Scatterplots: Basics, enhancements, problems and solutions

Peter L. Flom

Peter Flom Consulting, LLC

PhilaSUG, June, 2016

Outline

Outline

Outline

Outline

Outline

Introduction

In this paper, I discuss scatterplots. I start with a very basic example, and then illustrate some enhancements. Next, I show some problems that can occur, and illustrate some solutions.

The SG PROCS

With the new SG PROCs, introduced as experimental PROCS in v. 9 of SAS, and now production, SAS allows us to make good scatterplots relatively easily. However, there are many options, and applying them well is not always obvious. And, for specialized purposes, the SGRENDER PROC can produce highly customized graphics, but its use is not entirely straightforward.

Outline

Introduction to SGPLOT

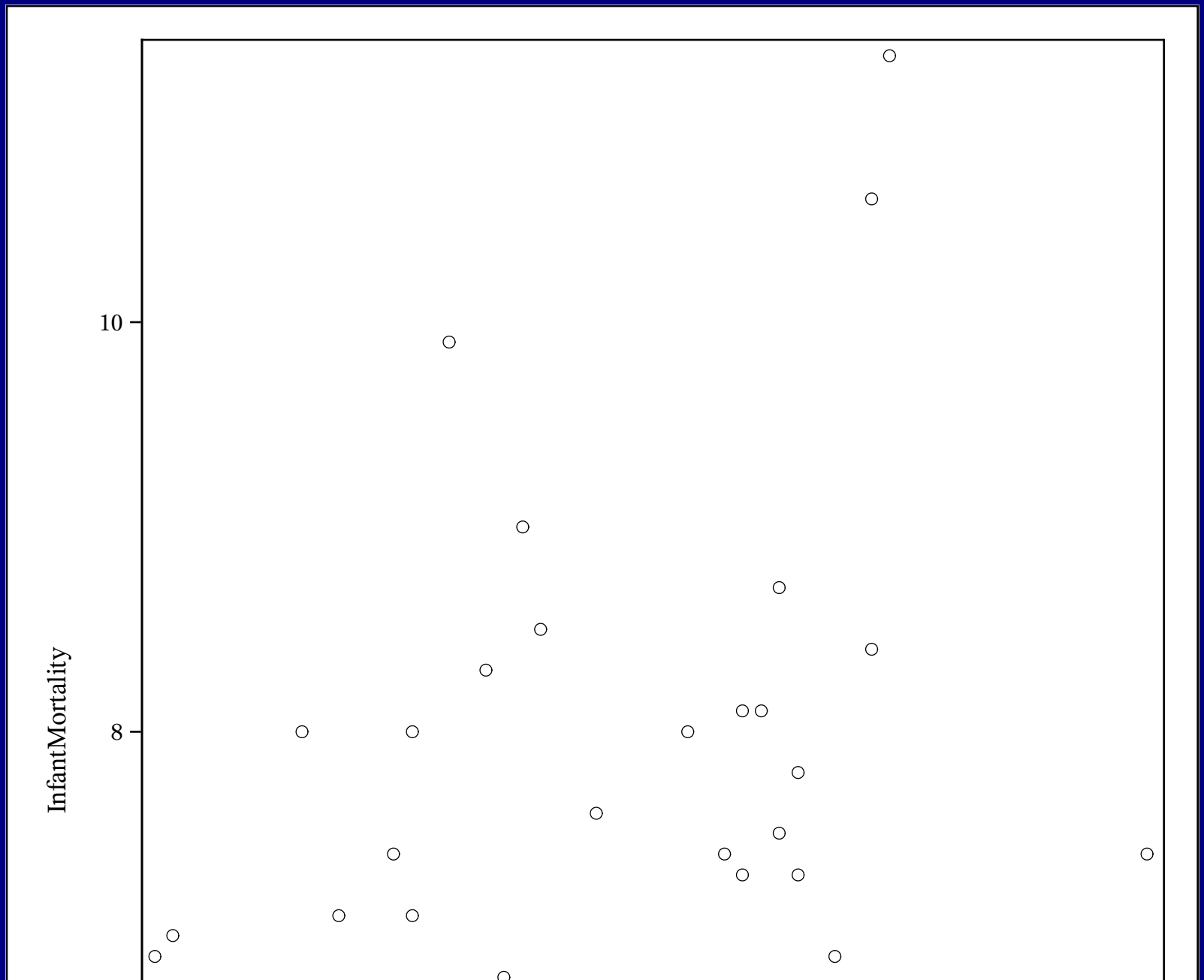
The PROC for basic scatterplots is PROC SGPLOT. Scatterplot matrices can be generated with PROC SGPANEL but I will not discuss that in this paper. Rather than list a lot of the options and syntax for this PROC (all of which can be looked up) I will give examples.

A starting example

As a starting example, let's plot unemployment rate and infant mortality for each of the 50 states plus the District of Columbia. This can be done with the following code (assuming the data have been read in).

```
ods pdf file = 'filename.pdf';
proc sgplot data = UnempIM;  *STARTS THE PROC;
 scatter x = Unemployment y = InfantMortality;
  *CREATES A PLOT, NOTE THE USE OF X = AND Y =;
run;
ods pdf  close;
```

