# Quantitative Data Processing Procedures

Version 2.0

PUBLIC VERSION

12 October 2010

T +44 (0)1206 872001
E help@esds.ac.uk
www.data-archive.ac.uk

UK DATA ARCHIVE
UNIVERSITY OF ESSEX
WIVENHOE PARK
COLCHESTER
ESSEX, CO4 3SQ

WE ARE SUPPORTED BY THE UNIVERSITY OF ESSEX, THE ECONOMIC AND SOCIAL RESEARCH COUNCIL, AND THE JOINT INFORMATION SYSTEMS COMMITTEE

# Contents



# Scope

**What is in this guide?**

This document contains detail of data processing procedures currently in use at the UK Data Archive.

Within this document:

- All document titles referenced appear in italics
- All typed commands appear in a `preformatted text` font

**What is not covered by this guide?**

This guide does not contain procedures for running the automated quantitative SPSS processing script. These are contained in the document *UKDA-DSS-Processing Script Procedures.*

This guide does not contain procedures for qualitative data collection processing. These are contained in the document *UKDA-DSS-Qualitative Data Processing Procedures.*

This guide does not contain procedures for documentation processing. These are contained in the document *UKDA-DSS-Documentation Processing Procedures.*

This guide does not contain details of Archive study processing standards A*-C. These are covered in the document *UKDA-DSS-Data Processing Standards*.

For a brief guide to study processing, the document *UKDA-DSS-Processing Quick Reference* should be consulted.

It should be noted that some of the documents referenced within the text below are not publicly available, but external readers may of course contact the Archive in case of query.

# 1. Quantitative data procedures in SPSS

**Introduction**

Successful data archiving requires the careful choice of formats in order to strike a balance between effective archival preservation and the provision of data in widely-available and well-supported software formats to enable easy secondary use. The ‘Statistical Package for Social Sciences’ (SPSS) software is well-used within the academic and social science user communities, and thus it is not surprising that the majority of ESDS/Archive depositors deposit data in SPSS format, and the majority of ESDS/Archive users choose to receive data in that format. Therefore, the Archive will create SPSS versions of deposited data files where possible and appropriate, and the bulk of Archive ingest processing work is undertaken using SPSS. Note that a dissemination copy of each data file **must** be created and data edits **only** undertaken on the copy, leaving the ‘original’ deposited file(s) to be archived as received.

Depending on the nature and condition of the individual study, assessment and processing work in SPSS may include a combination of one or more of any of the checking procedures set out below. These may include the generation of descriptive statistics, a data dictionary and a set of variable frequency distributions. Every study is unique, and the content and combination of checking procedures used in SPSS will depend entirely on the nature of the data in question and the processing standard allocated to the study (see document *UKDA-DSS-Data Processing Standards*).

Data checking and processing is normally conducted using an SPSS syntax file in preference to menu-

generated commands (which are also available for most common SPSS procedures). Not only is it easier to run multiple procedures using a single syntax file, but the syntax file will provide a record of data edits undertaken that can be archived with the study as appropriate and can also facilitate easy re-use of complex syntax for future deposits in a data series.

The SPSS routines described below are in alphabetical rather than any preferential order.

## 1.1. Generating descriptive statistics

In SPSS for Windows, to generate descriptive statistics for all variables, the following command may be used in a syntax window:

```
DESC all.
```

(SPSS is not case sensitive. Where the main SPSS command is shown in uppercase, it is for emphasis only.)

**Specimen Descriptives output:**

| | N | Minimum | Maximum | Mean | Std. Deviation |
|---|---|---|---|---|---|
| AGE3 Age | 11471 | 15 | 65 | 39.67 | 12.99 |
| SEX Sex | 11471 | 1 | 2 | 1.50 | .50 |
| QUOTA Stint number where interview took place | 11471 | 1 | 221 | 109.84 | 65.25 |
| WEEK Week number when interview took place | 11471 | 1 | 13 | 7.12 | 3.69 |
| W1YR Year that address first entered survey | 11471 | 5 | 5 | 5.00 | .00 |
| QRTR Quarter that address first entered survey | 11471 | 3 | 3 | 3.00 | .00 |
| ADD Address number on interviewer address list' | 11471 | 1 | 11 | 3.56 | 1.81 |
| Valid N (listwise) | 11471 | | | | |

Greater output can be obtained from the descriptives command using the statistics subcommand:

```
DESC all
/stats=all.
```

Selecting the `/all` subcommand will provide the standard error of the mean, variance, skewness and kurtosis. Note, however, that the default output (minimum, maximum, mean and standard deviation) is normally sufficient for Archive processing purposes.

The descriptives output provides the basis for checking anomalous values, especially for nominal (categorical) variables. For example, consider a nominal (categorical) variable that is (according to the depositor's documentation and/or value labels in the SPSS file) supposed to range between 1 and 8. When one examines the descriptives output, it shows a maximum value of 9. There is, therefore, an additional value that has not been defined by the depositor. Either this represents an undefined code or one or more errors in the data. The frequency distribution of that variable can then be examined to see the extent of the problem.

