https://udrive.oit.umass.edu/statdata/spss.zip

# IBM SPSS Statistics Syntax

## Keeping Your Sanity While Managing Large Files

March 2012

# Introduction

Information presented here is intended for SPSS Statistics version 17 or higher. The Syntax and principles apply to earlier SPSS versions as well, but the Graphical User Interface items have changed significantly over releases, so these cannot necessarily be used in earlier versions.

Most things you do in SPSS Statistics have an associated set of commands (syntax). You can begin learning syntax by looking at the syntax generated by your menu choices. The generated syntax is visible in your output window:

## *Why use Syntax?*

Syntax provides a valuable record of what you did. In particular, if you use data transformations such as Compute or Recode from the menu system, it is difficult to trace how new variables are defined or how existing variables are changed. Saving the syntax for such data transformations provides the necessary documentation of how new variables were defined. It also enables you to check whether you did everything you intended to, make corrections easily, and insures that you will be able to duplicate the computations, should that be necessary.

If you need to do the same (or very similar) analysis many times, you can do it more quickly and with fewer opportunities for errors by using syntax. Once you have written and checked that the syntax is doing what you want, you can run it repeatedly, in this or future SPSS Statistics sessions, without having to make any menu choices. You can make minor modifications without having to go through the menus again. The syntax can include a whole series of procedures.

Syntax is not dependent on your operating system; thus, analyses started on one machine can easily be re-created and continued on another machine by transferring the syntax and data files.

Finally, some advanced features are not available from the menus. For example, General Linear Models can do factorial and repeated measures models from the menus. For nested models, you must use syntax.

## *Pasting Syntax*

The easiest way to begin using syntax is to complete all your choices and options from the menus, and click **`Paste`**. A `Syntax` window will be opened, showing the syntax generated by the choices you made. Most SPSS Statistics dialog boxes have a Paste button. Using Paste, you can create a syntax file for your entire analysis.

The Define Variable Properties dialog can be used to Paste Syntax for defining missing values, variable labels, value labels, formats, and other variable attributes. Changes made directly in the Variable View window do not generate syntax, so should be avoided if you are trying to collect syntax.

# Reading Data from External Files

The gss93.dat data is a small subset of a large social/demographic survey. You can get information about it, and download the data file from http://www.umass.edu/statdata/statdata/data. Use the following variable names and locations to read the data:

ID - integer in columns 1-4
MARSTAT - Marital Status - columns 5
AGE - columns 6-7
DEGREE. - columns 8-9
SEX - column 10
ETHNIC - column 11
RELIG - column 12
CAPPUN - column 13
GUNLAW - column 14
SEXEDUC - column 15
LETDIE - column 16
READNEWS - column 17
TVNEWS - column 18
TVPBS - column 19
BIGBAND - column 20
BLUEGRAS - column 21
CW - column 22
BLUES - column 23
MUSICAL - column 24
FOLK - column 25
JAZZ - column 26

Observe that the variables are not delimited by blanks (or anything else). Also, some missing information has been left blank. Therefore, the data cannot be read as freefield. However, since the variables are aligned in columns, we can use Fixed Format to read this data.

Choose **File/Read Text Data**. Go through the Import data wizard dialogs, filling them in as follows:

Look In: Folder where you saved the file. For the workshop, select **Desktop →SPSSwork**
Filename: gss93.dat
Files of type: Text (*.txt, *.dat)
Click **Open**

**Does your text file match a predefined format?** No. Click Next
**How are your variables arranged?** Fixed width
**Are variable names included at top of your file?** No. Click Next
**First case of data begins on which line?** 1
**How many lines represent a case?** 1
**How many cases do you want to import?** All the cases. Click Next
**The vertical lines in data represent breakpoints between variables?** Click between the numbers to separate variables, using the variable location information above. Click Next
**Specification for variable selected in data preview?** Click a column heading in the data preview to select it, type the variable name for it in the Variable Name box, using the names listed above. Repeat for each columns. Click Next
**Would you like to save the file format?** No
**Would you like to paste the syntax?** Yes. Click **Finish**

A Syntax window opens, with the SPSS Statistics command equivalent of the menu choices you made.

# Working with Syntax

## *Running Syntax*

To run all or part of the commands in the Syntax window:

From the SPSS Statistics Syntax editor menu → **Run**
Choose:
**All** (to run everything in the syntax window)
**Selection** (runs only highlighted syntax)
**To end** (runs syntax from cursor position to end)

## *Saving Syntax*

In order to use the same syntax in a future SPSS Statistics session, you need to save it.

