Learning to Create and Modify Pie Charts with Stata: A Step-by-Step Guide

Authored by

Mohammed looti

November 8, 2025

The Role of Pie Charts in Statistical Analysis

A pie chart is a fundamental graphical representation tool in statistics, specifically designed to display the proportional distribution of **categorical data**. This intuitive circular chart divides a dataset into "slices," where the area of each slice is mathematically proportional to the quantity it represents. By illustrating the relative sizes of different components within a whole, pie charts offer an immediate visual understanding of distribution.

While simple in concept, effective generation and customization are crucial for professional reports. This comprehensive guide, tailored for users of Stata, the powerful statistical software package, explains in detail how to effectively generate, customize, and refine pie charts for superior data visualization.

Data Preparation and Loading the Census Dataset

The first step in any visualization process is ensuring the data is correctly loaded and structured. For this tutorial, we will utilize a built-in dataset provided standard with Stata installations, known as *census*. This particular dataset is ideal for demonstrating population distribution across different regions, which is a classic application for pie charts.

To begin, load the required data into Stata's active memory by executing the following command in the Command window:

sysuse census

The **sysuse** command is specifically engineered to load example datasets distributed alongside the Stata installation. Once the data is loaded, it is considered best practice to review the dataset's structure. We can quickly inspect the variables, observations, and obtain basic descriptive statistics by entering the **summarize** command:

summarize

This output provides essential information regarding the integrity and structure of the variables, confirming that we are working with the correct data structure before proceeding to the graph generation phase.

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
state	0				
state2	0				
region	50	2.66	1.061574	1	4
pop	50	4518149	4715038	401851	2.37e+07
poplt5	50	326277.8	331585.1	35998	1708400
pop5_17	50	945951.6	959372.8	91796	4680558
pop18p	50	3245920	3430531	271106	1.73e+07
pop65p	50	509502.8	538932.4	11547	2414250
popurban	50	3328253	4090178	172735	2.16e+07
medage	50	29.54	1.693445	24.2	34.7
death	50	39474.26	41742.35	1604	186428
marriage	50	47701.4	45130.42	4437	210864
divorce	50	23679.44	25094.01	2142	133541

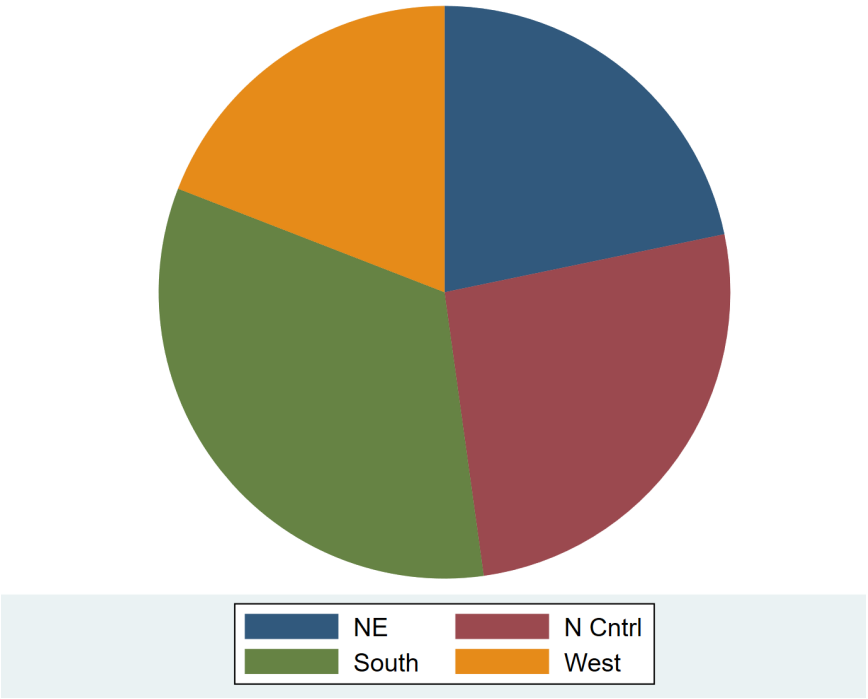
Generating the Foundational Pie Chart

The core of pie chart creation in Stata relies on the **graph pie** command. Our objective here is to visualize the total population size (represented by the variable `pop`) segmented by geographic region (represented by the variable `region`). The `over()` option is fundamentally important, as it instructs Stata which categorical variable should be used to define the individual slices of the chart.