First scatterplot



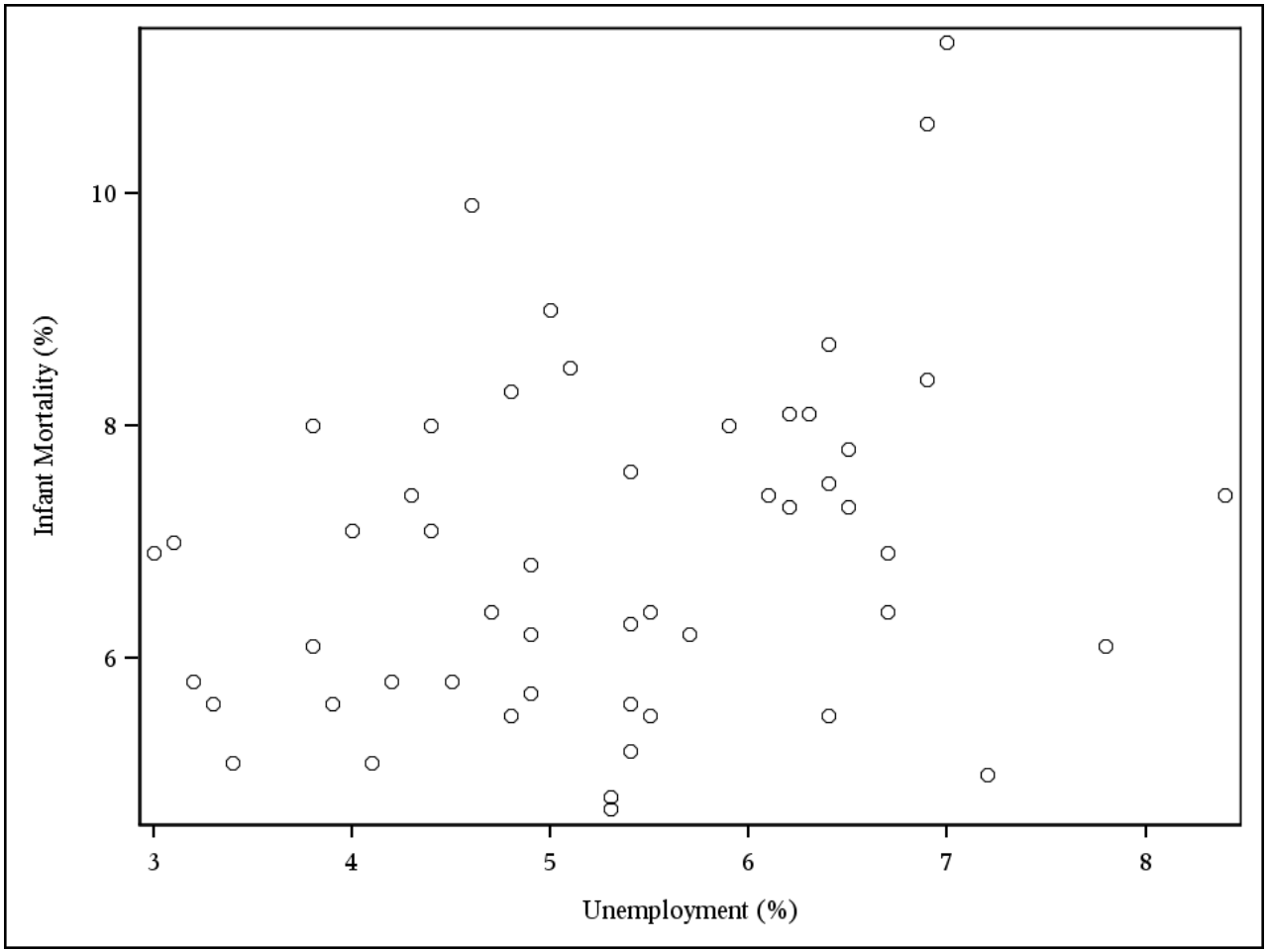
Clarifying axes

Next, we should probably make the axes clearer:

```
ods pdf file = 'filename2.pdf';
proc sgplot data = UnempIM;
 xaxis label = "Unemployment (%)";
  *THIS SHOULD BE SELF EXPLANATORY,
   THERE ARE OTHER AXIS OPTIONS AS WELL;
  yaxis label = "Infant Mortality (%)";
  scatter x = Unemployment y = InfantMortality;
run;
ods pdf close;
```

Scatterplot with axes fixed

This creates figure



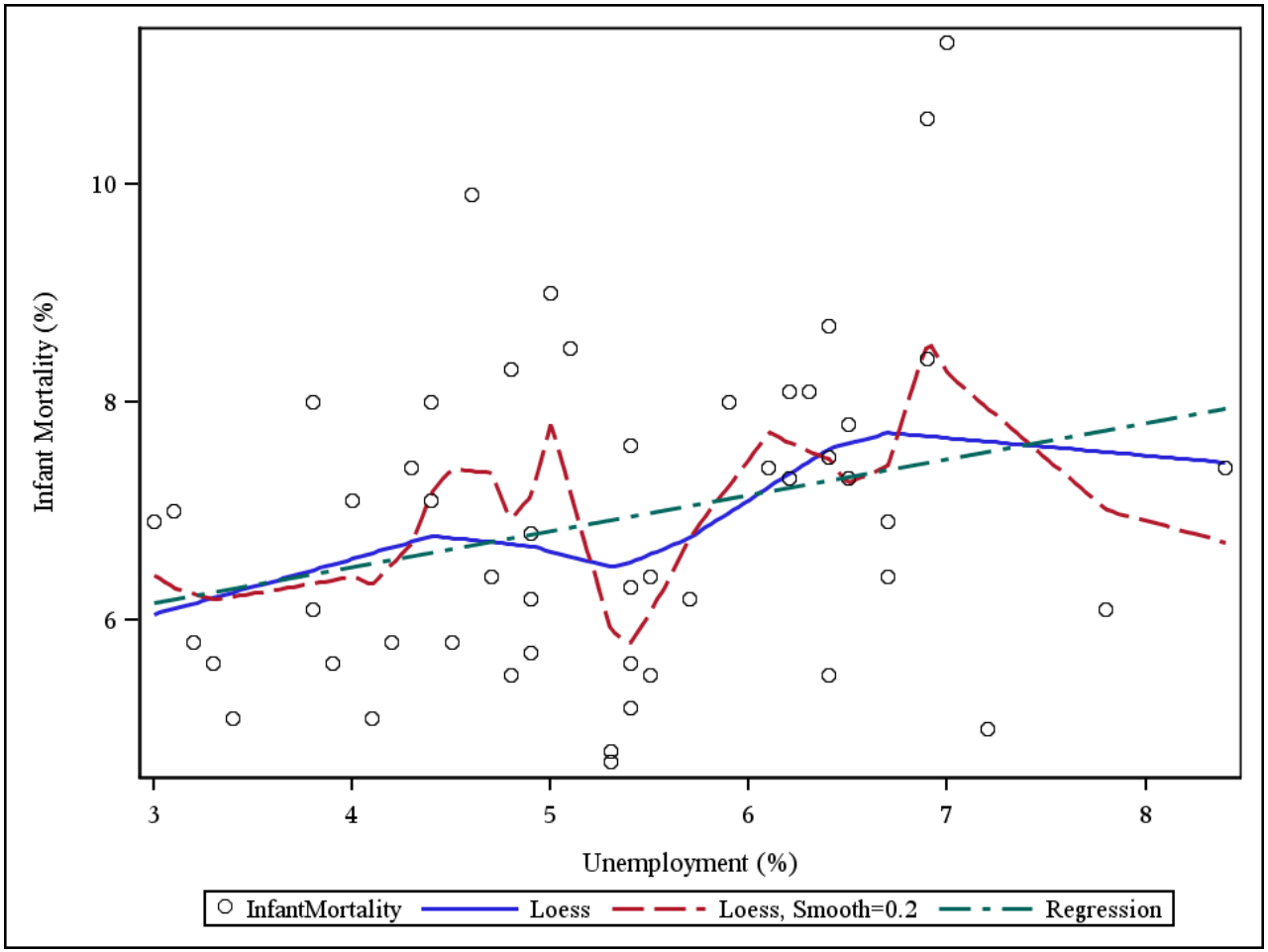
Adding information

The scatterplot isn't bad, but we can easily include more information. One thing we might want to do is add a smoothed line for the relationship between the two variables; in fact, we might want more than one, with different amounts of smooth. We can add loess lines as follows:

