How the DATA step works, PROC SGPLOT

Shannon Pileggi

STAT 330

OUTLINE

How DATA steps work

The purpose of the DATA step is *read*, *modify*, or *create* data. How does it work? SAS executes the DATA step line by line **and** observation by observation.

SAS has a built in loop!

- all lines of the data step are executed on observation 1
- all lines of the data step are executed on observation 2
- all lines of the data step are executed on observation 3
- etc.

Example data

SAS Code

```
DATA class;
  INPUT name $ GPA dob MMDDYY10. salary COMMA8.;
  DATALINES;
  Bill  3.4  10/13/1995 $18,000
  Susan 2.7  6/24/1993  $535,000
  ;
RUN;
```

SAS Code

DATA step actions

When you submit a DATA step, it goes through the

1. Compile phase Jump to it
2. Execution phase Jump to it

http://support.sas.com/documentation/cdl/en/basess/58133/HTML/default/viewer.htm#a001290590.htm

Compile phase

During the compile phase, SAS compiles statements, checks syntax, and creates:

1. descriptor information More info
2. an input buffer More info
3. a Program Data Vector (PDV) More info

Back to DATA step actions

Descriptor information

- Descriptor information is information that SAS creates and maintains about each SAS data set
- This includes data set attributes and variable attributes
- for example, the date data set was created, names and types of variables.
- To see some descriptor information, submit

```
PROC CONTENTS DATA = class; RUN;
```

Back to compile phases

Input buffer

- logical area in memory into which SAS reads each record of raw data when executing an input statement
- only used for reading "raw" data (not a SAS data set)

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
B	i	l	l		3	.	4		1	0	/	1	3	/	1	9	9	8		$	1	8	,	0	0	0			

Back to compile phases

Program Data Vector - Compile phase

- The PDV is as a virtual row of the SAS data set.
- The PDV begins with all missing entries.
- Two temporary variables are created during the processing of every data step as part of the PDV.
  1. `_N_` is the DATA step iteration counter
  2. `_ERROR_` indicates the data error status

     0 = no data error on that record
     1 = at least one data error on that record

_N_	_ERROR_	name	GPA	dob	salary
1	0		.	.	.

PDV in execution phase Back to compile phases

Execution phase

- counts iterations of the DATA step (at the top)
- sets all values in PDV to missing
- reads input record (if relevant)
- executes additional manipulation/computation

Program Data Vector in the Execution Phase

- writes record to output data set

Back to DATA step actions

Program Data Vector - Execution phase

- PDV entries are specified one column at a time.
- the PDV copies its contents into the SAS data set matrix filling a complete row (observation).
- the PDV empties and the process begins again.

_N_	_ERROR_	name	GPA	dob	salary
1	0	Bill	3.4	13069	18000

PDV in compile phase Back to DATA step actions

Running the SAS DATA step

Compile SAS examines program for syntax errors and creates PDV with values set to missing

Execution 1 SAS reads Bill's record from the input buffer to the PDV one variable at a time

Execution 2 SAS calculates the day of the week Bill was born on and stores it in the PDV

Execution 3 SAS writes values in the PDV to the first observation in the `class` data set

Execution 4 SAS returns to the the top of the DATA step and resets values in the PDV to missing, `_N_` changes to 2

Execution 5 SAS reads Susan's record from the input buffer to the PDV one variable at a time

Execution 6 SAS calculates the day of the week Susan was born on and stores it in the PDV

Execution 7 SAS writes values in the PDV to the second observation in the `class` data set

Cycle Ends

Discussion

Suppose you run a program that causes three DATA step errors where the last iteration ends on an error. At the end of the data step, what is the value of the automatic variable _ERROR_?

0 1 2 3

Example error in PDV

SAS Code

```
DATA class;
  INPUT name $ GPA dob MMDDYY10. salary COMMA8.;
  DATALINES;
  Bill  3.4 10/13/1995  $18,000
  Susan 2.7 06/24/199  $535,000
  ;
RUN;
```

SAS Code

SAS log

```
NOTE: Invalid data for dob in line 106 13-22.
RULE:      ----+----1----+----2----+----3----+----4----+----5----+--
106          Susan 2.7 06/24/199  $535,000
name=Susan GPA=2.7 dob=. salary=53500 _ERROR_=1 _N_=2
```

SAS log

Discussion

Which output will this code generate?

SAS Code

```
DATA one;
  OUTPUT;
  DO i = 1 to 4;
        x = i**3;
        OUTPUT;
  END;
  OUTPUT;
RUN;
PROC PRINT DATA = one;
RUN;
```

SAS Code

Ouptut 1

i	x
.	.
1	1
2	8
3	27
4	64
5	64

Ouptut 2

i	x
1	1
2	8
3	27
4	64

Ouptut 3

i	x
1	1
2	8
3	27
4	64
4	64

How the DATA step works

PROC SGPLOT

PROC SGPLOT

PROC SGPLOT is a stand alone procedure for making graphics.

