Graphs

Statistics 426: SAS Programming

Module 9

2021

Graphs

One thing I always tell students is that if you do not look at your data, as in a picture (graph), you do not have a full understanding of your data, regardless how much number crunching you do. Most often in statistics, in order to choose the most appropriate model, you must know what the distribution of the data looks like; that is *look at a graph.* Graphs are crucial to data science, *all* sciences (all disciplines); even a small data table is nearly incomprehensible by itself. They allow scientists to visualize quantitative patterns.

Iris Dataset

Edgar Anderson's Iris Data

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *setosa*, *versicolor*, and *virginica.*

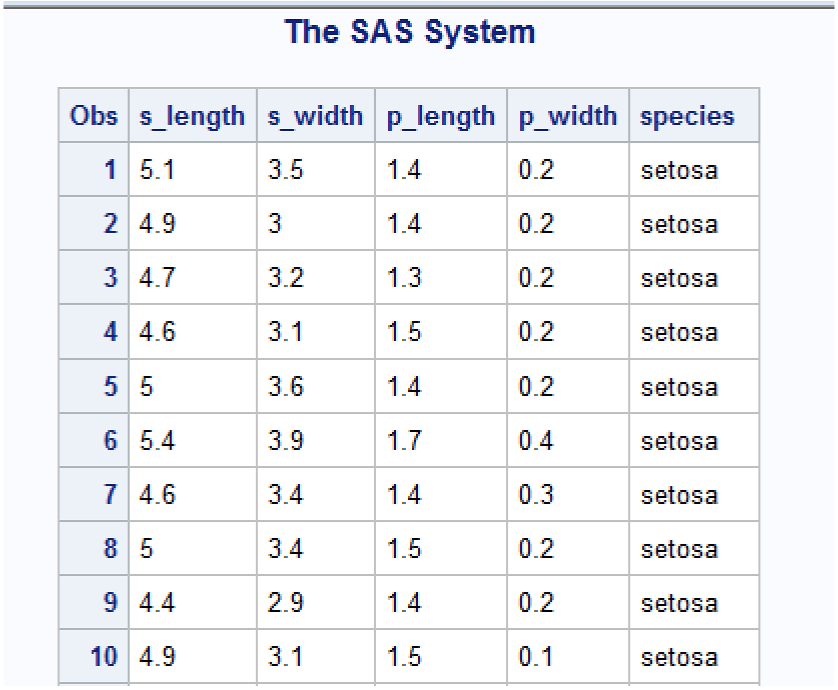
Sepal length is numeric, as are sepal width, petal length, and petal width. Species has no numbers in the variable at all, it is qualitative (categorical, character, etc.). Based on data types, we can figure out which graphs are appropriate given the data type.

Iris Dataset

```
filename flower url 'https://webpages.uidaho.edu/~renaes/Data/iris.csv';

data iris;
infile flower dsd missover firstobs=2;
input s_length s_width p_length p_width species$;
* could insert LABEL statement for variable names;
run;
proc print data=iris;
run;
```

Iris data



The SAS System

Obs	s_length	s_width	p_length	p_width	species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

print.png

PROC SGPLOT and PROC GCHART

PROC SGPLOT and PROC GCHART will be used; both have statement options for many types of graphs for quantitative and qualitative data

Quantitative graphs

(1) `DOT`
(2) `HISTOGRAM`
(3) `DENSITY` (only used with `HISTOGRAM`)
(4) `SCATTER`
(5) `SERIES` (can be used alone or with `SCATTER`)
(6) `HBOX`
(7) . . .

Qualitative graphs

(1) `HBAR/VBAR`
(2) `HLINE/VLINE` (used with `HBAR/VBAR`)
(3) `PIE`

PROC SGPLOT general form

```
PROC SGPLOT DATA=SASdataset;
STATEMENT;
<additional SAS statements>
RUN;
QUIT;
```

`STATEMENT`: graph type, variable(s), and options

Why the QUIT statement is (sometimes) needed

Run-group processing

Many SAS/GRAPH procedures can perform RUN-group processing, which means the procedure executes the group of statements following the PROC statement when a RUN statement is encountered, additional statements followed by another RUN statement can be submitted without resubmitting the PROC statement, and the procedure stays active until a PROC, DATA or QUIT statement is encountered. PROC SGPLOT, PROC GCHART, PROC REG, and PROC GLM are several that occasionally need the QUIT statement

```
PROC SGPLOT;
HISTOGRAM variable;
RUN;
VBOX variable;
RUN;
QUIT;
```

SGPLOT `STATEMENT` options

(a) `DOT`
(b) `HISTOGRAM`
(c) `DENSITY` (can be used alone or with `HISTOGRAM`)
(d) `SCATTER`
(e) `SERIES` (can be used alone or with `SCATTER`)

`HISTOGRAM var1;` as an example for the `STATEMENT` line

`DOT` statement in SGPLOT

Dotplot: A dotplot (also called a stripchart) is a simple visualization of the data; use with quantitative data

Dotplot `STATEMENT` line
`DOT var1 ... </options>;`

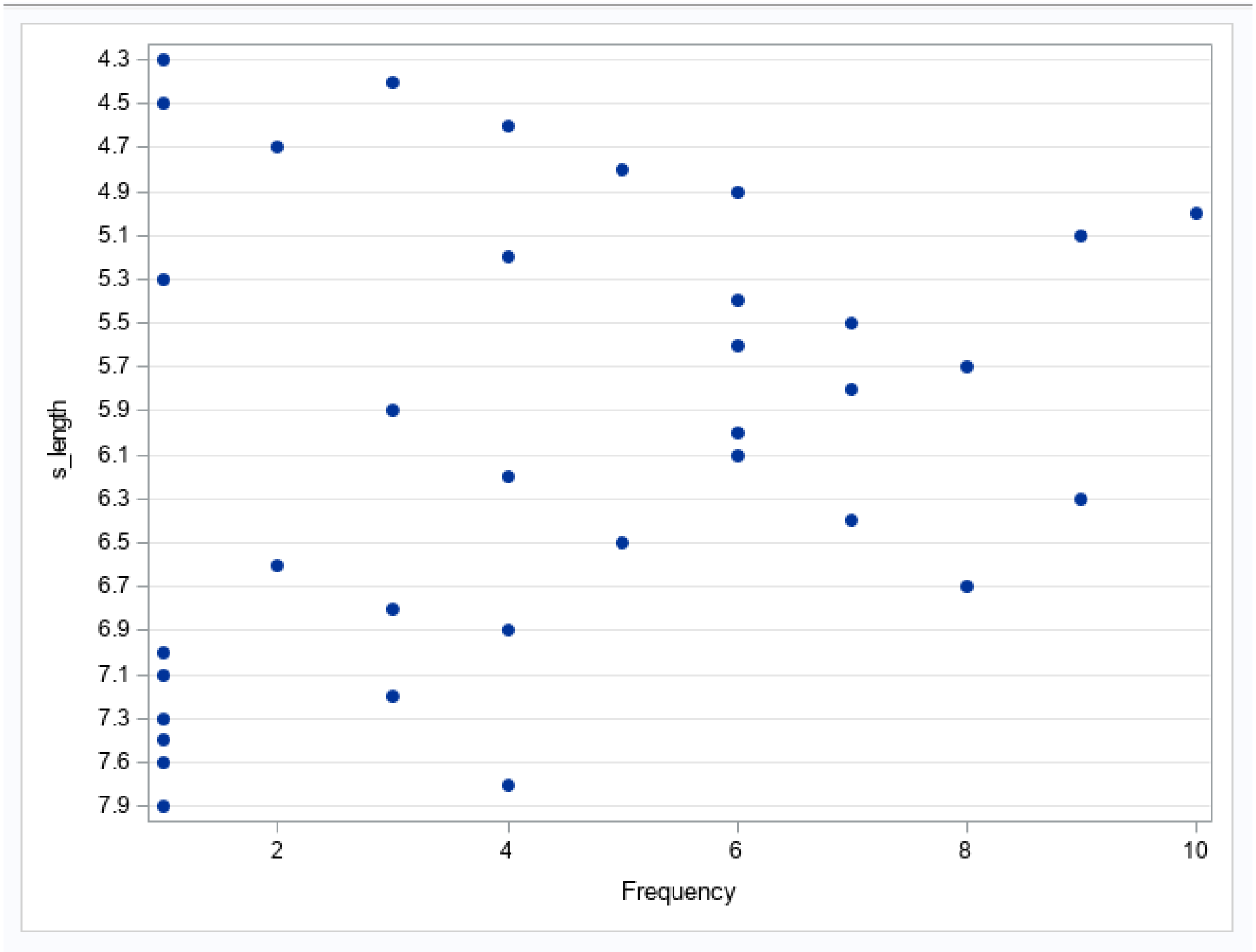
`var1 ...`: variable(s) of values
`</options>`: more options, soooo many more options

