Title

stata.com

tsreport — Report time-series aspects of a dataset or estimation sample

Description

`tsreport` reports time gaps in a dataset or in a subset of variables. By default, `tsreport` reports periods in which no information is recorded in the dataset; the time variable does not include these periods. When you specify *varlist*, `tsreport` reports periods in which either no information is recorded in the dataset or the time variable is present, but one or more variables in *varlist* contain a missing value.

Quick start

Report time gaps in a `tsset` time-series dataset

```
tsreport
```

Report time gaps for the variable `y`

```
tsreport y
```

Same as above, and report the beginning and ending times of each gap

```
tsreport y, detail
```

Report time gaps, ignoring panel changes, using `tsset` or `xtset` data

```
tsreport, panel
```

Menu

Statistics > Time series > Setup and utilities > Report time-series aspects of dataset

Syntax

```
tsreport [varlist] [if] [in] [, options]
```

options	Description
Main	
detail	list periods for each gap
casewise	treat a period as a gap if any of the specified variables are missing
panel	do not count panel changes as gaps

varlist may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
collect is allowed; see [U] **11.1.10 Prefix commands**.

Options

Main

detail reports the beginning and ending times of each gap.

casewise specifies that a period for which any of the specified variables are missing be counted as a gap. By default, gaps are reported for each variable individually.

panel specifies that panel changes not be counted as gaps. Whether panel changes are counted as gaps usually depends on how the calling command handles panels.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Basic examples
Video example

Basic examples

Time-series commands sometimes require that observations be on a fixed time interval with no gaps, or the command may not function properly. tsreport provides a tool for reporting the gaps in a sample.

▷ Example 1: A simple panel-data example

The following monthly panel data have two panels and a missing month (March) in the second panel:

```
. use https://www.stata-press.com/data/r18/tsrptxmpl
. list edlevel month income in 1/6, sep(0)
```

	edlevel	month	income
1.	1	1998m1	687
2.	1	1998m2	783
3.	1	1998m3	790
4.	2	1998m1	1435
5.	2	1998m2	1522
6.	2	1998m4	1532

Invoking `tsreport` gives us the following report:

```
. tsreport
Panel variable: edlevel
Time variable:  month
-------------------------
Starting period = 1998m1
Ending period   = 1998m4
Number of obs   =      6
Number of gaps  =      2 (includes panel changes)
```

Two gaps are reported in the sample. We know the second panel is missing the month of March, but where is the second gap? The note at the bottom of the output is telling us something about panel changes. Let's use the `detail` option to get more information:

```
. tsreport, detail
Panel variable: edlevel
Time variable:  month
-------------------------
Starting period = 1998m1
Ending period   = 1998m4
Number of obs   =      6
Number of gaps  =      2 (includes panel changes)
Gap report
```

Observations	edlevel	Start	End	N. obs
3	1	1998m4	.	.
5-6	2	1998m3	1998m3	1

We now see what is happening. `tsreport` is counting the change from the first panel to the second panel as a gap. Look at the output from the `list` command above. The value of `month` in observation 4 is not one month later than the value of `month` in observation 3, so `tsreport` reports a gap. (If we are programmers writing a procedure that does not account for panels, a change from one panel to the next represents a break in the time series just as a gap in the data does.) For the second gap, `tsreport` indicates that just one observation is missing because we are only missing the month of March. This gap is between observations 5 and 6 of the data.

In other cases, we may not care about changes in panels and not want them counted as gaps. We can use the `panel` option to specify that `tsreport` should ignore panel changes:

```
. tsreport, detail panel
Panel variable: edlevel
Time variable:  month
-------------------------
Starting period = 1998m1
Ending period   = 1998m4
Number of obs   =      6
Number of gaps  =      1
Gap report
```

Observations	edlevel	Start	End	N. obs
5-6	2	1998m3	1998m3	1

`tsreport` now indicates there is just one gap, corresponding to March for the second panel.

◁

Example 2: Variables with missing data

We asked two large hotels in Las Vegas to record the prices they were quoting people who called to make reservations. Because these prices change frequently in response to promotions and market conditions, we asked the hotels to record their prices hourly. Unfortunately, the managers did not consider us a top priority, so we are missing some data. Our dataset looks like this:

```
. use https://www.stata-press.com/data/r18/hotelprice
. list, sep(0)
```

	hour	price1	price2
1.	13feb2007 08:00:00	140	245
2.	13feb2007 09:00:00	155	250
3.	13feb2007 10:00:00	.	250
4.	13feb2007 11:00:00	155	250
5.	13feb2007 12:00:00	160	255
6.	13feb2007 13:00:00	.	.
7.	13feb2007 14:00:00	165	255
8.	13feb2007 15:00:00	170	260
9.	13feb2007 16:00:00	175	265
10.	13feb2007 17:00:00	180	.
11.	13feb2007 20:00:00	190	270

First, let's invoke `tsreport` without specifying `price1` or `price2`. We will specify the `detail` option so that we can see the periods corresponding to the gap or gaps reported:

```
. tsreport, detail
Time variable: hour
-------------------------------------
Starting period = 13feb2007 08:00:00
Ending period   = 13feb2007 20:00:00
Number of obs   =                  11
Number of gaps  =                   1
Gap report
```

Observations	Start	End	N. obs
10-11	13feb2007 18:00:00	13feb2007 19:00:00	2

One gap is reported, lasting two periods. We have no data corresponding to 6:00 p.m. and 7:00 p.m. on February 13, 2007.

