Ten SPSS tricks to make your peers jealous

Bernard Liew

2021-05-13

Contents

Chapter 1

Introduction

Most people interact with SPSS using what I call the "point and click" interface. A point-click method is not wrong per-se, I use it. The bad thing about the point-click method is that:

1) You cannot easily remember what you did. Try recalling the exact order of actions you undertook the last time you analyzed your data in SPSS. It is for this reason why one gets nervous reproducing the results.

2) It gets tedious really quickly, especially if you want to do the same thing repeatedly.

The alternative to the "point and click" interface, is using the **SPSS syntax**. Syntax is all about typing. You can open a fresh syntax like in Figure 1.1, populate it with commands to run, and save it. My rule of thumb is, when you spend so long getting something commands running, save your commands, not your results.

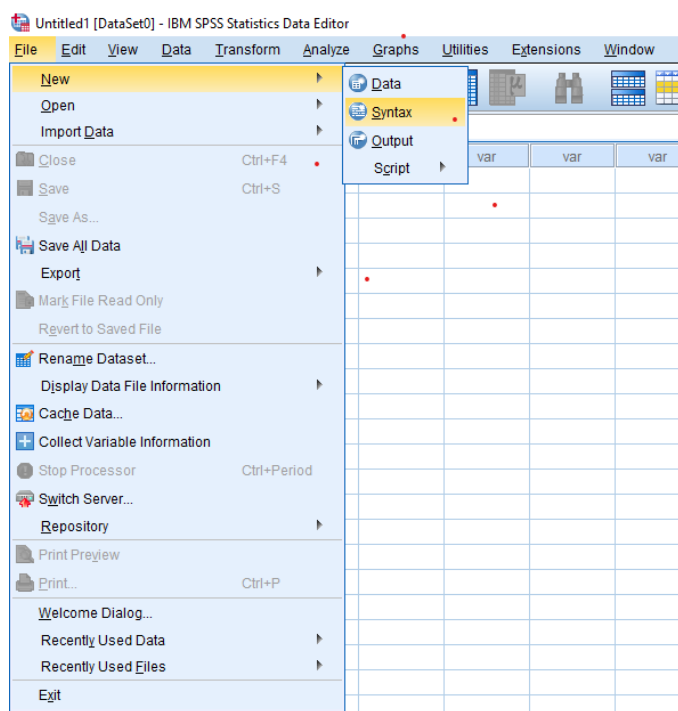Figure 1.2 shows what a full syntax looks like when you string together a bunch of commands.

Figure 1.1: Open your first syntax

Figure 1.2: What a full syntax looks like

Chapter 2

Trick 1 - Paste to syntax

If you can develop one habit, it is this!!! The good thing about interacting with SPSS using syntax, is that you do not have to be a total convert. You can continue using the "point and click" interface, but instead of executing the command, press **paste** as you see in Figure 2.1. Pasting automatically copies the word version of what you clicked into your opened syntax.
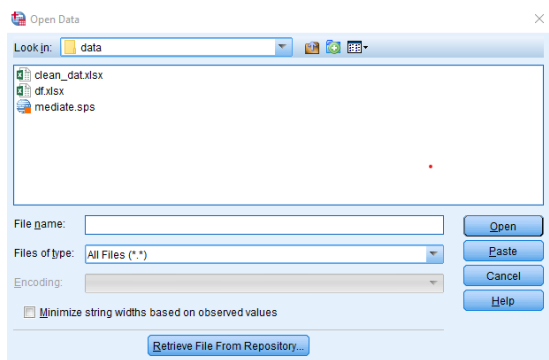
Figure 2.1: Paste the syntax you generated implicitly by clicking

I have not ever encountered an SPSS function that does not eventually have a paste function. Try finding it and saving what you intend to do like in Figure 2.2.