The descriptives command can also throw up erroneous values for interval variables. In many instances, one will have little knowledge of the possible range of values for an interval variable, but in the case of some variables, such as 'age' (age in years), values below 0 or above, for example, 120 would indicate the presence of possible errors.

## 1.2. Generating a data dictionary

The display dictionary command generates what SPSS calls the 'data dictionary', which is a detailed description of the contents of the data file, giving details of each variable in terms of the variable name, variable label, variable type, format, missing values, and value labels.

To run the display dictionary command, the following syntax can be used:

```
DISPLAY DICTIONARY.
```

(This command will display the data dictionary for the whole file, and can be modified to display a list of selected variables. see online syntax guide within spss for details.)

For versions of SPSS prior to 12.0 (and in Unix) the data dictionary output looks like this:

**Specimen data dictionary output**

List of variables on the working file

| Name | | Position |
|---|---|---|
| AGE | Age | 1 |
| | Measurement Level: Scale | |
| | Column Width: 8 Alignment: Right | |
| | Print Format: F8 | |
| | Write Format: F8 | |
| | Missing Values: -9, -8 | |
| SEX | Sex | 2 |
| | Measurement Level: Ordinal | |
| | Column Width: 8 Alignment: Right | |
| | Print Format: F8 | |
| | Write Format: F8 | |
| | Missing Values: -9, -8 | |
| | Value Label | |
| | 1 Male | |
| | 2 Female | |
| QUOTA | Stint number where interview took place | 3 |
| | Measurement Level: Scale | |
| | Column Width: 8 Alignment: Right | |
| | Print Format: F8 | |
| | Write Format: F8 | |
| | Missing Values: -9, -8 | |

Data dictionary display in SPSS versions 12 and over is far less user-friendly: it splits the variable name and label output from the value label output. However, the data dictionary produced by the processing scripts currently in use at the Archive provide listed variable and value labels in a formatted RTF file, using a display similar to the one shown above. For further details, see the document *UKDA-DSS-Processing Script Procedures.*

### 1.2.1 Useful elements of the display dictionary command output

If the descriptives command forms the basis of checking the data, the display dictionary command forms the basis of checking the internal metadata. The most useful elements of data dictionary output are as follows:

**Variable name**

Lists the name of each variable.

**Variable label**

A description of the variable appears to the right of the variable name; this is the variable label (if one has been defined). Note that SPSS for Windows (version 7.x upwards) can store variable labels up to 255 characters, but some automated Computer Assisted Personal Interviewing (CAPI) questionnaire software, such as Blaise, may truncate labels or add random characters, such as @. These labels may need to be edited, especially for A* standard studies that are to be added to Nesstar (see document *UKDA-DSS-Data Processing Standards*).

**Measurement level**

For numeric variables this can be nominal, ordinal or scale (i.e. interval)[1]. However, this cannot always be taken as meaningful as SPSS defines these using a simple rule based on the number of unique values (calculated on the fly when the file is opened): numeric variables with fewer than 24 unique values and string variables are set to nominal, and numeric variables with 24 or more unique values are set to scale (SPSS settings can be changed if necessary).

**Print and Write format**

Unless the depositor has specified otherwise, these will be one and the same. If the Format is of the form FX (where X is a number, typically 8), the variable is almost certainly nominal (categorical) or ordinal (since no decimal places are defined). If the variable format is of the form FX.Y (where Y is the number of decimal places) then the variable may well be interval (though note that F8.2 is SPSS's default format).

**Missing values**

This line only appears if any user-defined missing values are present; system-missing values are not listed here.

**Value labels**

This line only appears if value labels have been defined for a variable (as for 'sex' in the specimen output above). Each value label is listed.

## 1.3. Making use of the descriptives and data dictionary output

In combination with a visual examination of the data, the descriptives and data dictionary output provides the basis for the following checks:

- unlikely or impossible values for interval variables;
- undefined or incorrect values for nominal (categorical) variables;
- completeness and interpretability of value labels for nominal (categorical) variables;
- missing values appear sensibly and consistently defined (for example, if 'refused' is defined as missing for one variable, is it defined as missing for other variables?).

## 1.4. Generating frequencies

The SPSS frequencies command provides useful information in addition to that provided by the descriptives command. For example, consider a nominal (categorical) variable that is supposed to range between 1 and 8, but the descriptives output shows a maximum value of 18. From the descriptives output it cannot be seen

[1] Useful definitions of quantitative variable types may be found on the GraphPad.com web site at: http://www.graphpad.com/faq/viewfaq.cfm?faq=1089 (retrieved October 12, 2010).

whether there is a single case with a value of 18 (in which case, it's most probably a data entry error), or whether there are many values between 8 and 18, in which case, a more substantial problem exists (either incorrect mapping of value labels or very 'dirty' data).

To see the number of cases (observations) in every value category the SPSS frequencies command must be run:

```
FRE all
```

- generates frequencies for all variables. (Note that SPSS may have an internal limit on the amount of variables for which frequencies may be generated in one command; this is typically 1,000 or even 500. in these cases, separate frequency statements that respect these limits will need to be written, using the 'to' command, as given in the example below.)

```
FRE var1 var2 var3 var4
```

- generates frequencies for the variables specified (substitute actual variable names for var1, var2, etc.).

```
FRE var1 to var321
```

- generates frequencies all variables that lie between var1 and var321 in SPSS file order (substitute actual variable names for var1 and var321).