From the SPSS Statistics Syntax editor window → **File** → **Save As**
Save in: *choose a drive or directory*
*(for the workshop, use* ***Desktop*** → ***SPSSwork****)*
Filename: **gss93**
Save as type:**SPSS Statistics**(***.sps**)

## *Other sources of syntax*

You can copy syntax from the log output in the viewer and paste it into a syntax window.

Finally, if all else fails, all of the syntax from each SPSS Statistics session is saved in the SPSS journal file, statistics.jnl. The name and default location of this file varies depending on your version of Windows, and your version of SPSS Statistics. Look in **Edit →Options →File Locations (or General)** to see where it is on your system.

If you want to extract syntax for a session from the journal, you must do this promptly, as it will become increasingly difficult to identify as syntax from later sessions is appended. Alternatively, if you have set your journal option to overwrite, the syntax from a session is destroyed as soon as you re-open SPSS Statistics. Open the journal file with any text editor (e.g.Wordpad, Notepad, etc), and copy the syntax you want. Save the extracted syntax to your personal directory (folder), and add some comments so you can associate the commands with the corresponding output file. In line with SPSS Statistics conventions, the syntax file should have the .sps extensions. To add comments without disturbing the syntax, write the word **Comment** at the left margin. Then type any text you want. You can continue on as many lines as you need. The comment should not have any internal periods, but must end with a period.

Using the journal is a last-ditch solution, as you will likely need to be very selective in what you take from it, and may need to do considerable editing.

## *Using Saved Syntax*

You can reuse syntax saved to a file in a new SPSS Statistics session.

From the Menu → **File** → **Open** → **Syntax**
From the **Open file** dialog box:
Look in: *drive or directory where syntax was saved*
*(for the workshop,* ***Desktop*** → ***SPSSwork****)*
Filename: **gss93**
File of type: **SPSS Statistics (*.sps)**

You can now run it to repeat your previous analysis or edit it if you need to make some changes.

# Useful Options

Under **Edit - Options** you can choose whether dialog boxes will present variable lists using **variable names** or **variable labels**, and whether the variables in the dialog boxes will be in the **file order** or in **alphabetic order**. For large files in which the variables have been arranged in some logical order, file order is generally more useful.

Under **Data**, you can set the 100-year interval SPSS Statistics will assume if you enter a date using only two digits for the year.

You can also choose from a variety of standard table and chart formats, what is displayed or hidden in the Output, and many other features. When you make changes to Options, your latest choices are saved, and will remain in effect for future sessions.

# Defining Variable Properties

## *Using menus to Paste Syntax*

The gss93 data has several groups of variables that are coded the same way:

CAPPUN, GUNLAW, SEXEDUC, LETDIE
- 0,8,9 represent different kinds of missing values
- 1 - Favor
- 2 - Oppose.

READNEWS, TVNEWS, TVPBS
- 0,8,9, are missing
- 1 - every day
- 2 - a few times per week
- 3 - once a week
- 4 - rarely
- 5 - never

BIGBAND to JAZZ ( 8 variables)
- 0,8,9 are missing
- 1 - like very much
- 2 - like
- 3 - mixed feelings
- 4 - dislike
- 5 - dislike a lot

We can make the job of defining missing values and value labels much easier by setting up a the correct values for one variable in each group, then copying it to the remaining variables.

In the **Define Variable Properties** dialog, select MARSTAT, DEGREE to JAZZ to Scan. (Shift-Click or Control-Click are easy ways to select many variables in a dialog.) Click **Continue**. Define the **Missing Values**, **Value Labels**, **Type**, **Measurement Level, Width** and **Decimals** for each of MARSTAT to RELIG. *Do NOT press OK after each variable definition.* Simply click another variable and proceed with its definition. When you get to CAPPUN, after defining its properties, under Copy Properties, click **To Other Variables**. Select GUNLAW, SEXEDUC, LETDIE, and click **Copy**. Similarly, define the variable properties for the first variable from each of the other groups, and copy it to all the other variables in the group. When ALL variables properties are defined, click **Paste**.

Here is the first the set of commands generated for the CAPPUN to LETDIE group:

```
*cappun.
VARIABLE LABELS cappun  'Capital Punishment'.
MISSING VALUES cappun ( 0, 8, 9 ).
VALUE LABELS  cappun
     1  'Favor'
     2  'Oppose'  .
```

A similar MISSING VALUES and VALUE LABELS command is generated for each variable in the group. The VARIABLE LABELS is not copied to the other variables, but we can use the above as a model for adding VARIABLE LABELS to any additional variables that need them.