To run each command, highlight and press the play button like in Figure 2.3.
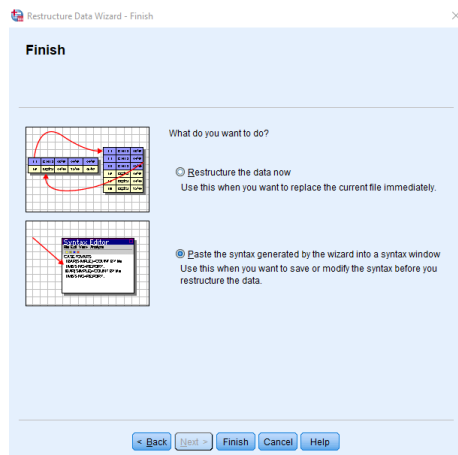
Figure 2.2: Always try to paste your syntax rather than executing
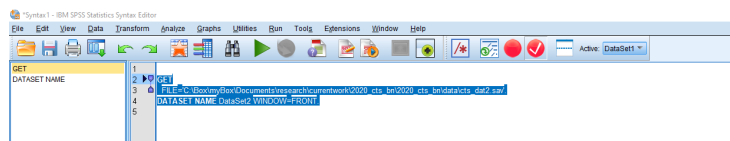
Figure 2.3: Executing a command

Chapter 3

Trick 2 - Setting your work directory

When importing data from deeply nested folders, it can be a pain to do so. In addition, when exporting files or figures, many often scratch their head as to where the files went to. The solution to this is to **explicitly** define the folder on your computer where every thing starts and ends. All the files will be exported to that folder. And importing becomes a breeze.

You can set the working directory by pasting the command below into your SPSS syntax, and change the folder path to your path. Just make sure to enclose it in " ".

```
cd 'C:\Box\myBox\Documents\teaching\Statistics\SPSS\wrangling\Essex_SPSS_summer\data'.
```

Chapter 4

Trick 3 - Importing data

The most common starting point for data analysis is Excel, so I will demonstrate how to easily import an Excel file. Paste the command below into your syntax, and simply change the file name `df.xlsx` to whatever your file is named. Remember not to change the file extension `.xlsx`. Notice I did not to do this `"'C:\Box\myBox\Documents\teaching\Statistics\SPSS\wrangling\Essex_SPSS_summer\data\df.xlsx'"`. It was because I already defined the path the file was in trick 3. If your Excel file has multiple sheets, change the sheet name `jump` to whatever it is named. If you only have one sheet, you can leave it blank.

```
GET DATA
  /TYPE=XLSX
  /FILE= 'df.xlsx'
  /SHEET=name 'jump'
  /CELLRANGE=FULL
  /READNAMES=ON
  /DATATYPEMIN PERCENTAGE=95.0
  /HIDDEN IGNORE=YES.
```

Using the sample data I provide, this is what the data looks like in Figure 4.1.

