

Block 1: From questionnaire to SPSS saved file

1.3.3.3 First shot at writing SPSS syntax

[Updated 7 Sep 2010]

For this next bit, you need to have hard copies of the [pre-course questionnaire](#) and the [data transfer sheet](#) in front of you (or open on the computer if you don't mind flipping between screens).

We start by deciding which variables we want to define and use in our analysis and then give them [names](#)¹. Variable names in SPSS can be any combination of letters and numbers, but the first character must always be a letter of the alphabet. For beginners it may be easier to use names which look like plain English (e.g. **SEX, AGE, HEIGHT**). These are known as **mnemonic** names (after the Greek for memory) and may be easier to remember.

However, in a survey with a very large number of variables, several hundred variable names may be needed, plus the generation of new and unique names for derived variables such as grouped age and income or scores on attitude scales. Remembering all their names, not to mention where they are located inside the file, and to which bit of the questionnaire they relate, constitutes a prodigious and, to me, pointless effort which virtually guarantees errors and makes smooth working on large data files tedious and cumbersome, especially when using the drop-down menus in SPSS.

This is why I always use syntax instead of the menus and **positional** rather than sequential or mnemonic variable names, but sometimes I have to use question numbers if that's all the information to hand, as in modern surveys such as the [British Social Attitudes](#) or [European Social Survey](#) series, conducted with computer assisted personal interviews (CAPI).

How do we now get SPSS to read the raw data from file **myclass.txt** and save them as variables for later use?

Syntax to read raw data for pre-course questionnaire

Let's take it one step at a time for just 3 variables, **serial** number, **sex** and **age**.

Here's an extract from the raw data file with a template indicating the column layout. The **serial** number is in columns 1 - 2, **sex** is in column 22 and **age** in columns 26 -27.

0	1	2	3	4
1234567890123456789012345678901234567890				
11	24531	112 4	1235	2 1 39 1.68
12	25431	323 3	12	2 1 5 8
13	25413	2 3 3 2 4	1 1 28	5 10
14	42312	322 5	12 5	1 1 22 1.80
15	15324	213 4	23	2 1 36 5 1
16	14523	223 5	12345	1 3 28 1.74

¹ See separate document by John Hall & Jim Ring, [Conventions for Naming Variables in SPSS](#)

Step 1:

Give each of your variables, including the serial number, a name. Use mnemonics for this once! Later, we'll use positional names for most of the remaining variables. The names must start with a letter of the alphabet, and should be kept fairly short to make them easier to find in SPSS and to avoid cluttering up the screen. Thus:

SERIAL **SEX** **AGE**

Step 2

Work out for each variable whether it has been entered in the data as numeric or alphabetic. If numeric, does it have any decimal places? Very few social surveys ever collect data with decimals unless they are financial or medical, so we don't need to bother ourselves with that just yet, but later on we shall need to read the data for height in metres to 2 decimal places. In this example the data for **sex** have already been entered as 1 for male and 2 for female, so we don't need to bother with alphabetic formats either.

Step 3

For each variable, work out which line of data it is on and which position it occupies in the line. In this example there is only one line of data (or record) per case, but later examples will involve data sets with as many as 23 records per case.

There is only 1 line (record) of data per case. It helps to have a table to refer to.

Line	Variable name	Column(s)	Type
1	SERIAL	1-2	Integer
1	SEX	22	Integer
1	AGE	26-27	Integer

[NB: Type **integer** is a number with no decimal places]

Step 4

To read the data for our chosen variables we need to know the line (record) number in which the first variable occurs. There is only record per case, so by definition all three variables are on line 1, and so we specify the line we want by writing a forward slash / followed by the number of the line:

/ 1

Step 5

Next we write down the name of the first variable we want from that line. We want three variables altogether, and they are all on the same line. The first one is **SERIAL**.

/ 1 SERIAL

Step 6

Write in the column(s) within the line where that variable is coded. If it is coded across more than one column, write the start column followed by a hyphen, then the end column. The serial number is in a two-column field 1 - 2.

```
/1 SERIAL 1 - 2
```

Well, that's our first variable specified: now let's do the rest. Data for sex are in column 22 and for age in the two-column field 26 - 27. Repeating steps 1 - 6 above we get:

```
/1 SERIAL 1 - 2 SEX 22 AGE 26 - 27
```

[NB: If you have more than one variable on the same data line you only need to give the line number once.]

What we have just done is to specify our three variables and their respective locations in the data file exactly as they will be specified in SPSS command syntax (the set of instructions you have to compile telling SPSS what to do). However, before we can use this specification, we need to tell SPSS where to look for the data and how many records there are for each case.

To read our raw data into SPSS we are going to use the **DATA LIST** command, followed by two sub-commands, the first, **FILE =** to specify the location and filename of the external data file and the second, **RECORDS =** to specify the number of records per case.

In formal syntax (which will be used from now on for introducing new commands in SPSS):

Command	Keyword	Specification <user-supplied>
DATA LIST	FILE = RECORDS =	<location and name of external data file> <number of records per case>

A "record" in SPSS can be a line of up to 255 columns, but in practice we tend to use only 80 because that is how many columns there used to be on the industry-standard terminal with a visual display unit (VDU) which was based on 80-column Hollerith cards.

