Title

# demandsys postestimation — Postestimation tools for demandsys

## Postestimation commands

The following postestimation commands are of special interest after `demandsys`:

| Command | Description |
|---|---|
| `estat elasticities` | price and expenditure elasticities |
| `estat parameters` | estimated parameter vectors and matrices |

The following standard postestimation commands are also available:

| Command | Description |
|---|---|
| `estat ic` | Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC) |
| `estat summarize` | summary statistics for the estimation sample |
| `estat vce` | variance–covariance matrix of the estimators (VCE) |
| `estimates` | cataloging estimation results |
| `lincom` | point estimates, standard errors, testing, and inference for linear combinations of coefficients |
| `lrtest` | likelihood-ratio test |
| `margins` | marginal means, predictive margins, marginal effects, and average marginal effects |
| `marginsplot` | graph the results from margins (profile plots, interaction plots, etc.) |
| `nlcom` | point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients |
| `predict` | predicted shares, residuals, quantities, and functions |
| `predictnl` | point estimates, standard errors, testing, and inference for generalized predictions |
| `test` | Wald tests of simple and composite linear hypotheses |
| `testnl` | Wald tests of nonlinear hypotheses |

## predict

### Description for predict

predict creates a new variable containing statistics such as predicted shares and residuals based on the demandsys estimation results currently active.

### Menu for predict

Statistics > Postestimation

### Syntax for predict

```
predict [type] newvar [if] [in] [, statistic options]
```

| *statistic* | Description |
|---|---|
| Main | |
| shares | predicted shares; the default |
| quantities | predicted quantities |
| residuals | residuals |
| iuf | indirect utility function |
| ef | expenditure function |

| *options* | Description |
|---|---|
| Main | |
| equation(#*eqno*) | display only specified goods; available only with shares, quantities, and residuals |
| prices(*varlist*$_p$) | specify set of counterfactual price variables |
| expenditures(*varname*) | specify a counterfactual total expenditure variable |
| utilities(*varname*) | specify the variable containing the utilities at which to compute the expenditure function; available only with and required with ef |
| demographics(*varlist*) | specify a set of counterfactual demographic variables |

### Options for predict

Main

shares, the default, calculates the predicted shares.

quantities calculates the predicted quantities given actual prices, actual expenditures, and predicted shares.

residuals calculates the residuals.

iuf calculates the indirect utility function $v(\mathbf{p}, m)$.

ef calculates the expenditure function $e(\mathbf{p}, u)$. You must also specify option utilities() with this option. This option is not available after translog and generalized translog models.

`equation(#`*eqno*`)` specifies to which good you are referring. `equation(#1)` would mean the calculation is to be made for the first good, `equation(#2)` would mean the second, and so on. If you do not specify `equation()`, results are the same as if you specified `equation(#1)`.

`prices(`*varlist*$_p$`)` specifies a set of counterfactual price variables at which to compute the requested statistic. You must specify $G$ variables, where $G$ is the number of goods in the demand system. By default, `predict` uses the same price variables that were used to fit the demand system. This option aids in the computation of welfare and other policy measures by allowing you to contrast predicted values at different price levels.

`expenditures(`*varname*`)` specifies a counterfactual total expenditure variable at which to compute the requested statistic. By default, `predict` uses the same expenditure variable that was used to fit the demand system. This option aids in the computation of welfare and other policy measures by allowing you to contrast predicted values at different expenditure levels.

`utilities(`*varname*`)` specifies the variable containing the utilities at which the expenditure function is to be computed. Almost surely, *varname* is a variable that was created by calling `predict` ..., `iuf` .... This option aids in the computation of welfare and other policy measures by allowing you to contrast predicted values at different levels of utility.

`demographics(`*varlist*`)` specifies a set of counterfactual demographic variables at which to compute the requested statistic. If you specified $D$ demographic variables at estimation, then you must specify precisely $D$ variables here. By default, predict uses the same demographic variables (if any) that were used to fit the demand system.

## margins

### Description for margins

`margins` estimates margins for the predicted share of the first good.

### Menu for margins

Statistics > Postestimation

### Syntax for margins

`margins` [ , *options* ]

`margins, predict(`*statistic* ...`)` [ *options* ]

| *statistic* | Description |
|---|---|
| `shares` | predicted shares; the default |
| `quantities` | predicted quantities |
| `residuals` | residuals |
| `iuf` | indirect utility function |
| * `ef` | expenditure function |

*You must specify the `predict` option `utilities()` with statistic `ef`.

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

## estat

### Description for estat

`estat elasticities` computes expenditure and price elasticities after fitting a demand system using `demandsys`. Elasticities are calculated at the price and expenditure levels in the data. However, you may compute elasticities at the means of prices and expenditures or use other counterfactual prices or expenditures.