There are many (MANY) options for graphs, for a complete list of all options and descriptions, the official SAS website has complete information available, most with detailed examples of code and output

Dotplot

```
proc sgplot data=iris;
dot s_length;
run;
quit;
```

Dotplot Sepal Length



dot.png

HISTOGRAM statement in SGPLOT

Histogram: represents the frequency (or sometimes relative frequency/density/percents) of the data points in an interval as a rectangle over the interval, with the area of the rectangle equal to the frequency. The histogram statement in PROC SGPLOT usually does a decent job at creating the classes (bins) for the data on its own; there are ways to control more of what the graphs can do with more extensive options.

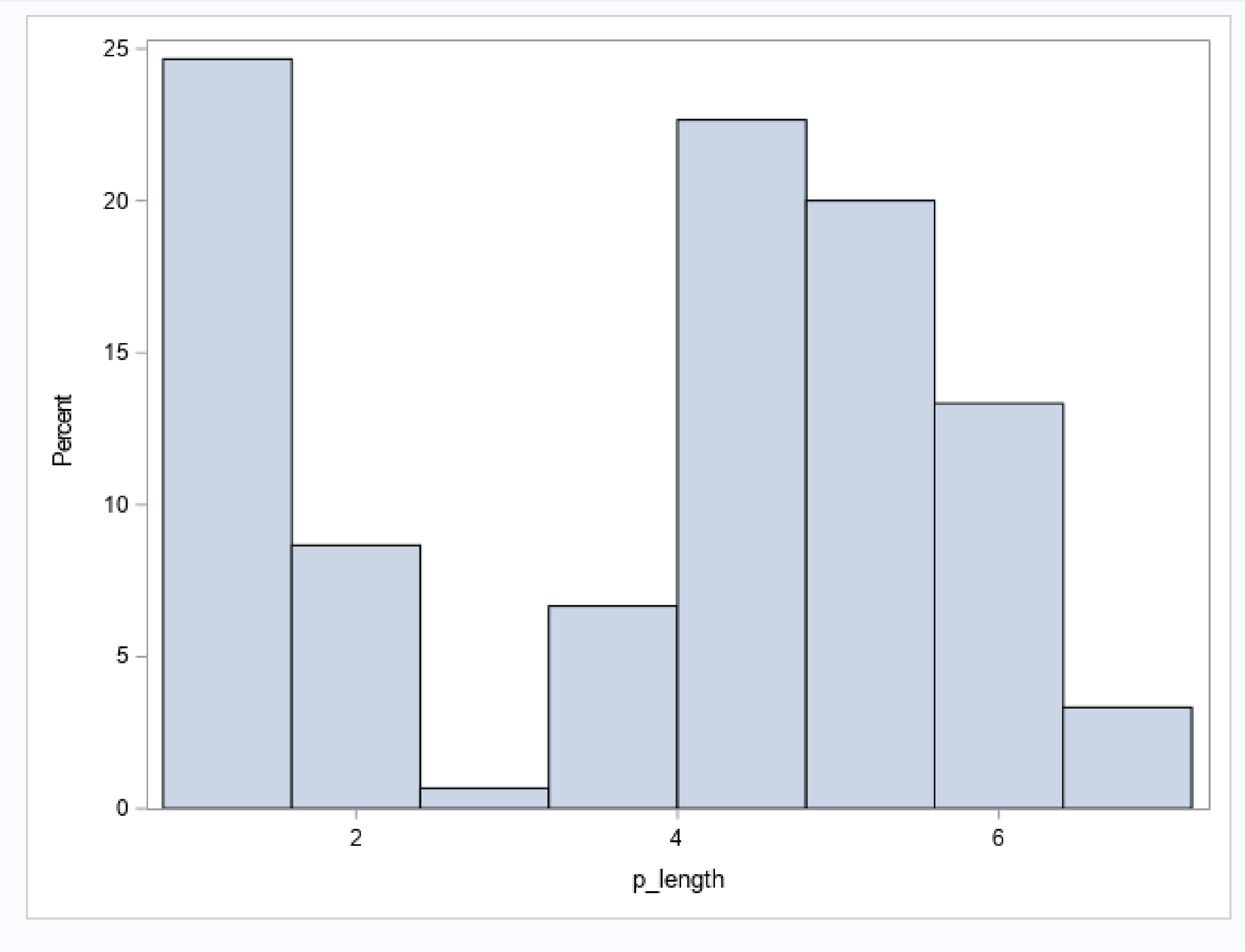
Histogram `STATEMENT` line
`HISTOGRAM var1 ... </options>;`