Code for fixing axes

```
proc sgplot data = UnempIM;
 xaxis label = "Unemployment (%)";
 yaxis label = "Infant Mortality (%)";
 scatter x = Unemployment y = InfantMortality;
 loess x = Unemployment y = InfantMortality
    /nomarkers;
 loess x = Unemployment y = InfantMortality
   /smooth = 1 nomarkers;
 *LOESS WORKS ON THE SAME DATA AS SCATTER,
  SMOOTH CAN BE ADJUSTED.
  NOMARKERS PREVENTS SAS FROM PLOTTING
   EACH POINT 3 TIMES;
run;
```

Scatterplot with smooths



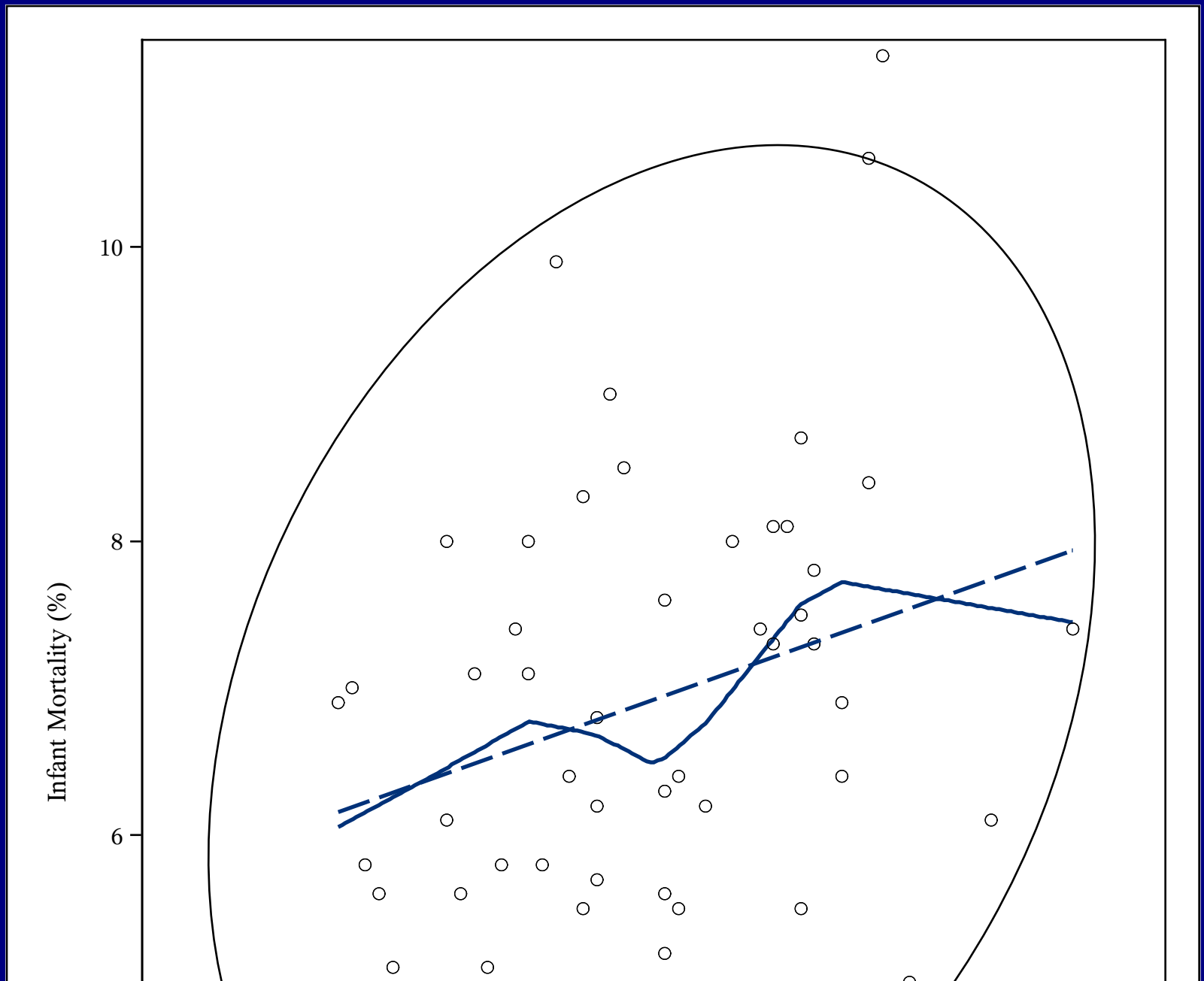
Adding an ellipse

And we might also want to add a confidence ellipse around the points:

```
proc sgplot data = UnempIM;
 xaxis label = "Unemployment (%)";
 yaxis label = "Infant Mortality (%)";
 scatter x = Unemployment y = InfantMortality;
 loess x = Unemployment y = InfantMortality
    /nomarkers;
 loess x = Unemployment y = InfantMortality
    /smooth = 1 nomarkers;
  ellipse  x = Unemployment y = InfantMortality;
 *ELLIPSE, LIKE LOESS, OPERATES ON THE SAME DATA;

run;
```