`estat parameters` returns the estimated parameter vectors and matrices based on the currently active `demandsys` estimation results. Results are stored in `r()` and vary depending on the demand model fitted. See [U] **18.8 Accessing results calculated by other programs**.

### Menu for estat

Statistics > Postestimation

### Syntax for estat

*Compute price and expenditure elasticities*

estat elasticities [*if*] [*in*] , { expenditure | compensated | uncompensated }
  [ [ atmeans | at(*atspec*) ] generate(*stub*) ]

*Report estimated parameter vectors and matrices*

estat parameters

### Options for estat elasticities

Main

`expenditure`, `compensated`, and `uncompensated` indicate the type of elasticities to be reported. You must specify one of these.

`expenditure` requests that the $G$ expenditure elasticities be calculated, where $G$ is the number of goods in the system.

`compensated` requests that compensated price elasticities be calculated. There is a total of $G^2$ such elasticities because there are $G$ goods, each of whose quantity consumed can be affected by any one of the $G$ prices changing.

`uncompensated` requests that the $G^2$ uncompensated price elasticities be calculated.

`atmeans` requests that elasticities be calculated at the means of the price, expenditure, and demographic variables. By default, elasticities are computed for each observation and then averaged.

`at(`*atspec*`)` specifies the prices, expenditures, and demographic variables' values at which elasticities are to be calculated. If you do not specify this option, elasticities are calculated at the observed values unless you specify option `atmeans`, in which case they are calculated at the means of the variables. See example 1 below. You may not specify option `at()` if you request compensated price elasticities.

`generate(`*stub*`)` requests that new variables be generated holding the observation-level elasticities. For expenditure elasticities, $G$ new variables *stub*`_1`, *stub*`_2`, ..., *stub*`_`*G* will be created, where $G$ is the number of goods. For price elasticities, $G^2$ new variables will be created of the form *stub*`_`*g*`_`*h* for $g = 1, \dots, G$ and $h = 1, \dots, G$. Variable *stub*`_`*g*`_`*h* will contain the percentage change in the quantity of good $h$ with respect to the price of good $g$.

## Remarks and examples

stata.com

Remarks are presented under the following headings:

### Introduction

`estat elasticities` allows you to compute expenditure and price elasticities after you fit a demand model and is much more flexible than the options available with `demandsys` to display elasticities in lieu of coefficient estimates. With `estat elasticities`, you can calculate elasticities at the same prices, expenditures, and demographic variables' levels as was used at estimation time, or you can specify your own values. For example, you could calculate price elasticities for every observation in your dataset assuming each household had two children, even if the number of children varies by household. Example 1 below shows you how that can be done.

`predict` allows you to obtain values of the indirect utility function and of the expenditure function underlying the demand system you fit. Example 3 leverages that ability and shows you how to compute compensated and equivalent variation in response to a price change.

`estat parameters` allows you to obtain the estimated parameter vectors and matrices for the demand system based on the coefficient vector stored in `e(b)`. This may be useful if you wish to conduct further policy experiments or calculate your own welfare measures. For instance, in example 2 below, we show how `estat parameters` with just a bit of coding can be used to check whether a fitted generalized translog model can be interpreted as having subsistence quantities for each good.

### Elasticities

Although one can have `demandsys` report expenditure or price elasticities for the estimation sample, the `estat elasticities` command gives you much more flexibility. We illustrate features of `estat elasticities` in the next example. We often request expenditure elasticities to keep the output shorter, but all our commands would work regardless of the type of elasticity requested.

#### ▷ Example 1

Before we can obtain elasticities, we first need to fit a demand system to our data. Here we include the `quietly` prefix command to omit the results of fitting a quadratic almost ideal demand system (QUAIDS) model with demographic translation, and we use the `labels()` option to provide short names for each of our goods. We type

```
. use https://www.stata-press.com/data/r18/food_consumption
(Food consumption)

. quietly demandsys quaids w_dairy w_proteins w_fruitveg w_flours w_misc,
> prices(p_dairy p_proteins p_fruitveg p_flours p_misc)
> expenditures(expfd)
> demographics(n_kids n_adults)
> labels(Dairy Meats FruitVeg Flours Misc)
```

Suppose we want the average expenditure elasticity among households with two children. We type

```
. estat elasticities if n_kids == 2, expenditure
Expenditure elasticities                              Number of obs = 751

─────────────┬────────────────────────────────────────────────────────────────
 Expenditure │  Elasticity   Std. err.      z    P>|z|     [95% conf. interval]
─────────────┼────────────────────────────────────────────────────────────────
       Dairy │    .9028921   .0240869    37.48   0.000     .8556826    .9501016
       Meats │    1.087763   .0156852    69.35   0.000      1.05702    1.118505
    FruitVeg │    .9978958   .0215996    46.20   0.000     .9555614     1.04023
      Flours │    .8523383   .0220511    38.65   0.000     .8091189    .8955576
        Misc │    .9971274   .0265993    37.49   0.000     .9449938    1.049261
─────────────┴────────────────────────────────────────────────────────────────
```

The header of the output reports that there are 751 households in the dataset with two children. Had we not specified the `labels()` option when we fit our model, `estat elasticities` would have simply numbered the goods from one through five.

