

xpose — Interchange observations and variables[Description](#)[Options](#)[Also see](#)[Quick start](#)[Remarks and examples](#)[Menu](#)[Methods and formulas](#)[Syntax](#)[References](#)

Description

`xpose` transposes the data, changing variables into observations and observations into variables. All new variables—that is, those created by the transposition—are made the default storage type. Thus any original variables that were strings will result in observations containing missing values. (If you transpose the data twice, you will lose the contents of string variables.)

Quick start

Replace dataset in memory with transposed variables and observations

```
xpose, clear
```

Add `_varname` containing the original variable names

```
xpose, clear varname
```

Use the most compact data type that preserves accuracy in the transposed data

```
xpose, clear promote
```

Menu

Data > Create or change data > Other variable-transformation commands > Interchange observations and variables

Syntax

```
xpose, clear [options]
```

<i>options</i>	Description
* clear	reminder that untransposed data will be lost if not previously saved
format	use largest numeric display format from untransposed data
format(<i>%fmt</i>)	apply specified format to all variables in transposed data
varname	add variable <code>_varname</code> containing original variable names
promote	use the most compact data type that preserves numeric accuracy

* `clear` is required.

Options

`clear` is required and is supposed to remind you that the untransposed data will be lost (unless you have saved the data previously).

`format` specifies that the largest numeric display format from your untransposed data be applied to the transposed data.

`format(%fmt)` specifies that the specified numeric display format be applied to all variables in the transposed data.

`varname` adds the new variable `_varname` to the transposed data containing the original variable names. Also, with or without the `varname` option, if the variable `_varname` exists in the dataset before transposition, those names will be used to name the variables after transposition. Thus transposing the data twice will (almost) yield the original dataset.

`promote` specifies that the transposed data use the most compact numeric data type that preserves the original data accuracy.

If your data contain any variables of type `double`, all variables in the transposed data will be of type `double`.

If variables of type `float` are present, but there are no variables of type `double` or `long`, the transposed variables will be of type `float`. If variables of type `long` are present, but there are no variables of type `double` or `float`, the transposed variables will be of type `long`.

Remarks and examples

[stata.com](http://www.stata.com)

► Example 1

We have a dataset on something by county and year that contains

```
. use http://www.stata-press.com/data/r15/xposexmpl
. list
```

	county	year1	year2	year3
1.	1	57.2	11.3	19.5
2.	2	12.5	8.2	28.9
3.	3	18	14.2	33.2

Each observation reflects a county. To change this dataset so that each observation reflects a year, type

```
. xpose, clear varname
. list
```

	v1	v2	v3	_varname
1.	1	2	3	county
2.	57.2	12.5	18	year1
3.	11.3	8.2	14.2	year2
4.	19.5	28.9	33.2	year3

We would now have to drop the first observation (corresponding to the previous county variable) to make each observation correspond to one year. Had we not specified the `varname` option, the variable `_varname` would not have been created. The `_varname` variable is useful, however, if we want to transpose the dataset back to its original form.

```
. xpose, clear
. list
```

	county	year1	year2	year3
1.	1	57.2	11.3	19.5
2.	2	12.5	8.2	28.9
3.	3	18	14.2	33.2

4

Methods and formulas

See [Hamilton \(2013, chap. 2\)](#) for an introduction to Stata's data management features.

References

- Baum, C. F. 2016. *An Introduction to Stata Programming*. 2nd ed. College Station, TX: Stata Press.
- Hamilton, L. C. 2013. *Statistics with Stata: Updated for Version 12*. 8th ed. Boston: Brooks/Cole.

Also see

- [D] [reshape](#) — Convert data from wide to long form and vice versa
- [D] [stack](#) — Stack data