Block 1: From questionnaire to SPSS saved file

1.3.1 Conventions [1] used for naming variables in SPSS

[Updated 18 March 2020 from 2010 original]

Imagine a small survey with questions about political party support, age last birthday, sex and support for trade unions (on a five-point scale from 1: "make them illegal" to 5: "give my life to keep them in existence"). These are only a few of the questions asked and the data for each completed questionnaire fills more than one line. Thus we need to know:

how the data for each variable has been coded
which line it is on
whereabouts in the line it is (which column it starts in)
how many columns it occupies (which column it ends in)

Let us suppose that our variables have been coded as follows:

Variable	**Codes**	**Line number**	**Column(s)**
Party support	Con,Lab,LibDem, Other,None,Refuse	1	27
Union support	1,2,3,4,5,8,9	2	34
Sex	M,F	3	07
Age	16 - 90,99	3	08-09

Let us also suppose that the party support categories were hand-coded from their categories into numeric codes (1=Con, 2=Lab, 3=LibDem, 4=Other, 8=Refused, 0=None), but that sex has been coded in its original alphabetic form. As well as the scale of support for trade unions, some people have refused to give an opinion, others have insisted on "Don't know": these have been coded respectively 8=Refused and 9=Don't know. Age has been recorded as the actual age last birthday in the range 16 to 90. Some people have not given their ages and these have been coded as 99.

Having decided which variables we want for our analysis we must first give them all names for our SPSS run. Variable names in SPSS can be any combination of letters and numbers (and certain special characters) but the first character must always a letter of the alphabet. For beginners it may be easier to use a variable name which looks like plain English (e.g. SEX, AGE, PARTY, UNION). These are known as **mnemonic** names (after the Greek for memory) and are easier to remember.

However, in a survey with a very large number of variables, several hundred names may be needed, and the generation of new and unique variable names, not to mention the ability to remember them all and which questions they relate to, constitutes a prodigious and pointless effort which virtually guarantees errors and makes smooth working on large data files all but impossible.

[1] **Author's note:**

This document was updated 2 September 2010 from a 1981 original and, given subsequent developments in SPSS some parts may now be outdated, but the essential logic remains valid as do all of the procedures. Data capture has advanced by leaps and bounds with the arrival of portable computers which contain questionnaires, direct entry and automatic quality controls. Manual data preparation is now obsolete for many large scale surveys, but is still used by individual students and researchers for their own projects. It is not clear whether the notion of a raw data matrix in 80-column format still has relevance now that data entry is made directly into packages such as SPSS, but this could in fact make errors much harder to spot in some instances.

The SPSS files for the SCPR British Social Attitudes series (compiled by Prof. John Curtice's team at Liverpool University and distributed by the UK Data Archive at Essex University) are all of this type and there is a dedicated and vehement school of "Mnemonicism" among the SPSS user community. These mnemonic variable names can be confusing for teaching purposes and for research within a single year of the series, although they can be useful for cross-year comparisons as the same variable names are retained when questions are repeated in successive waves. However, they cannot be used in conjunction with facsimiles of the original questionnaires (which only print information for data preparation).

To make it easier to use data from these surveys for teaching on the Survey Analysis Worshop, we changed the names of the variables (using the SPSS **RENAME VARIABLES**[2] command) to comply with the following variable naming convention, which everyone found much easier to follow and use in conjunction with facsimile copies of the original questionnaires.

At SSRC and PNL we developed a different approach to the naming of variables. Because we handled literally hundreds of questionnaire surveys, because we could not afford errors or lost time, and because we frequently had to deal with "naïve" clients, many of whom had minimal documentation and resources other than their own time, we developed a convention which (except for a few standard variables such as SEX, AGE and MARITAL status) eschews mnemonic names in favour of **positional** variable names.

SPSS variable names must all start with a letter of the alphabet. In the early releases, as well as using mnemonic variable names, SPSS introduced a facility for automatic generation of variable names. This enabled the specification of large numbers of names without having to write them all down one at a time. This involved specifying two variable names starting with the letters **VAR** followed by three digits (eg **VAR001**, **VAR051**). Everything had to be in **UPPER CASE**.