Scatterplot with ellipse



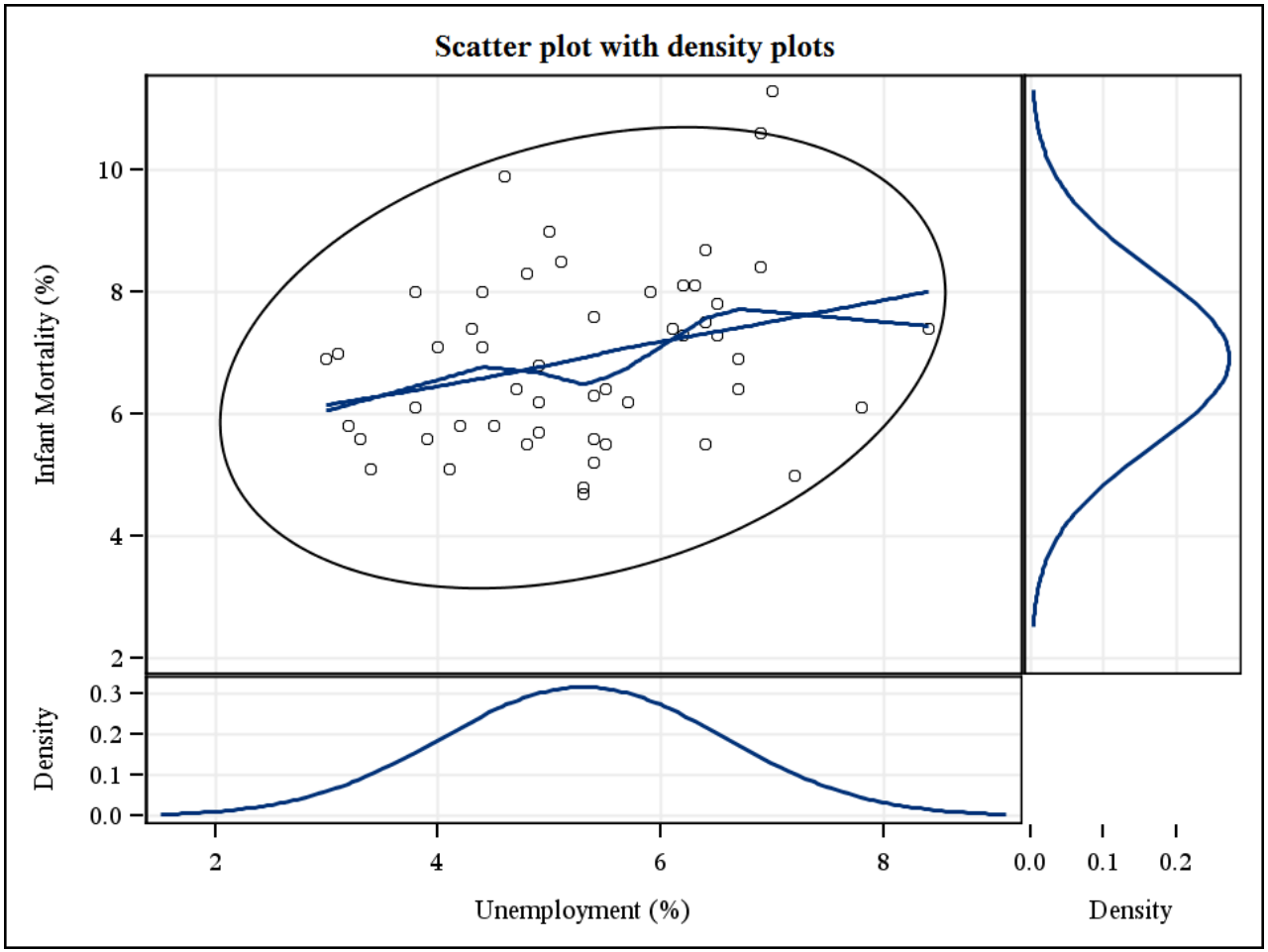
Getting fancy

That's all simple enough, and certainly adds information. But we can add more; we can look at the distribution of each variable separately and plot these in the margins. This requires use of the graph template language.

Introduction to graph template language

The graph template language allows very fine control over every aspect of a graph. It is also the language that SAS uses to create graphics. To use the GTL, you begin with a PROC TEMPLATE. You then use PROC SGRENDER to render that template. A very good reference that includes many starting templates is Kuhfeld. In addition, the GTL Users' Guide and GTL Reference Manual are quite useful. There are a great many possible options, and I will cover only a few.

A fancy scatterplot



PROC TEMPLATE for a fancy graph (part 1)

```
proc template;
 define statgraph scatdens2;
 begingraph;   *BEGIN DEFINING THE GRAPH;
   entrytitle "Scatter plot with density plots";
    *CREATE A TITLE;
   layout lattice/columns = 2 rows = 2
    columnweights = (.8 .2) rowweights = (.8 .2)
      columndatarange = union rowdatarange = union;
   *LAYOUT LATTICE...SETS UP A GRID OF GRAPHS;
   *COLUMNWEIGHTS AND ROWWEIGHTS SETS
    THE RELATIVE SIZE OF THE INDIVIDUAL
     COLUMNS AND ROWS;
```

PROC TEMPLATE for a fancy graph (part 2)

```
 columnaxes;
    columnaxis /label = 'Unemployment (%)'
     griddisplay = on;
columnaxis /label = '' griddisplay = on;
  endcolumnaxes;
  *COLUMNAXES SETS PARTICULAR
    CHARACTERISTICS OF COLUMNS;
  *THE SECOND ONE HAS NO LABEL (NONE WOULD FIT)
  rowaxes;
    rowaxis /label = 'Infant Mortality (%)'
       griddisplay = on;
rowaxis /label = '' griddisplay = on;
  endrowaxes;
```

PROC TEMPLATE for a fancy graph (part 3)

```
          layout overlay;
           *STARTS THE ACTUAL GRAPHING OF DOTS AND SU
            scatterplot x = unemployment
             y = infantmortality; *GRAPHS THE DOTS;
        loessplot x = unemployment y = infantmortalit
               /nomarkers;
        loessplot x = unemployment y = infantmortalit
                /smooth = 1 nomarkers;
    ellipse  x = unemployment y = infantmortality
               /type = predicted;
         endlayout;
     densityplot infantmortality/orient = horizontal;
     densityplot unemployment;
     endlayout;
endgraph;
end;
run;
```

PROC SGRENDER for the template

```
proc sgrender data = UnempIM template = scatdens2;
  *NOW WE RENDER THE TEMPLATE WE CREATED;
run;
```

SGDESIGN

Rick Wicklin of SAS has pointed out that, in his words ‘For people who do not like to program, the %sgdesign macro brings up a GUI interface that allows you to create the second image using drag-and-drop and menus. For details and examples, see

`support.sas.com/documentation/cdl/en/grstatdesignug`

but I have not used this feature.

Outline

Introduction to overplotting

Although scatterplots are very useful, they can have problems. The most important of these is *overplotting* which occurs when more than one observation has the same values. The proper solution depends on the type and amount of overplotting. In some cases, overplotting is due to the discrete nature of the way the data are recorded; for example, when asked their weights and heights, people respond with weights in pounds (or kilograms) and heights in feet and inches (or centimeters), rounded to the nearest unit, or sometimes even to the nearest multiple of 5. In other cases, there is so much data that overplotting occurs even when the data are recorded accurately to several decimal places.