`var1 ...`: variable(s) of values
`</options>`: more options

Histogram

```
proc sgplot data=iris;
histogram p_length;
run;
quit;
```

Histogram of Iris petal length



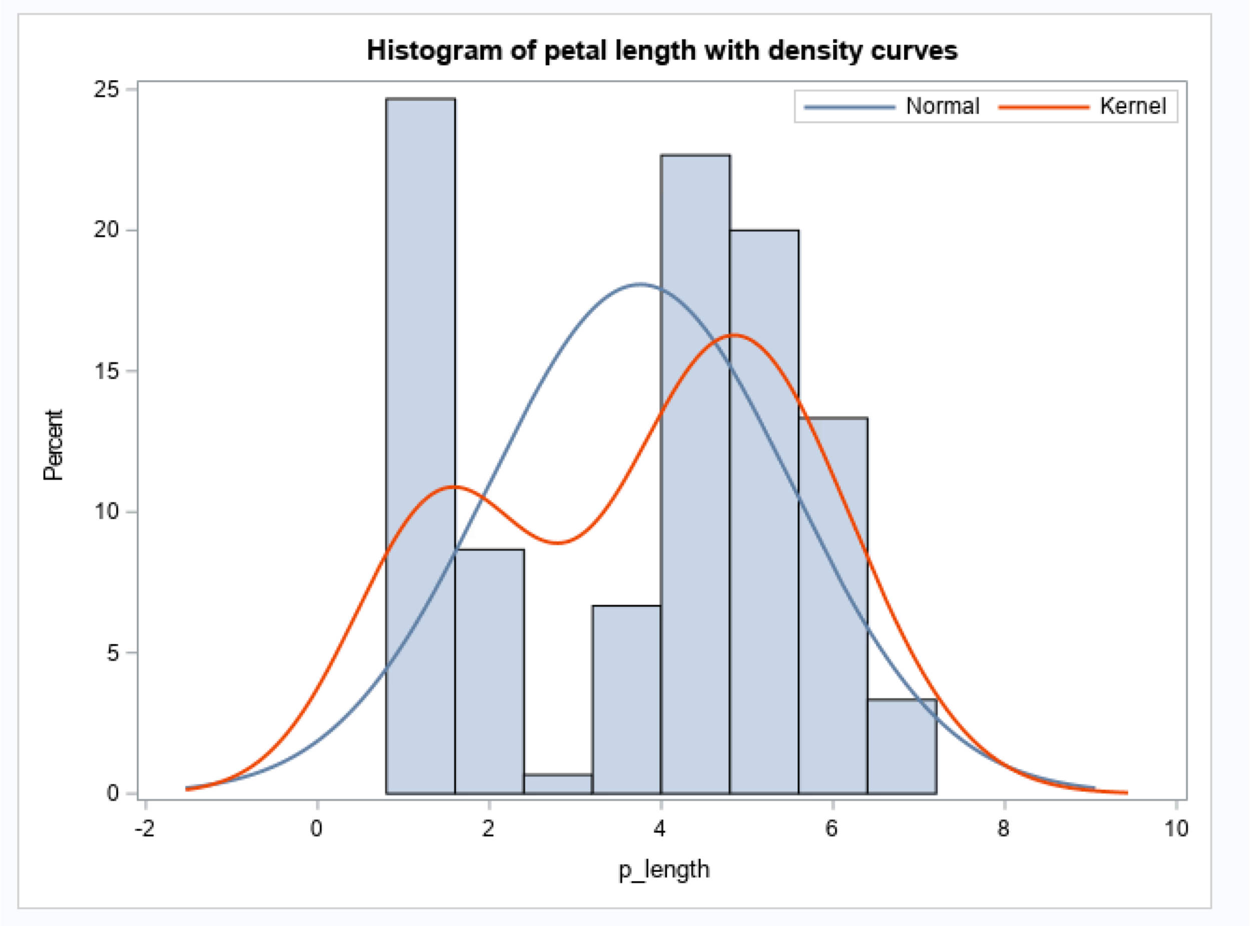
hist1.png

Histogram with density curve

The DENSITY statement can only be used with other statements (like HISTOGRAM)

```
proc sgplot data=iris;
title 'Histogram of petal length with density curves';
histogram p_length;
density p_length;
density p_length / type=kernel;
  keylegend / location=inside position=topright;
run;
quit;
title;
```

Histogram and density curve



hist2.png

Boxplots (box-and-whisker plots)

A boxplot relies on five numbers to summarize all of the data in the variable. The "5 Number summary" consists of the minimum, quartile 1, median, quartile 3, and the maximum. There are commands, individual and within PROCs, to find the values of the 5 number summary, most often with PROC UNIVARIATE (others do have these calculations as well)

5 number summary

minimum: (min) the smallest observation
quartile 1: (Q1) the 25th percentile; 25% of the data points are less than Q1 and 75% are greater than Q1
median: aka 50th percentile; the middle observation; 50% of the data points are less than the median and 50% are greater than the median
quartile 3: (Q3) the 75th percentile; 75% of the data points are less than Q3 and 25% are greater than Q3
maximum: (max) the largest observation

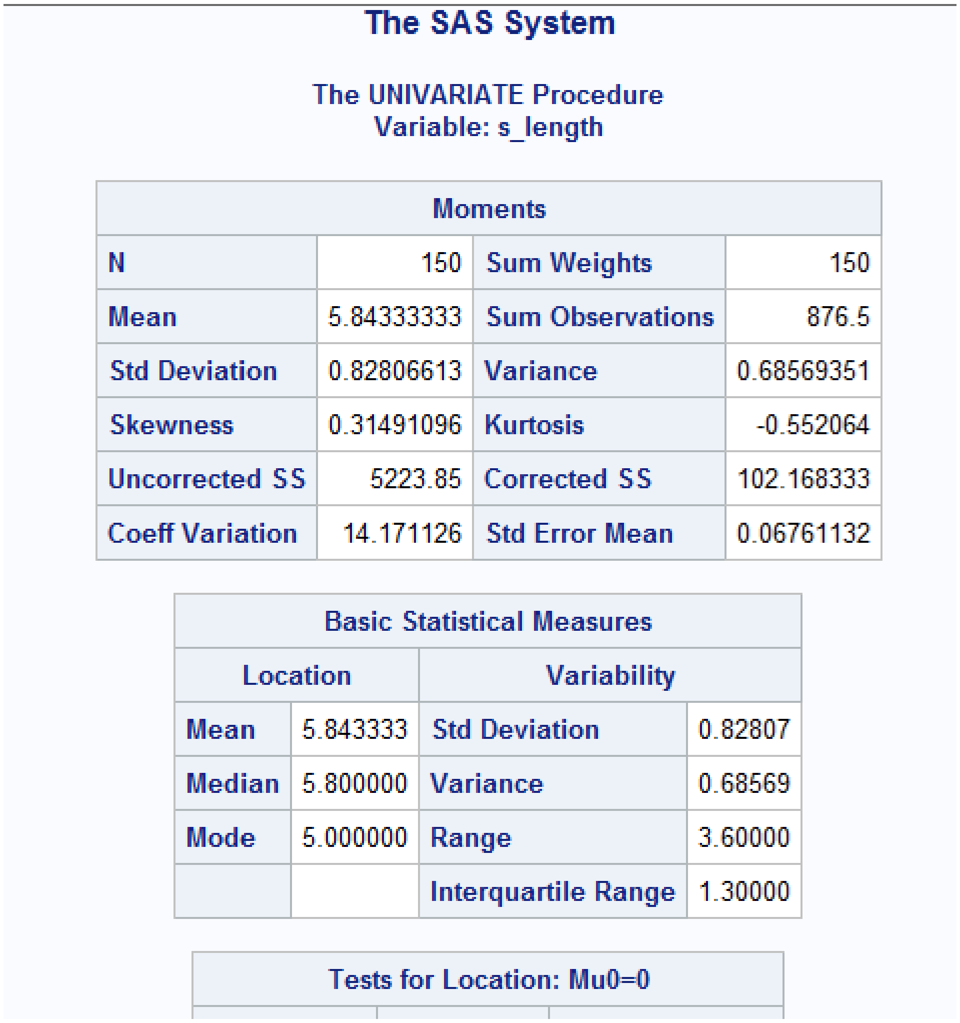
PROC UNIVARIATE

PROC UNIVARIATE was used for data validation and again here for the five number summary, to show for relating to boxplots. PROC UNIVARIATE will calculate more than just the five number summary, but not our focus

```
proc univariate data=iris;
```

```
var s_length;
run;
```

Univariate output



The SAS System

The UNIVARIATE Procedure
Variable: s_length

Moments			
N	150	Sum Weights	150
Mean	5.84333333	Sum Observations	876.5
Std Deviation	0.82806613	Variance	0.68569351
Skewness	0.31491096	Kurtosis	-0.552064
Uncorrected SS	5223.85	Corrected SS	102.168333
Coeff Variation	14.171126	Std Error Mean	0.06761132

Basic Statistical Measures			
Location		Variability	
Mean	5.843333	Std Deviation	0.82807
Median	5.800000	Variance	0.68569
Mode	5.000000	Range	3.60000
		Interquartile Range	1.30000