What is displayed in the table of output is an average elasticity. Behind the scenes, `estat elasticities` calculated the five expenditure elasticities for each of the 751 households in this sample. What is reported as the expenditure elasticity for dairy products is the average over the 751 individual expenditure elasticities calculated. The column labeled `Std. err.` contains the standard error of the estimated mean. Looking to the right, we see the 95% confidence interval for the mean dairy elasticity is [0.856, 0.950]. More importantly, the column labeled `Std. err.` is not the sample standard deviation of the 751 individual elasticities, nor does the 95% confidence interval report something resembling the central 95% of the individual elasticities.

To save the individual elasticities for the 751 observations, you can use the `generate()` option like so:

```
. quietly estat elasticities if n_kids == 2, expenditure generate(ee)

. summarize ee*
    Variable │        Obs        Mean    Std. dev.       Min        Max
─────────────┼─────────────────────────────────────────────────────────
        ee_1 │        751    .9028921    .0753773   .5095888    1.44884
        ee_2 │        751    1.087763    .0664746   .8448506   1.525114
        ee_3 │        751    .9978958    .1755427   .7566877    5.07766
        ee_4 │        751    .8523383    .1053903   .3864381   1.110392
        ee_5 │        751    .9971274     .082875   .3257591   1.296199
```

The means of these variables match the point estimates in the previous `estat elasticities` output. The standard deviations reported here measure the spread of the individual observations' elasticities rather than the standard errors of the means as in the previous output.

A second way to compute elasticities is to first compute the means of the price variables, the expenditure variable, and any demographics and then compute the elasticity for this hypothetical household. To have `estat elasticities` do that, you use the `atmeans` option. Here we obtain the expenditure elasticities at the means for the subset of households with two children:

```
. estat elasticities if n_kids == 2, expenditure atmeans
Expenditure elasticities                              Number of obs = 751

─────────────┬────────────────────────────────────────────────────────────
Expenditure  │ Elasticity   Std. err.      z    P>|z|     [95% conf. interval]
─────────────┼────────────────────────────────────────────────────────────
       Dairy │   .9155493   .0232877    39.31   0.000     .8699063    .9611923
       Meats │   1.078329   .0149179    72.28   0.000      1.04909    1.107567
    FruitVeg │   .9804481   .0201581    48.64   0.000     .9409389    1.019957
      Flours │   .8695964   .0212882    40.85   0.000     .8278722    .9113206
        Misc │   1.004098   .0246939    40.66   0.000     .9556994    1.052498
─────────────┴────────────────────────────────────────────────────────────
Note: Elasticities are calculated at prices', demographic variables', and
      expenditure means.
```

With the `atmeans` option, we calculate the elasticities for a hypothetical household that may have, for example, 0.87 children and 1.93 adults. Without the `atmeans` option, we are calculating elasticities for households that do exist in the data, though taking the average of household elasticities poses its own conceptual issues.

Consider the following table:

Table 1. Average expenditure elasticities

| Consumer | $m$ | $E^m$ | $Q$ | $m'$ | $Q'$ |
|---|---|---|---|---|---|
| $A$ | 100 | 2 | 20 | 110 | 24 |
| $B$ | 200 | 1 | 30 | 220 | 33 |
| $C$ | 400 | 0.5 | 30 | 440 | 31.5 |
| Sum | 700 | | 80 | 770 | 88.5 |
| Mean | | 1.17 | | | |

Column $m$ represents each consumer's initial total expenditure, $E^m$ is the consumer's expenditure elasticity for the good in question, and $Q$ is the quantity consumed. Column $m'$ represents a 10% increase in each consumer's expenditure, and $Q'$ represents the level of consumption based on the increases in total expenditure and the expenditure elasticities. The mean elasticity, analogous to what `estat elasticities` reports without the `atmeans` option, is 1.17. However, based on the aggregate data, the expenditure elasticity works out to 1.06.