## *Modifying syntax*

The generated syntax contains an identical set of MISSING VALUES and VALUE LABELS for each variable in each group, which makes it rather lengthy. You can often write much more compact code by generating the definition for the first variable in each group, and modifying the code to include other variables. For example the code for CAPPUN to LETDIE could be shortened to:

```
MISSING VALUES CAPPUN TO LETDIE(0,8,9).
VAR LABELS CAPPUN 'Capital Punishment'/ GUNLAW 'Gun Control'/
   SEXEDUC 'Sex Education'/ LETDIE 'Let Terminal Patients Die'.
VALUE LABELS CAPPUN TO LETDIE 1 'FAVOR' 2 'OPPOSE'.
```

Notice that the MISSING VALUES and VALUE LABELS commands both accept lists of variables. Groups of contiguous variables can be specified using the keyword TO; non-contiguous variables can be listed separated by blanks or commas. You can even specify variables with different definitions on a single MISSING VALUES or VALUE LABELS command, by separating them with a slash. The complete set of MISSING VALUES and VALUE LABELS might look like this:

```
MISSING VALUES MARSTAT(9)/AGE (0,98,99)/ DEGREE (7,8,9)/
   RELIG(8,9)/CAPPUN TO JAZZ(0,8,9).
VALUE LABELS MARSTAT 1 "Married" 2 "Widowed" 3 "Divorced"
     4 "Separated" 5 "Never Married" /
  DEGREE 0 'Less than High School' 1 'High School'
    2 'Junior College' 3 'Bachelor' 4 'Graduate' /
  SEX 1 'Male' 2 'Female' /  ETHNIC 1 'White' 2 "Black" 3 'Other'/
  RELIG 1 'Protestant' 2 'Catholic' 3 'Jewish' 4 'None'  5 'Other'/
  CAPPUN TO LETDIE 1 "favor" 2 "oppose"/
  READNEWS TO TVPBS 1 "every day" 2 "few times per week"
     3 "once a week" 4 "rarely" 5 "never"/
  BIGBAND TO JAZZ 1 "like a lot" 2 "like" 3 "mixed"
     4 "dislike" 5 "dislike a lot".
```

Upper/lower case is strictly cosmetic. You can use either apostrophes (') or quotes (") in label definitions, provided the label does not itself contain these characters – for example the label "don't know" requires quotes. Indentation is used to make code easier to read – it is not required.

Select the MISSING VALUES, VAR LABELS, VALUE LABELS and FORMAT commands and **Run** them. Check your Output window for any error messages, and look in the **Variable View** window to confirm that all variables have the correct properties. When you are done, your syntax file should contain everything you need to read the gss93 text file and define all essential variable properties – GET DATA, MISSING VALUES, VAR LABELS, and VALUE LABELS commands.

### *Syntax Rules*

The contents of the syntax window can be edited as with any text editor. You can add or change options to suit your needs. Use the Syntax Reference Manual to look up details of command options.
The general rules are:

- commands begin at the left margin
- continuation lines are indented at least one space
- text must not be longer than 80 characters on any line.
- each command must end with a period.
- options or different parts of a command are usually separated by a slash (/)
- Variable lists of the form VAR1 TO VARn imply that the command applies to all the variables in the data file between VAR1 and VARn (inclusive). The order implied is data *file order*, not alphabetic order, regardless of the Display Order setting in Edit/Options.
- You can use variable lists of the form VARa TO VARk, VARm, VARp TO VARs to make lists that combine groups of contiguous and non-contiguous variables.

If you Copy and Paste a command, be sure to start the selection at the beginning of the command (where it is at the left margin), and go to the period at the end. (Pasted syntax usually has the period on a line by itself.) Then you can make modifications between these two points, keeping all intermediate lines indented, and leaving any slashes where they are. Use the dialogs to generate the Syntax as much as possible, so you can use it as a model. If in doubt, you can check the precise syntax of any command in Online Help.

# Checking the Data

Having created a new SPSS Statistics file, we should do some simple descriptive statistics to make sure that the data has been read correctly. All the variables other than age are categorical, so we will begin with Frequencies on these variables.

Run Frequencies on all the variables except ID and Age. When you've filled out the dialog, click **Paste**. Observe that the Frequencies command is added to your Syntax window, but the command is not run. As before, pasted syntax is considerably more verbose than necessary. It particular, each selected variable is named. If we were to write our own FREQUENCIES command we would use the TO convention to select contiguous variables:

```
FREQUENCIES VARIABLES=MARSTAT, DEGREE TO JAZZ.
```

To run the Frequencies from the Syntax window, place the cursor anywhere within the Frequencies command. Notice the blue triangle marking the line with the cursor. If the command takes more than one line, an indicator shows the beginning and end of the command. Choose **Run/Selection** (or click the **right facing triangle on the toolbar**).