### 1.4.1. Useful frequencies subcommands

To generate frequencies for nominal (categorical) variables only, the formats limit sub-command can be used to suppress output for variables that have more than a specified number of values. In the example below, only frequencies for variables with up to 30 unique values will be generated - which will suppress output for the interval variable age (which has more than 30 unique values), and generates output only for the variable sex, which has two unique values.

```
FRE age sex
/format limit=30.
```

**Specimen output**

SEX

| | sex | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | 1 Male | 5750 | 50.1 | 50.1 | 50.1 |
| | 2 Female | 5721 | 49.9 | 49.9 | 100.0 |
| | Total | 11471 | 100.0 | 100.0 | |

## 1.5. Display Labels

This command generates variable labels. These are also generated by the display dictionary command (along with value labels and variable format information), but to simply generate variable names with their SPSS data file position and variable labels, use the DISPLAY LABELS command.

```
DISP labels
```

**Specimen output:**

Variable Labels

| Name | Position | Label |
|---|---|---|
| AGE3 | 1 | Age |
| SEX | 2 | Sex |
| QUOTA | 3 | Stint number where interview took place |
| WEEK | 4 | Week number when interview took place |
| W1YR | 5 | Year that address first entered survey |
| QRTR | 6 | Quarter that address first entered survey |
| ADD | 7 | Address number on interviewer address list |

Display labels can be used as a quick guide to highlight which variables have no label, which may be useful for studies to be processed to A/A* standard, where missing variable labels are routinely added.

## 1.6. Weighted studies

Many studies contain variables that are used to perform weighted statistical analysis (a technique typically used to make a sample representative of some important criteria, such as population figures). However, it is desirable to take off the weighting for ingest processing and archiving, unless the depositor has specifically requested that the weight remain on (this should be recorded in the Note file for information).

If an SPSS study is weighted, 'Weight on' will be displayed in the bottom right hand corner of the SPSS data window. However, this does not always display when the data are in SPSS portable format (a .por file), so the syntax command to show whether the study is weighted is:

```
SHOW weight.
```

For unweighted data, the output should read 'the file is not weighted'

If the output reads 'weighting variable = <variable name>', the data are currently weighted by the variable shown. Weighting data will not alter the core data matrix (i.e. the actual data values SPSS displays in its data viewer) but will alter any statistical output, such as descriptive statistics. The weight should therefore be removed, using the command:

```
Weight=off.
```

Documentation should be checked to ensure that comprehensive information on weighting variables and their construction and use is given. The depositor should be contacted and extra information requested if that provided is not adequate.

For a useful guide to the principles of data weighting, see the ESDS guide, *Weighting the Social Surveys,* at http://www.esds.ac.uk/government/docs/weighting.pdf .

## 1.7. Identifying cases with anomalous values

The commands described above, particularly descriptives and frequencies, will show any anomalous values, particularly for nominal (categorical) variables. An anomalous value of a nominal variable is one that is unexpected (according to the documentation and/or internal metadata). For example, the documentation and value labels in SPSS for the variable mstatus (marital status) may be described both in the documentation and the SPSS value labels as 1=married, 2 = divorced, 3 = single; yet the descriptives command reveals a maximum value of 6, and the frequencies command reveals a substantial number of cases of values 4 and 5 as well. As a result, the meaning of codes 4-6 is unknown. They may either be invalid (i.e. data errors) or

they may be incorrectly labelled (i.e. the data are correct but the codes 4 to 6 have not been defined by value labels or in the documentation).

Assessment of an anomalous value for an interval variable may rest on logic and knowledge alone, rather than whether or not a value label or code is defined. For variables that are inherently comprehensible – like age or income – checks can be made for values that lie outside the realm of probability or possibility (such as less than 0 or greater than 120 for an age variable).

Whenever an anomalous value is found, the offending case(s) may be examined to see if the other variables for that case provide any clue as to the source of errors. To do this, the Temporary command followed by the Select If command should be used. A logical statement must be provided that selects out cases that meet a given condition, such as a certain variable having a certain value. The Select If command should **always** be preceded by the Temporary command, otherwise a permanent selection of cases will be made, and if the file is saved all non-selected cases will be lost! (For safety reasons, processing work must always be undertaken on a processing copy of the file rather than the original.)

The following example temporarily selects all cases for which the variable Mstatus is greater than 3 (for example, values may be 1=single, 2=married, 3=living as married). The LIST command will then provide the number/name of any variable(s) specified. In the example below, the Hhno (household number variable) of all cases that record an Mstatus value in excess of 3 will also be given alongside the value of Mstatus.

```
TEMP.
SELECT IF mstatus>3.
LIST mstatus hhno.
```

Using the value of the Hhno variable (or whatever the data file's unique identifier variable may be), the cases in the data file with an anomalous Mstatus value can be examined to see if there is any discernible reason for the anomaly. There may be some distinct pattern, for example, all cases in a certain geographical area contain an undefined code for a location variable, or all people aged under 16 display an undefined marital status variable. Even if such detective work does not provide a definitive answer by itself, the more information that can be discerned during processing, the more likely a speedy solution can be sought from the depositor, or information provided for the secondary user.