The basic syntax required to visualize the population distribution across each region is concise and direct:

graph pie pop, over(region)

This command sequence directs Stata to calculate the sum of the `pop` variable for every unique value found within the `region` variable, then create a proportional slice for each summation. The resulting visualization offers an immediate, though often unrefined, view of how the population is distributed across the specified categories.



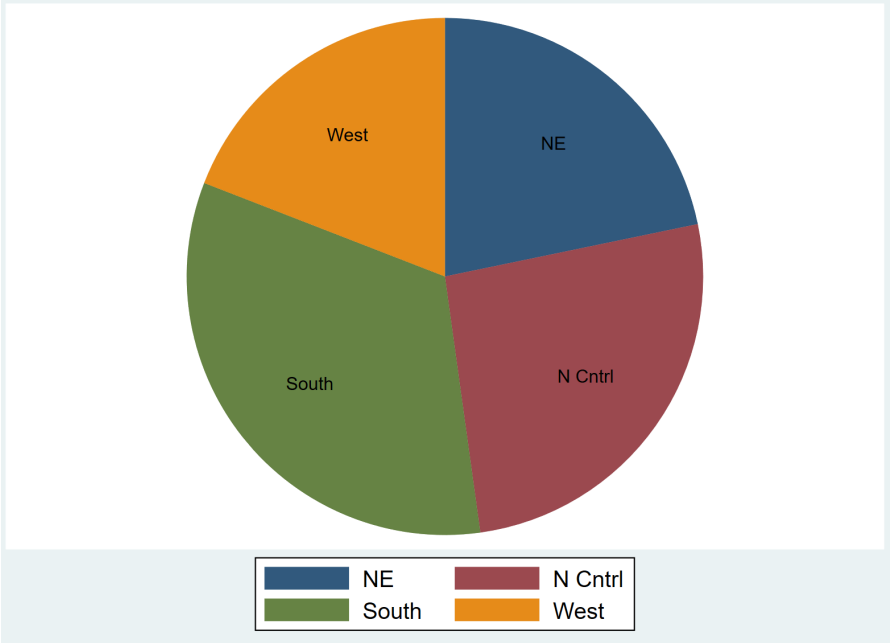
Implementing Direct Slice Labels for Clarity

While the initial pie chart is functional, interpreting which slice corresponds to which category typically requires viewers to cross-reference colors with an accompanying legend. This step can be inefficient and reduces immediate readability. Stata provides powerful tools for direct labeling of slices, significantly improving the chart's clarity and information accessibility.

To integrate descriptive labels directly onto the slices, we introduce the `plabel()` option. Specifically, using `plabel(_all name)` instructs Stata to display the name of the category (which is the value of the `over()` variable, `region` in this context) on every slice:

graph pie pop, over(region) plabel(_all name)

This simple modification makes the chart substantially more informative by removing the dependency on the legend. However, a potential issue arises when using the default black text against darker default slice colors, which can sometimes lead to poor contrast and legibility issues.



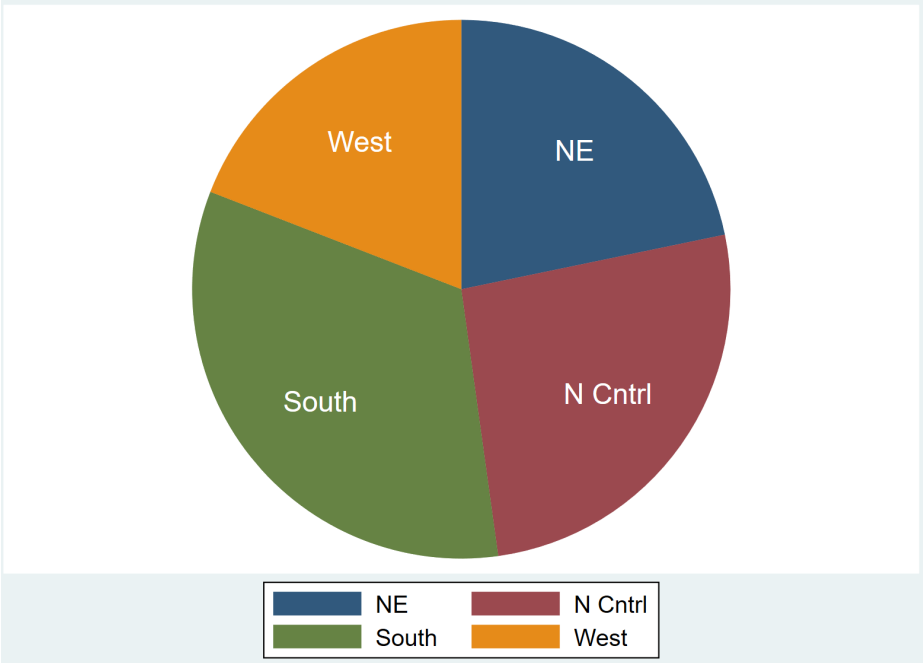
Optimizing Contrast and Removing Redundancy