**Save** the syntax file as **gss93.sps**, and the data file as **gss93.sav**.

# Corrections with Syntax

Looking over the Frequencies output, and comparing it to the data dictionary, we realize that something is seriously wrong! Variables DEGREE, SEX, ETHNIC, and LETDIE have many cases with values that are not in the codebook. Checking the codebook we realize that DEGREE should have been read from column 8 only, not columns 8-9. As a result of this error, all the subsequent variables were read from the wrong location. In addition, we left out one variable, CLASSICL, which should be between MUSICAL and FOLK. We will need to re-read the data file using the correct locations for the variables.

We have already spent considerable time naming the variables, defining missing values, value labels, etc. Using the dialogs alone, we would have to start all over! In addition to the burden of going through all the work again, we risk making other mistakes. Using the saved syntax it is easy to make the necessary correction, while preserving the parts we got right.

In the gss93.sps syntax window, go to the GET DATA command, and change the column locations of all the variables, starting with DEGREE 7-7. *Note that SPSS* Statistics *starts numbering positions at 0, rather than 1*. As a result, all variable locations are one less than on the codebook. Insert CLASSICL 23-23 between MUSICAL and FOLK.

Look through the remaining commands to see whether these corrections have further ramifications. Note that the MISSING VALUES, VAR LABELS, and VALUE LABELS are correct for all previously defined variables. Add a VAR LABEL for CLASSICL. If you defined MISSING VALUES and VALUE LABELS using the list BIGBAND TO JAZZ, the newly defined variable CLASSICL is included in those commands; otherwise you'll need to add MISSING VALUES and VALUE LABELS for the new variable.

The Frequencies command generated by Paste lists variables individually. You need to add CLASSICL to that list in the desired location, or change the list to the more compact version DEGREE TO JAZZ. When you are finished, save the revised syntax file, and choose **Run/All**. If the Frequencies output now shows that the data is correct, **Save** the datafile, gss93.sav, and the modified syntax file, gss93.sps.

# Transformations

Even if you don't need to use syntax for most purposes, if you create or recode variables or do any substantive data transformations, you should save the syntax. Without it you will have a data file with a bunch of variables without a record of how they were created, nor any evidence that they are what they were intended to be. It will be impossible to do any troubleshooting, should the need arise. Furthermore, if for any reason you need to start again, the saved syntax can save you much time and grief.

The gss93 survey contains four variables that ask the respondents' view on some social issues. A response of "Favor" to three of the questions, GUNLAW, SEXEDUC, LETDIE, and a response of "Oppose" to CAPPUN is considered to be at the "socially liberal" end of the spectrum. The opposite set of responses will be considered "socially conservative". Our task it to create an "index" variable that will score each respondent on a social outlook scale.

We need to first make sure the four social issue questions are coded in the same "direction"; three of the four have codes that can be interpreted as 1 for "liberal" and 2 for "conservative". The codes for CAPPUN need to be reversed. When the four questions are coded in the same direction, counting the number of 1's for each respondent will give a 0-4 score, with 0 being most "conservative", and 4 most "liberal".

To reverse the coding for CAPPUN, on the Data View screen:

From the **Transform menu,** Select:
**Recode→Into Different Variables**
Select **CAPPUN** and move it to **Input Variable->Output Variable** box.
Type a new name, say **CAPPUNR,** in the **Output Variable** name box.
Optionally, type a **Label** for the new variable.
Click **Change.**
Click **Old and New Values.**
**Old Value 1; New Value 2;** Click **Add.**
**Old Value 2; New Value 1;** Click **Add.**
Click **System or User-Missing** on the left side,
**Copy Old Values** on the right side; Click **Add.**
Click **Continue.**
Click **Paste.**

You should get the following syntax:

```
RECODE  cappun  (1=2)  (2=1)  (MISSING=Copy)  INTO  cappunr .
VARIABLE LABELS cappunr 'Capital Punishment Reversed'.
EXECUTE .
```

Add a **Value Labels** command for the new variable CAPPUNR to the syntax file.

Select and Run the syntax segment that creates the new variable CAPPUNR. Run Frequencies on both CAPPUN and CAPPUNR to check that the coding has been reversed as intended and all missing values and labels are correct.

When CAPPUNR has been correctly defined you are ready to create the "index" that counts the number of "liberal" responses to the four questions. On the Data View screen menu:

Select **Transform→Count Values within Cases**
Type a name for the new variable in **Target Variable:** LIBSCORE
Type a label in the **Target Label** box: Liberal Social View Score
**Select the four variables**: GUNLAW, SEXEDUC, LETDIE, CAPPUNR. Do NOT select CAPPUN!
**Click the arrow** to move the selected variables into the **Numeric Variables** box.
Click **Define Values.**
On the left side, select **Value** and type **1** into the Value box.
Click **Add** to move it to the Values to Count box.
Click **Continue.**
Click **Paste.**

Here is the pasted syntax:

```
COUNT  libscore = cappunr gunlaw sexeduc letdie  (1)  .
VARIABLE LABELS libscore 'Liberal Views Score' .
EXECUTE .
```

Add VALUE LABELS for the new variable, LIBSCORE, to define 0 as "Most Conservative" and 4 as "Most Liberal". Select and run the newly added syntax.