What about observations 3, 6, and 10? We are missing data on one or both of the price variables for those observations, but the time variable itself is present for those observations. By default, `tsreport` defines gaps as periods in which no information, not even the time variable itself, is recorded.

If we instead want to obtain information about when one or more variables are missing information, then we specify those variables in our call to `tsreport`. Here we specify `price1`, first without the `detail` option:

```
. tsreport price1
Gap summary report
```

Variable	Start	End	Number of obs	gaps
price1	13feb2007 08:00:00	13feb2007 20:00:00	9	3

The output indicates that we have data on price1 from 8:00 a.m. to 8:00 p.m. However, we only have 9 observations on price1 during that span because we have 3 gaps in the data. Let's specify the detail option to find out where:

```
. tsreport price1, detail
Variable: price1
Time variable: hour
-------------------------------------
Starting period = 13feb2007 08:00:00
Ending period   = 13feb2007 20:00:00
Number of obs   =                   9
Number of gaps  =                   3
Gap report
```

Observations	Start	End	N. obs
2–4	13feb2007 10:00:00	13feb2007 10:00:00	1
5–7	13feb2007 13:00:00	13feb2007 13:00:00	1
10–11	13feb2007 18:00:00	13feb2007 19:00:00	2

The three gaps correspond to observations 3 and 6, for which price1 is missing, as well as the two-period gap in the evening when not even the time variable is recorded in the dataset.

◁

When you specify multiple variables with tsreport, by default, it summarizes gaps in each variable separately. Apart from combining the information into one table, typing

```
. tsreport price1 price2
```

is almost the same as typing

```
. tsreport price1
. tsreport price2
```

The only difference between the two methods is that the former stores results for both variables in r-class macros for later use, whereas if you were to type the latter two commands in succession, r-class macros would only contain results for price2.

In many types of analyses, including linear regression, you can only use observations for which all the variables contain nonmissing data. Similarly, you can have tsreport report as gaps periods in which any of the specified variables contain missing values. To do that, you use the casewise option.

▷ Example 3: Casewise analyses

Continuing with our hotel data, we specify both price1 and price2 in the variable list of tsreport. We request casewise analysis, and we specify the detail option to get information on each gap tsreport finds.

```
. tsreport price1 price2, casewise detail
Variables: price1 price2
Time variable: hour
-------------------------------------
Starting period = 13feb2007 08:00:00
Ending period   = 13feb2007 20:00:00
Number of obs   =                  8
Number of gaps  =                  3
Gap report
```

Observations	Start	End	N. obs
2-4	13feb2007 10:00:00	13feb2007 10:00:00	1
5-7	13feb2007 13:00:00	13feb2007 13:00:00	1
9-11	13feb2007 17:00:00	13feb2007 19:00:00	3

The first gap reported by `tsreport` corresponds to observation 3, when `price1` is missing, and the second gap corresponds to observation 6, when both `price1` and `price2` are missing. The third gap spans 3 observations: the 5:00 p.m. observation is missing for `price2`, and as we discovered earlier, not even the time variable is present at 6:00 p.m. and 7:00 p.m.

◁

Video example

Formatting and managing dates

Stored results

`tsreport`, when no *varlist* is specified or when `casewise` is specified, stores the following in `r()`:

Scalars

`r(N_gaps)`	number of gaps
`r(N_obs)`	number of observations
`r(start)`	first time in series
`r(end)`	last time in series

Macros

`r(panelvar)`	name of panel variable
`r(timevar)`	name of time variable
`r(tsfmt)`	%*fmt* of time variable

Matrices

`r(table)`	matrix containing start and end times of each gap, if `detail` is specified

`tsreport`, when a *varlist* is specified and `casewise` is not specified, stores the following in `r()`:

Scalars

`r(N_gaps#)`	number of gaps for variable #
`r(N_obs#)`	number of observations for variable #
`r(start#)`	first time in series for variable #
`r(end#)`	last time in series for variable #

Macros

`r(panelvar)`	name of panel variable
`r(timevar)`	name of time variable
`r(tsfmt)`	*%fmt* of time variable
`r(var#)`	name of variable #

Matrices

`r(table#)`	matrix containing start and end times of each gap for variable #, if `detail` is specified

When k variables are specified in *varlist*, # ranges from 1 to k.

Also see

[TS] **tsset** — Declare data to be time-series data