**Description**

`cumul` creates `newvar`, defined as the empirical cumulative distribution function of `varname`.

**Quick start**

Create new variable `ecd` containing the empirical cumulative distribution of `v`

```stata
cumul v, gen(ecd)
```

Use frequency as the unit for `v` to generate `ecdf`

```stata
cumul v, gen(ecdf) freq
```

Give equal values of `v` the same value in generated `ecde`

```stata
cumul v, gen(ecde) equal
```

Graph the empirical cumulative distribution of `v`

```stata
line ecd v, sort
```

Graph the distributions of variables `v1` and `v2`

```stata
cumul v1, gen(ecd1) equal
cumul v2, gen(ecd2) equal
stack ecd1 v1 ecd2 v2, into(ecd v) wide clear
line ecd1 ecd2 v, sort
```

**Menu**

Statistics > Summaries, tables, and tests > Distributional plots and tests > Generate cumulative distribution
cumul — Cumulative distribution

Syntax

cumul varname [if] [in] [weight], generate(newvar) [options]

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td></td>
</tr>
<tr>
<td>*generate(newvar)</td>
<td>create variable newvar</td>
</tr>
<tr>
<td>freq</td>
<td>use frequency units for cumulative</td>
</tr>
<tr>
<td>equal</td>
<td>generate equal cumulatives for tied values</td>
</tr>
</tbody>
</table>

*generate(newvar) is required.
by is allowed; see [D] by.
aweights are allowed; see [U] 11.1.6 weight.

Example 1

cumul is most often used with graph to graph the empirical cumulative distribution. For instance, we have data on the median family income of 957 U.S. cities:

use https://www.stata-press.com/data/r16/hsng
(1980 Census housing data)
cumul faminc, gen(cum)
sort cum
line cum faminc, ylab(, grid) ytitle("") xlab(, grid)
> title("Cumulative of median family income")
> subtitle("1980 Census, 957 U.S. Cities")
It would have been enough to type `line cum faminc`, but we wanted to make the graph look better; see [G-2] *graph twoway line*.

If we had wanted a weighted cumulative, we would have typed `cumul faminc [w=pop]` at the first step.

**Example 2**

To graph two (or more) cumulatives on the same graph, use `cumul` and `stack`; see [D] *stack*. For instance, we have data on the average January and July temperatures of 956 U.S. cities:

```
. use https://www.stata-press.com/data/r16/citytemp, clear
   (City Temperature Data)
. cumul tempjan, gen(cjan)
. cumul tempjuly, gen(cjuly)
. stack cjan tempjan cjuly tempjuly, into(c temp) wide clear
. line cjan cjuly temp, sort ylab(, grid) ytitle("") xlab(, grid)
   > xtitle("Temperature (F)")
   > title("Cumulatives: " "Average January and July Temperatures")
   > subtitle("956 U.S. Cities")
```
As before, it would have been enough to type `line cjan c July temp, sort`. See `[D] stack` for an explanation of how the `stack` command works.

**Technical note**

According to Beniger and Robyn (1978), Fourier (1821) published the first graph of a cumulative frequency distribution, which was later given the name “ogive” by Galton (1875).

Jean Baptiste Joseph Fourier (1768–1830) was born in Auxerre in France. As a young man, Fourier became entangled in the complications of the French Revolution. As a result, he was arrested and put into prison, where he feared he might meet his end at the guillotine. When he was not in prison, he was studying, researching, and teaching mathematics. Later, he served Napoleon’s army in Egypt as a scientific adviser. Upon his return to France in 1801, he was appointed Prefect of the Department of Isère. While prefect, Fourier worked on the mathematical basis of the theory of heat, which is based on what are now called Fourier series. This work was published in 1822, despite the skepticism of Lagrange, Laplace, Legendre, and others—who found the work lacking in generality and even rigor—and disagreements of both priority and substance with Biot and Poisson.

**Acknowledgment**

The `equal` option was added by Nicholas J. Cox of the Department of Geography at Durham University, UK, and coeditor of the *Stata Journal* and author of *Speaking Stata Graphics*.
References


Also see

[R] **Diagnostic plots** — Distributional diagnostic plots

[R] **kdensity** — Univariate kernel density estimation

[D] **stack** — Stack data