Exercise Lab 4

Mina Moeini
Mina_moeini@sfu.ca

October 4, 2024

Introduction

In SAS, you can create a scatter plot using the `PROC SGPLOT` procedure. Below is a step-by-step explanation of how to generate and customize scatter plots.

Basic Syntax

The simplest way to create a scatter plot is by specifying the `x` and `y` variables in the `scatter` statement:

```
proc sgplot data=your_dataset;
   scatter x=x_variable y=y_variable;
run;
```

Example: Scatter Plot Using `sashelp.class`

For instance, let's use the `sashelp.class` dataset to visualize the relationship between height and weight.

```
proc sgplot data=sashelp.class;
   scatter x=height y=weight;
run;
```

Customizing the Scatter Plot

Add a Title

You can include a title for your graph by using the `title` statement:

```
title "Scatter Plot of Height vs Weight";
proc sgplot data=sashelp.class;
   scatter x=height y=weight;
run;
title;
```

Labeling the Axes

Custom labels for the axes can be added using the `xaxis` and `yaxis` statements:

```
proc sgplot data=sashelp.class;
   scatter x=height y=weight;
   xaxis label="Height (inches)";
   yaxis label="Weight (pounds)";
run;
```

Change Marker Attributes

You can customize the markers' appearance, such as changing color, size, or symbol, by using the `markerattrs` option:

```
proc sgplot data=sashelp.class;
   scatter x=height y=weight
   / markerattrs=(symbol=circlefilled color=red size=10);
run;
```

Group by Categories

Grouping the data by a categorical variable (e.g., `sex`) allows you to show different colors for each group:

```
proc sgplot data=sashelp.class;
   scatter x=height y=weight / group=sex;
run;
```

Adding a Regression Line

To add a regression or fit line to your scatter plot, use the `reg` statement along with `scatter`:

```
proc sgplot data=sashelp.class;
   scatter x=height y=weight;
   reg x=height y=weight;
run;
```

Overlaying Multiple Scatter Plots

You can overlay multiple scatter plots by adding more `scatter` statements. For example:

```
proc sgplot data=sashelp.class;
   scatter x=height y=weight;
   scatter x=height y=age
   / markerattrs=(symbol=trianglefilled color=blue);
run;
```

Conclusion

To create a scatter plot in SAS:

- Use `PROC SGPLOT` with the `scatter` statement.
- Specify the variables for the `x` and `y` axes.
- Customize the plot with titles, axis labels, marker attributes, or group by categorical variables.
- Add features like regression lines or overlay multiple scatter plots for comparison.

Bar chart

In SAS, you can create a bar chart using the `PROC SGPLOT` procedure. Bar charts are useful for visualizing categorical data by displaying the frequency or a related statistic for each category as a bar.

Basic Syntax

The simplest way to create a bar chart is by using the `VBAR` statement (for vertical bars) or `HBAR` statement (for horizontal bars):

```
proc sgplot data=your_dataset;
   vbar categorical_variable;
run;
```

Example: Bar Chart Using `sashelp.class`

For example, let's create a simple bar chart showing the number of students by gender in the `sashelp.class` dataset:

```
proc sgplot data=sashelp.class;
   vbar sex;
run;
```

Customizing the Bar Chart

Add a Title

You can include a title for your chart using the `title` statement:

```
title "Distribution of Students by Gender";
proc sgplot data=sashelp.class;
   vbar sex;
run;
title;
```

Labeling the Axes

You can add custom labels to the axes using the `xaxis` and `yaxis` statements:

```
proc sgplot data=sashelp.class;
   vbar sex;
   xaxis label="Gender";
```

```
   yaxis label="Frequency";
run;
```

Show Frequency or Other Statistic

By default, SAS displays the frequency (count) for each category. You can change this to display another statistic, such as the mean of a numerical variable, using the `response` and `stat` options:

```
proc sgplot data=sashelp.class;
   vbar sex / response=height stat=mean;
run;
```

This code displays a bar chart where the height of the bars represents the mean of the `height` variable for each gender.

Change the Appearance of the Bars

You can customize the appearance of the bars using the `fillattrs` option:

```
proc sgplot data=sashelp.class;
   vbar sex / fillattrs=(color=lightblue);
run;
```

This will change the bar color to light blue.

