



The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

Stata tip 143: Creating donut charts in Stata

Andrew Musau
Molde University College
Molde, Norway
amus@himolde.no

Pie charts are circular charts cut by radii into segments illustrating relative magnitudes or frequencies. In Stata, one can create this graphic using the **graph pie** command (see [G-2] **graph pie**). A closely related graphic is the donut chart, essentially a pie chart with an area of the center cut out. The argument for pie charts or donut charts seems simple and can be summarized into the following three points:

- i. They are widely used and widely familiar. In many places, pie charts are taught at very early ages. Familiar beats unusual, with nothing else said.
- ii. The general idea of subdividing a total is very easy to grasp with a pie chart. Everyone can think of slices of a pie, or cake if you prefer, even small children or senior managers.
- iii. The specific logic of a pie chart is that slice (more professionally, sector) angle, area, and arc length on the circumference are all proportional to quantity shown. The exception would be if pie slices are themselves divided radially, which gets very messy very quickly.

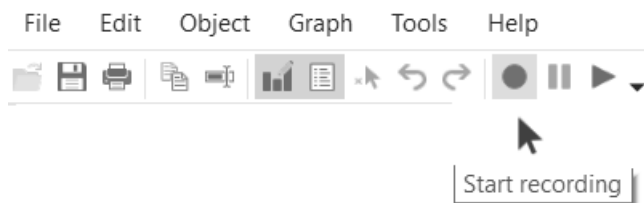
Conversely, the arguments against these graphics boil down to one argument: that other designs, bar or dot charts, work better and are not more difficult to understand. For a review of bar and dot charts and Stata techniques for producing them, see Cox (2008).¹ As of Stata 17, donut charts are not officially supported. However, there is some demand for this graphic by Stata users, as evidenced by posts on Statalist. Here we show that by first creating a pie chart, one can subsequently retrieve code generated from the Graph Editor (see [G-1] **Graph Editor**) to create a donut chart. Going through this process will not be necessary, because the code that I present here will directly create the chart, but the intermediate steps are set forth for replicability. For illustration, we will use **census.dta**, where we show proportions of population across census regions of the United States.

```
. sysuse census
(1980 Census data by state)
. graph pie pop, over(region) scheme(sj) legend(off)
```

1. I thank Nick Cox and an anonymous referee for detailed and constructive comments and suggestions.

The illustration is based on Windows 10 (Stata 17) but can be adapted for other operating systems. After running the code creating the pie chart, do the following:

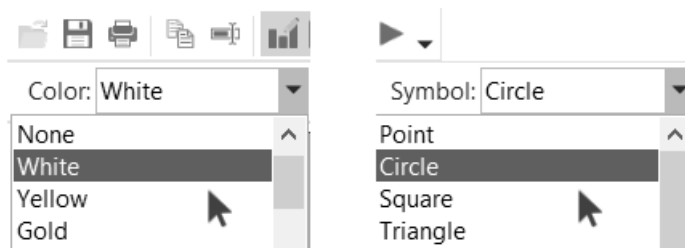
1. Right-click on the pie chart, and click on **Start Graph Editor**.
2. On the Standard Toolbar, click on the **Start recording** button.



3. On the Tools Toolbar, click on **Add marker** button.



4. On the Contextual Toolbar, choose **White** as the *Color* and **Circle** as the *Symbol*.



5. Place the marker at the center of the pie chart by placing the cursor there and clicking.
6. On the Standard Toolbar, click on the **End recording** button (same button as **Start recording** in number 2).
7. You will be prompted to provide a *Recording name*;, and you can choose where you want it to be saved.

The maximum area of the inner circle that can be specified using the Graph Editor is about a tenth of the area of the pie chart, not sufficient to create a donut chart. Therefore, we must modify the code retrieved from the Graph Editor to increase the inner circle's area. The recording is saved with the file extension `.grec`. On a Windows PC, use Notepad to open and access the code. The undocumented command `gr_edit` can be used to run lines of code retrieved from the Graph Editor. Below, we present a do-file code where we specify the size of the inner circle as a percentage of the pie chart's area in a local macro named `csize`. Note that a macro is simply a container holding text and that the preceding word `local` implies that the macro exists solely within the do-file or program in which it is defined (see [U] **18 Programming Stata** and Cox [2020] for an introduction to macros in Stata). Because of the existence of the local macro and the length of the code derived from the Graph Editor resulting in multiple line breaks, it is necessary to always run the entire do-file code in one go.

```
. local csize= 70
. gr_edit .plotregion1.AddMarker added_markers editor
> .0715713654788885 -.0361049561829105
. gr_edit .plotregion1.added_markers_new = 1
. gr_edit .plotregion1.added_markers_rec = 1
. gr_edit .plotregion1.added_markers[1].style.editstyle marker( symbol(circle)
> linestyle( width( sztype(relative) val(.2) allow_pct(1)) color(white)
> pattern(solid) align(inside)) fillcolor(white) size( sztype(relative)
> val('csize'*(93/100)) allow_pct(1)) angle(stdarrow) symangle(zero)
> backsymbol(none) backline( width( sztype(relative) val(.2) allow_pct(1))
> color(black) pattern(solid) align(inside)) backcolor(black)
> backsize( sztype(relative) val(0) allow_pct(1)) backangle(stdarrow)
> backsymangle(zero)) line( width( sztype(relative) val(.2) allow_pct(1))
> color(black) pattern(solid) align(inside)) area( linestyle( width(
> sztype(relative) val(.2) allow_pct(1)) color(ltblueishgray) pattern(solid)
> align(inside)) shadestyle( color(ltblueishgray) intensity(inten100)
> fill(pattern10))) label( textstyle( horizontal(center) vertical(middle)
> angle(default) size( sztype(relative) val(2.777) allow_pct(1)) color(black)
> position() margin( gleft( sztype(relative) val(0) allow_pct(1))
> gright( sztype(relative) val(0) allow_pct(1)) gtop( sztype(relative)
> val(0) allow_pct(1)) gbottom( sztype(relative) val(0) allow_pct(1)))
> linestyle( width( sztype(relative) val(.2) allow_pct(1)) color(black)
> pattern(solid) align(inside))) position(6) textgap( sztype(relative)
> val(.6944) allow_pct(1)) format('""') horizontal(default) vertical(default))
> dots( symbol(circle) linestyle( width( sztype(relative) val(.2) allow_pct(1))
> color(black) pattern(solid) align(inside)) fillcolor(black)
> size( sztype(relative) val(.1) allow_pct(1)) angle(horizontal) symangle(zero)
> backsymbol(none) backline( width( sztype(relative) val(.2) allow_pct(1))
> color(black) pattern(solid) align(inside)) backcolor(black)
> backsize( sztype(relative) val(1.52778) allow_pct(1)) backangle(horizontal)
> backsymangle(zero)) connect(direct) connect_missings(yes) editcopy
```