Some solutions to overplotting

In the first situation, one excellent solution is *jittering* or adding small amounts of random noise to the data. In the latter case, there are various solutions. If the data set is not enormous, changing the plotting character or its size may be enough. If there is an enormous number of points, then we can change to a parallel box plot.

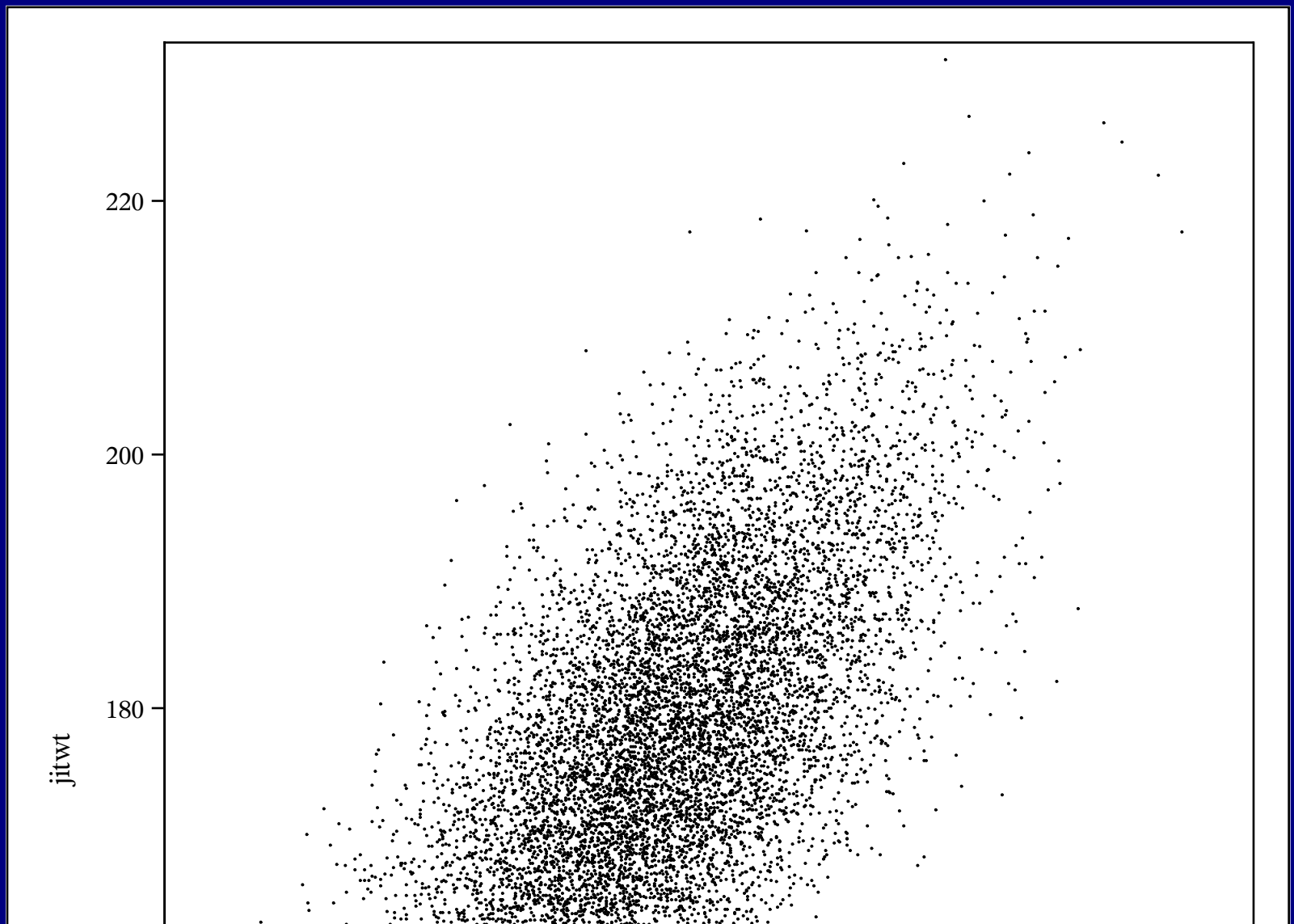
A data set

Here I create a data set of actual heights and weights (realht and realwt), rounded to the nearest inch and pound (ht and wt). I also jitter these (jitht and jitwt).

```
data htwt;
  do i = 1 to 50000;
  realht = rannor(1828282)*3 + 66;
  realwt = realht*2 + realht**2*.01
    + 10*rannor(12802194);
  ht = round(realht,1);
  wt = round(realwt,1);
  jitht = ht+rannor(1818282);
  jitwt = wt+rannor(199328282);
  output;
end;
run;
```

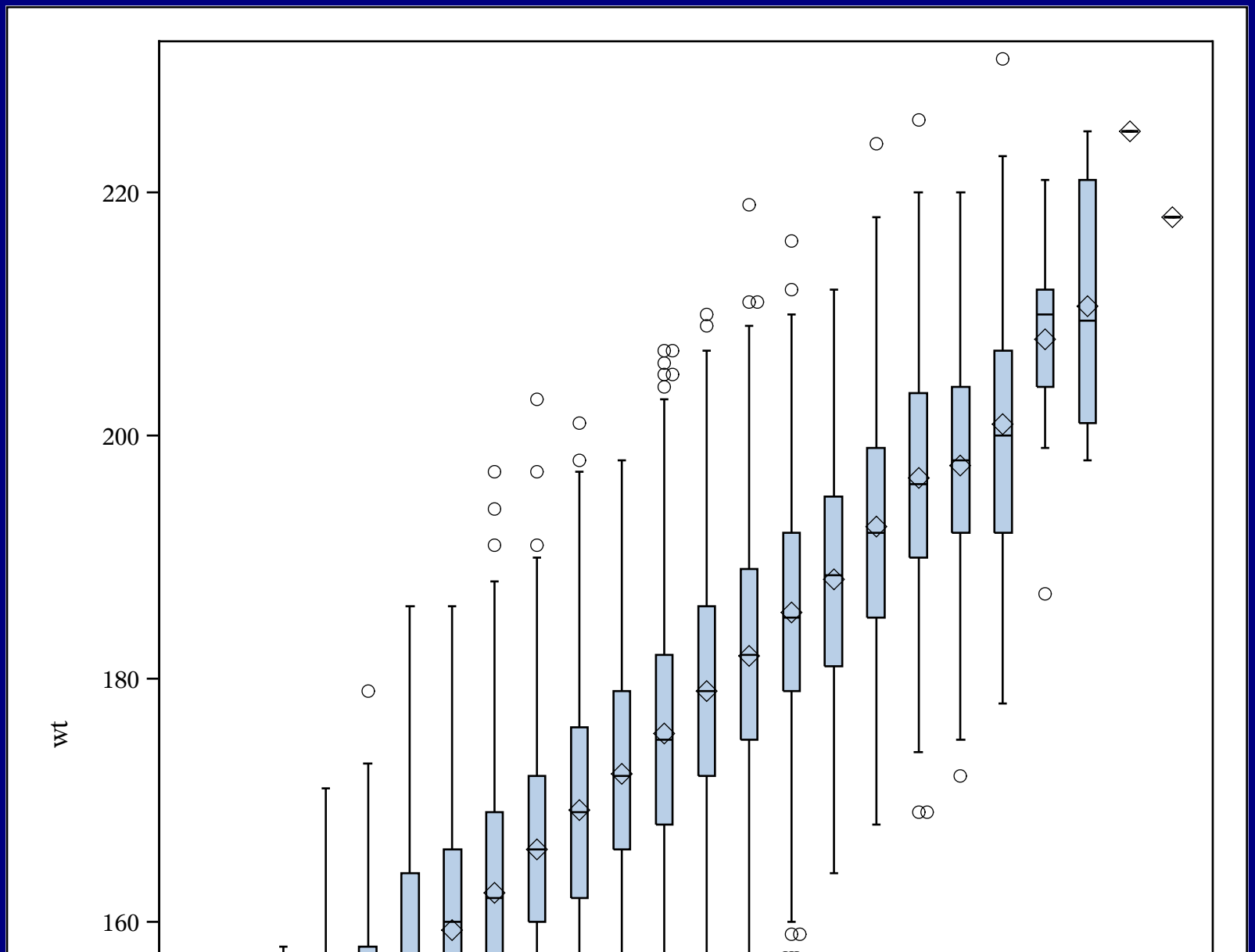
Moderate overplotting due to discretization