By using the keyword **TO** between a pair of variable names (and provided the second name contained a number greater than that in the first (eg **VAR001 TO VAR051**) SPSS automatically generated names for all the implied variables in between as well as the two named (eg **VAR001 VAR002** .... **VAR050 VAR051**). Many users gave the first variable in their data the name **VAR001** and produced automatically generated names for each variable until they got to the last one. Thus a survey with 240 variables would have had variables specified as **VAR001 TO VAR240**, which is a lot quicker and easier than typing out 240 separate variable names!

We dubbed this convention **sequential** naming of variables.

Later versions of SPSS allowed more flexible naming of this kind so that any letter(s) could be used together with any digit(s). Some people immediately started calling their variables **Q1 TO Q69** and the like, which was certainly an improvement as the names at least bore a more transparent relationship to the questionnaire from which they were purportedly derived! Later releases allowed the use of **lower case** in variable names and syntax (eg **q1 to q69**)

However even this can get cumbersome once we get questionnaires with large classification sections (i.e. household grids with no question numbers) and questions with multiple sub-questions. More than 30 years ago, we put in a request for automatic generation of variable names whose sequence is defined by alphabetical order of their last letter (e.g. Q21a, Q21b, Q21c etc) but this facility is not offered, even in 2010.

When I worked at the Social Science Research Council Survey Unit, we were attached to the London School of Economics. Through the LSE we had access to the CDC2000 at the University

[2] The renaming of variables for the 1989 British Social Attitudes survey was done by Graham Farrant, then a student on the BSc Social Science Social Research Pathway at PNL, later a researcher, then fieldwork manager at the National Centre for Social Research. The original variable names can easily be recovered by switching the variable lists on each side of the **=** sign in the RENAME command.

of London Computer Centre (ULCC) and it was here (in the early 70's) that we had our first taste of SPSS. Before SPSS became available, we had to use commercial services (eg Donovan Data Systems at Research Services Ltd) which could produce client-friendly tables (with labels and full text) or LSE's own **s**urvey **d**ata **tab**ulation program, **SDTAB** written by Peter Wakeford.

The latter was limited in its facilities, but had a convention of naming variables by their column position on an 80-column Hollerith card. SDTAB variables could only consist of a three digit number in which the first digit indicated which card, and the second and third digits which column(s), the data were to be read from. Thus **151** would be the name of the variable on column **51** of card **1** and **237** would be on card **2**, column **37**. Remember in those days everything, including SPSS programs, had to be punched on cards and read into the computer through a card reader. At least the results came back as line-printer output: prior to that even the tabulations were performed on a card sorter. Because of the fixed 3-digit variable names, data were limited to 9 cards per case.

SDTAB's variable naming had the distinct advantage that computer printout could instantly be related to a questionnaire, and vice-versa, provided, of course, that the questionnaire had been properly laid out, not just to record respondents' answers, but also for use as a data-entry tool.

What we started doing at the SSRC Survey Unit, and further developed at PNL, was to combine the automatic name generating capacity of SPSS to the variable naming logic of SDTAB. At PNL we made one major departure from sequential naming so that the numeric part of the variable name relates, not to its **sequential** position in the questionnaire, but to its **actual location** in the coded data. This means that names immediately relate to the line of data within each case, and to the position, or column(s), within each line. In all our variable names, the last two digits tell us which column(s) in a data line contain(s) the coded data for the variable, and the digit (or digits) before them tells us which line of data the variable is on.

Thus **V106** indicates the variable which is coded on line **1**, column **6** of the data (It does not mean the 106th variable in the file!). Conversely, if we have to name a variable, the data for which is to be entered on line **3** in column **44**, we should call it **V344**. Ah, you may say. But what if it's a variable like age which you have coded as two digits on line **3** columns **14** - **15**? Well, when a variable is spread across more than one column (in computer jargon, a "**field**" more than one column wide) we always call the variable by the first column in the field. In our example, **age** would be called **V314** and there would not be a variable V315. The next variable after age would then be **V316**.

All variables in the final version of the SPSS saved file will have names beginning with a letter **V** followed by three or more digits. Exceptions will be derived variables such as scores from attitude scales or complex derived variables such as income when it has been computed from several other variables. Even the standard classification variables such as sex, age and income groups will start life in **Vddd** or **Vdddd** form.

This **positional** convention is much preferable to sequential naming for one other important reason. Most decent questionnaires already had printed on them a data layout template indicating how data were to be punched on 80-column cards or keyed directly into the computer as "card images". Either way, the raw data appeared on line printer output (and later on computer screens) as 80-column lines exactly as indicated by the data layout template (normally printed in the right hand margins of the questionnaire pages).

