matrix score — Score data from coefficient vectors

Syntax

```
matrix score [type] newvar = b [if] [in]
[ , equation(#|eqname) missval(#) replace forcezero ]
```

where b is a 1 x p matrix.

Description

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

Options

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

```
misval(#) 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 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 http://www.stata-press.com/data/r13/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

\[
1.7465592 \text{ weight} - 49.512221 \text{ mpg} + 1946.0687
\]

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
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>lc</td>
<td>74</td>
<td>6165.257</td>
<td>1597.606</td>
<td>3406.46</td>
<td>9805.269</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th></th>
<th>price:</th>
<th>price:</th>
<th>price:</th>
<th>displacement:</th>
<th>displacement:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>weight</td>
<td>mpg</td>
<td>_cons</td>
<td>weight</td>
<td>_cons</td>
</tr>
<tr>
<td>y1</td>
<td>1.7358275</td>
<td>-51.298248</td>
<td>2016.5101</td>
<td>.10574552</td>
<td>-121.99702</td>
</tr>
</tbody>
</table>

```
.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
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcnoeq</td>
<td>74</td>
<td>6165.257</td>
<td>1598.264</td>
<td>3396.859</td>
<td>9802.336</td>
</tr>
<tr>
<td>lca</td>
<td>74</td>
<td>6165.257</td>
<td>1598.264</td>
<td>3396.859</td>
<td>9802.336</td>
</tr>
<tr>
<td>lc1</td>
<td>74</td>
<td>6165.257</td>
<td>1598.264</td>
<td>3396.859</td>
<td>9802.336</td>
</tr>
<tr>
<td>lcb</td>
<td>74</td>
<td>197.2973</td>
<td>82.18474</td>
<td>64.1151</td>
<td>389.8113</td>
</tr>
<tr>
<td>lc2</td>
<td>74</td>
<td>197.2973</td>
<td>82.18474</td>
<td>64.1151</td>
<td>389.8113</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th></th>
<th>price:</th>
<th>price:</th>
<th>displacement:</th>
<th>price:</th>
<th>displacement:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>weight</td>
<td>mpg</td>
<td>_cons</td>
<td>weight</td>
<td>_cons</td>
</tr>
<tr>
<td>y1</td>
<td>1.7358275</td>
<td>-51.298248</td>
<td>.10574552</td>
<td>2016.5101</td>
<td>-121.99702</td>
</tr>
</tbody>
</table>

```
.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*

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>badnoeq</td>
<td>74</td>
<td>4148.747</td>
<td>1598.264</td>
<td>1380.349</td>
<td>7785.826</td>
</tr>
<tr>
<td>bada</td>
<td>74</td>
<td>4148.747</td>
<td>1598.264</td>
<td>1380.349</td>
<td>7785.826</td>
</tr>
<tr>
<td>bad1</td>
<td>74</td>
<td>4148.747</td>
<td>1598.264</td>
<td>1380.349</td>
<td>7785.826</td>
</tr>
<tr>
<td>badb</td>
<td>74</td>
<td>319.2943</td>
<td>82.18474</td>
<td>186.1121</td>
<td>511.8083</td>
</tr>
<tr>
<td>bad2</td>
<td>74</td>
<td>319.2943</td>
<td>82.18474</td>
<td>186.1121</td>
<td>511.8083</td>
</tr>
<tr>
<td>bad3</td>
<td>74</td>
<td>2016.51</td>
<td>0</td>
<td>2016.51</td>
<td>2016.51</td>
</tr>
<tr>
<td>bad4</td>
<td>74</td>
<td>-121.997</td>
<td>0</td>
<td>-121.997</td>
<td>-121.997</td>
</tr>
</tbody>
</table>

Coefficient vectors created by Stata estimation commands will have equation names together.

Also see

[P] matrix — Introduction to matrix commands
[U] 14 Matrix expressions