## 1.8. Identifying non integer values

In addition to frequencies output, a test for non–integer values can be run to check that a variable that should only have integer values does indeed do so. This uses the modulus function (termed 'mod' in the SPSS command language), which enables a check on whether there is a remainder when the variable is divided by one. There will be no remainder if the variable is always an integer, so no cases will be selected by the syntax below, where <varname> is the name of the variable concerned:

```
Temporary.
Select if mod (<varname>, 1) ne 0.
List <varname>.
```

Any cases listed will contain non-integer values, and will need to be examined if the variable is nominal or ordinal. Again, the Temporary command should precede 'Select if'.

## 1.9. Print and write formats, variable width and decimal places

SPSS print formats control how variables are displayed on the screen in the SPSS data viewer (the 'variable width' column in variable view). The default format for a numeric variable is defined as **F8.2**, which allows up to 8 digits to the left of the decimal point and two digits to the right (i.e. two decimal places). An **F8.2** format will display numbers up to 99999999.99 and will use scientific notation (E+X) for larger numbers.

Print and write formats can be problematic when importing data into SPSS. In some instances, variables with many decimal places may be displayed as an integer (e.g. 1.348726 being displayed as 1), or with only one or two decimal places. This may confuse naïve end users of the data, though the data are correct. More importantly however, print and write formats should also be considered as data export formats, as they can affect certain conversions, most importantly from SPSS portable (.por) to STATA, whereby the resulting data file only holds the data to the number of decimal places given in the SPSS print format, rather than the number contained in the SPSS data file itself. For safety, SPSS to STATA conversions should always be performed using the SPSS system file (.sav) rather than portable format. The processing script currently in use at the Archive creates STATA from .sav format.

The variable width in SPSS can present similar problems. A value of 10 or -1 can exist in a variable defined as having a width of 1 (e.g. defined as F1.0), but this can also affect export formats, depending on how conversion is performed. For example, two-character missing values (e.g. -1) will display as * if the format is set to F1.0. This can create problems for both Nesstar and STATA conversion, so all quantitative data files should have the formats for variables set to F2.0, or the appropriate display format length before the processing script is run.

If such formats inevitably lead to rounding, truncation or loss of data upon conversion to other preservation or dissemination formats, they must be altered prior to conversion. Therefore, all file transfers must be checked thoroughly to ensure data integrity is preserved.

The SPSS command to change or set formats is as follows:

```
FORMATS age sex marstat (F2.0).
```

This command will set both print and write formats for the numeric variables ‘age’, ‘sex’ and ‘marstat’ to two characters, with no decimal places.

```
FORMATS income (F10.6).
```

… will set both print and write formats for the numeric variable ‘income’ to 10 characters and 5 decimal places.

```
FORMATS council (A20).
```

… will set both print and write formats for the string (text) variable ‘council’ to 8 characters.

For further information on formats and similar commands, see SPSS online help or manuals.

## 1.10. SPSS 14 record display issues

Occasionally, SPSS files may not display the full number of cases (records) in the Data View screen of SPSS version 14.0. The cause of the problem is currently uncertain, but it may be due to driver file or computer display issues. The correct number of cases are usually still contained within the file, even if they do not show; the problem may first be detected when descriptives or frequencies statistics are generated, because more cases will be displayed in the output results than appear in the Data View screen. As the depositor does not always provide information on the correct numbers of cases and variables for each file, it is very important to run at least one test on each SPSS file before running the processing script to check the number of cases (ensure that the data file weight is off (see section 1.5 above)). If the file contains only numeric variables that are not suitable for frequency generation, descriptive statistics can be run instead to display the numbers of cases (see table in section 1.1).

As an extra safeguard, in case the problem is not apparent beforehand, full checks of case numbers vs. SPSS Data View displays should also be made on all files and outputs after the processing script has run.

If the display problem is found to be present, it can usually be cured by converting the file to SPSS portable format, checking that the numbers of cases appear correctly in the portable file, and if all is well, converting it back to SPSS .sav format, which should then display the correct number of cases. If unsure, please consult the Data Services Manager.

## 1.11. Rounding in Microsoft Excel and Microsoft Access

All file transfers should be checked thoroughly for rounding. The issue of 'rounding' affects Microsoft Excel display formats. Depositor-specified limitation of decimal places (to make the data or output more easily interpretable) should be treated as a data export format. Export to, for example, tab-delimited text, will result in data rounded to the displayed number of decimal places rather than the number in the underlying data. Similarly, export of tables from Microsoft Access can result in rounding of decimal places to two digits, due to the internal software settings. See section 7 for information on the processing of Access databases.