So far in this example, we computed the expenditure elasticities for the subset of households with two children. We can also calculate the average elasticity for all households in our dataset under the counterfactual that all of them have two children. One way to do that is to type

```
. generate n_kids_backup = n_kids
. replace n_kids = 2
(3,409 real changes made)
. estat elasticities, expenditure
Expenditure elasticities                            Number of obs = 4,160

─────────────┬────────────────────────────────────────────────────────────
Expenditure  │ Elasticity   Std. err.      z    P>|z|     [95% conf. interval]
─────────────┼────────────────────────────────────────────────────────────
       Dairy │   .8464408   .0266003    31.82   0.000     .7943051    .8985764
       Meats │    1.20188   .0252998    47.51   0.000     1.152294    1.251467
    FruitVeg │   1.130297   .0323216    34.97   0.000     1.066948    1.193646
      Flours │   .7604338   .0229378    33.15   0.000     .7154766    .8053911
        Misc │   .9082412   .0287833    31.55   0.000     .8518269    .9646554
─────────────┴────────────────────────────────────────────────────────────
```

```
. replace n_kids = n_kids_backup
(3,409 real changes made)
```

Here we first created a backup variable holding the true values of `n_kids`, and then we set `n_kids` equal to two for all observations. After calling `estat elasticities`, we restored the true values.

◁

## ❑ Technical note

In principle, a second way to obtain the expenditure elasticities under the counterfactual that all households have two children would be to type

```
estat elasticities, expenditure at(n_kids=2)
```

The key is the `at()` option, which allows us to request that variables be set to the values we specify before calculating the elasticities. For more information about the `at()` option, see [R] **margins**, especially *Syntax of at()*.

However, in this example, if you issue the previous command, you would obtain the error message

```
. estat elasticities, expenditure at(n_kids=2)
Option at() specified; using margins.
could not calculate numerical derivatives -- discontinuous region with missing
values encountered
r(459);
```

The issue is that for most demand systems the predicted quantities are nonlinear functions of prices. As the output indicates, when you specify the `at()` option, `estat elasticities` calls `margins` to obtain the desired results. Depending on the exact demand system and demographic specification, `margins` may not be able to obtain all the required derivatives numerically.

When you do not use the `at()` option, `estat elasticities` uses analytics formulas for the expenditure and price elasticities; *Methods and formulas* of [R] **demandsys** includes the elasticity formulas for all the demand systems implemented. Evaluating analytic formulas is quick, and most of the time is spent computing the variance–covariance matrix of the estimated mean elasticities, because that does use numerical derivatives. Because those derivatives are for a sample mean, rather than individual observations, you are unlikely to obtain the same message as `margins` issued in the previous command.

In contrast, `margins` uses numerical derivatives for all its calculations, and, moreover, it must incur the added expense of having to call `predict` to evaluate predicted quantities. One case where you would need to use `margins` is to obtain, say, a quantity elasticity with respect to a change in a demographic variable, because those elasticities are not programmed into `demandsys`.

❑

In the text after example 4 and in example 6 of [R] **demandsys**, we cautioned that after fitting a generalized model that allows for subsistence or committed quantities, you should check whether that interpretation actually holds. If for some household $\sum_g p_g(\mu_g + \boldsymbol{\nu}_g'\mathbf{d}) > m$, then either the household is not consuming enough to meet what we think is a minimum subsistence or our assumption that this quantity represents a minimum subsistence is wrong. Moreover, if $\mu_g + \boldsymbol{\nu}_g'\mathbf{d} < 0$, then we are saying that the minimum required level of $g$ is less than 0, which implies that good $g$ is in fact not needed for subsistence.

## ▷ Example 2

This example is a continuation of example 6 in [R] **demandsys**. In that example, we fit a generalized QUAIDS model, which we refit here:

```
. use https://www.stata-press.com/data/r18/food_consumption, clear
(Food consumption)

. demandsys gquaids w_dairy w_proteins w_fruitveg w_flours w_misc,
> prices(p_dairy p_proteins p_fruitveg p_flours p_misc)
> demographics(n_kids n_adults) piconstant(3) expenditure(expfd) nolog
  (output omitted )
```

Next, we compute variables containing each household's implied minimum quantity of each good conditional on that household's demographics. Given those variables, we can then calculate the expenditure needed to acquire the minimum quantities given the prices each household faces. We type

```
. estat parameters

. return list

macros:
              r(names) : "alpha beta Gamma lambda mu Nu"

matrices:
                 r(Nu) :  5 x 2
                 r(mu) :  5 x 1
             r(lambda) :  5 x 1
              r(Gamma) :  5 x 5
               r(beta) :  5 x 1
              r(alpha) :  5 x 1

. matrix mu = r(mu)

. matrix Nu = r(Nu)

. forvalues g = 1/5 {
  2.     generate minq_`g' = mu[`g', 1] + Nu[`g',1]*n_kids + Nu[`g',2]*n_adults
  3. }

. generate sub_exp = p_dairy*minq_1 + p_proteins*minq_2
> + p_fruitveg*minq_3 + p_flour*minq_4 + p_misc*minq_5
```

We first issued the command `estat parameters` to obtain the estimated parameter matrices from our demand system. The following command shows that six vectors and matrices were stored, following the layout of the coefficient table from `demandsys`. To compute minimum quantities, we need matrices $\boldsymbol{\mu}$ and $\mathbf{N}$, so we transferred those `r()` results to Stata matrices.