subj	group	age	wt	ht	time	side	set	rep	aexttorq	aextwork	aextpow	task	height
1	G	29	82.26	182	PRE	R	1	1	70	34	58	cmjbw	1.3909740000000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	cmjbw	1.2776400000000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	cmjbw	1.3741570000000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	sjbw	1.3472660000000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	sjbw	1.3436980000000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	sjbw	1.3543050000000000
1	G	29	82.26	182	PRE	L	1	3	66	33	66	sjbw	1.3543050000000000
1	G	29	82.26	182	PRE	L	1	4	63	32	65	cmjbw	1.3909740000000000
1	G	29	82.26	182	PRE	L	1	4	63	32	65	cmjbw	1.2776400000000000
1	G	29	82.26	182	PRE	L	1	4	63	32	65	cmjbw	1.3741570000000000
1	G	29	82.26	182	PRE	L	1	4	63	32	65	sjbw	1.3472660000000000
1	G	29	82.26	182	PRE	L	1	4	63	32	65	sjbw	1.3436980000000000
1	G	29	82.26	182	PRE	L	1	4	63	32	65	sjbw	1.3543050000000000
1	G	29	82.26	182	PRE	L	2	1	64	33	63	cmjbw	1.3909740000000000
1	G	29	82.26	182	PRE	L	2	1	64	33	63	cmjbw	1.2776400000000000
1	G	29	82.26	182	PRE	L	2	1	64	33	63	cmjbw	1.3741570000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	cmjbw	1.2020900000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	cmjbw	1.2119450000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	cmjbw	1.1813080000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	sjbw	1.2081710000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	sjbw	1.2119480000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	sjbw	1.2022780000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	cmjbw	1.2020900000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	cmjbw	1.2119450000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	cmjbw	1.1813080000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	sjbw	1.2081710000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	sjbw	1.2119480000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	sjbw	1.2022780000000000
2	G	34	42.30	161	PRE	R	1	3	60	28	52	cmjbw	1.2020900000000000
2	G	34	42.30	161	PRE	L	1	2	44	22	42	sjbw	1.2081710000000000
2	G	34	42.30	161	PRE	L	1	2	44	22	42	sjbw	1.2119480000000000
2	G	34	42.30	161	PRE	L	1	2	44	22	42	sjbw	1.2022780000000000
2	G	34	42.30	161	PRE	L	1	3	42	20	37	cmjbw	1.2020900000000000
2	G	34	42.30	161	PRE	L	1	3	42	20	37	cmjbw	1.2119450000000000
2	G	34	42.30	161	PRE	L	1	3	42	20	37	cmjbw	1.1813080000000000
2	G	34	42.30	161	PRE	L	1	3	42	20	37	sjbw	1.2081710000000000

Figure 4.1: Original data

Chapter 5

Trick 4 - Rename columns

Rename columns is a common task. Paste the command below into your syntax and run it. Put all the original column names before the = and all the new names after. There must be a spacing between each name, and the order preceding and proceding the = must be identical.

```
RENAME VARIABLES (subj group wt  = id grp weight).
```

After renaming this is what the data looks like in Figure 5.1.

id	grp	age	weight	ht	time	side
1	G	29	82.26	182	PRE	R
1	G	29	82.26	182	PRE	R
1	G	29	82.26	182	PRE	R
1	G	29	82.26	182	PRE	R
1	G	29	82.26	182	PRE	R
1	G	29	82.26	182	PRE	R
1	G	29	82.26	182	PRE	L
1	G	29	82.26	182	PRE	L
1	G	29	82.26	182	PRE	L

Figure 5.1: New column names

Chapter 6

Trick 5 - Filter rows

You might want to keep rows in your data based on some conditions. I tend to prefer to keep whatever rows I want and discard the remaining. Discarding your data does not harm your original data in Excel. If you change your mind, just highlight all the commands from the start and press the play button. In the example below, I want to keep rows where the variable `task` is equal `"cmjbw"` AND `side` is equal to "R". Below are some of the operators you can mix and match to powerfully filter our your data.

Symbols	Meaning
=	Equal
~=	Not equal
<	Less than
<=	Less than or equal
>	More than
>=	More than or equal
&	AND
|	OR

Highlight this command and press the play button.

```
FILTER OFF.
USE ALL.
SELECT IF (task = "cmjbw" &  side = "R").
EXECUTE.
```

After filtering this is what the data looks like in Figure 6.1.

id	grp	age	weight	ht	time	side	set	rep	aexttorq	aextwork	aextpow	task	height
1	G	29	82.26	182	PRE	R	1	1	70	34	58	cmjbw	1.39097400000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	cmjbw	1.12776400000000
1	G	29	82.26	182	PRE	R	1	1	70	34	58	cmjbw	1.37415700000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	cmjbw	1.20209000000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	cmjbw	1.21194500000000
2	G	34	42.30	161	PRE	R	1	1	52	25	52	cmjbw	1.18130800000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	cmjbw	1.20209000000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	cmjbw	1.21194500000000
2	G	34	42.30	161	PRE	R	1	2	54	27	56	cmjbw	1.18130800000000
2	G	34	42.30	161	PRE	R	1	3	60	28	52	cmjbw	1.20209000000000

