

tsrevar — Time-series operator programming command

Description Remarks and examples	Quick start Stored results	Syntax Also see	Options
-------------------------------------	-------------------------------	--------------------	---------

Description

`tsrevar`, `substitute` takes a *varlist* that might contain *op.varname* combinations and substitutes equivalent temporary variables for the combinations.

`tsrevar`, `list` creates no new variables. It returns in `r(varlist)` the list of base variables corresponding to *varlist*.

Quick start

Create temporary variables containing the first lag and difference of `y` using `tsset` data, and store the temporary variable names in `r(varlist)`

```
tsrevar l.y d.y
```

Store the name of the base variable, `y`, in `r(varlist)`, and do not create any temporary variables

```
tsrevar l.y d.y, list
```

Syntax

```
tsrevar [varlist] [if] [in] [ , substitute list ]
```

You must `tsset` your data before using `tsrevar`; see [TS] `tsset`.

Options

`substitute` specifies that `tsrevar` resolve *op.varname* combinations by creating temporary variables as described above. `substitute` is the default action taken by `tsrevar`; you do not need to specify the option.

`list` specifies that `tsrevar` return a list of base variable names.

Remarks and examples

`tsrevar` substitutes temporary variables for any *op.varname* combinations in a variable list. For instance, the original *varlist* might be “`gnp L.gnp r`”, and `tsrevar`, `substitute` would create `newvar = L.gnp` and create the equivalent varlist “`gnp newvar r`”. This new varlist could then be used with commands that do not otherwise support time-series operators, or it could be used in a program to make execution faster at the expense of using more memory.

`tsrevar`, substitute might create no new variables, one new variable, or many new variables, depending on the number of *op.varname* combinations appearing in *varlist*. Any new variables created are temporary. The new, equivalent varlist is returned in `r(varlist)`. The new varlist corresponds one to one with the original *varlist*.

`tsrevar, list` returns in `r(varlist)` the list of base variable names of *varlist* with the time-series operators removed. `tsrevar, list` creates no new variables. For instance, if the original *varlist* were “`gnp l.gnp l2.gnp r l.cd`”, then `r(varlist)` would contain “`gnp r cd`”. This is useful for programmers who might want to create programs to keep only the variables corresponding to *varlist*.

▷ Example 1

```
. use http://www.stata-press.com/data/r15/tsrevar.ex
. tsrevar l.gnp d.gnp r
```

creates two temporary variables containing the values for `l.gnp` and `d.gnp`. The variable `r` appears in the new variable list but does not require a temporary variable.

The resulting variable list is

```
. display "r(varlist)"
__00014P __00014Q r
```

(Your temporary variable names may be different, but that is of no consequence.)

We can see the results by listing the new variables alongside the original value of `gnp`.

```
. list gnp 'r(varlist)' in 1/5
```

	gnp	__00014P	__00014Q	r
1.	128	.	.	3.2
2.	135	128	7	3.8
3.	132	135	-3	2.6
4.	138	132	6	3.9
5.	145	138	7	4.2

Temporary variables automatically vanish when the program concludes.

If we had needed only the base variable names, we could have specified

```
. tsrevar l.gnp d.gnp r, list
. display "r(varlist)"
gnp r
```

The order of the list will probably differ from that of the original list; base variables are listed only once and are listed in the order that they appear in the dataset.

□ Technical note

`tsrevar`, substitute avoids creating duplicate variables. Consider

```
. tsrevar gnp l.gnp r cd l.cd l.gnp
```

`l.gnp` appears twice in the varlist. `tsrevar` will create only one new variable for `l.gnp` and use that new variable twice in the resulting `r(varlist)`. Moreover, `tsrevar` will even do this across multiple calls:

```
. tsrevar gnp l.gnp cd l.cd
. tsrevar cpi l.gnp
```

`l.gnp` appears in two separate calls. At the first call, `tsrevar` creates a temporary variable corresponding to `l.gnp`. At the second call, `tsrevar` remembers what it has done and uses that same temporary variable for `l.gnp` again.

□

Stored results

`tsrevar` stores the following in `r()`:

```
Macros
  r(varlist)    the modified variable list or list of base variable names
```

Also see

[P] [syntax](#) — Parse Stata syntax

[P] [unab](#) — Unabbreviate variable list

[U] [11 Language syntax](#)

[U] [11.4.4 Time-series varlists](#)

[U] [18 Programming Stata](#)