

Stata Cheat Sheet

The core commands for this course

Prof. Jack Reilly

Fall 2026

A short reference for the Stata commands we actually use in PAI 721. Every command has more options than shown here; type `help` followed by any command name (for example, `help summarize`) to see them all. Words like `var`, `group`, `y`, and `x` are placeholders for your own variable names.

Don't forget - for a more extensive references, you also have:

- Solutions to every problem set (course drive) and code embedded in class slides
- The [UCLA Stata guide](#), linked in the syllabus
- The Acock "Gentle Introduction to Stata" optional/recommended reference book

Setup & Files

Command	What it does
<code>cd "folder"</code>	Set the working directory (where Stata looks for files)
<code>use "file.dta", clear</code>	Open a Stata dataset, closing whatever is already open
<code>import delimited "file.csv", clear</code>	Open a .csv file
<code>save "file.dta", replace</code>	Save the data in memory, overwriting the old copy
<code>clear</code>	Wipe the data in memory
<code>browse</code>	Open the spreadsheet view of your data
<code>describe</code>	List the variables in the dataset

`.do` files

A `.do` file is a plain-text list of commands that Stata runs top to bottom. Open the editor with `doedit`. Lines starting with `*` or `//` are comments; use them to explain what you're doing and to label which question each chunk of code answers. Put `///` at the end of a line to continue a long command onto the next line.

Looking at Data

Command	What it does
<code>sum var</code>	Mean, standard deviation, min, max, and n (summarize)
<code>sum var, detail</code>	Adds the median and percentiles (25th = Q1, 75th = Q3)
<code>tab var</code>	Frequency table: how many cases take each value
<code>tab var1 var2</code>	Cross-tabulation of two variables
<code>tabstat var, stats(mean sd n)</code>	Choose your own summary statistics
<code>tabstat var, by(group)</code>	The same, separately for each group
<code>bys group: sum var</code>	Run any command separately for each group
<code>corr var1 var2</code>	Correlation coefficient
<code>sort var</code>	Sort cases from low to high (gsort -var for high to low)

Creating & Changing Variables

Command	What it does
<code>gen new = pts / g</code>	Create a new variable from a formula
<code>replace var = 0 if var == 9</code>	Change values that meet a condition
<code>recode var 7=., gen(newvar)</code>	Recode values into a new variable (here, 7 becomes missing)
<code>keep if year == 2010</code>	Keep only the cases that meet a condition
<code>drop if year == 1996</code>	Drop the cases that meet a condition
<code>keep var1 var2</code>	Keep only these <i>variables</i> (drop var1 to drop one)
<code>sample 20</code>	Keep a random 20% of cases (sample 50, count for exactly 50)
<code>set seed 12345</code>	Run before sample so your random draw comes out the same each time
<code>collapse (mean) var, by(group)</code>	Replace the data with one row per group
<code>preserve ... restore</code>	Take a snapshot of the data, do something to it, then get the original back

Conditions use == (equals), != (not equal), >, <, >=, <=, & (and), and | (or). Missing values are stored as . and Stata treats them as *larger than any number*, so keep if x > 10 also keeps the missings. Add & x < . to leave them out.

Graphs

Command	What it does
<code>histogram var</code>	Histogram of a continuous variable
<code>histogram var, discrete percent</code>	Histogram of a variable with only a few whole-number values
<code>graph bar (count), over(var)</code>	Bar chart of a categorical variable
<code>scatter y x</code>	Scatterplot
<code>scatter y x lfit y x</code>	Scatterplot with a best-fit line
<code>twoway (scatter y x) (lfit y x)</code>	Scatterplot with a best-fit line (alternate)
<code>graph box y, over(group)</code>	Box plots, one per group
<code>graph export "plot.png", replace</code>	Save the most recent graph to a file

Add titles and axis labels to any graph with options at the end of the command: `, title("...") xtitle("...") ytitle("...")`. Doing it in the command rather than the graph editor means it ends up in your `.do` file.

Regression & Tests

Command	What it does
<code>reg y x</code>	Regress <i>y</i> on <i>x</i> (regress)
<code>reg y x1 x2</code>	Multiple regression
<code>predict yhat</code>	After <code>reg</code> : the predicted value for every case
<code>predict resid, residuals</code>	After <code>reg</code> : the residual for every case
<code>ci means var</code>	Confidence interval for a mean (add <code>, level(90)</code> for 90%)
<code>ttest var == 30</code>	One-sample t-test against a specific number
<code>ttest var, by(group)</code>	Difference-in-means t-test (add <code>unequal</code> if the spreads differ)
<code>tab var1 var2, chi2 expected</code>	Chi-square test on a crosstab, showing expected counts
<code>tabi 16 4 \ 10 9 \ 6 18, chi2</code>	Chi-square test on a table you type in (\ starts a new row)

Stata as a Calculator

display (or just di) prints the result of any expression, so you can do hand calculations without leaving Stata.

Command	What it does
di (1820 - 1509) / 321	Arithmetic (here, a z-score)
di normal(0.97)	Area to the <i>left</i> of $z = 0.97$ under the Normal curve
di 1 - normal(0.97)	Area to the <i>right</i> of $z = 0.97$
di invnormal(0.975)	The z-score with 97.5% of the area below it (1.96)
di invttail(35, 0.025)	Critical t with $df = 35$ and 2.5% in the upper tail (for a 95% CI)
di sqrt(36)	Square root
di _b[x]	After <code>reg</code> : the slope on x (<code>_b[_cons]</code> is the intercept)