Because we specified `n_kids` first in the `demographics()` option of `demandsys`, that variable's coefficients are stored in the first column of our matrix `r(Nu)`, which we converted to `Nu`. The second column corresponds to coefficients associated with `n_adults`. We looped over the five goods in our system and created variables named `minq_`*g* containing the minimum quantities. The final `generate` statement calculates the minimum expenditure for each household given the prices it faced and our calculated minimum quantities.

Now we can look at the results.

```
. summarize minq_*

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
      minq_1 |      4,160   -1.604524    .9955551  -7.287278  -.2078816
      minq_2 |      4,160   -2.893996    1.743902  -13.71866  -.6731596
      minq_3 |      4,160   -5.344703    3.802853  -29.30052  -.5518757
      minq_4 |      4,160    .3094949    .1354879  -.0171549   1.328885
      minq_5 |      4,160   -.1878336    .2030023   -1.34631   .0894621
```

For the first three goods in our system, the calculated minimum quantity is negative for every single household in our sample. That is hardly encouraging if we wish to take the subsistence interpretation seriously, because it implies three of the five categories of food we have are not required for subsistence. Moreover, on average, good 5 is not required either. Even for good 4, for some households, the minimum quantity is negative. Finally, we look at the expenditure needed to purchase the minimum quantities of the five goods:

```
. summarize sub_exp

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
     sub_exp |      4,160    -8.10118    5.480523   -43.8327   .2678578
```

Not surprisingly given the previous summary statistics, the minimum required expenditure for most households is less than zero.

◁

What are we to make of the results in the previous example? We think there are at least four explanations:

1. There are minimum quantities of each good consumers must purchase for subsistence, but our demand system is misspecified. If you believe this to be true, you might try changing the demographic variables for which you control. Alternatively, try a different model; for example, use a generalized QUAIDS model rather than a generalized translog or AIDS model.
2. The data are simply not compatible with a minimum subsistence interpretation. That does not imply the results of the fitted demand system must be ignored. As Pollak and Wales (1992, 75) write, any demand system that includes demographic translation should include a constant term in the demand equations—which is the key characteristic of the models we refer to as "generalized". For example, rather than using the translog model with demographic translation, you should instead use the generalized translog model with demographic translation. In other words, the $\mu$ parameters do belong in the model, but they should not be construed as minimum quantities required for subsistence.
3. Items for which the calculated minimum quantities are positive for all observations must be purchased to achieve subsistence, but items for which the minimum quantities are negative need not be consumed for subsistence. Under this interpretation, in our example, households must consume breads, cereals, and pastas but need not consume any other foods for minimal subsistence. That is not particularly palatable to us.
4. The procedure we used is useful as a diagnostic tool, but it is not a formal statistical test. Our estimated parameters are subject to sampling variation, and therefore our calculated minimum quantities are also subject to sampling variation. We have not accounted for this. If we were to conduct a test of the hypothesis that all minimum quantities are zero or more versus the alternative that some are negative, we may not be able to reject the null hypothesis.

## Evaluating elasticities

The elasticities that are reported with `estat elasticities` refer to the slope of the fitted demand curve. Let us explain what we mean. Look at figure 1. We have plotted a fitted demand curve $D^{\mathrm{F}}$ as a straight line for simplicity, and the large dots represent observed (price, quantity) pairs for two observations. The dotted lines represent each observation's residual, and the plus signs represent the predicted quantities given the two observations' prices. While quantity is a function of price here, following the influential English economist Alfred Marshall, we plot price on the ordinate; hence, the lines connecting observations to the regression line are horizontal, not vertical.