The graph obtained after running the do-file code from the previously created pie chart is shown in figure 0.1.

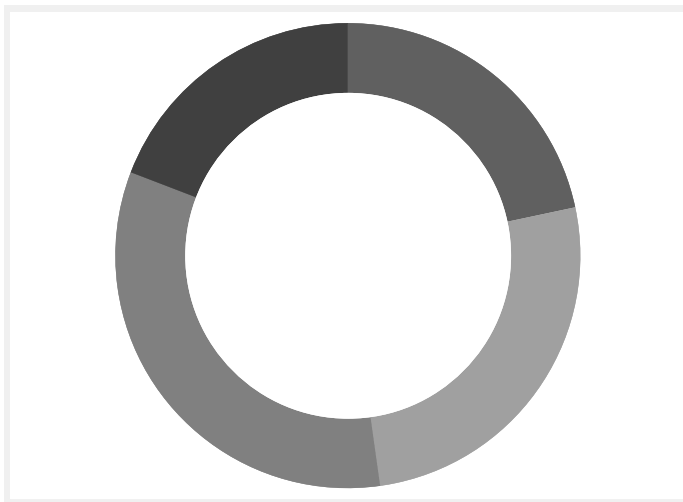


Figure 0.1. Donut plot without proportions and labels displayed

Because the inner circle is placed on top of the pie chart, it becomes apparent that proportions and labels must be placed outside the inner circle within the pie chart or completely outside the pie chart to be visible. Stata allows one to explicitly include either category labels or proportions. If proportions are included, category labels are shown in the legend, and if category labels are included, then proportions are not shown and one forgoes the legend.

Figure 0.2 illustrates a variant of each case. The `gap()` suboption specified within `graph pie`'s `plabel()` option determines the additional radial distance for the labels to appear on the pie chart's slices. When combining charts, note that the command `graph combine` (see [G-2] `graph combine`) does not shrink the sizes of the individual donut charts' inner circles; therefore, depending on how many charts need to be combined, one has to specify an appropriate value for the local `csize` in the do-file code. For the pie charts creating figure 0.2, we specify `csize=60`, just over 10% less than the value specified for figure 0.1.

```
. graph pie pop, over(region) scheme(sj)
> legend(ring(0) pos(2) bmargin(zero) col(1))
> plabel(_all percent, gap(10) color(white) size(medlarge))
. * run do-file code, specifying csize=60
  (output omitted)
. graph save gr1, replace
(file gr1.gph not found)
file gr1.gph saved
```

```

. graph pie pop, over(region) scheme(sj)
> legend(ring(0) pos(5) bmargin(zero) col(1))
> plabel(_all percent, gap(22) size(medlarge))
. * run do-file code, specifying csize=60
  (output omitted)
. graph save gr2, replace
(file gr2.gph not found)
file gr2.gph saved
. graph combine gr1.gph gr2.gph, scheme(sj)

```

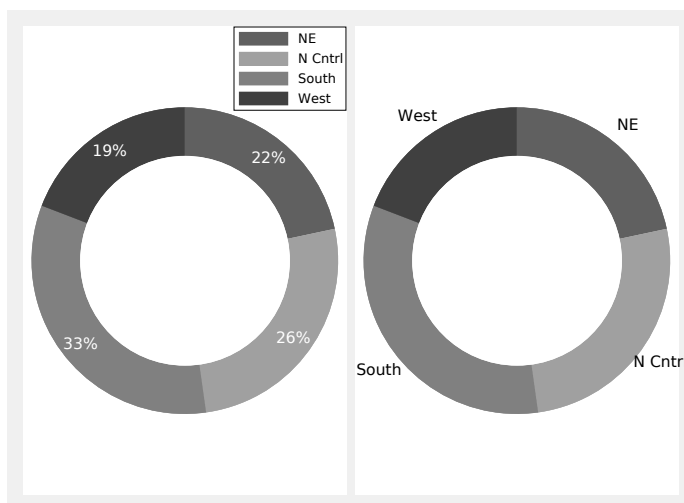


Figure 0.2. Combined donut plots with legends and proportions displayed

Having both proportions and category labels displayed requires that one insert each manually using the `ptext()` option.

References

- Cox, N. J. 2008. Speaking Stata: Between tables and graphs. *Stata Journal* 8: 269–289. <https://doi.org/10.1177/1536867X0800800208>.
- . 2020. Stata tip 138: Local macros have local scope. *Stata Journal* 20: 499–503. <https://doi.org/10.1177/1536867X20931028>.