Title

matrix score — Score data from coefficient vectors

Description

`matrix score` creates $newvar_j = \mathbf{x}_j\mathbf{b}'$ ($\mathbf{b}$ being a row vector), where $\mathbf{x}_j$ is the row vector of values of the variables specified by the column names of $\mathbf{b}$. The name `_cons` is treated as a variable equal to 1.

Syntax

`matrix score` [*type*] *newvar* = **b** [*if*] [*in*]

[, `equation(`*##* | *eqname*`)` `missval(`*#*`)` `replace` `forcezero`]

where $\mathbf{b}$ is a $1 \times p$ matrix.

Options

`equation(`*##* | *eqname*`)` specifies the equation—by either number or name—for selecting coefficients from $\mathbf{b}$ to use in scoring. See **[U] 14.2 Row and column names** and **[P] matrix rownames** for more on equation labels with matrices.

`missval(`*#*`)` specifies the value to be assumed if any values are missing from the variables referred to by the coefficient vector. By default, this value is taken to be missing (.), and any missing value among the variables produces a missing score.

`replace` specifies that *newvar* already exists. Here observations not included by `if` *exp* and `in` *range* are left unchanged; that is, they are not changed to missing. Be warned that `replace` does not promote the storage type of the existing variable; if the variable was stored as an `int`, the calculated scores would be truncated to integers when stored.

`forcezero` specifies that, should a variable described by the column names of $\mathbf{b}$ not exist, the calculation treat the missing variable as if it did exist and was equal to zero for all observations. It contributes nothing to the summation. By default, a missing variable would produce an error message.

Remarks and examples

Scoring refers to forming linear combinations of variables in the data with respect to a coefficient vector. For instance, let's create and then consider the vector `coefs`:

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)
. quietly regress price weight mpg
. matrix coefs = e(b)
. matrix list coefs
coefs[1,3]
        weight         mpg       _cons
y1   1.7465592  -49.512221   1946.0687
```

Scoring the data with this vector would create a new variable equal to the linear combination

$$1.7465592\,\texttt{weight} - 49.512221\,\texttt{mpg} + 1946.0687$$

The vector is interpreted as coefficients; the corresponding names of the variables are obtained from the column names (row names if `coefs` were a column vector). To form this linear combination, we type

```
. matrix score lc = coefs
. summarize lc
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
          lc |         74    6165.257    1597.606    3406.46   9805.269
```

If the coefficient vector has equation names, `matrix score` with the `eq()` option selects the appropriate coefficients for scoring. `eq(#1)` is assumed if no `eq()` option is specified.

```
. quietly sureg (price weight mpg) (displacement weight)
. matrix coefs = e(b)
. matrix list coefs
coefs[1,5]
          price:       price:       price:  displacem~t:  displacem~t:
         weight          mpg        _cons        weight         _cons
y1    1.7358275   -51.298248    2016.5101     .10574552    -121.99702
. matrix score lcnoeq = coefs
. matrix score lca = coefs , eq(price)
. matrix score lc1 = coefs , eq(#1)
. matrix score lcb = coefs , eq(displacement)
. matrix score lc2 = coefs , eq(#2)
. summarize lcnoeq lca lc1 lcb lc2
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
      lcnoeq |         74    6165.257    1598.264   3396.859   9802.336
         lca |         74    6165.257    1598.264   3396.859   9802.336
         lc1 |         74    6165.257    1598.264   3396.859   9802.336
         lcb |         74    197.2973    82.18474    64.1151   389.8113
         lc2 |         74    197.2973    82.18474    64.1151   389.8113
```

❑ Technical note

If the same equation name is scattered in different sections of the coefficient vector, the results may not be what you expect.

```
. matrix list bad
bad[1,5]
          price:       price:  displacem~t:        price:  displacem~t:
         weight          mpg       weight         _cons         _cons
y1    1.7358275   -51.298248    .10574552     2016.5101    -121.99702
. matrix score badnoeq = bad
. matrix score bada = bad , eq(price)
. matrix score bad1 = bad , eq(#1)
. matrix score badb = bad , eq(displacement)
. matrix score bad2 = bad , eq(#2)
. matrix score bad3 = bad , eq(#3)
```

```
. matrix score bad4 = bad , eq(#4)
. summarize bad*
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
     badnoeq |         74    4148.747    1598.264   1380.349   7785.826
        bada |         74    4148.747    1598.264   1380.349   7785.826
        bad1 |         74    4148.747    1598.264   1380.349   7785.826
        badb |         74    319.2943    82.18474   186.1121   511.8083
        bad2 |         74    319.2943    82.18474   186.1121   511.8083
-------------+---------------------------------------------------------
        bad3 |         74     2016.51           0    2016.51    2016.51
        bad4 |         74    -121.997           0   -121.997   -121.997
```

You do not need to worry about a bad matrix score when working with coefficient vectors created by Stata estimation commands. These commands always return coefficient vectors that are appropriately ordered according to equation names.

❑

Also see

[P] **matrix** — Introduction to matrix commands

[U] **14 Matrix expressions**

For suggested citations, see the FAQ on citing Stata documentation.