To mitigate potential contrast problems and enhance the overall visual impact, Stata allows for meticulous control over label aesthetics, including font size and color. We can embed formatting options directly within the `plabel()` command. For instance, we can specify the desired size (e.g., `size(*1.5)` for 1.5 times the default size) and color (e.g., `color(white)`) to ensure maximum visibility against the default dark slice colors.

The updated syntax for a refined label appearance that prioritizes high contrast is:

graph pie pop, over(region) plabel(_all name, size(*1.5) color(white))

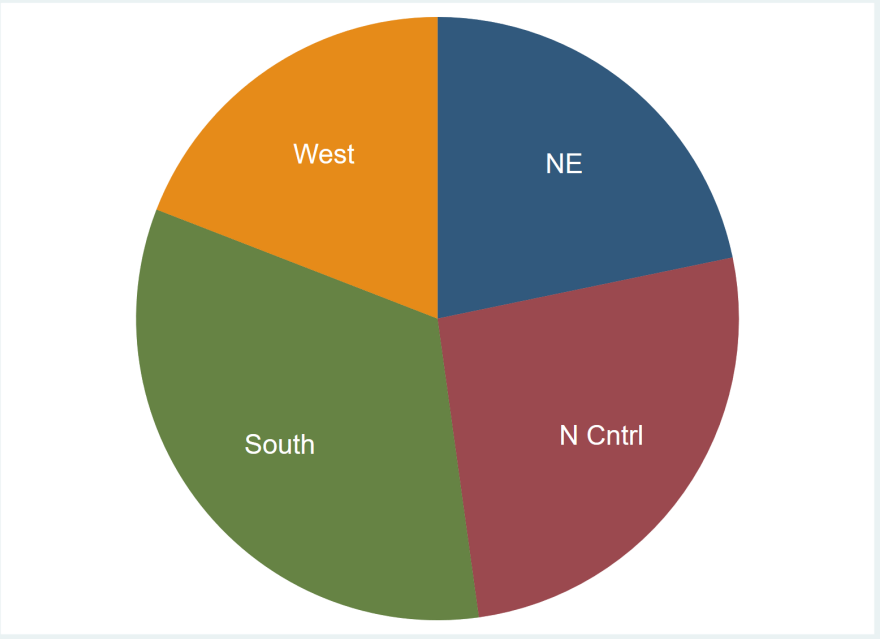
This professional approach ensures that the category names are large enough to be easily readable and possess sufficient contrast against the colored backgrounds, significantly elevating the quality of the visualization.



Once labels are successfully incorporated directly onto the slices, the graphical legend often becomes redundant. Retaining a redundant legend consumes valuable visual space and can clutter the graph unnecessarily. Stata provides a straightforward mechanism to disable the legend using the `legend(off)` option. By combining the enhanced labeling options with the legend suppression command, we achieve a clean, self-contained, and highly readable chart:

graph pie pop, over(region) plabel(_all name, size(*1.5) color(white)) legend(off)

This command sequence represents a best practice in visualization design, prioritizing data clarity and minimizing unnecessary graphical elements. The resulting graph focuses the viewer's attention solely on the proportional relationship between the regions.