If we have a data set of 500 people with rounded height and weight, the plot will not show all the points clearly



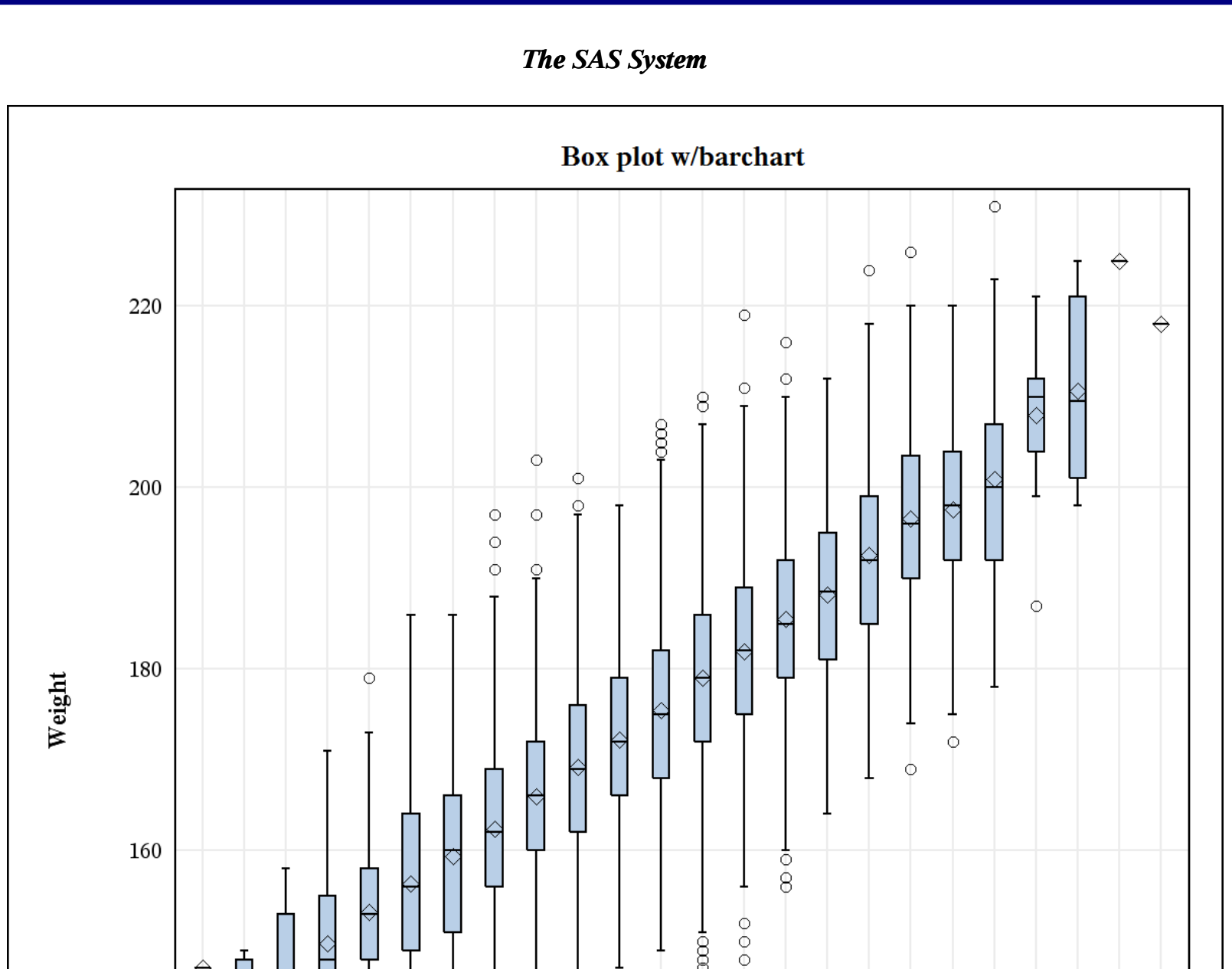
Jittering for moderate overplotting

Here, simply jittering the data works well



More severe overplotting

However, if we have 10,000 points, jittering is not enough

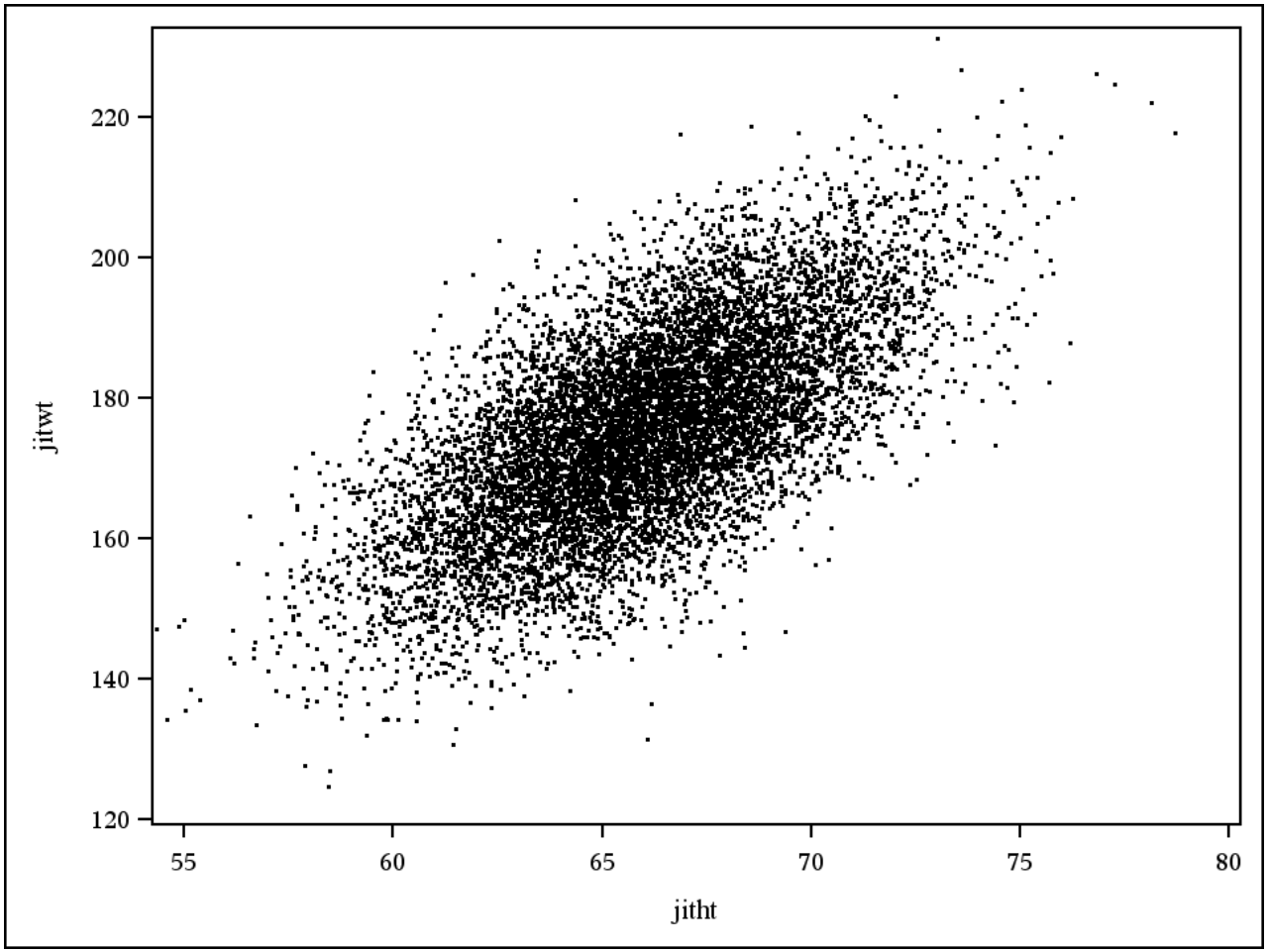


Changing the plotting symbol

We can change the plotting character and its size with the following program:

```
ods pdf file ="filename6.pdf";
proc sgplot  data = htwt;
   scatter x = jitht y = jitwt/
    markerattrs = (size = 2 symbol = circlefilled);
   where i < 10000;
run;
ods pdf close;
```