You now have a complete syntax file for reading the gss93.dat data file, defining new variables, and setting up missing values and labels for all variables. You can do a bit of clean-up to remove extraneous commands (if you pasted some things you didn't intend) put things in sensible order and add some comments. You can safely remove all but the last EXECUTE. Save the final syntax file, gss93.sps.

# Split Files

Split File is very useful when you want to repeat a series of analyses for several subgroups. For example, suppose you want to test whether the average age of males differs according to their "liberal" score. You will then want to do the same for the females. You also want to display the comparisons with bar graphs.

You could accomplish this by using **Data→Select Cases** to select the males and run the analysis, then select the females and re-run the same analyses. But it would be quicker and simpler to use **Split Files**. On the Data Editor menu:

**Data→Split File.**
Click **Compare Groups** or **Organize Output by Groups**
From the list of variables, **Select** SEX
**Click the arrow** to move **SEX** into the **Groups Based on Box.**
(Notice that on the bottom of the dialog, the **Sort the File** button is on. In order to use Split Files, the data file must be sorted on the grouping variable. )
Click **Paste.**

To test for differences in age among the 5 values of the liberal beliefs score, select

**Analyze→Compare Means→Oneway ANOVA.**
Select **AGE; Click** the arrow to move it to **Dependent List**
Select **LIBSCORE; Click** the arrow to move it to **Factor.**
Click **Options.**
Check **Descriptives.**
Click **Continue.**
Click **Paste.**

To make the bar graphs, select

**Graphs→Chart Builder (do we need to define variables??) →Bar.**
**Drag** Bar Chart from the Gallery to the Chart Preview area
**Drag** AGE to the y-axis.
** **Right_Click** LIBSCORE; **Select** Ordinal.
**Drag** LIBSCORE to the x-axis.
Optionally, under the **Title** tab, add a title.
Click **Paste.**

Now Select and **Run** these commands, beginning with Split File. Note that the two graphs have different scales, and are underneath each other so they are not easily compared visually. You can get better visual comparisons by turning Split Files Off and using Column Panels or Clustering.

# Reuse Syntax for Similar Analyses

If you want to repeat the above tests and charts for the individual opinion questions (GUNLAW, SEXEDUC, LETDIE), you can modify the syntax generated above. Copy the ONEWAY, GPL and GGRAPH syntax and Paste it below the original. Change LIBSCORE to GUNLAW in each procedure (there is one reference in Oneway, one in GGRAPH, three in GPL (2 in DATA, 1 in ELEMENT)). If you used a TITLE, you'll need to change that also. Now you can run the identical analysis for this variable.

# Split File Off

Split File stays in effect until you turn it off. To turn Split File off, on the Data sheet menu select:

**Data→Split File**
Click **Analyze All Cases.**
Click **Paste**

**Run** the SPLIT FILE OFF command from the Syntax window.

# Chart Builder vs. Legacy Dialogs

SPSS Statistics has two distinct set of graphing routines, Chart Builder and Legacy Dialogs. Although there is some overlap between their functionality, each offers some choices not available in the other. Where they overlap there is no general rule for which procedure to use. A few observations:

- To get side-by-side graphs for visual comparison, use
  - Chart Builder with a Columns Panel variable
  - Legacy Dialog with a COLUMN variable
- To get line charts with proportional axis for scale variables with unequal spacing, use Chart Builder. See example 1.
- Legacy Dialogs can create multiple-variable clustered bar charts. Chart Builder does not have this feature. See Example 2.
- Chart Builder lets you choose the base for computing percentages. Legacy Dialogs does not. See Example 3.

You may have to experiment with both procedures to see which is better for any given situation.