Adding Essential Contextual Annotations (Title, Subtitle, Note)

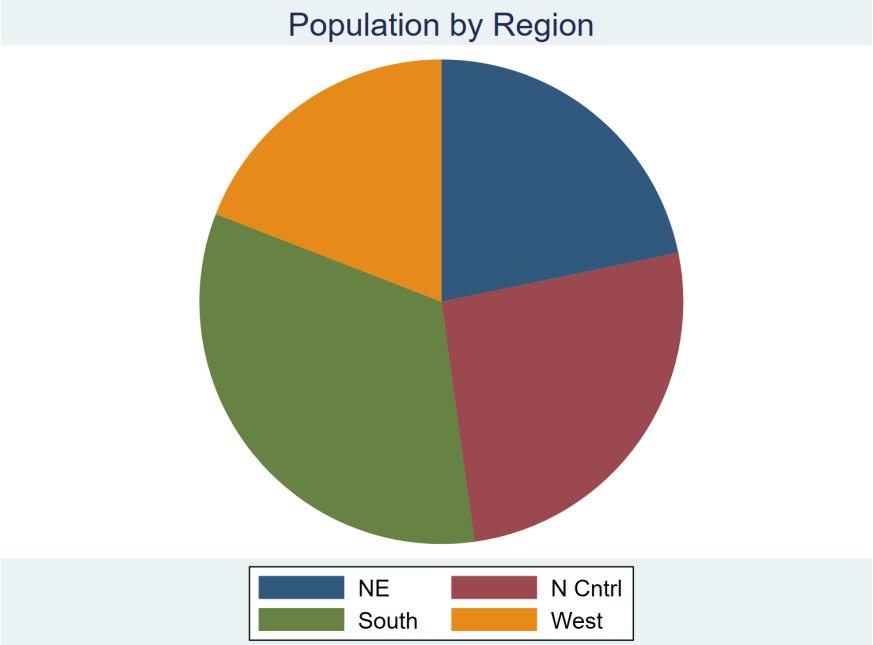
Beyond the core visual data, a professional statistical graph requires clear contextual information, typically provided through titles, subtitles, and notes. Stata offers dedicated options to control these metadata elements, allowing analysts to fully contextualize their visualizations for any audience.

A descriptive title is essential for immediately informing the viewer about the chart's specific content. The `title()` option allows you to place a main heading prominently above the chart area. This title must be concise yet fully explanatory.

To add the title "Population by Region" to the pie chart, we integrate the following syntax:

graph pie pop, over(region) title("Population by Region")

This addition immediately elevates the professionalism of the graph, making it suitable for formal reports and presentations where context is paramount.



In cases where additional detail is necessary, but should not clutter the main title, the `subtitle()` option is utilized. Subtitles typically convey information such as sample size, specific time periods, or statistical aggregates relevant to the data being displayed. For instance, to clarify the number of regions included in the aggregation, we can use the subtitle option:

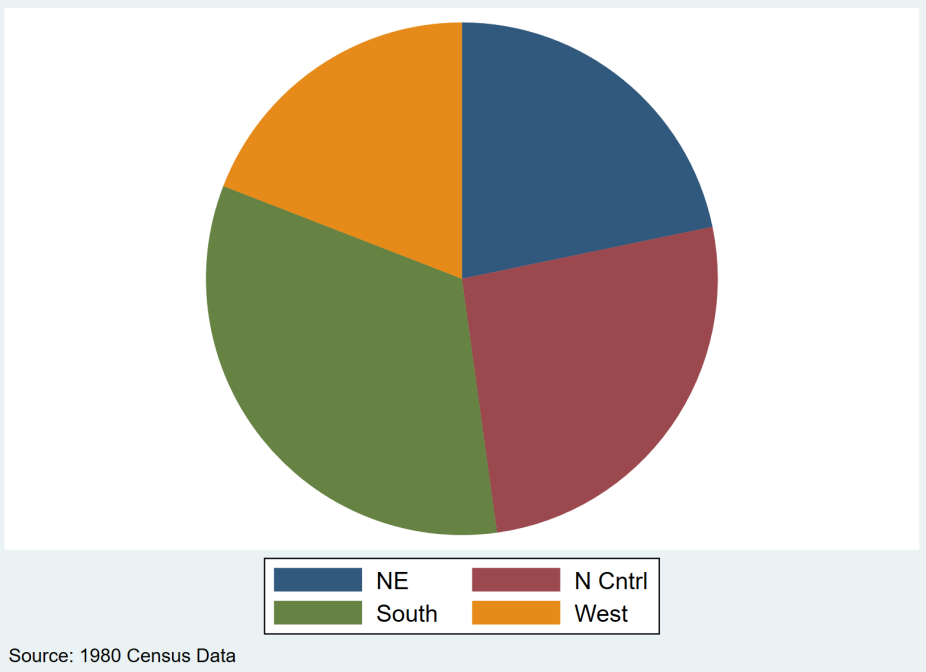
graph pie pop, over(region) title("Population by Region") subtitle("n = 4 total regions")

The subtitle is positioned directly beneath the main title, providing secondary but necessary information without distracting from the primary visual data.