# 2. Adding variable and value labels

One of the most common processing tasks undertaken at the Archive is the addition/editing of variable and value labels. The extent to which this is undertaken depends on the condition of the deposited data file (i.e. how many labels are missing) and the Archive processing standard allocated to the study (see document *UKDA-DSS-Data Processing Standards*). Labelling is always carried out on the dissemination copy of the data file, not the deposited 'original' (see section 1 above), and is usually carried out in SPSS prior to data format conversion (see section 3 below). In order to enable 'tracking' of the edits undertaken, the labels are added using a syntax file rather than being added directly to the file via the SPSS graphical interface. The syntax file is then archived with the study. The archiving of syntax files also has another advantage; in the case of series studies, the same labels are often missing or need editing at each wave. Therefore, archived syntax from the previous wave may often need only minor editing to enable it to run on the latest wave, reducing subsequent work. Of course, thorough checks should be made against the latest wave's data and documentation to ensure that the labels are still valid.

When variable and value labels have been added/edited as necessary, the results should be checked by running frequencies to check the amended variables. If the additions/edits have been successful and all other errors in the data fixed or noted, conversion from SPSS to dissemination and archival formats (usually STATA and tab-delimited text) may be undertaken (see section 3 below).

## 2.1. Adding variable labels

To add/edit variable labels in SPSS, open the data file required, and a syntax file. Use the 'variable labels' command, and list underneath the variables that require (re)naming alongside the label required (see example below). At the end of the list, use a full stop as the command terminator.

*Adding/editing variable labels*

```
variable labels

nb7pint     Pint equivalent

fuh         Family unit head

fuhage      Family unit head age

fuhsex      Family unit head sex

husbage     Age in years of male partner
```

```
husband     Person number of male partner

husbmar     Marital status of male partner

mothage     Age in years of mother

mother      Person number of mother

partage     Age in years of partner

.
```

Once labelling is complete, the syntax file should be saved and archived with the study in the dp/code/spss (or appropriate software format) directory. For practical purposes, variable/value label and other edits may be combined into one syntax file.

## 2.2. Adding value labels

To add/edit value labels in SPSS, open the data file required, and then open a syntax file. The procedure is similar to that used for variable labelling, with the variable names and required values specified (see example below). Two SPSS commands may be used: ‘add value labels’ or ‘value labels’:

- ‘add value labels’ will add labels or change only those specified in the syntax. This option is often more appropriate for Archive processing, as in many cases, only one or two value labels from an individual variable may be truncated or missing, while others are correct.
- ‘value labels’ will erase all existing value labels within the variable and add/edit only those specified in the syntax. Therefore, to provide adequate value labelling, the whole set of value labels for the variable must be specified.

Note: where the label contains a character that SPSS usually interprets as a command, e.g. / or \ , the label should be enclosed in single quotation marks.

Similarly, to label ‘string’ or textual variables, single quotation marks may need to be used. If unsure, refer to the help guide for the version of SPSS in use.

```
add value labels

grifuh grifp grhheq ntquint pfempgr1 traingr1

1     Yes

2     No

-9    'DNA/Child/Proxy/No int'

-8    NA

-7    Refused section

-6    'CHILD/MS/PROXY'

.
```

Once labelling is complete, the syntax file should be saved and archived with the study in the dp/code/spss (or appropriate software format) directory. As described in section 2.1 above, the variable/value label and other edits may be combined into one syntax file.

# 3. Converting from SPSS to STATA and tab-delimited text

The conversion process from SPSS to the current Archive standard dissemination formats (STATA and tab-delimited text) is currently undertaken on SPSS system files (.sav), using an automated script that runs in SPSS. The procedure for installing and using the script is contained within a separate document. Once the dissemination formats have been created, they should be checked using analogous procedures to the SPSS checks already undertaken, to ensure data integrity during transfer. Obviously, the checks undertaken on tab-delimited files are more limited than those that can be undertaken in STATA. A useful guide to simple data analysis with STATA, written by the ESDS Government team and using Labour Force Survey (LFS) data, may be found at http://www.esds.ac.uk/government/resources/analysis/ .

# 4. Checking SPSS to STATA transfers

Transfer between SPSS and STATA can be problematic due to the differing nature of the two software packages. Thorough checks must be carried out on all STATA files generated from SPSS, either via the Archive processing script, via proprietary software such as StatTransfer, or exported direct from SPSS (versions 14 and above). Most problems are evident from very basic post-transfer checks, but not all; information on more complex problems that may result during transfer (and their solutions) are available internally for Archive staff, covering incorrect value transposition, incorrect missing value transfer and problems with variable name transfer.

Basic information on potential truncation during transfer to STATA is given in the RTF ‘SPSS to STATA’ RTF document generated as part of the processing script outputs, based on transfer to STATA version 8.0. The likely effects of transfer are as follows:

1. String variables in the SPSS file with a defined width of >80 characters (the standard STATA limit) or >244 characters (the STATA Special Edition (SE) limit) will be truncated.
2. Variable labels in the SPSS file of >80 characters (the STATA limit) will be truncated.
3. Value labels in the SPSS file of >32 characters (the STATA limit) will be truncated.
4. String variables that have value labels in the SPSS file will lose these in STATA.
5. Non-integer values that have value labels in the SPSS file will lose these in STATA.

## 4.1. Analogous checking procedures in STATA

The STATA statistical software package has a command language similar to that of SPSS. STATA is currently the second most popular statistical software package among the Archive’s user community, and is preferred by many economists. Some of the key STATA commands useful for Archive data processing are detailed below.