Scatterplot with different symbol

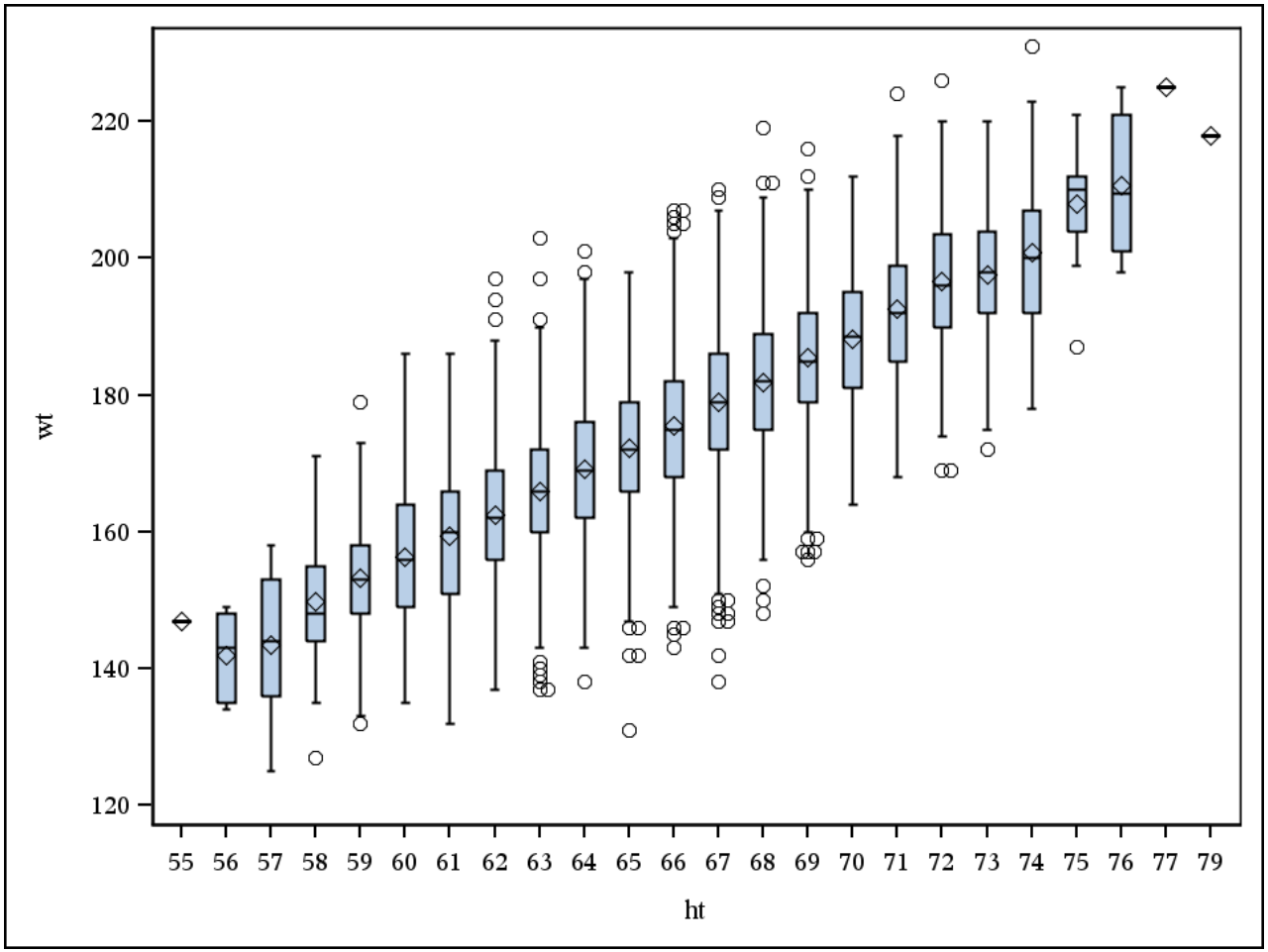


Parallel boxplots as an alternative

An alternative is to abandon the scatterplot and use parallel boxplots:

```
ods  pdf file ="filename7.pdf";
proc sgplot  data = htwt;
  vbox wt/category = ht spread;
   *THE SPREAD OPTION PREVENTS OVERLAP;
  where i < 10000;
run;
ods pdf close;
```

Parallel boxplots



Getting fancy again

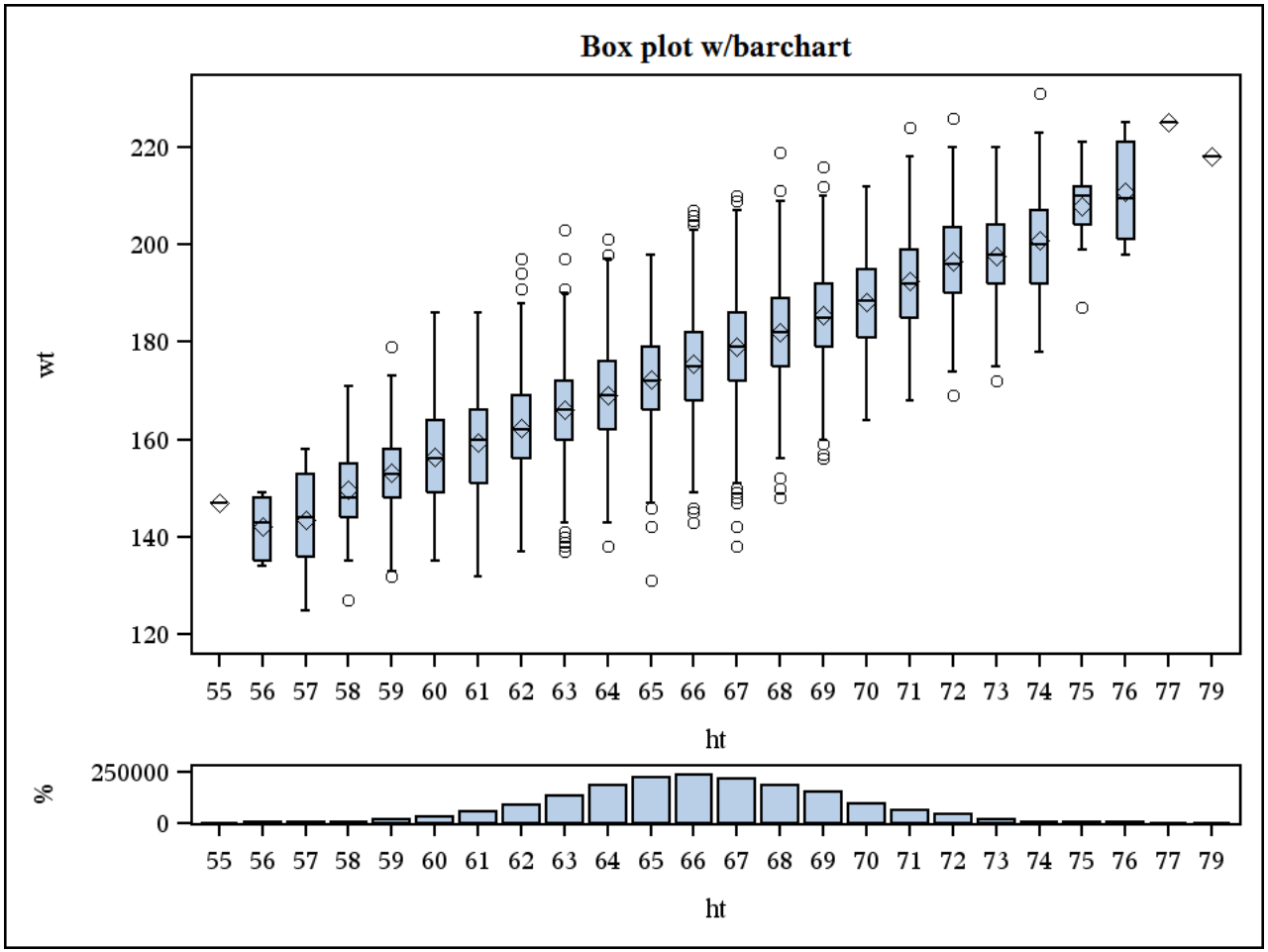
One problem with this is that it does not give any indication of the density of height. Using the GTL, we can add a histogram:

```
proc template;
 define statgraph fancybox;
 begingraph;
   entrytitle "Box plot w/histogram";
   layout lattice/rows = 2 columns = 1
    order = columnmajor rowweights = (.8 .2);
    columnaxes;
     columnaxis /griddisplay = on;
columnaxis /label = '' griddisplay = on;
   endcolumnaxes;
      boxplot x = ht y = wt;
   histogram ht;
endlayout;
endgraph;
end;
run;
```

Rendering the plot

```
ods  pdf file ="filename8.pdf";
proc sgrender data = htwt template = fancybox;
run;
ods pdf close;
```

Parallel boxplot with histogram



Outline

Summary

Scatterplots are a very valuable graphical tool. The SG PROCs in SAS allow many scatterplots to be produced easily, and the graph template language allows very fine control over all aspects of a graph.

Contact information

Peter L. Flom
515 West End Ave
Apt 8C
New York, NY 10024
peterflomconsulting@mindspring.com
(917) 488 7176
www.statisticalanalysisconsulting.com

SAS stuff

SAS® and all other SAS Institute Inc., product or service names are registered trademarks ore trademarks of SAS Institute Inc., in the USA and other countries. ® indicates USA registration. Other brand names and product names are registered trademarks or trademarks of their respective companies.