Tests for Location: Mu0=0

univ.png

`HBOX/VBOX` statement in SGPLOT

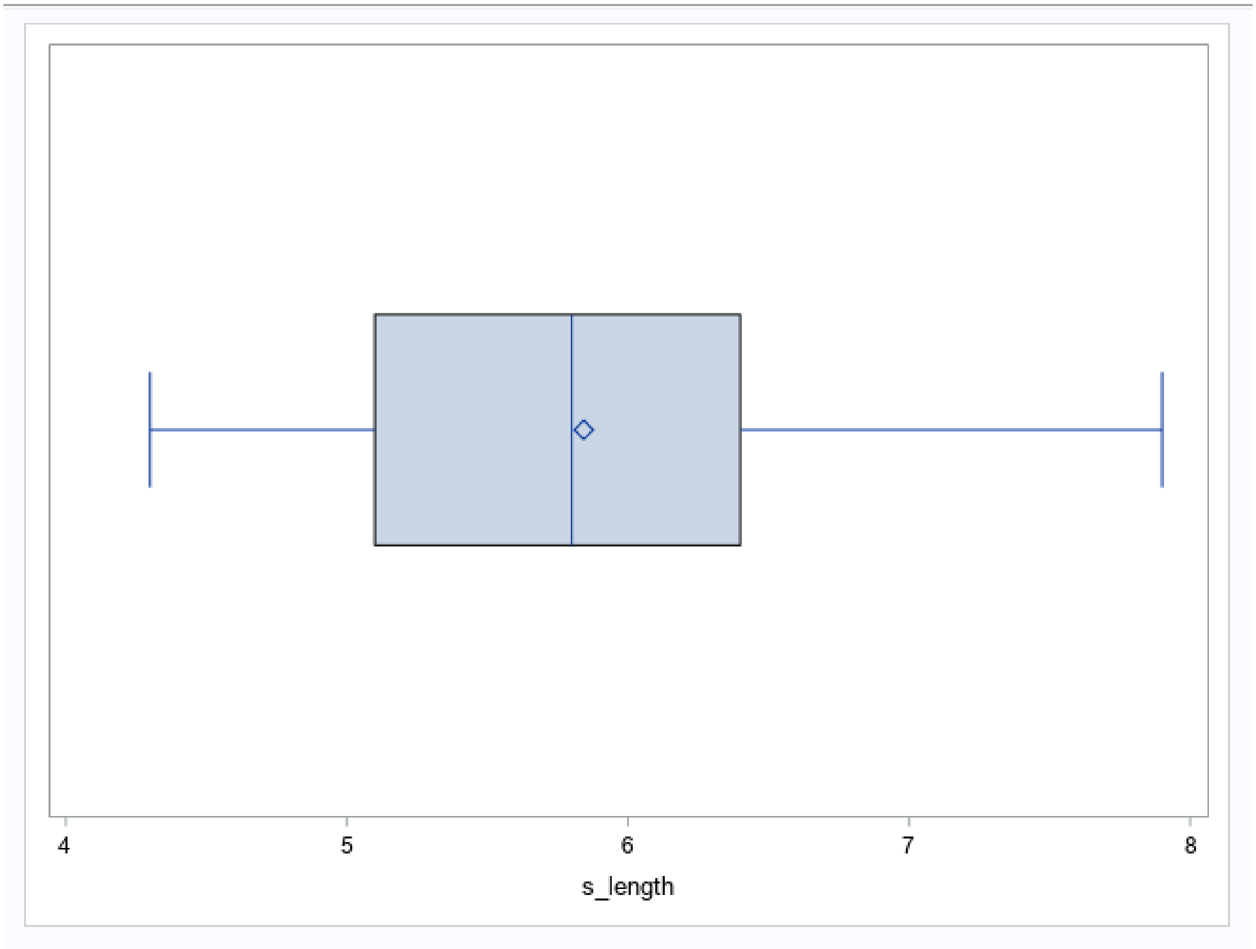
HBOX/VBOX `STATEMENT` line
`HBOX var1 ... </options>;` or `VBOX var1 ... </options>;`

`var1 ...`: variable(s) of values
`</options>`: more options

Boxplot

```
proc sgplot data=iris;
hbox s_length;
run;
quit;
```

Sepal Length boxplot

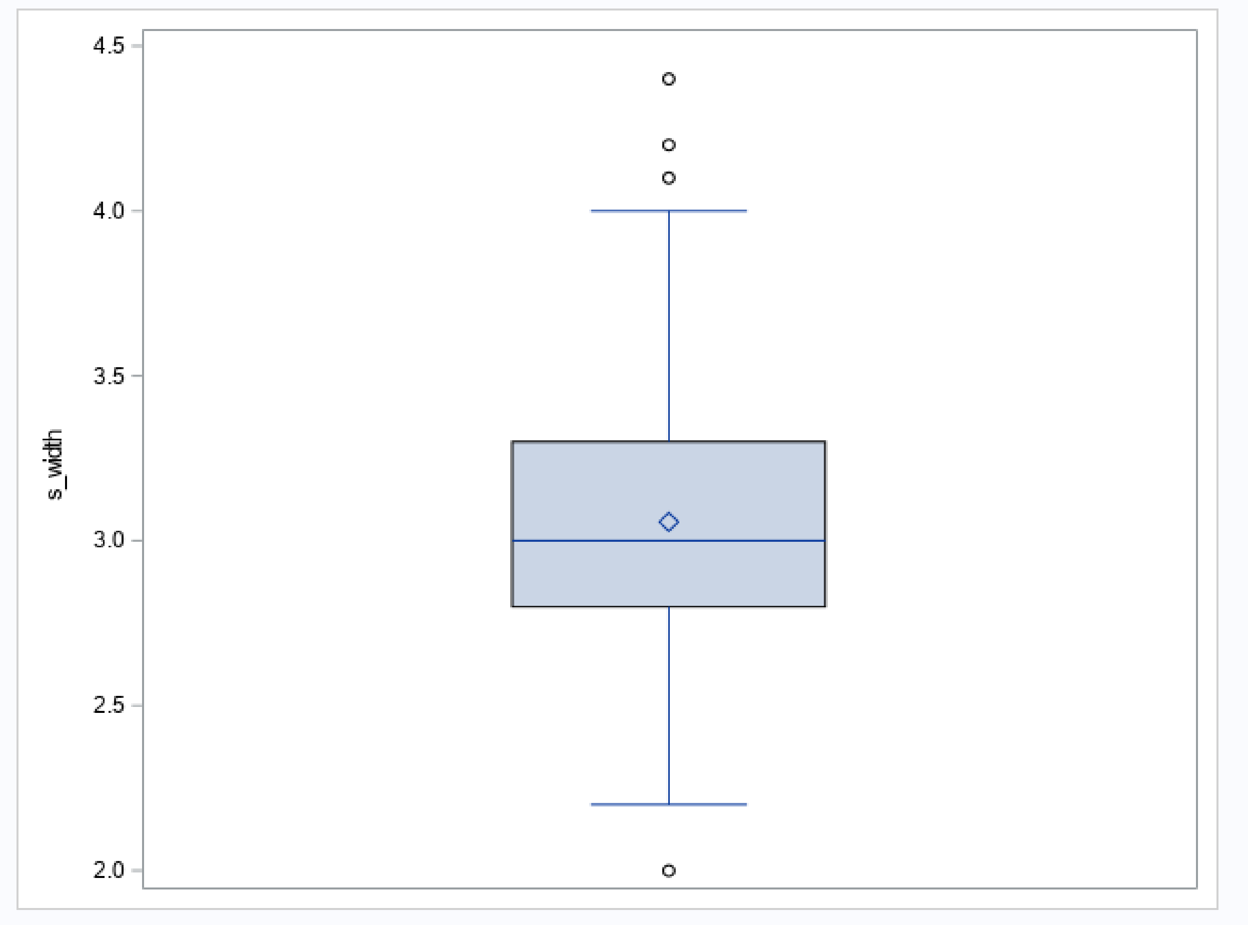


hbox.png

Boxplot

```
proc sgplot data=iris;
vbox s_width;
run;
quit;
```

Sepal Width boxplot



vbox.png

Graphs of two variables

Various types of graphs are helpful for investigating relationships between two variables.