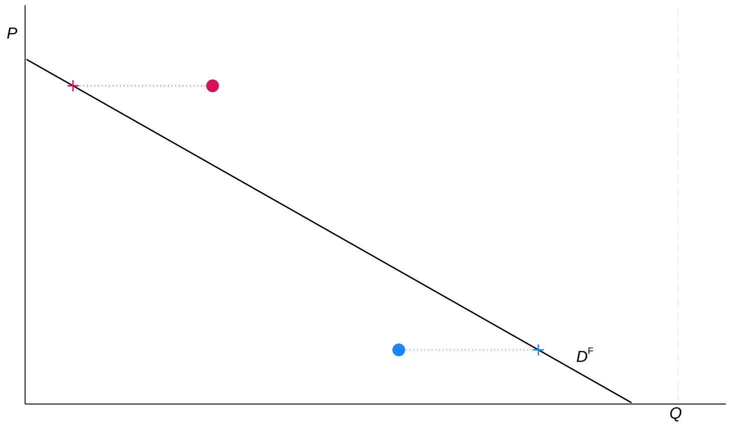

Figure 1. Actual and predicted quantities

For the AIDS model without demographic variables included, we can show that the expenditure elasticity for the $g$th good is given by

$$E^g = 1 + \frac{\beta_g}{w_g}$$

We do not observe the true value $\beta_g$, so to compute the elasticity for household $i$, we have no choice but to use the estimated $\widehat{\beta}_g$. However, we observe both household $i$'s true expenditure share $w_{gi}$, and we can compute its predicted expenditure share $\widehat{w_{gi}}$.

In terms of figure 1, the issue is whether we evaluate the expenditure or any other elasticity formula at the red and blue dots or at their corresponding fitted values shown by the pluses on the demand curve. We believe the correct procedure is to evaluate elasticity formulas at their fitted values. That is, `estat elasticities` evaluates the expenditure share elasticity for household $i$ as

$$\widehat{E_i^g} = 1 + \frac{\widehat{\beta_g}}{\widehat{w_{gi}}}$$

After all, for policy or other analyses, what we want to know is essentially the slope of the fitted demand curve. We need to be able to predict the effect of a policy on a household, and in general to do that, we predict the household's response with the policy imposed and compare it with the predicted response in the absence of the policy. In other words, our elasticity computations measure the slope of the fitted demand curve, not the slope of an imaginary line that passes through one of our dots that may or may not be parallel to the fitted demand curve.

In contrast, the postestimation tools accompanying the community-contributed `quaids` command of Poi (2012) use the actual values $w_g$ to calculate elasticities. Thus, users of that command will see minor differences when refitting models using `demandsys` and `estat elasticities`. We believe, however, that evaluating the formulas at the predicted expenditure shares, or equivalently, evaluating the elasticity formulas along the demand curve rather than at points in space that need not lie on the demand curve, is conceptually more sound. Moreover, the elasticity results reported after `quaids` cannot be replicated using the `margins` command, but the results reported by `estat elasticities` are identical to those one would obtain using `margins`.

## Compensating and equivalent variation

Although students in introductory economics courses learn of consumer's surplus as a way to measure whether consumers are better or worse off in response to a policy such as a tax, it has limitations. Perhaps most importantly, consumer's surplus is not well defined in most cases; see Auerbach (1985). Also see Harberger (1964), who develops a method for calculating consumer welfare when multiple prices change but assumes the timing of the price changes is scattered rather than occurring simultaneously. Hines (1999) provides a historical overview of consumer's welfare measurement.

The workhorse metrics used to evaluate a policy are compensating and equivalent variation (CV and EV, respectively). These metrics assign a monetary amount to a household's change in welfare and can accommodate policies that cause multiple prices to change simultaneously. Because `predict` after `demandsys` allows us to obtain the predicted values of the expenditure and indirect utility functions, these metrics are easy to obtain.

CV measures how much a social planner would have to compensate a consumer to offset the effect of, say, a tax that causes prices to rise. Consider figure 2, which illustrates CV for two goods and would be familiar to those who have studied intermediate microeconomics.

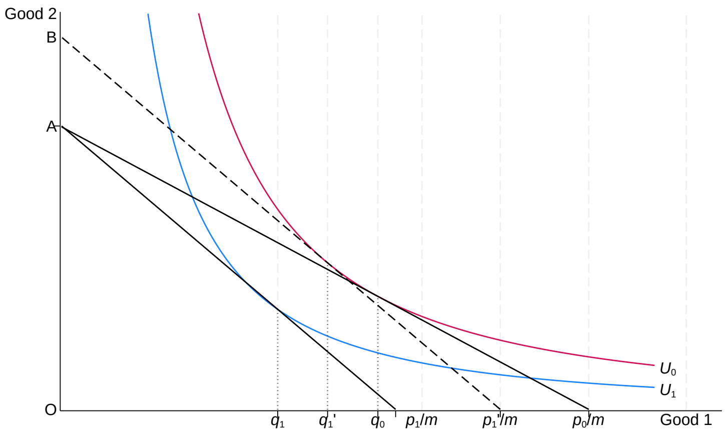

Figure 2. CV if the price of good 1 rises

Initially, the price of good 1 is $p_0$, and the line segment $\overline{A \cdot p_0/m}$ represents the budget constraint. The consumer purchases $q_0$ units of good 1 and enjoys utility level $U_0$. Now suppose the price of good 1 rises to $p_1$: consumption of good 1 falls to $q_1$ and utility falls to $U_1$; line segment $\overline{A \cdot p_1/m}$ represents the new budget constraint.