**Example 1:** Proportional axis scale.
In order to illustrate what happens with unequally spaced scale variables, we need to create such a variable. We will do this by computing the square of LIBSCORE. We will call the new variable LIBSCOR2. It will have codes 0,1,4,9, and 16. The syntax is:

```
COMPUTE LIBSCOR2=LIBSCORE**2.
```

Notice that LIBSCOR2 is a scale variable. Now let's make a line graph of average AGE against LIBSCOR2, comparing Legacy Dialog and Chart Builder results. (The same is true for Bar graphs, but there is less of an expectation for bar spacing to reflect a scale.) From the Graph menu:

**Legacy Dialogs →Line**
Select **Simple**
Select **Summaries for Groups of Cases**
Click **Define**
Select **Other Summary Function**
Select **Age**
**Click the arrow** to put **Mean(Age)** into the **Variable** box
Select **Libscor2**
**Click the arrow** to move it to the **Category Axis** box
Click **Paste**

Run the generated GRAPH command. Observe that the values of 0,1,4,9, and 16 are equally spaced on the x-axis, even though the measurement level of LIBSCOR2 is Scale. If the values of the x-variable have any quantitative meaning you would likely want the x-scale to reflect the magnitude of the x-variable.

Compare the results to Chart Builder:

**Graphs → Chart Builder → Line**
**Drag Simple Line to the Preview area**
**Drag Age** to the y-axis.
**Drag Libscor2** to x-axis. (If it does not have the ruler icon, right-click and select Scale.)
Click **Paste**

Run the generated commands. The x-axis is now proportional to the values of LIBSCOR2.

Re-do the graph, but this time right-click Libscor2 and select **Ordinal**, rather than Scale. Compare the resulting graphs for the two commands.

**Example 2:** Multi-variable clustered bars.

We would like to make a graph comparing the percent in favor of the 4 social issue questions among males to the percent in favor among females. Using Legacy Dialogs,

```
Legacy Dialogs → Bar
   Click Clustered
   Click Summaries of Separate Variables
   Click Define
   Select CAPPUN, GUNLAW, SEXEDUC, LETDIE
   Click the arrow to move them to the Bars Represent box
   Click Change Summary
      Select Percentage Inside
      Enter 1 in both the Low and High boxes
      Click Continue
   Select SEX
   Click the arrow to move it to the Category Axis box
   Click Titles
      Line 1: Percent in Favor
      Line 2: By Gender
      Click Continue
   Click Paste
```

Run the Graph syntax.

Chart Builder does not have a feature for displaying several variables on one chart.

**Example 3:** Choosing the base for percents

Let's take another look at percent in favor of gun control laws, comparing the two genders' opinions.

```
Legacy Dialogs → Bar
   Click Clustered
   Click Summaries of Groups of Cases
   Click Define
      Under "Bars Represent", select % of Cases
      Select GUNLAW
      Click the arrow to move it to the Category Axis box
      Select SEX
      Click the arrow to move it to the Define Clusters By box
      Click Titles
          Line 1: Opinion on Gun Control Law
          Line 2: By Gender
          Click Continue
      Click Paste
```

Observe that when we choose "% of Cases" we get no information on (or control over) what the base for the percents is. In fact, what we get is percent in favor or opposed to gun control laws, within gender. Suppose we want to look at "what percent of those in favor or opposed are male/female"? Using Legacy Dialogs, we cannot do this without reversing the roles of the "category axis" and "cluster by" variables.

Using Chart Builder, we have control over the base of percents:

```
Graphs → Chart Builder
   Drag Clustered Bar to the Preview area
   Drag GUNLAW to the x-axis.
   Drag Sex to Cluster On box.
   In the Element Properties Dialog:
      Select Bar1
      Under Statistics, select Percentage(?)
      Click Set Parameters
          Select Total for Each Legend Variable (to get %
favor/oppose within gender)
       OR… Select Total for Each X-Axis Category (to get %
male/female within opinion
          Click Continue
      Click Apply
   Click Titles/Footnote
      Click Title1 checkbox
          In the Element Properties contents box, enter a title
      Click Apply
   Click Paste
```

Run the generated GGRAPH and GPL commands. Compare the results to the previous Graph. If, in Set Parameters, you chose "Total for Each legend Variable", the graph should be identical to the one you got from legacy Graphs. If you chose "Total for Each X-Axis Category", the two graphs will be quite different.

# Chart Templates

While the content of a chart is largely controlled by syntax, many of the visual design aspects can be standardized using " Chart Templates". Use Chart Templates to apply a consistent set of colors, fonts, symbols, etc. to your graphs.

When you apply a chart template to a different chart style, only those properties of the template that make sense for the new chart are used. For example, if you apply a template you create for a bar graph to a pie chart, only the colors in the template are used. Axes scales and tick marks do not apply to a pie chart.