- `HBAR/VBAR` = bar chart
- `HISTOGRAM` = histogram
- `VBOX/HBOX` = box plot
- `SCATTER` = scatter plot
- `SERIES` = time series plots

Getting started

The `O2012.sas7bdat` data set contains information on athletes that won medals in the 2012 olympics, including the country that they represented, the sport they competed in, and the total number of medals earned.

On your own:

1. Write a libname statement to access the `O2012` data set.
2. Examine the contents of the `O2012` data set. How many observations are there?
3. Explore the data set.

Review question

Which code can be used to obtain the number of athletes who won medals in `Swimming` for the participating countries?

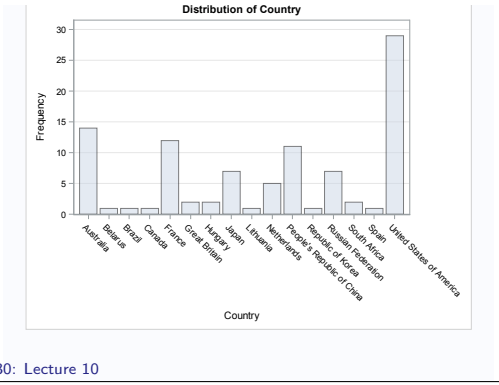
1. `proc freq data=flash.o2012; tables country; where sport="Swimming"; run;`
2. `proc means data=flash.o2012; var country; where sport="Swimming"; run;`
3. `proc freq data=flash.o2012; tables country; if sport="Swimming"; run;`
4. `proc freq data=flash.o2012; tables country; class sport="Swimming"; run;`
5. `proc univariate data=flash.o2012; tables sport; class country; run;`

Bar plot in PROC FREQ

SAS Code

```
PROC FREQ DATA = flash.o2012;
   TABLES country / PLOTS = freqplot;
   WHERE sport = "Swimming";
RUN;
```

SAS Code

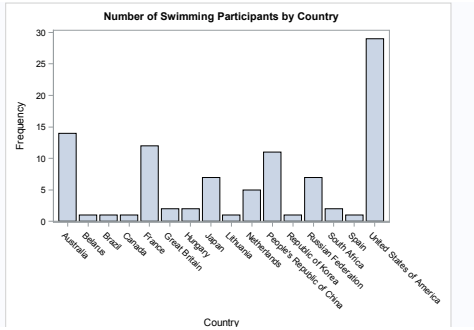


Bar plot in PROC SGPLOT

SAS Code

```
PROC SGPLOT DATA = flash.o2012;
        WHERE sport="Swimming";
        VBAR country ;
RUN;
```

SAS Code

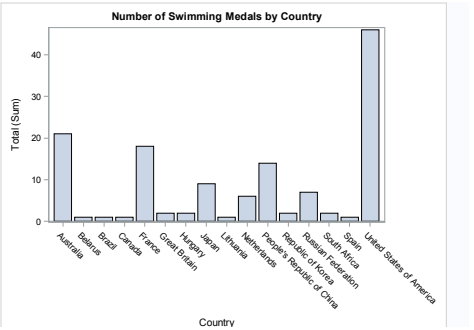


Bar plot, weighted count

SAS Code

```
PROC SGPLOT DATA = flash.o2012 ;
        where sport = "Swimming";
        VBAR country / RESPONSE = total ;
RUN ;
```

SAS Code

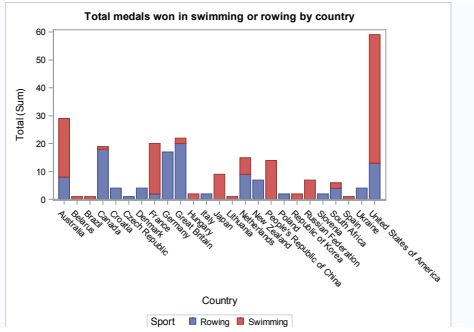


Bar plot, stacked

SAS Code

```
PROC SGPLOT DATA = flash.o2012;
        WHERE sport="Swimming" or sport="Rowing";
        VBAR country / GROUP = sport RESPONSE = total ;
RUN;
```

SAS Code

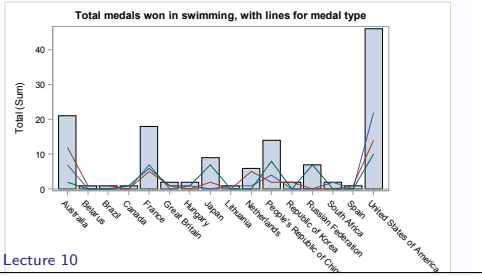


Bar plot, add vertical lines

SAS Code

```
PROC SGPLOT DATA = flash.o2012;
   WHERE sport="Swimming" ;
   VBAR country / RESPONSE = total ;
   VLINE country / RESPONSE = gold ;
   VLINE country / RESPONSE = silver ;
   VLINE country / RESPONSE = bronze ;
RUN ;
```