Group by a Categorical Variable

You can group the data by another categorical variable (e.g., `age`) to show stacked or clustered bars:

```
proc sgplot data=sashelp.class;
   vbar sex / group=age;
run;
```

This will group the bars by the `age` variable, showing multiple bars for each gender based on the age categories.

Horizontal Bar Chart

You can create a horizontal bar chart using the `HBAR` statement instead of `VBAR`:

```
proc sgplot data=sashelp.class;
   hbar sex;
run;
```

Adding a Legend

When grouping or stacking bars, you may want to add a legend to explain what the colors represent. This can be done using the `keylegend` statement:

```
proc sgplot data=sashelp.class;
   vbar sex / group=age;
   keylegend / position=topright;
run;
```

This code places a legend in the top-right corner of the plot to indicate the age categories.

Bar Chart with Response Data

You can also create bar charts based on continuous response data. For example, to visualize the average height for each gender:

```
proc sgplot data=sashelp.class;
   vbar sex / response=height stat=mean;
run;
```

This will display the average height for each gender.

Conclusion

To create a bar chart in SAS:

- Use `PROC SGPLOT` with the `VBAR` or `HBAR` statement.
- Customize the plot by adding titles, labels, colors, and legends.
- Display different statistics such as frequency, mean, sum, etc.

Introduction

In SAS, you can create a histogram using the `PROC SGPLOT` procedure. Histograms are useful for visualizing the distribution of a continuous variable. Below is a guide on how to create and customize a histogram in SAS.

Basic Syntax

The simplest way to create a histogram is by using the `HISTOGRAM` statement:

```
proc sgplot data=your_dataset;
   histogram continuous_variable;
run;
```

Where:

- `your_dataset` is the dataset you are working with.
- `continuous_variable` is the continuous variable you want to visualize.

Example: Histogram Using `sashelp.class`

To create a histogram of the `weight` variable from the `sashelp.class` dataset, use the following code:

```
proc sgplot data=sashelp.class;
   histogram weight;
run;
```

Customizing the Histogram

Add a Title

You can add a title to the histogram using the `title` statement:

```
title "Distribution of Weight";
proc sgplot data=sashelp.class;
   histogram weight;
run;
title;
```

Labeling the Axes

To provide custom labels for the x-axis and y-axis, you can use the `xaxis` and `yaxis` statements:

```
proc sgplot data=sashelp.class;
   histogram weight;
   xaxis label="Weight (pounds)";
   yaxis label="Frequency";
run;
```

Adjusting the Number of Bins

SAS chooses the number of bins by default, but you can manually adjust the number of bins using the `nbins` option:

```
proc sgplot data=sashelp.class;
   histogram weight / nbins=10;
run;
```

Change the Color of the Histogram

You can customize the color of the histogram bars using the `fillattrs` option:

```
proc sgplot data=sashelp.class;
   histogram weight / fillattrs=(color=lightblue);
run;
```

Overlay a Density Curve

To visualize the underlying distribution more smoothly, you can overlay a density curve on the histogram using the `density` statement:

```
proc sgplot data=sashelp.class;
   histogram weight;
   density weight;
run;
```

You can also customize the appearance of the density curve:

```
proc sgplot data=sashelp.class;
   histogram weight;
   density weight / lineattrs=(color=red pattern=solid);
run;
```

Specifying Kernel Density Estimation

Instead of a normal density curve, you can use kernel density estimation for a non-parametric estimate of the probability density function:

```
proc sgplot data=sashelp.class;
   histogram weight;
   density weight / type=kernel;
run;
```

Histogram with a Group Variable

You can create separate histograms for different groups using the `group` option:

```
proc sgplot data=sashelp.class;
   histogram weight / group=sex transparency=0.5;
run;
```

This generates two histograms (one for each gender) overlaid on the same graph with transparency to highlight overlaps.

Stacked Histograms (Side-by-Side)

To display histograms side by side for different groups, you can use the `panelby` statement:

```
proc sgplot data=sashelp.class;
   histogram weight;
   panelby sex / layout=rowlattice;
run;
```

Conclusion

In summary, you can create histograms in SAS using `PROC SGPLOT`. You can:

- Add titles and axis labels.
- Adjust the number of bins.
- Change the color of the bars.
- Overlay density curves or use kernel density estimation.
- Create grouped histograms or side-by-side comparisons.