select() — Select rows, columns, or indices

**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).

**Syntax**

```plaintext
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)
```

**Remarks and examples**

Remarks are presented under the following headings:

*Examples*

*Using st_select()*

**Examples**

1. To select rows 1, 2, and 4 of a \( 5 \times c \) matrix X,
   
   ```plaintext
   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 a \( r \times 5 \) matrix X,
   
   ```plaintext
   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,
   \[ \text{submat} = \text{select}(X, X[.,1] > 0) \]

4. To select columns of $X$ for which the first element is positive,
   \[ \text{submat} = \text{select}(X, X[1,.] > 0) \]

5. To select rows of $X$ for which there are no missing values,
   \[ \text{submat} = \text{select}(X, \text{rowmissing}(X) == 0) \]

6. To select rows and columns of square matrix $X$ for which the diagonal elements are positive,
   \[
   \begin{align*}
   \text{pos} & = \text{diagonal}(X) > 0 \\
   \text{submat} & = \text{select}(X, \text{pos}) \\
   \text{submat} & = \text{select}(\text{submat}, \text{pos}')
   \end{align*}
   \]

   or, equivalently,
   \[
   \begin{align*}
   \text{pos} & = \text{diagonal}(X) > 0 \\
   \text{submat} & = \text{select}(\text{select}(X, \text{pos}), \text{pos}')
   \end{align*}
   \]

7. To select column indices for which $v[j] \neq 0$,
   \[
   \begin{array}{cccccc}
   \hline
   v & 1 & 2 & 3 & 4 & 5 \\
   \hline
   1 & 6 & 0 & 7 & 0 & 8 \\
   \hline
   \end{array}
   \]
   \[ : \text{selectindex}(v) \]
   \[
   \begin{array}{c}
   1 \ 2 \ 3 \\
   \hline
   1 \ 1 \ 3 \ 5 \\
   \hline
   \end{array}
   \]

8. To select row indices for which $v[i] \neq 0$,
   \[
   \begin{array}{c}
   w \\
   \hline
   1 \\
   \hline
   1 \ 0 \\
   2 \ 3 \\
   3 \ 0 \\
   4 \ 2 \\
   5 \ 1 \\
   \hline
   \end{array}
   \]
   \[ : \text{selectindex}(w) \]
   \[
   \begin{array}{c}
   1 \\
   \hline
   1 \ 2 \\
   2 \ 4 \\
   3 \ 5 \\
   \hline
   \end{array}
   \]
Using `st_select()`

Coding

\[
\text{st_select(submat, X, v)} \quad (1)
\]

produces the same result as coding

\[
\text{submat = st_select(X, v)} \quad (2)
\]

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

\[
\text{select(X, v):}
\]

\[
\begin{align*}
X: & \quad r_1 \times c_1 \\
v: & \quad r_1 \times 1 \quad \text{or} \quad 1 \times c_1 \\
\text{result:} & \quad r_2 \times c_1 \quad \text{or} \quad r_1 \times c_2, \quad r_2 \leq r_1, c_2 \leq c_1
\end{align*}
\]

\[
\text{st_select(A, X, v):}
\]

\[
\begin{align*}
\text{input:} & \\
X: & \quad r_1 \times c_1 \\
v: & \quad r_1 \times 1 \quad \text{or} \quad 1 \times c_1 \\
\text{output:} & \\
A: & \quad r_2 \times c_1 \quad \text{or} \quad r_1 \times c_2, \quad r_2 \leq r_1, c_2 \leq c_1
\end{align*}
\]

\[
\text{selectindex(v):}
\]

\[
\begin{align*}
v: & \quad r_1 \times 1 \quad \text{or} \quad 1 \times c_1 \\
\text{result:} & \quad r_2 \times 1 \quad \text{or} \quad 1 \times c_2, \quad r_2 \leq r_1, c_2 \leq c_1
\end{align*}
\]

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