’*ln(‘p5’)
forvalues i = 1/5 {
forvalues j = 1/5 {
replace ‘lnpindex’ = ‘lnpindex’ + ///
0.5*‘g‘i’‘j’’*ln(‘p‘i’’)*ln(‘p‘j’’)
}
}
replace ‘w1’ = ‘a1’ + ‘g11’*ln(‘p1’) + ‘g12’*ln(‘p2’) + ///
‘g13’*ln(‘p3’) + ‘g14’*ln(‘p4’) + ‘g15’*ln(‘p5’) + ///
‘b1’*(ln(‘exp’) - ‘lnpindex’)
replace ‘w2’ = ‘a2’ + ‘g21’*ln(‘p1’) + ‘g22’*ln(‘p2’) + ///
‘g23’*ln(‘p3’) + ‘g24’*ln(‘p4’) + ‘g25’*ln(‘p5’) + ///
‘b2’*(ln(‘exp’) - ‘lnpindex’)
replace ‘w3’ = ‘a3’ + ‘g31’*ln(‘p1’) + ‘g32’*ln(‘p2’) + ///
‘g33’*ln(‘p3’) + ‘g34’*ln(‘p4’) + ‘g35’*ln(‘p5’) + ///
‘b3’*(ln(‘exp’) - ‘lnpindex’)
replace ‘w4’ = ‘a4’ + ‘g41’*ln(‘p1’) + ‘g42’*ln(‘p2’) + ///
‘g43’*ln(‘p3’) + ‘g44’*ln(‘p4’) + ‘g45’*ln(‘p5’) + ///
‘b4’*(ln(‘exp’) - ‘lnpindex’)
}
end
```

Program your own demand system – AIDS

- Then we can feed this program to nlsur

```
webuse food_consumption, clear

nlsur myaids @ w_dairy w_proteins w_fruitveg w_flours ///
 p_dairy p_proteins p_fruitveg p_flours p_misc expfd, ///
 parameters(a1 a2 a3 a4 b1 b2 b3 b4 g11 g12 g13 ///
 g14 g22 g23 g24 g33 g34 g44) neq(4) ifgnls
```

Program your own demand system – AIDS

- You might notice the same results as

```
demandsys aids w_dairy w_proteins w_fruitveg w_flours w_misc, ///
 prices(p_dairy p_proteins p_fruitveg p_flours p_misc) ///
 expenditure(expfd) piconstant(0)
```

- I manually input value 0 for α_0 used in `lnpindex` ($\ln a(\mathbf{p})$)
- We can recover those normalized parameters using `lincom` and adding-up restrictions
- Concerning elasticities, derive the formulas for them and use `nlcom`

Program your own demand system – QUAIDS

- What if we would like to add the quadratic terms? We just need to modify the program according to equation (2)
  - We need to define λs; add the Cobb-Douglas price index; and quadratic terms
  - See `dmdsys.do` for an example
- You can add other flavors to the demand system, such as, seasonality, demographic variables, different functional forms, linearized price index, etc

Conclusions

- Fit demand systems using `demandsys` introduced in Stata 18
- Calculate sensitivity measured using postestimation tools
- Program your own demand systems

Web source

- `demandsys` pdf manual
  www.stata.com/manuals/rdemandsys.pdf

- `demandsys postestimation` pdf manual
  www.stata.com/manuals/rdemandsyspostestimation.pdf

- Feature page
  www.stata.com/new-in-stata/flexible-demand-system-estimation

- `nlsur` pdf manual
  www.stata.com/manuals/rnlsur.pdf

Need help?

- Contact tech support
  www.stata.com/support/tech-support/contact/
- Interact with other users
  www.statalist.org
- Upcoming webinars and training
  www.stata.com/training/webinar/
  www.stata.com/training/classroom-and-web/

Reference

1. Banks, J., R. Blundell, and A. Lewbel. 1997. Quadratic Engle curves and consumer demand. *Review of Economics and Statistics* 79: 527–539.
2. Christensen, L. R., D. W. Jorgenson, and L. J. Lau. 1975. Transcendental logarithmic utility functions. *American Economic Review* 65: 367–383.
3. Deaton, A. S., and M. Irish. 1984. Statistical models for zero expenditures in household budgets. *Journal of Public Economics* 23: 59–80.
4. Poi, B. P. 2002. From the help desk: Demand system estimation. *Stata Journal* 2: 403–410.
5. Roy, P. R. 1943. La hiérarchie des besoins et la notion de groupes dans l'économie de choix. *Econometrica* 11: 13–24.
6. Shephard, R. W. 1970. *Theory of Cost and Production Functions*. Princeton, NJ: Princeton University Press.

STATA® 18