To create a chart template, edit a chart to set the elements you wish to standardize:

```
Double-click the last clustered bar chart you made. The Chart Editor opens:
   Double-Click the Y-Axis. In the Properties dialog
      Select the Labels and Ticks tab.
          Under Minor ticks, Check Display Ticks
          For 'Number of Minor Ticks', enter 3.
          Click Apply.
      Select the Scale tab.
          Under Range, uncheck Auto boxes, set Minimum=0,
Maximum=100, Major Increment=20
          Click Apply.
   In the chart, Click the Female legend box
      In the Properties dialog, under Fill & Border
          Click Fill
          Choose a different Color for Female bar
          Click Apply
      Close the Properties dialog
   Under Options
      Select Transpose Chart
      Select Show Grid Lines
   From the Chart Editor menu, select File→Save Chart Template
   Select the properties of the template to be saved:
      Layout, Styles, Axes (Do not check Text Content)
      Click Continue
      Choose a folder and filename (the extension is .sgt)
      Click Save
Close the Chart Editor.
```

To apply this look to other charts:

- For *existing charts*, double click the chart. On the **File** menu, select **Apply Chart Template**. Use Browse and **select the template** file you want to apply. Click **Open**.
- For *new charts using Chart Builde*r:
  - *in the Chart Builder dialog*, click **Options**. Under **Template**, click **Add**. Browse to the **saved chart template** to use for the new graph. Click **Open**. Click **OK**.
  - *using syntax*, add **TEMPLATE=["path\filename.sgt"]** to the existing syntax, between SOURCE=INLINE and the period after INLINE.
- For *new charts using Legacy Dialogs:*
  - *in the Legacy Dialogs*, **check** Use Chart Specifications From: Click **File**. Use Browse to find and **select the chart template** to use for the new graph. Click **Open**.
  - *using syntax*, add the option **/TEMPLATE='path\filename.sgt.'** to the existing syntax.

**NOTE:** Templates made from Legacy Dialogs graphs may not apply well to Chart Builder graphs, and vice versa.

# Date Variables

Open the SPSS Statistics file dates.sav. It has three variables. AVGTEMP is the average temperature during the time period from STARTDAT to ENDATE. Observe in Variable View that STARTDAT and ENDATE are defined as type DATE.

Sort the data by STARTDAT. From the Data View menu, choose **Data/Sort Cases**. Select STARTDAT and move it to the **Sort By** box. Click OK.

Calculate the number of days in each time period. SPSS Statistics stores dates as number of seconds from Jan 1, 1900. Try changing one of the Date variables to Numeric and you will see the underlying number. Change it back to type DATE before proceeding. To calculate elapsed time in days, we need to use the formula DAYS=(ENDATE-STARTDAT)/(60*60*24). Use this formula in **Transform/Compute** to calculate the number of days in each time period.

Make plots of AVGTEMP against STARTDAT using **Legacy Dialogs** and **Chart Builder** and compare the results. For Legacy Dialogs, select **Line/Simple/Values of Individual Cases/Define**. Put AVGTEMP in the Line Represents box. Under Category Labels, click Variable, and move STARTDAT into the Variables box. For Chart Builder, select **Line**. Drag AVGTEMP to the Y-axis, and STARTDAT to the X-axis. Observe that the two graphs look quite different. Why??

# Pivot Tables

SPSS Statistics Table output is in the form of Pivot Tables. This means you can re-arrange the rows and columns to suit your fancy.

**Double-click** the **Descriptives** table from the Oneway output. The slashed border shows that the table is now in edit mode, and the Pivot item appears on the menu.

From the **Pivot** Menu, select **Pivoting Trays**. Note the icons in the Row, Columns and Layers.
Drag the icon representing **Groups** from **Rows to Columns**.
Drag the **Statistics** icon from **Columns to Rows**. The table is re-arranged accordingly.
Close the Pivot Tray.
In the table, **double-click Mean**. Change it to **Average Age**.
**Drag** the pointer across the row of average ages to select them.
**Right-click** the selected average ages, select **Cell Properties**.
Under the **Format Value** tab, change the number of **decimal places**.

# Export Results

**SPSS** Statistics can Export output to Word, Excel, pdf, and other formats. If you don't want the entire SPSS Statistics output exported, use the left panel in the Output Viewer window to select the table(s) and/or charts to export. Use **ctrl-click** to select non-contiguous objects, or **shift-click** to select many contiguous objects. Select:

**File→Export**
Under Export: Select **Output Document**
Choose: **All, Visible Objects** or **Selected Objects.**
Under File Type: **Select Word/RTF file (*.doc)**
Click Browse: **Select a folder and name for the Word file.**
Click: **OK**

