Title

stata.com

select( ) — Select rows, columns, or indices

Syntax

transmorphic matrix `select(`*transmorphic matrix X*`,` *real vector v*`)`

void `st_select(`*A*`,` *transmorphic matrix X*`,` *real vector v*`)`

real vector `selectindex(`*real vector v*`)`

Description

`select(`*X*`,` *v*`)` returns *X*

1. omitting the rows for which *v*`[`*i*`]==0` (*v* a column vector) or
2. omitting the columns for which *v*`[`*j*`]==0` (*v* a row vector).

`st_select(`*A*`,` *X*`,` *v*`)` does the same thing, except that the result is placed in *A* and, if *X* is a view, *A* will be a view.

`selectindex(`*v*`)` returns

1. a row vector of column indices *j* for which *v*`[`*j*`]!=0` (*v* a row vector) or
2. a column vector of row indices *i* for which *v*`[`*i*`]!=0` (*v* a column vector).

Remarks and examples

stata.com

Remarks are presented under the following headings:

Examples

1. To select rows 1, 2, and 4 of 5 × *c* matrix `X`,

```
submat = select(X, (1\1\0\1\0))
```

   See [M-2] **subscripts** for another solution, `submat = X[(1\2\4), .]`.

2. To select columns 1, 2, and 4 of *r* × 5 matrix `X`,

```
submat = select(X, (1,1,0,1,0))
```

   See [M-2] **subscripts** for another solution, `submat = X[., (1,2,4)]`.

3. To select rows of X for which the first element is positive,

```
submat = select(X, X[.,1]:>0)
```

4. To select columns of X for which the first element is positive,

```
submat = select(X, X[1,.]:>0)
```

5. To select rows of X for which there are no missing values,

```
submat = select(X, rowmissing(X):==0)
```

6. To select rows and columns of square matrix X for which the diagonal elements are positive,

```
pos    = diagonal(X):>0
submat = select(X, pos)
submat = select(submat, pos')
```

or, equivalently,

```
pos    = diagonal(X):>0
submat = select(select(X, pos), pos')
```

7. To select column indices for which *v*[*j*]!=0,

```
: v
        1   2   3   4   5
    +-------------------------+
  1 |  6   0   7   0   8      |
    +-------------------------+

: selectindex(v)
        1   2   3
    +-------------+
  1 |  1   3   5  |
    +-------------+
```

8. To select row indices for which *v*[*i*]!=0,

```
: w
        1
    +-------+
  1 |  0    |
  2 |  3    |
  3 |  0    |
  4 |  2    |
  5 |  1    |
    +-------+

: selectindex(w)
        1
    +-------+
  1 |  2    |
  2 |  4    |
  3 |  5    |
    +-------+
```

Using st_select()

Coding

```
st_select(submat, X, v)                (1)
```

produces the same result as coding

```
submat = st_select(X, v)               (2)
```

The difference is in how the result is stored. If X is a view (it need not be), then (1) will produce `submat` as a view or, if you will, a subview, whereas in (2), `submat` will always be a regular (nonview) matrix.

When X is a view, (1) executes more quickly than (2) and produces a result that consumes less memory.

See [M-5] **st_view( )** for a description of views.

Conformability

`select(`X`, `v`)`:

X:	$r_1 \times c_1$			
v:	$r_1 \times 1$	or	$1 \times c_1$	
result:	$r_2 \times c_1$	or	$r_1 \times c_2$,	$r_2 \le r_1$, $c_2 \le c_1$

`st_select(`A`, `X`, `v`)`:

input:

X:	$r_1 \times c_1$			
v:	$r_1 \times 1$	or	$1 \times c_1$	

output:

A:	$r_2 \times c_1$	or	$r_1 \times c_2$,	$r_2 \le r_1$, $c_2 \le c_1$

`selectindex(`v`)`:

v:	$r_1 \times 1$	or	$1 \times c_1$	
result:	$r_2 \times 1$	or	$1 \times c_2$,	$r_2 \le r_1$, $c_2 \le c_1$

Diagnostics

None.

Also see

[M-5] **st_subview( )** — Make view from view

[M-2] **op_colon** — Colon operators

[M-2] **subscripts** — Use of subscripts

[M-4] **utility** — Matrix utility functions