**st macroexpand() — Expand Stata macros in string**

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
<th>Remarks and examples</th>
<th>Conformability</th>
</tr>
</thead>
</table>

**Description**

`st macroexpand(s)` returns `s` with any quoted or dollar sign–prefixed macros expanded.

`_st macroexpand(S, s)` places in `S` the contents of `s` with any quoted or dollar sign–prefixed macros expanded and returns a Stata return code (it returns 0 if all went well).

**Syntax**

```
string scalar  st macroexpand(string scalar s)
real scalar    _st macroexpand(S, string scalar s)
```

Note: The type of `S` does not matter; it is replaced and becomes a string scalar.

**Remarks and examples**

Be careful coding string literals containing quoted or prefixed macros because macros are also expanded at compile time. For instance, consider

```
s = st macroexpand("regress ‘varlist’")
```

`‘varlist’` will be substituted with its value at compile time. What you probably want is

```
s = st macroexpand("regress " + "’" + "varlist" + "’")
```

**Conformability**

```
st macroexpand(s):
  s: 1 × 1
  result: 1 × 1

_st macroexpand(S, s):
  input:
    s: 1 × 1
  output:
    S: 1 × 1
    result: 1 × 1
```
Diagnostics

\texttt{st\_macroexpand(s)} aborts with error if \textit{s} is too long (exceedingly unlikely) or if macro expansion fails (also unlikely).

\texttt{\_st\_macroexpand(S, s)} aborts with error if \textit{s} is too long.

Also see

[M-4] \texttt{Stata} — Stata interface functions