Figure 6.1: Keeping rows based on some conditions

Chapter 7

Trick 6 - Create new variables

You might want to create new variables, such as calculating BMI from height and mass. Add as many `COMPUTE` function below as needed. The command reads as `COMPUTE height = ht/100.`, make a new variable called `height` by dividing the original variable `ht` by 100. You can either create a new variable or replace the existing variable by using a new name or the original name, respectively.

Math functions

`+` Add

`-` Subtract

`/` Divide

`*` Multiply

`**` Power

Highlight this command and press the play button.

```
COMPUTE height = ht/100.
COMPUTE weight = weight/100.
COMPUTE BMI = weight/(height**2).
EXECUTE.
```

After computing new variables this is what the data looks like in Figure 7.1.

id	grp	age	weight	ht	time	side	set	rep	aexttorq	aextwork	aextpow	task	height	BMI
1	G	29	.82	182	PRE	R	1	1	70	34	58	cmjbw	1.8200000000000	.25
1	G	29	.82	182	PRE	R	1	1	70	34	58	cmjbw	1.8200000000000	.25
1	G	29	.82	182	PRE	R	1	1	70	34	58	cmjbw	1.8200000000000	.25
2	G	34	.42	161	PRE	R	1	1	52	25	52	cmjbw	1.6100000000000	.16
2	G	34	.42	161	PRE	R	1	1	52	25	52	cmjbw	1.6100000000000	.16
2	G	34	.42	161	PRE	R	1	1	52	25	52	cmjbw	1.6100000000000	.16
2	G	34	.42	161	PRE	R	1	2	54	27	56	cmjbw	1.6100000000000	.16
2	G	34	.42	161	PRE	R	1	2	54	27	56	cmjbw	1.6100000000000	.16
2	G	34	.42	161	PRE	R	1	2	54	27	56	cmjbw	1.6100000000000	.16
2	G	34	.42	161	PRE	R	1	3	60	28	52	cmjbw	1.6100000000000	.16

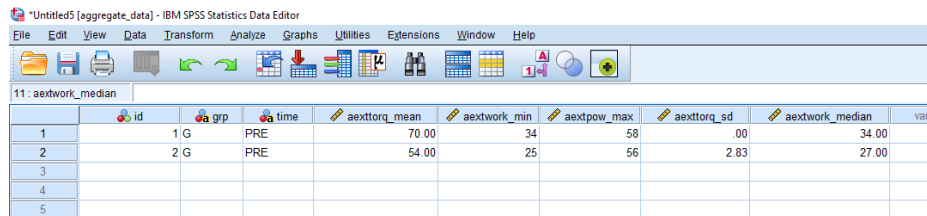Figure 7.1: New variables created

Chapter 8

Trick 7 - Aggregate data

You may want to find some summary statistics for each level of a grouping variable. In the example below, I want to calculate for each id, in each group, at each time, find the mean (`MEAN`), minimum (`MIN`), maximum (`MAX`), standard deviation (`SD`), and median (`MEDIAN`) of the following variables (`aexttorq`, `aextwork`, `aextpow`). When you see this argument, `/aexttorq_mean=MEAN(aexttorq)`, the value before `=` represents the new variable name. However many aggregate values you desire, add as many arguments on a separate line. In this command, the aggregate data is pasted onto a separate data window called `aggregate_data` - you can give the new data window any name.

```
DATASET DECLARE aggregate_data.
AGGREGATE
  /OUTFILE='aggregate_data'
  /BREAK=id grp time
  /aexttorq_mean=MEAN(aexttorq)
  /aextwork_min=MIN(aextwork)
  /aextpow_max=MAX(aextpow)
  /aexttorq_sd=SD(aexttorq)
  /aextwork_median=MEDIAN(aextwork).
```

After aggregating this is what the data looks like in Figure 8.1.