The dotted line segment $\overline{B \cdot p_1'm}$ represents a budget constraint based on price $p_1$ but shifted outward so that the consumer can still achieve utility level $U_0$ when facing price $p_1$, by consuming $q_1'$. The line segment $\overline{B \cdot A}$ represents the CV. It tells us, in the face of a higher price for good 1, how much the social planner would have to compensate this consumer so that he or she can enjoy the same level of utility as before the price increase. In this case, CV is negative because the social planner must pay the consumer.

Notice the close similarity to the concept of the income and substitution effects that cause demand curves to slope downward. CV, and EV as we will see shortly, is a measure of the income effect.

Mathematically, CV is defined as

$$\begin{aligned}\text{CV} &= e(\mathbf{p}^1, u^1) - e(\mathbf{p}^1, u^0)\\ &= e(\mathbf{p}^0, u^0) - e(\mathbf{p}^1, u^0)\\ &= m - e(\mathbf{p}^1, u^0)\end{aligned}$$

where $\mathbf{p}^0$ and $\mathbf{p}^1$ are the price vectors before and after the policy is implemented and $u^0$ is the household's initial level of utility. Where do we get that level of utility, considering that utility is unobserved? The answer is the indirect utility function, which expresses utility as a function of observed prices and expenditure; we can evaluate the indirect utility function by calling `predict` with the `iuf` option. With the indirect utility variable in hand, we then call `predict` with the `ef` option and set the price variables to $\mathbf{p}^1$.

## ▷ Example 3

We compute household CV in response to a 10% increase in the price of dairy products, holding all other prices constant. See Poi (2002) for similar computations for an expanded look at the welfare implications of lower dairy and other prices. First, we fit a generalized QUAIDS model controlling for the number of children and adults in each household. We type

```
. use https://www.stata-press.com/data/r18/food_consumption, clear
(Food consumption)
. quietly demandsys gquaids w_dairy w_proteins w_fruitveg w_flours w_misc,
> prices(p_dairy p_proteins p_fruitveg p_flours p_misc)
> expenditures(expfd) demographics(n_kids n_adults) nolog
. estimates store gquaids
```

We stored our estimation results for later use by specifying the `estimates store` command. Next, we obtain each household's initial utility $u^0$ by evaluating the indirect utility function:

```
. predict u0, iuf
```

Now, we create a new dairy price variable containing the price each household will face after the imposition of the tax. Given that price variable and initial utility $u^0$, we can calculate each household's new level of expenditure. Finally, we compute the CV. We type

```
. generate ptax_dairy = p_dairy * 1.10
. predict ep1u0, ef utilities(u0)
> prices(ptax_dairy p_proteins p_fruitveg p_flours p_misc)
. generate cv = expfd - ep1u0
```

Here we used the `prices()` option of `predict` to specify the new prices $\mathbf{p}^1$ at which we want the expenditure function. Had we not specified that option, `predict` would have used the prices used at estimation, $\mathbf{p}^0$. Because we wanted to evaluate the function at the level of expenditures $m$, the same as used at estimation, we did not need to specify the `expenditures()` option of `predict`.

We summarize our results by looking at the average CV by household size:

```
. mean cv, over(n_kids)
Mean estimation                         Number of obs = 4,160

--------------------------------------------------------------
             |       Mean   Std. err.     [95% conf. interval]
-------------+------------------------------------------------
c.cv@n_kids  |
           0 |  -.5142129   .0066562     -.5272626   -.5011631
           1 |  -.7762653    .013486      -.802705   -.7498257
           2 |  -.9060989    .014986     -.9354795   -.8767183
           3 |  -1.067159   .0268302      -1.11976   -1.014557
           4 |   -1.22238   .0506472     -1.321675   -1.123084
           5 |  -1.550712    .158197     -1.860863   -1.240561
           6 |  -1.304841   .2223387     -1.740744   -.8689384
           7 |  -1.902748   .1828158     -2.261165   -1.544331
           8 |  -2.923615          .             .           .
          10 |  -1.743889          .             .           .
--------------------------------------------------------------
```

The standard errors of the means are missing for households with 8 and 10 children because only one household of each size exists in our data. For the most part, CV becomes more negative with the number of children, as we expected, because larger households are likely to consume more dairy products by virtue of there being more people within the household. To reiterate, CV is negative here, indicating that the social planner must transfer money to the consumer. For a price decrease, CV would be positive.