Finally, it is standard academic and professional practice to cite the source of the data used in a visualization. The `note()` command places text at the bottom of the graph, which is the perfect location for source citations, disclaimers, or brief methodological comments. To cite the origin of the census data, we incorporate the `note()` option:

graph pie pop, over(region) note("Source: 1980 Census Data")

This annotation is crucial for transparency, helping viewers understand the limitations and origin of the data presented within the visualization.



Granular Control Over Slice Colors

While Stata's default color schemes are functional, analysts often require specific colors to align with institutional branding, publication standards, or to visually differentiate specific categories for emphasis. Stata provides granular control over the color of each individual slice, allowing for complete aesthetic customization.

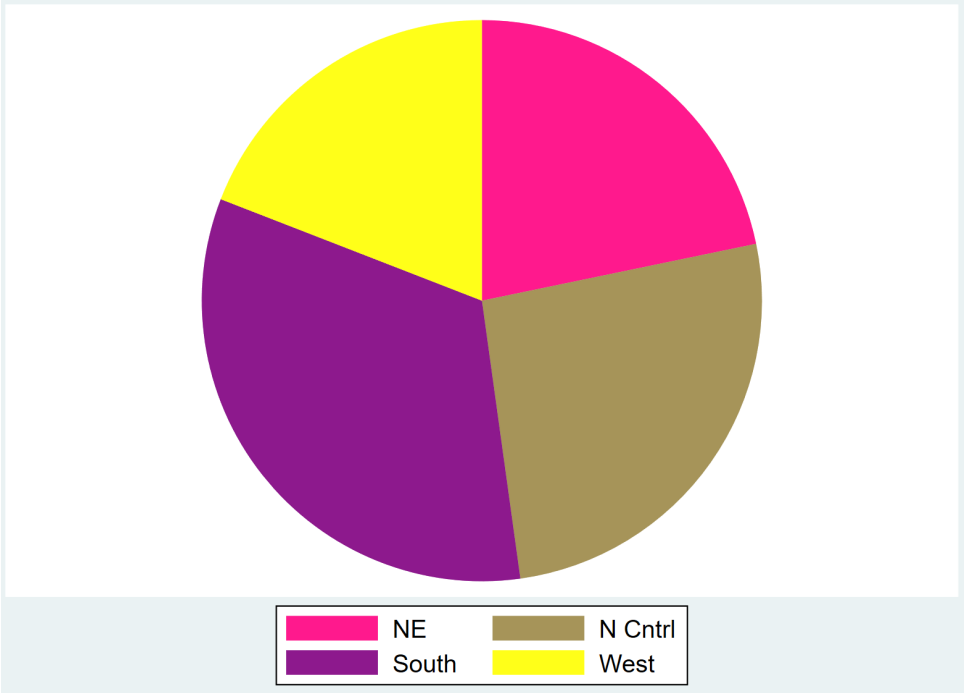
The customization of slice colors is achieved using the `pie()` option, which allows the user to target slices based on their numerical order (starting at 1) and apply a specified color. The general syntax for coloring a single slice is:

pie(slice #, color(specific_color))

When the `over()` option is used, Stata assigns slices based on the alphabetical or numeric order of the categories within the specified variable. In our *census* dataset, if the regions are ordered alphabetically, slice 1 would correspond to the first region, slice 2 to the second, and so on. We can apply distinct colors to the four regions in our dataset using a chain of `pie()` options:

graph pie pop, over(region) pie(1, color(pink)) pie(2, color(brown)) pie(3, color(purple)) pie(4, color(yellow))

This detailed level of control ensures that the visualization is not only statistically accurate but also aesthetically tailored to meet specific presentation requirements. Using custom colors is an effective way to highlight a particular segment of interest to the reader or stakeholder.



For users seeking to utilize a wider array of colors, Stata provides extensive documentation detailing acceptable named colors, RGB specifications, and CMYK codes. A comprehensive list of available colors and detailed documentation on advanced color schemes can be found in the official Stata Graphics Manual, ensuring complete graphical flexibility.