

SPSS Basics

SPSS is a program used to analyze statistical data. SPSS has three types of windows: (1) data windows, (2) syntax windows, and (3) an output window.

Data Windows

The data window shows you people’s responses to your questionnaires:

Each column is a *variable* (e.g., question in your survey)

Each row is an individual participant’s data

	male	age	single	married	anxietyg01	anxietyg02	anxietyg03
1	1	26	1	0	2	3	2
2	0	20	1	0	1	1	1
3	0	21	0	0	2	2	2
4	0	30	0	0	2	3	3
5	1	25	0	0	2	2	3
6	0	25	0	1	2	3	4
7	0	20	0	0	3	4	3
8	1	30	0	0	4	4	5
9	0	28	1	0	4	2	4
10	0	26	0	0	5	4	4

Participant #9 is 28 years old

Participant #7 has a score of 3 on the “anxietyg01” question

At the bottom of the window, you can toggle between “Data View” (depicted above) and “Variable View” (depicted below). Variable view shows an easy-to-read list of all of the variables in your study and lets you change their properties.

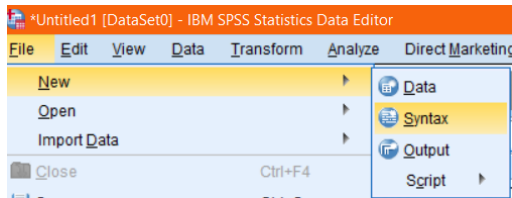
“Variable View” shows a list of all variables in the study

	Name	Type	Width	Decimals	Label
1	male	Numeric	1	0	
2	age	Numeric	2	0	
3	single	Numeric	1	0	
4	married	Numeric	1	0	
5	anxietyg01	Numeric	1	0	
6	anxietyg02	Numeric	1	0	
7	anxietyg03	Numeric	1	0	
8	avoidanceg01	Numeric	1	0	
9	avoidanceg02	Numeric	1	0	
10	avoidanceg03	Numeric	1	0	
11	avoidanceg04	Numeric	1	0	

Syntax Windows

Syntax is the language you use to give SPSS commands. For example, if you wanted to learn the mean and standard deviation of age in your sample, you would type the syntax command `descriptives age.`

To open a syntax window, where you can type in commands, go to “File > New > Syntax” on the main menu:



Writing Syntax

When writing syntax, every command must end with a period (.). For example, to run the correlations between two variables called “gender” and “height,” the syntax would be:

```
correlations gender height.
```

You can split a command along multiple *consecutive* lines if you’d like; e.g.:

```
correlations
  gender
  height.
```

If you’ve made a mistake that SPSS can automatically detect (e.g., not including a period at the end of a command, or including blank lines in a command), the command name will be highlighted in red; e.g.:

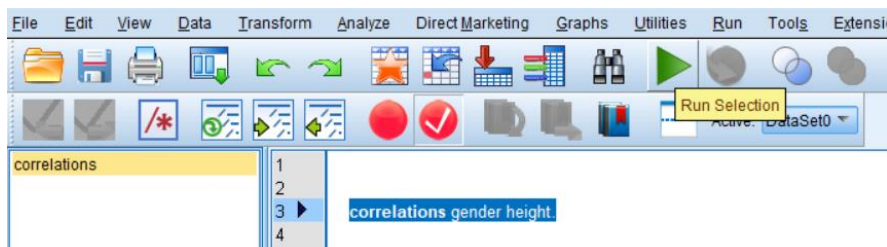
```
correlations gender height
```

```
correlations gender
```

```
height.
```

Running Syntax Commands

To run a command (or series of commands) that you’ve written, highlight the desired commands in the syntax window and then click the green “run selection” arrow.



Output Window

The final window SPSS uses is the Output window. Whenever you run a syntax command, the Output window will pop up, giving you the results of your statistical analyses.

Understanding Variables

Whenever you analyze data, you will be working with *variables*. When you first open a raw (i.e., brand new, unprocessed) data set, your variables will just be the questions in your survey:

Our survey has 3 questions measuring extraversion

	age	male	extraversi on1	extraversi on2	extraversi on3
1	20	0	2	3	2
2	19	0	3	2	4
3	22	1	3	3	4
4	25	0	5	4	4
5	18	1	3	2	2
6	23	1	4	5	3

Participant #2 provided a response of "4" ("agree") to extraversion question 3.

Participant #6 is 23 years old

SPSS can output information about the variables in your data set. For example, you could learn the *mean* for the "age" variable in your sample by using the following syntax command:

```
descriptives age.
```

You could also learn whether gender is *correlated* with age in your data by using the syntax command:

```
correlations age male.
```

Manipulating Variables

Oftentimes, in addition to gaining information about existing variables, we need to *create new variables* or *manipulate existing* variables. The `compute` syntax command creates new variables or changes existing variables.