◁

EV is similar to CV, except that we use the postpolicy level of utility and the initial price vector rather than the initial level of utility and postpolicy price vector. We have

$$\begin{aligned}\text{EV} &= e(\mathbf{p}^0, u^1) - e(\mathbf{p}^0, u^0)\\ &= e(\mathbf{p}^0, u^1) - e(\mathbf{p}^1, u^1)\\ &= e(\mathbf{p}^0, u^1) - m\end{aligned}$$

To aid in understanding EV, you may wish to sketch a diagram similar to figure 2. EV tells us how much consumers would be willing to pay to trade at the original price vector rather than the postpolicy one. In other words, EV tells us the change in expenditure that would yield $u^1$ if total expenditure were to change rather than the price vector. A negative value indicates that consumers are worse off.

## ▷ Example 4

Continuing the previous example, to calculate EV, we first need to calculate the new level of utility $u^1$ each household will enjoy based on the postpolicy dairy price. Because the preferences underlying all the demand systems implemented by `demandsys` are locally nonsatiated, each household will exhaust its full budget. Thus, the level of expenditure we supply as an argument to the indirect utility function is simply $m$, the value we used during estimation. To get $u^1$, we can therefore call `predict, iuf` at postpolicy prices.

```
. estimates restore gquaids
(results gquaids are active now)
. predict u1, iuf prices(ptax_dairy p_proteins p_fruitveg p_flours p_misc)
```

We had to restore our estimation results because the `means` command we used previously overwrites results in `e()`.

Now we evaluate the expenditure function at original prices and our computed level of utility $u^1$. Because the original prices were used at estimation, we do not need to supply alternative prices via the `prices()` option. We type

```
. predict ep0u1, ef utilities(u1)
```

We are now ready to compute EV and summarize the results:

```
. generate ev = ep0u1 - expfd
. mean ev, over(n_kids)
Mean estimation                          Number of obs = 4,160

--------------------------------------------------------------
             |       Mean   Std. err.     [95% conf. interval]
-------------+------------------------------------------------
 c.ev@n_kids |
           0 |  -.5073723   .0065622     -.5202378   -.4945068
           1 |  -.7660214    .013292     -.7920807    -.739962
           2 |  -.8940226    .014762      -.922964   -.8650811
           3 |  -1.052524   .0264206     -1.104322   -1.000725
           4 |  -1.205508   .0498826     -1.303304   -1.107711
           5 |  -1.528722   .1554489     -1.833485   -1.223959
           6 |  -1.285955    .218652      -1.71463   -.8572807
           7 |  -1.871663    .178062      -2.22076   -1.522567
           8 |   -2.87738          .             .           .
          10 |  -1.716629          .             .           .
--------------------------------------------------------------
```

◁

For small price changes, CV and EV will be similar; however, they will not match because CV is based on the original level of utility and postpolicy prices and EV is based on the postpolicy level of utility and original prices. Which one is more appropriate in any situation depends on the policy and goals. If the government is implementing, say, a carbon tax to reduce consumption of goods and services that emit carbon dioxide, CV may be a better choice because it tells how much the government must compensate consumers so that their overall utilities do not change. If one is evaluating the introduction of a new product competing with an existing product into the market, EV may be a better metric because it could be interpreted as a measure of how much better off consumers would be thanks to lower prices.

As with our analysis of estimated minimum quantities in example 2, here we have not attempted to obtain measures of the sampling variation associated with the CV or EV. One alternative to obtain, say, the sampling variation of the mean CV across households would be to use the bootstrap, being sure to bootstrap the entire procedure, including refitting the model at each bootstrap replication as well as obtaining the desired statistics.

## References

Auerbach, A. J. 1985. The theory of excess burden and optimal taxation. In Vol. 1 of *Handbook of Public Economics*, ed. A. J. Auerbach and M. Feldstein, 61–127. Amsterdam: Elsevier. https://doi.org/10.1016/S1573-4420(85)80005-7.

Harberger, A. C. 1964. The measurement of waste. *American Economic Review* 54: 58–76.

Hines, J. R. 1999. Three sides of Harberger triangles. *Journal of Economic Perspectives* 13: 167–188. https://doi.org/10.1257/jep.13.2.167.

Poi, B. P. 2002. From the help desk: Demand system estimation. *Stata Journal* 2: 403–410.

——. 2012. Easy demand-system estimation with quaids. *Stata Journal* 12: 433–446.

Pollak, R. A., and T. J. Wales. 1992. *Demand System Specification and Estimation*. New York: Oxford University Press.

Varian, H. R. 1992. *Microeconomic Analysis*. 3rd ed. New York: W. W. Norton.

## Also see

[R] **demandsys** — Estimation of flexible demand systems

[U] **20 Estimation and postestimation commands**