### 4.1.1. Analogous STATA command to ‘Descriptives’ in SPSS

The closest STATA command to the SPSS Descriptives command is the STATA **summarize** command. This will produce descriptive statistics for all numeric variables, displaying: number of (valid, i.e. non-missing) observations; mean; standard deviation; and minimum and maximum values.

Specimen output from the STATA **summarize** command:

| Variable | Obs | Mean | Std. Dev. | Min | Max |
|---|---|---|---|---|---|
| state | 0 | | | | |
| region | 50 | 2.66 | 1.061574 | 1 | 4 |
| pop | 50 | 4518149 | 4715038 | 401851 | 2.37e+07 |
| poplt5 | 50 | 326277.8 | 331585 | 35998 | 1708400 |

| | | | | | |
|---|---|---|---|---|---|
| pop5_17 | 50 | 945951.6 | 959373 | 91796 | 4680558 |
| pop18p | 50 | 3245920 | 3430531 | 271106 | 1.73e+07 |
| pop65p | 50 | 509502.8 | 538932 | 11547 | 2414250 |
| popurban | 50 | 3328253 | 4090178 | 172735 | 2.16e+07 |
| medage | 50 | 29.54 | 1.693445 | 24.2 | 34.7 |
| death | 50 | 39474.26 | 41742.35 | 1604 | 186428 |
| marriage | 50 | 47701.4 | 45130.42 | 4437 | 210864 |
| divorce | 45 | 23679.44 | 25094.01 | 2142 | 133541 |

**Note:** 'State' is a string variable in the example above, so STATA cannot give any output. Also, the 'divorce' variable must have five missing cases, as valid observations are given as 45 rather than 50.

Like SPSS, individual variables can be specified. For example, the command:

```
summarize death marriage
```

… will only generate descriptive statistics for these two variables.

Unlike SPSS and SAS, STATA commands are case sensitive (they are always lower case). The wildcard (*) can also be used for most STATA commands, for example:

```
summarize a* m*
```

will summarise all variables beginning with the letters a or m.

### 4.1.2. Analogous STATA command to 'Display Dictionary' in SPSS

There are two choices in STATA. As detailed below, the **describe** command produces output that is slightly less detailed than the SPSS data dictionary, while the **codebook** command produces output that is more detailed. These two options are discussed in turn:

The **describe** command generates variable names, storage formats (using the STATA definition, not SPSS formats); display formats (STATA rather than SPSS); value labels and variable labels.

**Specimen output from the `describe` command:**

| | | | | |
|---|---|---|---|---|
| obs: | 50 | 1980 Census data by state | | |
| vars: | 12 | 6 Jul 2000 17:06 | | |
| size: | 3,000 (99.5% of memory free) | | | |
| | | | | |
| variable name | storage type | display format | value label | variable label |
| | | | | |
| state | str14 | %-14s | | State |
| region | int | %63.0g | test | Census region |
| pop | long | %12.0gc | | Population |
| poplt5 | long | %12.0gc | | Pop, < 5 year |
| pop5_17 | long | %12.0gc | | Pop, 5 to 17 years |
| pop18p | long | %12.0gc | | Pop, 18 and older |
| pop65p | long | %12.0gc | | Pop, 65 and older |
| popurban | long | %12.0gc | | Urban population |
| medage | float | %9.2f | | Median age |
| death | long | %12.0gc | | Number of deaths |
| marriage | long | %12.0gc | | Number of marriages |

| | | | | |
|---|---|---|---|---|
| divorce | long | %12.0gc | | Number of divorces |
| highdiv | int | %63.0g | yesno | High divorce state? |

## 4.2. Guide to understanding STATA storage types:

**str**

string (text variable, of specified length); e.g. in the example above, ‘state’ is a string variable with 14 characters (str14).

**int** and **long**

integer variables, i.e. a categorical or ordinal variable; see ‘region’ above.

**float** and **double**

an interval (continuous) variable of up to 8 digits in accuracy, see ‘medage’ above.

Note also that value labels have only been defined for the highdiv variable. Yesno is the name of the value label (this is not the same as a ‘value label’ in SPSS; this feature does not exist there). The value labels are mapped to one or more variables sharing the same coding (e.g. several variables might have the ‘yesno’ value label.

To look at value labels (as defined in SPSS) themselves, use the **label list** command, e.g.:

```
Label list highdiv
0      No
1      Yes
```

To **generate** output that is more detailed than the SPSS data dictionary, use the **codebook** command:

**Specimen output from the** `codebook` **command:**

```
state ------------------------------------------------------------------- State
type: string (str14), but longest is str13
unique values:  50                coded missing:  0 / 50
examples:  "Georgia"
"Maryland"
"Nevada"
"S. Carolina"
warning:  variable has embedded blanks
region ------------------------------------------------------------ Census region
type:  numeric (int)
label:  regcode, but 3 values are not labelled
range:  [1,4]                     units:  1
unique values:  4                 coded missing:  0 / 50
tabulation:       Freq.    Numeric  Label
9                    1
12                   2         Tendring
16                   3
13                   4
pop      ------------------------------------------------------------------ Population
type:  numeric (long)
range:  [401851,23667902]          units:  1
unique values:  50                coded missing:  0 / 50
mean:   4.5e+06
```

| | | | | | |
|---|---|---|---|---|---|
| std. dev: 4.7e+06 | | | | | |
| percentiles: | 10% | 25% | 50% | 75% | 90% |
| | 67174 | 1.1e+06 | 3.1e+06 | 5.5e+06 | 1.1e+07 |

As can be seen, this gives extremely detailed information for each variable, almost as much as descriptives and the data dictionary combined in SPSS terms.