SAS Code

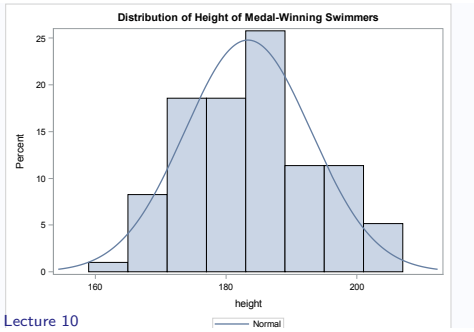


Histogram with normal curve

SAS Code

```
PROC SGPLOT DATA = flash.o2012;
   WHERE sport = "Swimming";
   HISTOGRAM height ;
   DENSITY height ;
RUN;
```

SAS Code

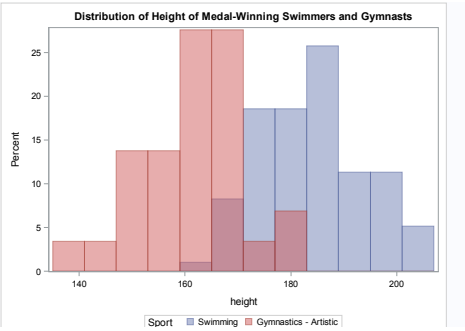


Histogram, density overlayed

SAS Code

```
PROC SGPLOT DATA = flash.o2012;
   WHERE sport="Swimming" OR sport="Gymnastics - Artistic";
   HISTOGRAM height/  GROUP = sport TRANSPARENCY = 0.5;
RUN ;
```

SAS Code



Discussion

To create overlaid histograms that shows the distribution of height and weight, you need (1/2/3); to create overlaid histograms that shows how the distribution of height among males and females you need (1/2/3).

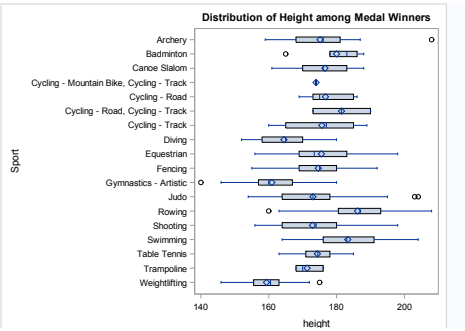
1. two HISTOGRAM statements
2. 1 HISTOGRAM statement with GROUP option
3. either

Side by Side Boxplot, with CATEGORY option

SAS Code

```
PROC SGPLOT DATA = flash.o2012;
   HBOX height / CATEGORY = sport ;
RUN ;
```

SAS Code

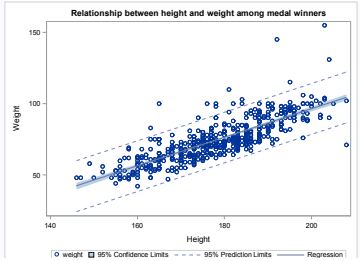


Scatterplot

SAS Code

```
PROC SGPLOT DATA = flash.o2012 ;
   SCATTER Y = weight X = height ;
   XAXIS LABEL = "Height" ;
   YAXIS LABEL = "Weight" ;
   REG Y = weight X = height / CLM CLI ;
RUN ;
```

SAS Code

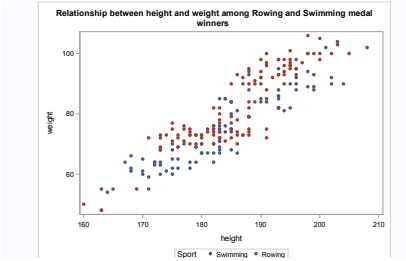


Scatterplot, points colored by categorical variable

SAS Code

```
PROC SGPLOT DATA = flash.o2012 ;
   WHERE sport = "Swimming" or sport = "Rowing" ;
   SCATTER Y = weight X = height /
      GROUP = sport
      MARKERATTRS = (symbol = CircleFilled) ;
RUN ;
```

SAS Code

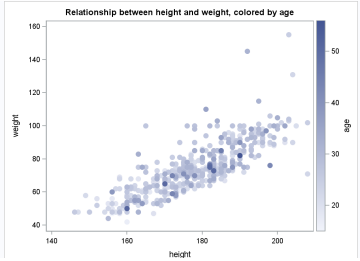


Scatterplot, points colored by quantitative variable

SAS Code

```
PROC SGPLOT DATA = flash.o2012 ;
   SCATTER Y = weight X = height /
     COLORRESPONSE = age
     MARKERATTRS = (symbol=CircleFilled size=10)
     COLORMODEL = TwoColorRamp;
RUN ;
```

SAS Code



Group vs Category Option

- In general, both `GROUP` and `CATEGORY` options can be used in the SGPLOT statements (ie, HBAR, SCATTER, etc).
- The output does look different with the two options.
- The group option results in graphical elements that have varying attributes
  - Each unique value of the grouping variable is drawn in a separate style element GraphData1 through GraphData*k*
  - Produces a legend matching graphical styles to group
- If you want **different colors** - try `GROUP`
- If you want **same color** and a legend - try `CATEGORY`
- More info: https://blogs.sas.com/content/iml/2012/08/22/categories-vs-groups-in-proc-sgplot.html