For example, imagine we were to run the following syntax on the above data:

```
compute double_age = age*2.  
execute.
```

What happens? Suddenly, a new variable, called “double_age” gets created in our data set:

	age	male	extraversi on1	extraversi on2	extraversi on3	double_age
1	20	0	2	3	2	40.00
2	19	0	3	2	4	38.00
3	22	1	3	3	4	44.00
4	25	0	5	4	4	50.00
5	18	1	3	2	2	36.00
6	23	1	4	5	3	46.00

As requested by our syntax,
“double_age” simply equals age*2.

Be careful not to overwrite existing variables! For example, if we ran this syntax:

```
compute double_age = 7.  
execute.
```

Note that it overwrites the “double_age” variable with a value of 7 for everyone!

	age	male	extraversi on1	extraversi on2	extraversi on3	double_age
1	20	0	2	3	2	7.00
2	19	0	3	2	4	7.00
3	22	1	3	3	4	7.00
4	25	0	5	4	4	7.00
5	18	1	3	2	2	7.00
6	23	1	4	5	3	7.00

Manipulating variables has many uses! For example, if you wanted to compute a person’s *average* response to all three extraversion items, you could use the following syntax command:

```
compute overall_extraversion = mean(extraversion1,extraversion2,extraversion3).  
execute.
```

	age	male	extraversi on1	extraversi on2	extraversi on3	double_age	overall_extraversion
1	20	0	2	3	2	7.00	2.33
2	19	0	3	2	4	7.00	3.00
3	22	1	3	3	4	7.00	3.33
4	25	0	5	4	4	7.00	4.33
5	18	1	3	2	2	7.00	2.33
6	23	1	4	5	3	7.00	4.00

Now we have an overall extraversion score
for each person that can be correlated with
age and/or gender!

Best Practice: Never Save Any Changes to Data Sets

As a general rule, you should never *save* edits or changes to your data sets. Instead, leave your data in its original, raw form. Why? If you save any changes you make to the data, any errors you make (e.g., overwriting variables with incorrect values) will be permanently saved into your data file!

What should you do instead? Keep your data in its original form but *save your syntax*. For example, if you wanted to correlate age with overall_extraversion in the data depicted above, your entire syntax file might look something like this:

```
compute overall_extraversion = mean(extraversion1,extraversion2,extraversion3).  
execute.  
  
correlations age overall_extraversion.
```

You can save a syntax file containing these commands. Whenever you want to compute this correlation again in the future, you would just run the entire block of syntax. Using this procedure, you don't *need* to save the data file with your new "overall_extraversion" variable in it. The syntax can easily recompute the overall_extraversion variable from scratch any time you need it!

Syntax FAQ:

Check your spelling!

If you receive errors when trying to run syntax, check to make sure that all command and variable names are spelled correctly.

What if I forget the syntax for an analysis I'd like to run?

SPSS does have a drop-down menu system. If you forget the syntax for running certain commands, you can use the drop-down menus to learn what the syntax is.

For example, if you forgot how to correlate two variables, you could explore the "Analyze" menu until you found the "Analyze > Correlate > Bivariate" option. You can then use the pop-up windows to set up the analyses and click the **PASTE** button. This causes SPSS to put the syntax for the commands you're trying to run at the end of your syntax window.

Although you *can* use the SPSS's "paste syntax" feature to quickly reference syntax, it's oftentimes a good idea to consult the handouts you've been given in this class instead. SPSS tends to paste overly long syntax with unnecessary options and commands included. For example, if you used the point-and-

click windows to paste the syntax for a correlation, it would look like this (notice that 75% of the words are completely unnecessary, as “correlations gender height.” is sufficient):

```
DATASET ACTIVATE DataSet0
CORRELATIONS
  /VARIABLES=gender height
  /PRINT=TWOTAIL NOSIG
  /MISSING=PAIRWISE.
```

Why should I use syntax at all?

If SPSS has a point-and-click interface, why should you even bother learning syntax? There are three answers. First, once you know the syntax, it is ***much*** quicker than using the drop-down menus. Relatedly, when you start working with big datasets containing many variables, finding and selecting the correct variables in the drop-down menus can be extremely difficult, frustrating, and time-consuming. It’s much easier to just type out some syntax with the correct variable names included.

Second, some of the advanced techniques that we’ll be teaching you (e.g., regression) are extremely simple to do with a few lines of syntax—but they require navigating through multiple confusing pop-up windows if you try to use the point-and-click interface.

Finally, when you use syntax, you can save your syntax so that you don’t have to redo all of your analyses every time you return to your data. If you understand the syntax, you can easily modify your saved scripts if you need to make some changes to your analyses, too.