## 5. Checks on tab-delimited text files

The Archive's current preservation format is tab-delimited text (either of the ASCII or UNICODE character set), though it is envisaged that this will move to XML (extensible markup language) in the future.

The limitation of tab-delimited format is the fact that it can only store the rectangular matrix of data points and variable names. What can be termed internal metadata - variable descriptions (labels), code descriptions (value labels), and whether certain codes are defined as 'missing', cannot be stored in the same file as the data. Such information needs to be stored in an additional series of tab-delimited files or in the documentation.

Tab-delimited text can be read into almost any statistical package, database, spreadsheet or word-processor, and so can also be considered a dissemination format, though most users request data in a software dependent format. Similarly, it is rare nowadays for data to be deposited in tab-delimited or other text format, unless in the form of database output files when no other alternative is possible.

Therefore, tab-delimited text will be encountered most frequently during checks on the outputs created by the processing script currently in use at the Archive. The following checks should be made, using the original SPSS .sav file and RTF data dictionary file (created by the script and archived under mrdoc/allissue) as a guide:

- all cases have transferred successfully
- the variable names are included in the top line of the tab-delimited file
- all variables have transferred successfully.

Note that the visual inspection of tab-delimited files during output checking may be more successful with a simple text-editing software package such as Pfe32 (or even NotePad for small files). More sophisticated text-editing software packages such as UltraEdit may wrap long lines and make it difficult to assess whether the correct number of cases has been transferred by SPSS. If unsure, please consult the Data Services Manager.

## 6. Analogous checking procedures in SAS

Occasionally, studies may be deposited in SAS format, although this is rare nowadays. SAS also has a full command language that allows similar (and in some areas greater) functionality to SPSS and STATA. Data deposited as .sd7 or .sasb7dat files can be read directly into recent versions of SPSS. SAS transport files (.ctp, .xpt) need to be converted using SAS 8.0 for Windows or above; for instructions on how to convert such files, consult the Data Services Manager.

Note that the transfer of some data files from SPSS to SAS may also be problematic; checks should be carried out accordingly.

# 7. Access databases

Microsoft Access is the most common ingest format for databases. It should be noted that each version of Access has some incompatibilities with former versions. Access databases may comprise tables only, or those with some functionality, such as formatting, queries, forms, reports and/or macros. For table-only Access databases, an Excel version may be created alongside the Access database, as an alternative dissemination format.

Where functionality exists, the tools (macros, reports etc.) should be examined to ascertain whether they are an aid to secondary analysis. If so, the usual dissemination format must remain an Access database – see also section 7.3 below.

The Archive preservation format for Access is almost always a tab-delimited version of each table, since Access databases may contain long (>255 character) textual strings.

Deposit in Access format is currently more common for data deposited with the History Data Service (HDS) and some qualitative and Rural Economy and Land Use Programme (RELU) data.

The following procedures should be read carefully before any conversion is attempted on an Access database.

## 7.1. Text strings of 255 characters and over

Access will only allow textual strings of >255 characters in ‘Memo’ fields. These have effectively unlimited length, and can contain embedded special characters such as tabs and carriage returns. Memo fields must be carefully checked for two reasons:

- When exported to tab-delimited text format, memo fields may be truncated to 255 characters when subsequently imported into other packages (including import back into Access).
- Where embedded tabs or carriage returns exist, data may not import as a rectangular matrix.

## 7.2. Dealing with memo fields and undertaking format conversions

To ascertain whether a table within Access contains memo fields, right-click on the table name, and choose ‘Design view’ from the list. This will show how each field is defined in Access. If there are no memo fields, conversion to tab-delimited format may proceed.

The quickest way to establish whether embedded carriage returns or tabs exist in the memo fields is to test-convert the table to tab-delimited format and reopen it in Access. If the data are no longer in the same rectangular matrix (i.e. the rows and columns no longer tally), they probably contain embedded tabs or carriage returns. To confirm this, the contents of cells can be cut and pasted into UltraEdit or Word to examine for embedded characters.

If embedded carriage returns and/or tabs are found to exist, they should be removed. Before any editing work is undertaken, a copy should be made of the Access database, and the editing work undertaken on the copy, to avoid any inadvertent damage.

Once the embedded carriage returns and tabs have been removed, the Access table should convert successfully to tab-delimited format. The long strings may need further checks when the tab-delimited version is converted to another format.