In this case we only have one record (line) per case, so **RECORDS = 1** will be needed. But what do we do about **FILE =**? First we need to tell SPSS the location and name (pathway) of the file in which the data are stored. Because such files are not actually part of SPSS, they are known as **external files**. To use this file, SPSS needs to be given the **full pathway name** enclosed in single primes (e.g. **'a:\myclass\myclass.txt'**). In the previous session you created folder **myclass** on your computer desktop (or on a floppy in **a:** or a RW CD in **e:**) and copied [myclass.txt](#) into that folder. If you haven't already done it, go back to [1.3.2.1 Preparing the ground](#) and do it now.

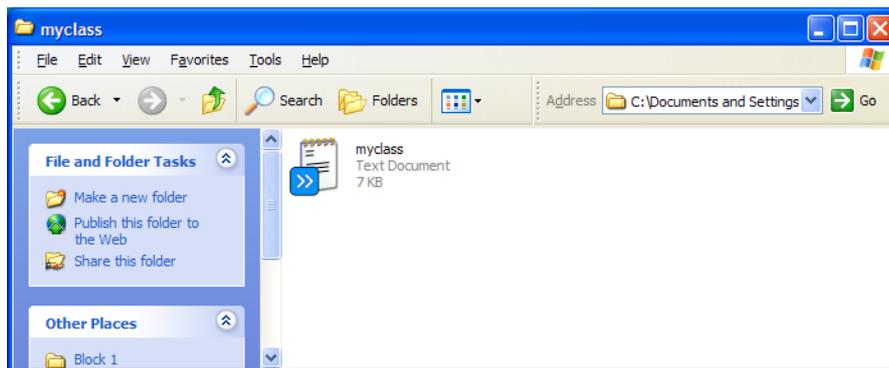
For simplicity, in this example, I'm using drive **a:** and the name **'a:\myclass\myclass.txt'**. I'm also using **lower case** as it's easier to read. The full **data list** command now looks like this:

```
data list file = 'a:\myclass\myclass.txt'  
records = 1  
/1 serial 1 - 2 sex 22 age 26 - 27 .
```

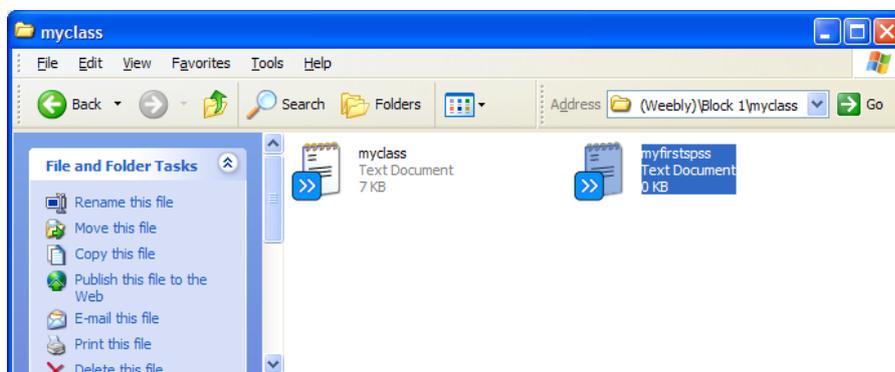
[Note the full stop (period) to end the command.] To read the data from folder **myclass** on your desktop the first line should read (more complex) but you only have to do it once!

```
data list file = 'C:\Documents and Settings\Owner\Desktop\myclass\myclass.txt'
```

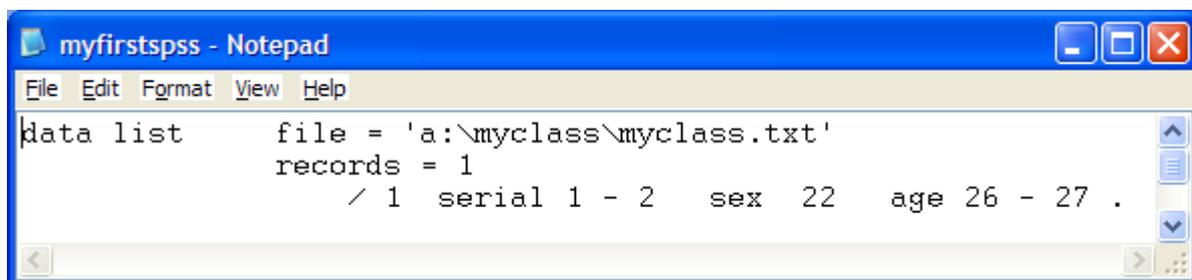
Go to your folder **myclass**



Click on **File** > **New** > [Text document] and call it **myfirstspss**:



...double-click to open the file and type in the syntax you have just written.



...then **File** > **Save As ..** to save the file in your **myclass** folder.

[NB: You have to do this in Notepad or Wordpad: if you do it in Word the primes (apostrophes) are sometimes interpreted as "smart" quotes and when you copy them to the syntax file, unless you change them manually, they cause errors when you run SPSS.]

You have just written your first sentence in SPSS language. Well done!

End of session

Next session: 1.3.3.4 First shot at running SPSS

[\[Back to Block 1 menu\]](#)