# Appendix: A Complete Syntax File

Here is the complete syntax file to do most of the tasks described in this document:

```
GET DATA  /TYPE = TXT /FILE = 'C:\SPSSWork\mytests\gss93.dat'
 /FIXCASE = 1 /ARRANGEMENT = FIXED /FIRSTCASE = 1 /IMPORTCASE = ALL
 /VARIABLES =
 /1  id 0-3 F4.2  marstat 4-4 F1.0  age 5-6 F2.1 degree 7-7 F1.0 sex 8-8 F1.0 ethnic 9-9 F1.0
 relig 10-10 F1.0 cappun 11-11 F1.0 gunlaw 12-12 F1.0 sexeduc 13-13 F1.0 letdie 14-14 F1.0
 readnews 15-15 F1.0 tvnews 16-16 F1.0 tvpbs 17-17 F1.0 bigband 18-18 F1.0 bluegras 19-19 F1.0
 cw 20-20 F1.0 blues 21-21 F1.0 musical 22-22 F1.0 classicl 23-23 F1.0 folk 24-24 F1.0
 jazz 25-25 F1.0.
format id to jazz(f4.0).
MISSING VALUES MARSTAT (9) / AGE(0,98,99) / DEGREE (7,8,9) /RELIG (8,9)/
 cappun TO letdie (0, 8, 9) / readnews TO tvpbs (0, 8, 9)/  bigband TO jazz (0, 8,9).
VALUE LABELS MARSTAT 1 "Married" 2 "Widowed" 3 "Divorced" 4 "Separated" 5 "Never Married" /
  degree 0 'Less than High Schoool' 1 'High School' 2 'Junior College' 3 'Bachelor' 4 'Graduate' /
  sex 1 'Male' 2 'Female' /  Ethnic 1 'White' 2 "Black" 3 'Other' /
  Relig 1 'Protestant' 2 'Catholic'  3 'Jewish' 4 'None'  5 'Other' /
  cappun TO letdie 1 "favor" 2 "oppose"/
  readnews TO tvpbs 1 "every day" 2 "few times per week" 3 "once a week" 4 "rarely" 5 "never"/
  bigband TO jazz 1 "like a lot" 2 "like" 3 "mixed" 4 "dislike" 5 "dislike a lot".

Comment Reverse code CAPPUN and make LIBSCORE the sum of "liberal" responses.
RECODE  cappun  (1=2)  (2=1)  (MISSING=Copy)  INTO  cappunr .
VARIABLE LABELS cappunr 'Capital Punishment Reversed'.
COUNT  libscore = cappunr gunlaw sexeduc letdie  (1)  .
VARIABLE LABELS libscore 'Liberal Views Score' .
VALUE LABELS cappunr 1 "Oppose" 2 "Favor"/ libscore 0 'Most Conservative' 4 "Most Liberal".
EXECUTE .

FREQUENCIES
  VARIABLES=marstat degree sex ethnic relig to jazz  .

Comment Run separate analyses for males and females.
SORT CASES BY sex .
SPLIT FILE  LAYERED BY sex .

ONEWAY  age BY libscore  /STATISTICS DESCRIPTIVES  /MISSING ANALYSIS .
* Chart Builder.
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=libscore MEAN(age)[name="MEAN_age"] MISSING=LISTWISE
    REPORTMISSING=NO
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
  SOURCE: s=userSource(id("graphdataset"))
  DATA: libscore=col(source(s), name("libscore"), unit.category())
  DATA: MEAN_age=col(source(s), name("MEAN_age"))
  GUIDE: axis(dim(1), label("Liberal Views Score"))
  GUIDE: axis(dim(2), label("Mean age"))
  SCALE: cat(dim(1), include(".00", "4.00"))
  SCALE: linear(dim(2), include(0))
  ELEMENT: interval(position(libscore*MEAN_age), shape.interior(shape.square))
END GPL.

Comment Stop separate analyses for males and females.
SPLIT FILE  OFF.

COMMENT Legacy Dialogs vs Chart Builder Example 1 - scaling the x-axis.
compute libscor2=libscore*libscore.
GRAPH  /LINE(SIMPLE)=MEAN(age) BY libscor2 .
* Chart Builder.
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=libscor2 MEAN(age)[name="MEAN_age"] MISSING=LISTWISE
    REPORTMISSING=NO
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
  SOURCE: s=userSource(id("graphdataset"))
```

```
  DATA: libscor2=col(source(s), name("libscor2"))
  DATA: MEAN_age=col(source(s), name("MEAN_age"))
  GUIDE: axis(dim(1), label("libscor2"))
  GUIDE: axis(dim(2), label("Mean age"))
  ELEMENT: line(position(libscor2*MEAN_age), missing.wings())
END GPL.

* Graph Example 2 - clustered bar graphs.
GRAPH  /BAR(GROUPED)=PIN(1 1)(cappun) PIN(1 1)(gunlaw) PIN(1 1)(sexeduc) PIN(1 1)(letdie) BY sex
  /MISSING=LISTWISE  /TITLE= 'Percent in Favor' 'By Gender'.
```