Stata commands currently supported by ACRO

We focus on Stata V17 and above as the syntax changed significantly from then, but support for Stata V16 and below is broadly the same.

This is an ongoing project we want to be driven by researchers' needs

So **please let us know** if there are commands or things you would like us to support: - by emailing sacro.contact@uwe.ac.uk - or adding an issue to github.com/ai-sdc/acro

Author james.smith@uwe.ac.uk

Last Updated 30 October 2025

Introduction and General Concepts

Aim

ACRO is designed to provide drop-in replacements for commonly-used Stata commands that add functionality to let researchers and TRE staff fulfil their collective duty to produce *Safe Outputs*.

Most data management is done exactly as before, the difference comes when you issue a *query* that produces an output you may want to take out of the TRE.

ACRO provides automated checks against risks you should be checking such as: *small group sizes* , *class disclosure*, *dominance* and *low residual degrees of freedom*.

Researchers are provided with this feedback immediately so they have the power to choose between: - **redesigning** their outputs to reduce risks, - **applying a mitigation strategy** (e.g. acro can provide automated suppression of table cells with low counts and adjust row/column totals), - **requesting an exception** to strict rules-based checking.

ACRO *sessions*

They key concept is that of an acro *session* : comparable to doing a day's work and requesting some outputs at the end. - Of course you can save your work at any stage using normal Stata functionality. - We anticipate that people may want to do preparatory work, and then add *acro* prefixes to queries, and session commands to their code and do a final run through.

Caveat

At present most acro-assessed outputs are saved and egressed as csv files or image files.

- So **we currently assume you will format your results later** e.g. once they have been released from the TRE and are producing your publication.
- We have open issues to support more formatted work - and welcome feedback on that.

ACRO session management commands

`acro init`: *starts an acro session*

`acro custom_output`: *adds a custom output i.e. code, or a paper version, or images of plots or results from analyses we can't auto-check (yet)*

`acro enable_suppression`: *turns on automatic suppression for subsequent outputs*

`acro disable_suppression`: *turns off automatic suppression for subsequent outputs.*

`acro print_outputs`: *prints list to screen of outputs in current session and their status*

`acro remove_output output_id`: *removes a named output from an acro session*

`acro rename_output output_id new_name`: *renames a named output from an acro session*

`acro add_comments output_id comment_string`: *adds a comment (string) to a named outputn*

`acro add_exception output_id exception_string`: *adds an exception request (string) comment to a named output*

`acro finalise [output_dir] [filetype]`: *wraps up the session, writing the outputs to a named directory (default= "stata_output") in json (default) or xlsx format.*

ACRO commands for creating tables

For Stata versions V17 and beyond you can make tables of frequencies (or other statistics) using Stata' *table* command.

`acro table (rowvars) (colvars) (tabvars) [if] [in] [,options]`

See Stata's built-in help for details of how to use this, **and there are some simple examples at the end of this document**

There are some minor differences, which will be addressed in the next version: - you can currently report on the following statistics: *count* (default, same as frequency), *mean*, *median*, *sum*, *std*, and *mode*. - weights are not currently supported within the options - formatting commands are not currently supported within the options (see above).

ACRO commands for creating regression models

`acro regress dependent_ var independent_vars [if] [in]`: *linear regression*

`acro logit dependent_ var independent_vars [if] [in]`: *logistic regression*

`acro probit dependent_ var independent_vars [if] [in]`: *probit regression*

Some simple examples of creating tables

All of these examples are using the standard Stata syntax with the prefix *acro*

These examples assume we have a simple data set with: - a numerical variable *n*, - and five categorical variables (*a,b,c,d,e*). - **For simplicity** the examples below assume each categorical variable has 2 possible values. - e.g. *a1* or *a2*.

Simple two-way table of frequencies:

`acro table a b` produces

	b1	b2
a1		
a2		

where the cells contain the number of records falling into each subgroup.

Two way table reporting statistics of variable *n*

`acro table a b, statistic(mean n)`

produces the table with the shape

	b1	b2
a1		
a2		

but now the cells will contain the mean value of n in each sub-group. See above for aggregation functions we currently support

You can request more than one statistic - but they must refer to the same numerical variable. For example: `acro table a b, statistic(mean n) statistic(std n)` produces

	mean		std	
	b1	b2	b1	b2
a1				
a2				

Adding hierarchies and complexity to tables

Add row or columns hierarchies by listing variables to group within parentheses.

As above, the cells will contain sub-group counts be default and other statistics on request

`acro table (a b) c` produces

		c1	c2
a1	b1		
	b2		
a2	b1		
	b2		

`acro table a (b c)` produces

	b1		b2	
	c1	c2	c1	c2
a1				
a2				

and `acro table (a b) (c d)` produces

		c1		c2	
		d1	d2	d1	d2
a1	b1				
	b2				
a2	b1				
	b2				

Creating multiple related tables

There may be cases where you want to separate your data out - for example if it makes the tables simpler to read

You can do this by providing a third (set of) variables alongside the row and column specifiers.

For examples `acro table a b c` produces:

c1		
	b1	b2
a1		
a2		

c2		
	b1	b2
a1		
a2		

Even more complexity ...

As before, in multiple tables the rows or columns could contain hierarchies, and you can specify different statistics to report on.

So you could combine our previous examples e.g.: `acro table (a b) (c d) e , statistic(mean n) statistic(std n)` and so on.