Scatterplot: used when both variables are quantitative; it shows the values of two variables recorded from each subject/unit as an ordered pair on an *x*-*y* plot.

Side-by-side boxplots: graph the values by categorical group(s).

Bar charts: graph depicting frequencies (or percents) of a numeric variable by a categorical variable

Pie charts: (are evil; avoid at all cost) graph depicting frequencies (or percents) of a numeric variable by a categorical variable

Scatterplots

There are many options to display the data with lines, points, and more. We will concentrate on lineplots (timeseries type plots) and point plots

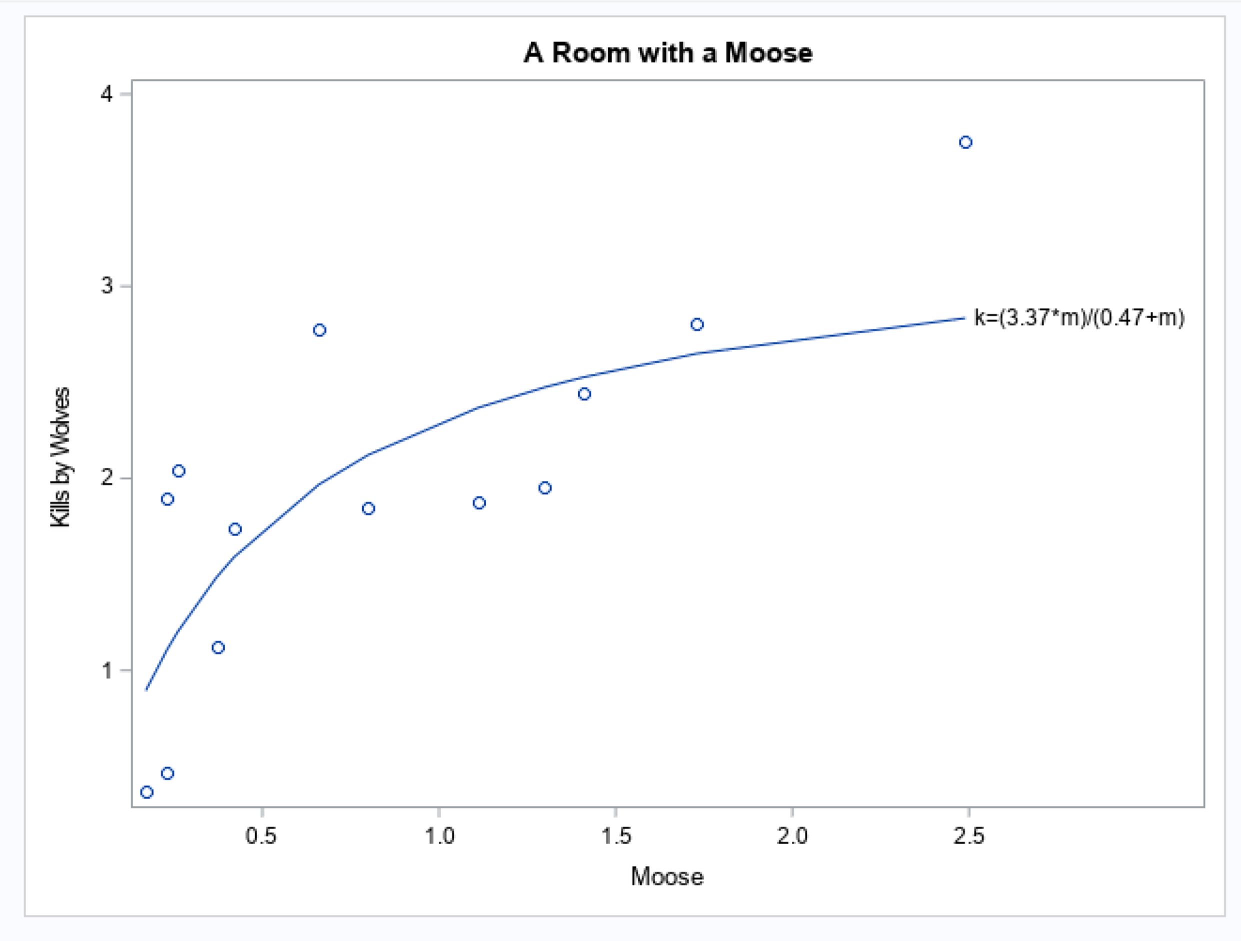
```
SCATTER x=x y=y;
RUN;
```

`SCATTER` for points, `SERIES` for lines, both for line overlay on points `x`: x values
`y`: y values

Revisit the room with a moose

```
proc sgplot data=moose_room2;
title 'A Room with a Moose';
scatter x=moose y=k_rate;
series x=moose y=k / curvelabel="k=(3.37*m)/(0.47+m)";
xaxis Label='Moose';
yaxis Label='Kills by Wolves';
run;
title;
```

Moose again



moose sgplot.png

Scatterplot matrix

To see a matrix of scatterplots (all (numeric) variables against all variables):

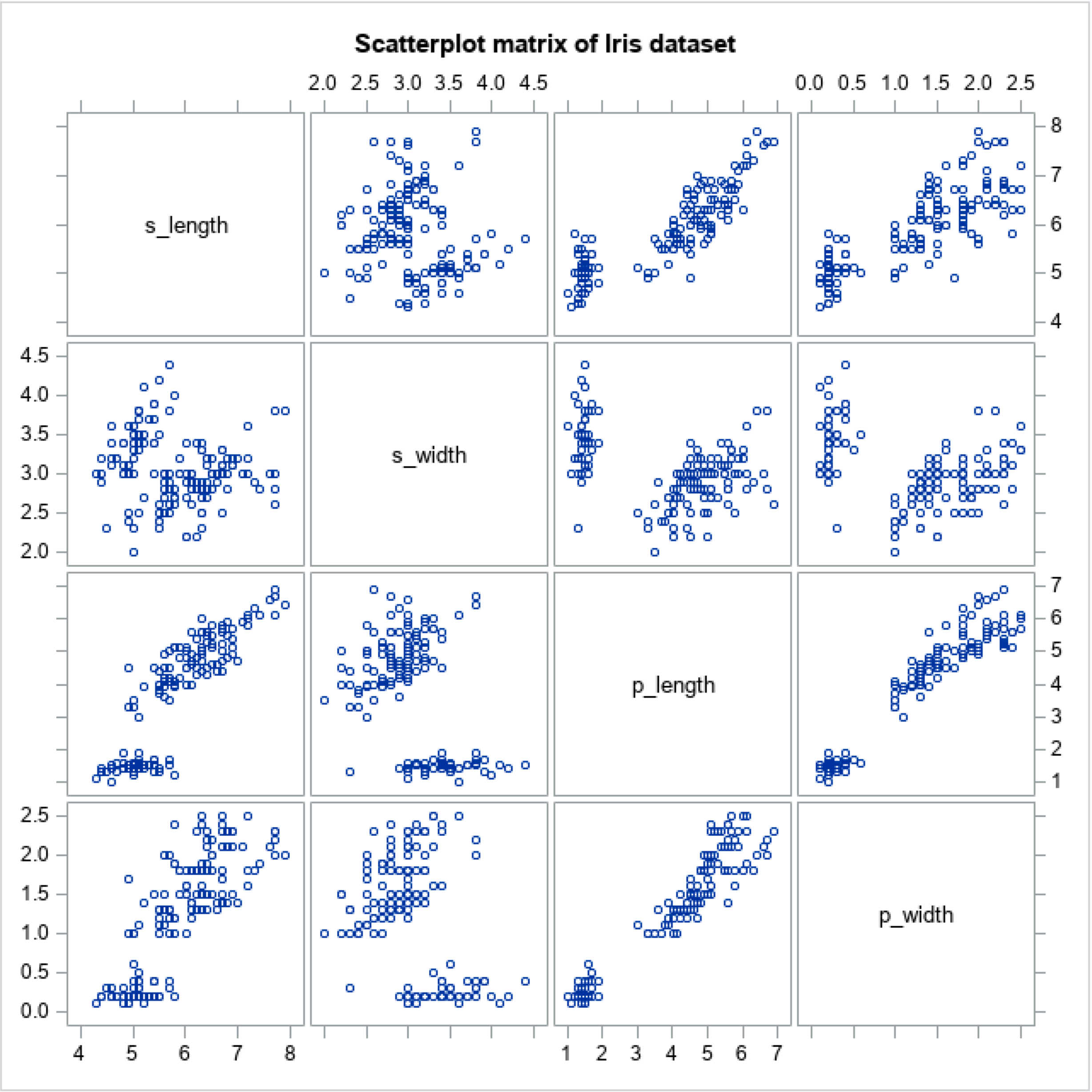
```
PROC SGSCATTER data=SASdataset;
MATRIX v1 v2 v3 ... ;
RUN;
```