This gives the immediate and enormous advantage that the original questionnaire can be used as a document for understanding what the variables will be called called and where they will be in the file. Thus **V528** can immediately be checked by looking for the question that has been coded on line **5** column **28**: conversely the question that has been coded on column **28** on line **5** can be expected to be called **V528** (at least on most surveys done by the Survey Research Unit at PNL)

To go back to our original example:

Variable	**Codes**	**Line number**	**Column(s)**
Party support	1 = Conservative 2 = Labour 3 = LibDem 4 = Other 0 = None of them 8 = Refused	1	27
Union support	1,2,3,4,5,8,9	2	34
Sex	“M”, “F”	3	07
Age	16 - 90, 99	3	08-09

These data have, let's say, been stored in a data file called **POLY.DAT** and there are three data lines per case. We want to read in the data for these four variables and then save them in a file called **POLY.SAV** for future use. Remember, we have coded party support using numeric codes, but left sex with its original alphabetic codes 'M' and 'F'.

How do we get SPSS to read these data and save them in an SPSS *.sav file? Let's take it one step at a time.

Step 1:

Give each of your variables a name. (Just this once you can use mnemonics!) The name must start with a letter of the alphabet, but must not be longer than eight characters. Thus: **PARTY UNION SEX** and **AGE** (In our convention these would be **V127 V234 V307** and **V308**)

Step 2

Work out for each variable whether it has been coded in alphabetic format or numeric. If it is numeric, does it have any decimal places? Very few surveys ever collect data with decimals unless they are financial or medical, so we don't need to bother ourselves with that just yet. We do need to specify for each variable what format it is to be read as, "alphabetic" (A) or "integer" numeric (0).

Step 3

For each variable, work out which line of data it is on and which position it occupies in the line.

Step 4

To specify our variables in SPSS we have to start with the line on which the first variable occurs. This is on line 1 so we specify the line by writing a forward slash "**/**" followed by the number **1**:

```
/1
```

Step 5

Next we write down the name of the first variable we want from that line. There is only one and we want to call it PARTY.

```
/1      PARTY
```

Step 6

Write in the column(s) within the line that the variable is coded

```
/1      PARTY       27
```

Step 7

Finally, write in brackets the type of data coded for the variable (in this case, an integer number with no decimals)

```
/1      PARTY       27      (0)
```

Well, that's our first variable specified: now let's do the rest.

A little table helps to show us what to do:

Line	Variable name	Position	Type
1	PARTY	27	Integer
2	UNION	34	Integer
3	SEX	7	Alphabetic
3	AGE	8-9	Integer

So, going over all the steps again we get:

```
/1    PARTY    27    (0)
/2    UNION    34    (0)
/3    SEX      7     (A)
      AGE      8-9   (0)
```

What we have just done is to specify a **DATA LIST** exactly as SPSS needs it before it can read raw data. Before we can use it, however, we need to give SPSS some more information. We need to tell SPSS **which file** the data are to be read from and **how many lines of data** there are for each case.

We do this by means of the **DATA LIST** command, followed by two sub-commands, **FILE =** and **RECORDS =** ("records" is SPSS-ese for "lines" or "cards")

General format:

```
DATA LIST    FILE =    '<external data file>'
             RECORDS = <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 80, partly because that was the number of columns on a standard Hollerith card, but mainly because that was the number of columns on a standard "line" on a computer terminal with a visual display unit (VDU).

In this case we have 3 records per case, so **RECORDS = 3** will be needed. But what do we do about **FILE =** ?

SPSS normally needs to have access to files elsewhere in the computer. These are known as **external** files. To use such a file, SPSS needs to be given the location and name of the external file enclosed in single primes (eg **'A:POLY.DAT'**).

Many files are more than 1.4 mb and too large for floppy **A:** so more often than not, it is necessary to specify the full pathway, eg:

```
DATA LIST    FILE = 'c:\Documents and Settings\Owner\Desktop\poly.dat'
```

So here we go! The full command will look like this:

```
DATA LIST    FILE = 'c:\Documents and Settings\Owner\Desktop\poly.dat'
             RECORDS = 3
             /1    PARTY    27    (0)
             /2    UNION    34    (0)
             /3    SEX      7     (A)
                   AGE      8-9   (0) .
```

Note the full stop (period) to end the command and also that, if there is more than one variable on the same line, you only need to specify the line number once.

If this command is submitted to SPSS, it will not actually do anything at this stage except to prepare the data dictionary (unless you override it, but that's later). In order to read in data from the external file, we must ask SPSS to do something such as execute the command, perform a statistical analysis or ask for some other procedure requiring it to read the actual data, such as to list the contents of a few cases or to display a list of variable names.

We can now ask for frequency tables or some other statistical analysis. For frequencies we could write:

```
        FREQUENCIES      VARIABLES = PARTY UNION SEX AGE .
```

Alternatively, we can simply **SAVE** the work we have done and store it our own private work area. To save the active file, which includes not only the data, but also all the dictionary information we specified on the **DATA LIST**, and put it safely away for future use we write:

```
        SAVE          OUTFILE =    ????
```

Oh dear! Now what? Well, it's back to the old "external file" again, but this time it's going to be a **saved file** with a **.SAV** extension which we shall call **POLY.SAV**, and which needs to be enclosed in single primes. SPSS also needs to know where to save the file. The simplest is to a floppy **A:**

```
        SAVE          OUTFILE = 'A:POLY.SAV' .
```

. . .but many files are too large for this and SPSS cannot save to drive **E:** so a full pathway will again need to be specified. The full program would then look like this....

```
        DATA LIST        FILE = 'c:\Documents and Settings\Owner\Desktop\poly.dat'
                         RECORDS = 3
                         /1      PARTY       27      (0)
                         /2      UNION       34      (0)
                         /3      SEX         7       (A)
                                 AGE         8-9     (0) .

        FREQUENCIES      VARIABLES = PARTY UNION SEX AGE .

        SAVE             OUTFILE = 'c:\Documents and Settings\Owner\Desktop\poly.sav'.
```

The above examples are all based on syntax for various earlier versions of SPSS-X on a 1980s mainframe. They all work with SPSS for Windows on a PC, but some requirements have now changed (eg the **(0)** for integers is no longer needed as it is the default). The Windows version now has a graphic user interface (GUI) with a range of drop-down menus. This makes some tasks more complex and cumbersome, but others, such as saving files, much easier.

Setting up **DATA LIST** for very large files can be time consuming and prone to error, especially when using mnemonic variable names. File building in SPSS is a job which is sometimes best left to experienced people or done under their close supervision. Even for them, we suggest that our positional variable naming will save time and money (and tears!). However, there will always be some researchers who are too proud or arrogant to ask for advice or assistance and who insist on trying to do everything themselves, even if it takes them ten times as long (and costs ten times as much!) and still have to come cap-in-hand for help later.

The advantages of the positional variable namimg convention will be immediately evident when you check the output from an SPSS **DATA LIST** command on which any errors in record number or column location will be immediately apparent as the variable names are printed out side by side with the record and column(s) from which they have been read.

Here's the SPSS output after reading in the **DATA LIST** command for the pre-course questionnaire:

```
Data List will read 1 records from A:\myclass.txt

Variable          Rec   Start     End  Format

serial              1       1       2  F2.0
v4                  1       4       4  F1.0
v5                  1       5       5  F1.0
v6                  1       6       6  F1.0
v7                  1       7       7  F1.0
v8                  1       8       8  F1.0
v10                 1      10      10  F1.0
v11                 1      11      11  F1.0
v12                 1      12      12  F1.0
v14                 1      14      14  F1.0
v16                 1      16      16  F1.0
v17                 1      17      17  F1.0
v18                 1      18      18  F1.0
v19                 1      19      19  F1.0
v20                 1      20      20  F1.0
sex                 1      22      22  F1.0
v24                 1      24      24  F1.0
age                 1      26      27  F2.0
metres              1      29      32  F4.0
feet                1      34      34  F1.0
inches              1      36      37  F2.0
```

Note that **positional** variable names make it much easier to check that they have been read from the correct (start) columns, but I used a few mnemonics as well (eg **sex** and **age**).

There's a lot more to setting up SPSS saved files, such as specifying **MISSING VALUES**, **VARIABLE LABELS**, **VALUE LABELS** and using **RECODE** to change any or all of your alpha variables into numeric, but this is probably enough for now. We'll be doing all this for real using data from the pre-course questionnaire and also real data from major surveys, but first we have to get the computer ready to download and store the files we need.

Next step: 1.3.3.1 Preparing the ground

[Back to Block 1 menu]