Figure 8.1: Aggregated summary

Chapter 9

Trick 8 - Group by function

Many times you want to do the same thing repeatedly on certain chunks of data. For example, you want to find the mean strength on each level of sex (male, female), or even each level of the combination of sex-side (male-right, male-left, female-right, female-left). Precede whatever function you want to execute, with the command below. In the command below, I want to do the same analysis for each level of the combination of `grp` and `side`. This works for categorical variables. Replace `grp` and `side` with however many variables you want to split the data by.

PS This means you can run the same stats on each split of the data by running the stats only once if you use this command.

Highlight this command and press the play button.

```
SORT CASES  BY grp side.
SPLIT FILE SEPARATE BY grp side.
```

After grouping the data, when I ran a descriptive analysis, this is what the results looks like in Figure 9.1.

▸ Descriptives

grp = G, id = 1

Descriptive Statistics[a]

	N	Minimum	Maximum	Mean	Std. Deviation
aexttorq	3	70	70	70.00	.000
Valid N (listwise)	3				

a. grp = G, id = 1

grp = G, id = 2

Descriptive Statistics[a]

	N	Minimum	Maximum	Mean	Std. Deviation
aexttorq	7	52	60	54.00	2.828
Valid N (listwise)	7				

a. grp = G, id = 2

Figure 9.1: Group-by descriptives

Chapter 10

Trick 9 - Wide to long

id	grp	time	aexttorq_pre	aexttorq_post
1	G	PRE	70.00	34
2	T	PRE	54.00	25

Figure 10.1: Wide data

Typically data is keyed into Excel in the wide format like in Figure 10.1. In > 90% of any data analysis, data should **NOT** be in this format. The only thing I know in SPSS that requires data to be in the wide format is when you want to use Repeated Measures Anova or a Paired t test. So it is useful to know how to convert a data to a long format since > 90% of SPSS function requires data to be in a long format.

`/MAKE val FROM aexttorq_pre aexttorq_post` says collect all the variables `aexttorq_pre`, `aexttorq_post` and stack their values on top of each other. The column containing the values is called `val`. The column where the names of the values are stored is called `mediator`. Remember, you can call the new columns anything you want. `/KEEP=id grp time` simple says keep the following columns in the new data.

Highlight this command and press the play button.

```
VARSTOCASES
  /MAKE val FROM aexttorq_pre aexttorq_post
  /INDEX=mediator(val)
```

```
/KEEP=id grp time
/NULL=KEEP.
```

After running the command, this is what the results looks like in Figure 10.2.

id	grp	time	mediator	val
1	G	PRE	aexttorq_pre	70.00
1	G	PRE	aexttorq_post	34.00
2	T	PRE	aexttorq_pre	54.00
2	T	PRE	aexttorq_post	25.00

Figure 10.2: Long data

Chapter 11

Trick 10 - Long to wide

Just in case you need to make the data wide again. The two key commands are `/ID=id grp time` and `/INDEX=mediator`. `/ID` represents how each data point will be uniquely identified. `/INDEX` represents the variable you want to spread by.

Notice that trick 11 and 10 are mirror images. the `/ID` command in trick 10 should be identical to the `/KEEP` in trick 10. The `/INDEX` in both tricks 11 and 10 should contain the variable name you want to spread by.

```
CASESTOVARS
  /ID=id grp time
  /INDEX=mediator
  /GROUPBY=VARIABLE.
```

After running the command, the data looks like in Figure 10.1 again.

Chapter 12

Conclusion

Notice how I did not even touch of saving any outputs. The reason for that is that one should first save all the commands ran rather than the results. Saving the process allows you to replicate your actions, saving your results will not.