Scatterplot matrix

```
proc sgscatter data=iris;
title 'Scatterplot matrix of Iris dataset';
```

```
matrix s_length s_width p_length p_width;
run;
title;
quit;
```

Iris matrix



matrix.png

Plots by a Categorical Variable

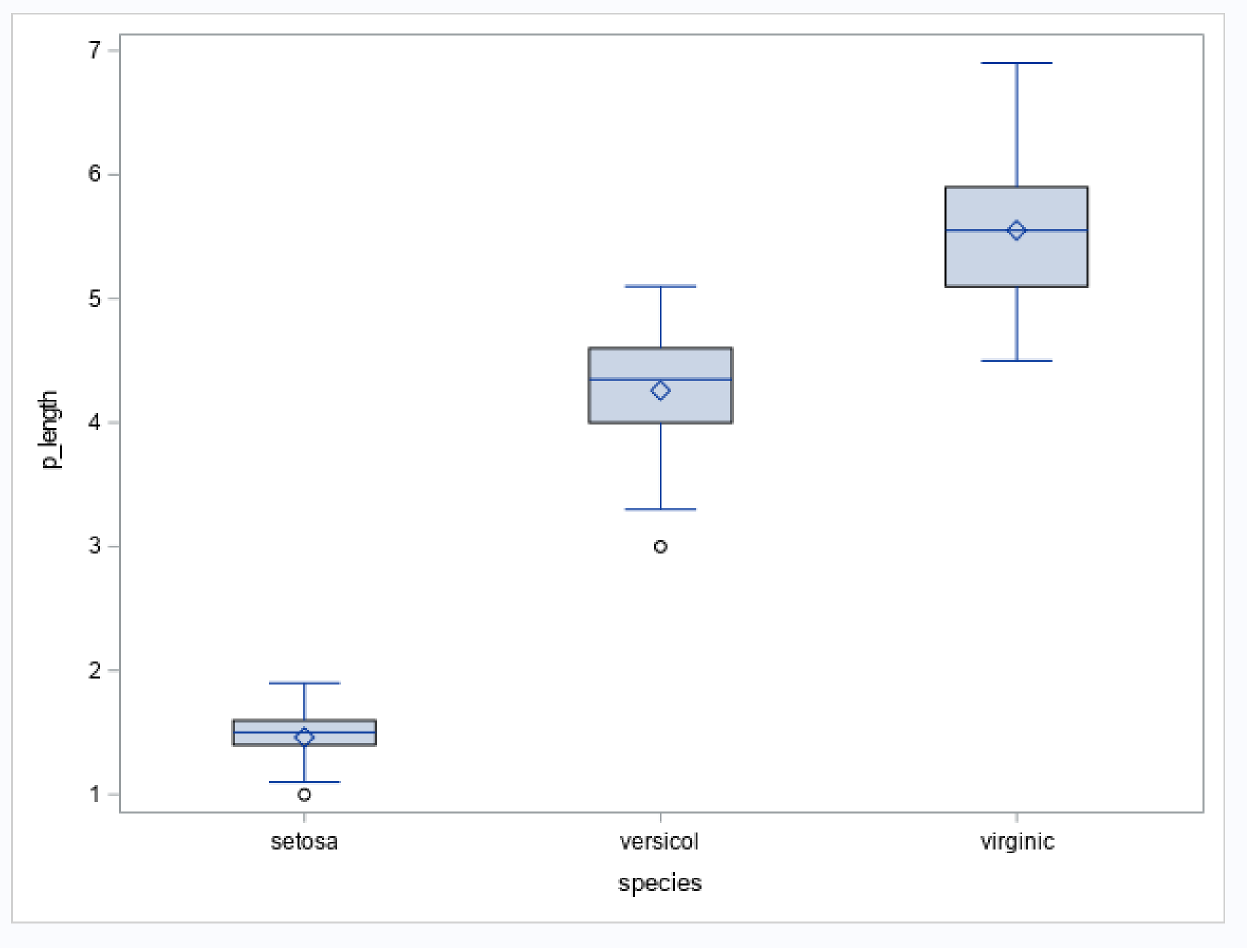
Creating plots by a grouping variable is easily done in PROC SGPLOT for plots by a categorical variable

```
proc sgplot;
HBOX var / category=var;
run;
```

Boxplot Petal Length by Species

```
proc sgplot data=iris;
vbox p_length / category=species;
run;
quit;
```

Boxplot Petal Length by Species



vbox bycat.png

Barplots

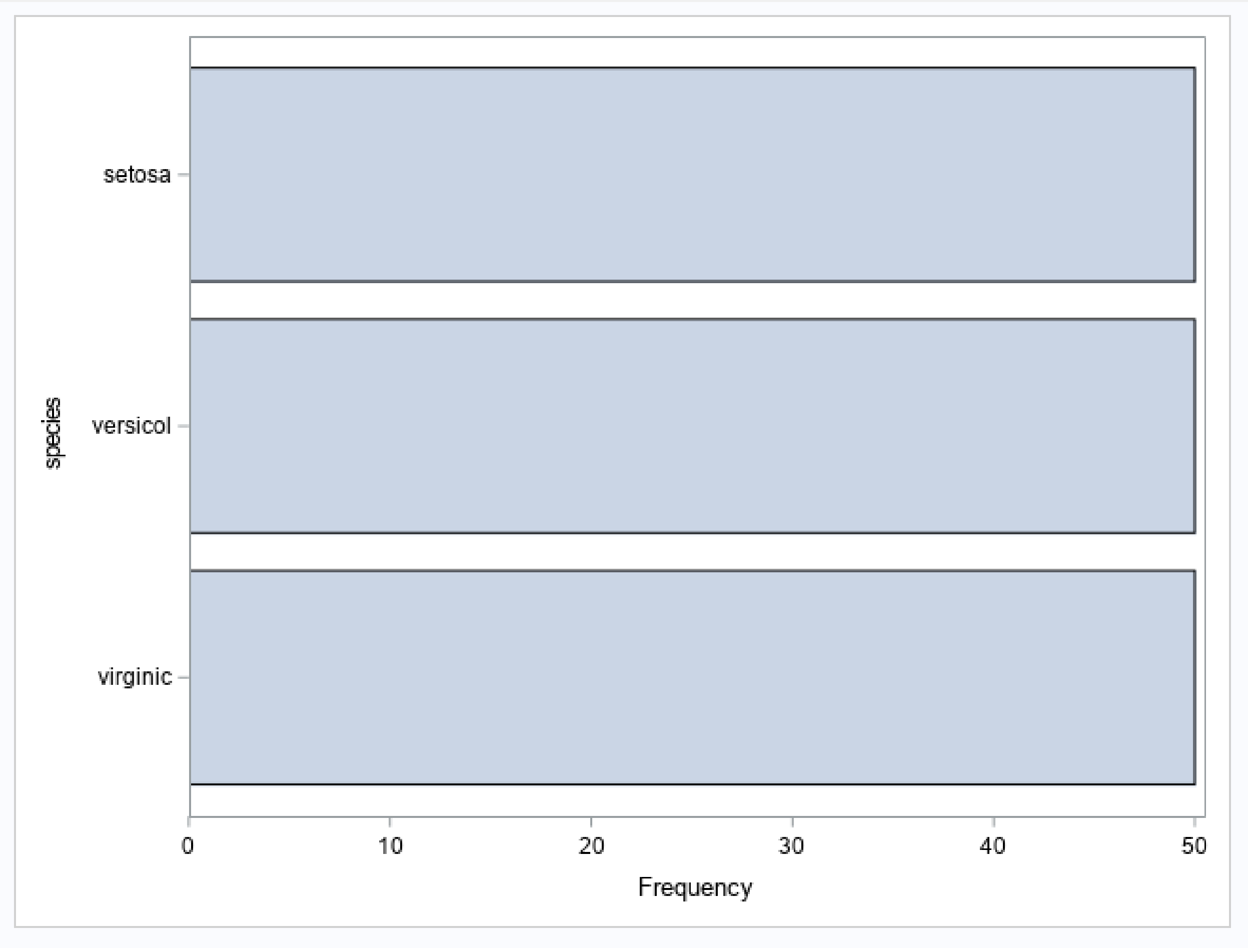
With HILINE/VLINE (but can be used without)

```
proc sgplot data=sasdataset;
   vbar cvar / response=nvar1;
   vline cvar / response=nvar2 y2axis;
run;
proc sgplot data=sasdataset;
   hbar cvar / response=nvar1;
   hline cvar / response=nvar2 y2axis;
run;
```

Iris barplots of species

```
proc sgplot data=iris;
hbar species;
run;
```

Iris species barplot



barplot.png

Pie charts

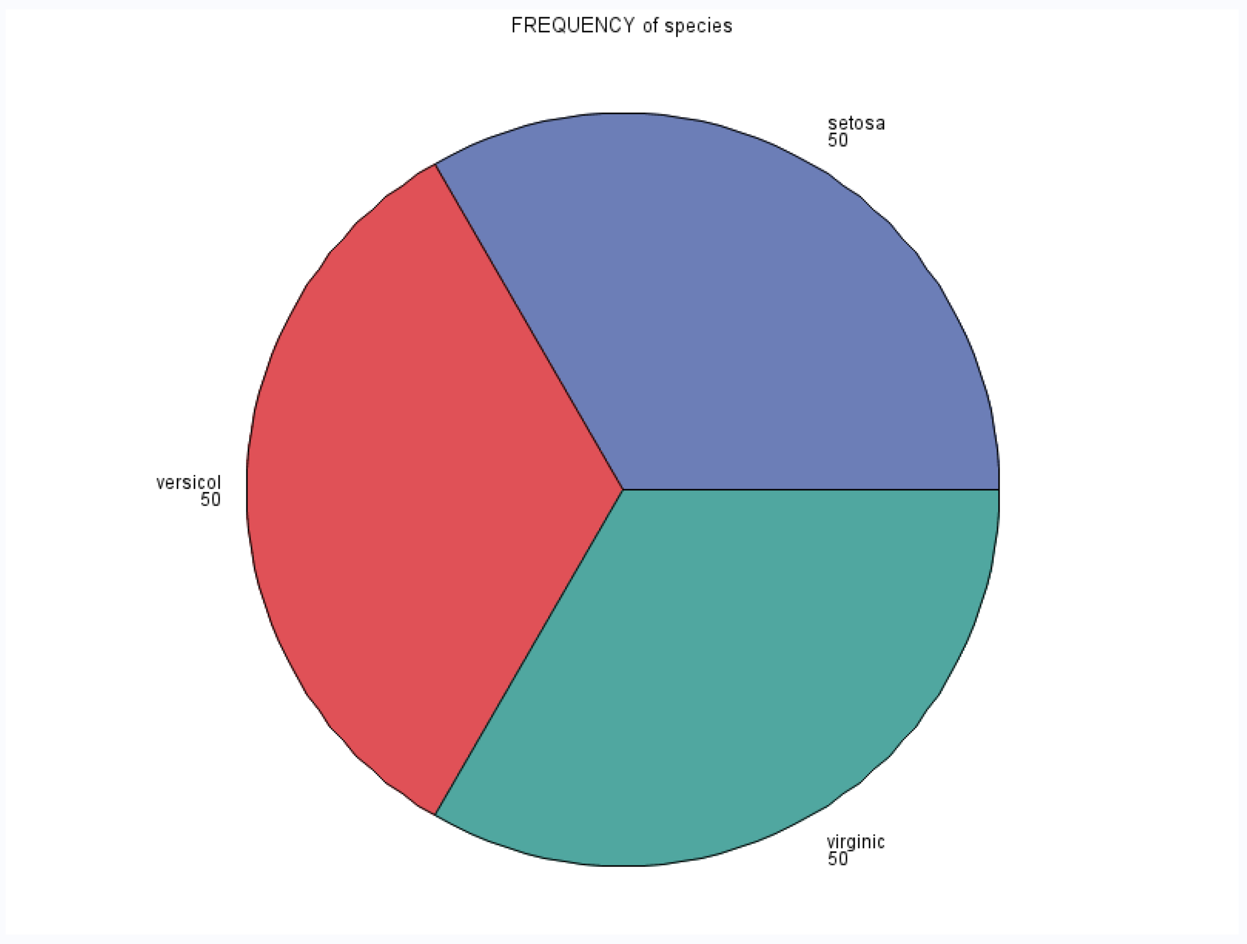
For pie charts, use PROC GCHART

```
proc gchart data=sasdataset;
  pie cvar / sumvar=nvar <options>;
run;
```

Pie chart petal width

```
proc gchart data=iris;
  pie species;
run;
quit;
```

Pie chart

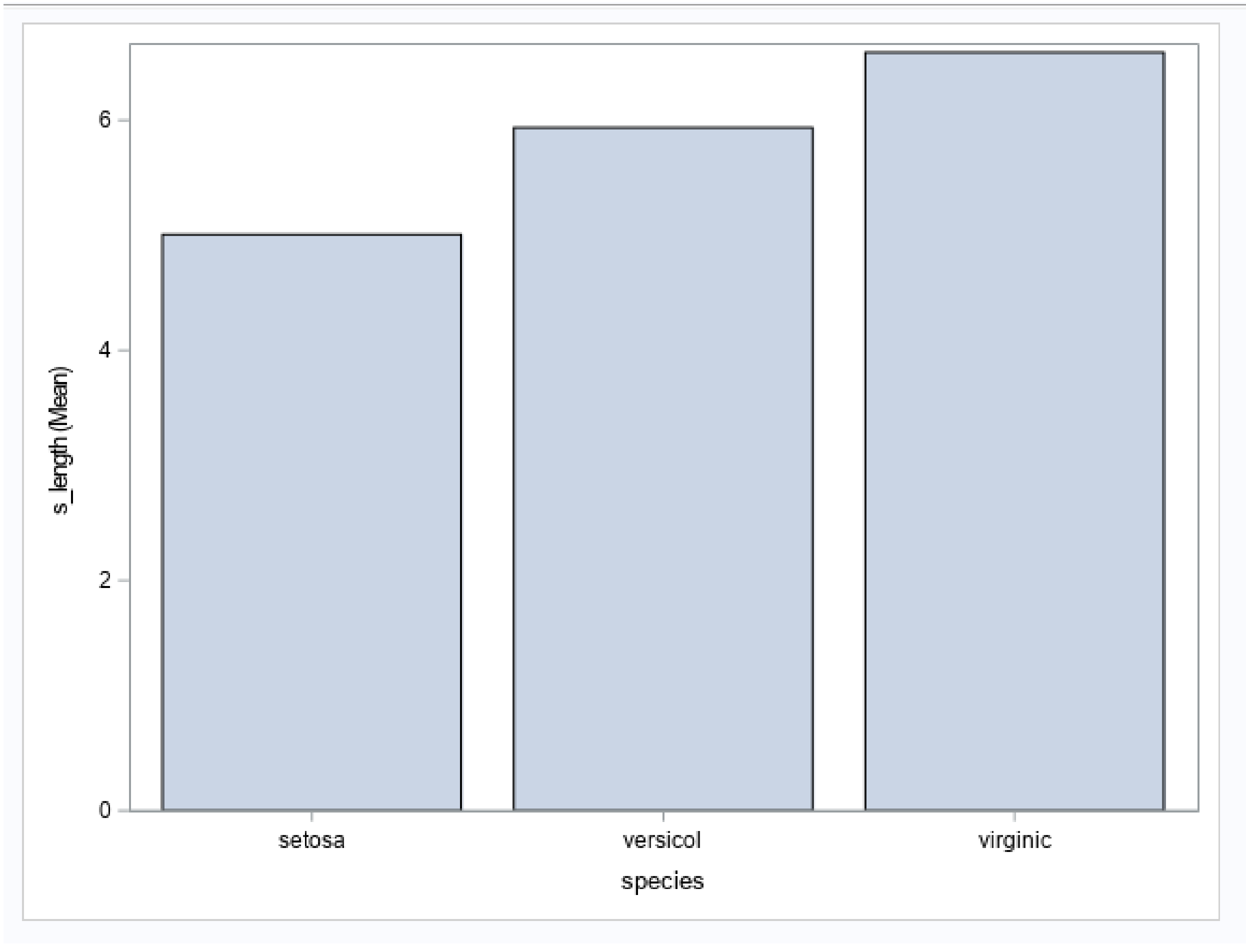


pie.png

Barplot of means (sepal lengths)

```
proc sgplot data=iris;
vbar species / response=s_length stat=mean;
run;
```

Barplot of mean sepal length



meanbarplot.png

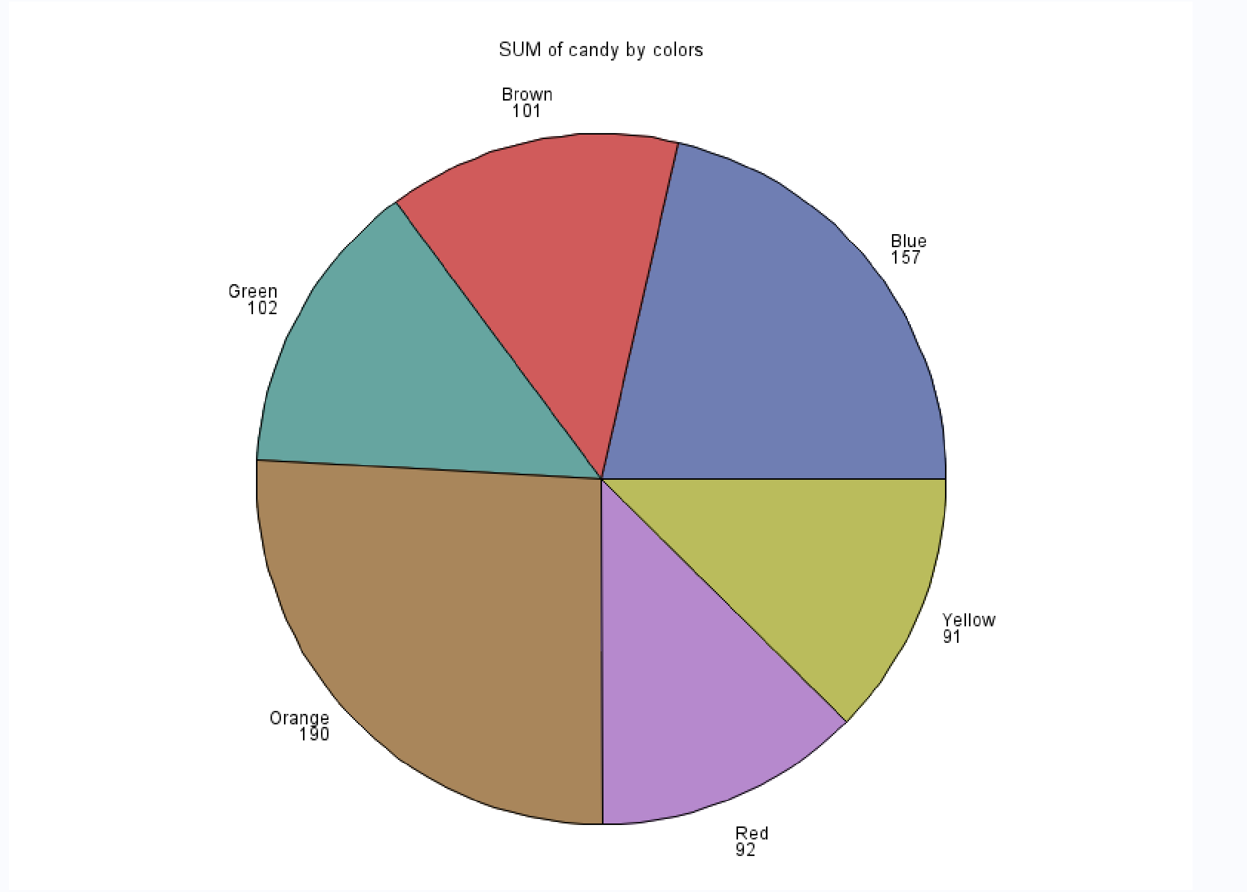
One last pie chart (yuk)

I *really* hate these things (because they are easy to manipulate) but this one is M&Ms, so chocolate makes it ok

(but seriously, avoid them if possible)

```
data mandms;
input colors$ candy @@;
cards;
Red 92 Blue 157 Green 102 Orange 190 Yellow 91 Brown 101
;
proc print data=mandms;
run;
proc gchart data=mandms;
pie colors / sumvar=candy;
run;
```

CHOCOLATE


SUM of candy by colors
Brown 101
Blue 157
Green 102
Yellow 91
Orange 190
Red 92