## 7.3. Converting Access tables to tab-delimited format

Following depositor and user feedback, the Access database should remain the primary dissemination format, but the tab-delimited text files generated via the procedures described below make a useful archival

format, and should be archived under ‘noissue’ in a ‘tab’ directory. The tab files may of course be supplied to users on request.

### 7.3.1. Decimal place rounding during table export from Access

Exporting Access tables directly into tab-delimited text format may automatically round/truncate all values to two decimal places. Therefore, if a table within the database contains values with more than two decimal places, a few extra steps should be undertaken to ensure this truncation does not occur.

Firstly, potential problems may be averted by the export of individual tables to MS Excel. This works for some Access databases, but not for others. If export to Excel works well, the resulting files may then be exported from Excel to tab-delimited text, taking care to ensure no further rounding/truncation occurs in that transfer.

If export to Excel is not successful in avoiding rounding, a copy should be made of the Access database, as for the editing of memo fields described in section 7.2 above. As changes are to be made to table construction, performing the work on a copy of the database is essential, to avoid passing changes on to secondary users of the Access database.

1. In Access, select the table for export, then click on the ‘Design’ button to open the Design view of the table.
2. Each Field Name should have a Data Type specified. To prevent decimal place truncation, the Data Type of the affected variable(s) should be changed to Text.
3. Click on the current Data Type and a drop-down menu should appear. Text should be selected.
4. Save the database.
5. The table can then be closed which will return to the Main view where all tables within the database may be seen.
6. The instructions for converting Access tables to tab-delimited format given in 7.3.2 and 7.3.3 below can then be followed.

### 7.3.2. Access 2003 (and 2002)

Each table must be saved separately. To convert a table, select it and then go to:

File Menu> **Export**

Select the appropriate location, then choose **Text Files (*.txt; *.csv; *tab;*asc)** from the drop-down menu. Click Export, then in the **Export Text Wizard** interface, make sure **Delimited** is selected, then click **Next**, then do the following:

Choose the delimiter that separates the fields: **tab**

Tick **Include Field Names on First Row**

Set **Text Qualifier** to **{none}**

Then click on **Next**, then **Finish**.

You may need to change the resulting file extensions from .txt to .tab.

### 7.3.3. Access 2000

To convert, choose the following settings:

File Menu: **Export**

Save as type: **Text files**

**Save all**

Make sure **Delimited** is checked, then click **Next**, then follow instructions as for Access 2002

**Notes**

- Text qualifiers should only be used where there are known to be embedded tabs in memo fields (and these have not been removed). If qualifiers are used this must be stated in the Note file.
- Always reopen the tab-delimited files back into Access to perform version control checks. Any import errors are stored as a table in the current database by Access.
- Although long text strings (memo fields) may need most work, the other version control checks - on date fields, decimal places, etc. - need to be carried out in accordance with the level of processing allocated to the study.

## 7.4. Converting Access tables to MS Excel format

Access tables may also be converted into Excel format for dissemination. If this is required, care should also be taken to avoid similar rounding and truncation of values, as described in section 7.3 above. Export of individual tables to Excel may be performed using the procedures given in 7.3.2 and 7.3.3, substituting Excel as the export format. The online help available in Access on exporting tables to Excel is comprehensive, but there are some points to remember.

When long text strings contain a certain combination of numeric characters, Excel cells may be truncated at 255 characters without warning. Some cells in a given field may be truncated, but not others. Therefore, extra checks should be undertaken on all fields, and to avoid further truncation problems with long text strings, all fields that were originally memo fields in Access **must** be assigned as text fields in Design View before export. Excel text fields, unlike Access, may be over 255 characters in length. If this is the case, a copy of the Access database must be made before any changes are saved, as for the avoidance of decimal place rounding, described in section 7.3.1 above.

## 7.5. ‘Documenting’ the Access database and archiving the data structure

### 7.5.1. Documenting variables

The Access ‘documenter’ command should be run for every table. This may be done as follows:

Under the **Tools** menu select **Analyze** and **Documenter.**

Select one table at a time (unless there are many tens of tables, in which case it is acceptable to select all and thereby generate and archive one block of output that covers all the tables).

To control the level of output select **options.** The following elements should be chosen:

**Properties**

**Relationships**

**Names data types, size and types**

**Names and fields**

The files (one for each data table converted to tab-delimited text format) should exported in RTF format and be named ‘<table name>_variableinformation.rtf’ (e.g. 5662_variableinformation.rtf). They should be archived

under the following structure:

SN/mrdoc/allissue/<table name>_variableinformation.rtf

### 7.5.2. Documenting table relationships

Larger Access databases may contain a complex network of relationships between tables. A diagram of the table relationships may be created, both to aid users (especially those who request the tab-delimited files) and as a preservation tool. Therefore, for those Access databases that need it, the table relationships may be documented as follows:

1. From the Access database top menu, select Tools > Relationships: a diagram of the table relationships will be displayed onscreen.
2. From the top menu, select File > Print Relationships
3. Choose the Adobe PDF printer from your list of printer options, to generate an Adobe PDF file. The file may be named as follows: ####_access_table_relationships.pdf , and included under mrdoc alongside the other PDF study documentation. (Note that some image editing may need to be done if the table relationships are extensive, as they may print across more than one page within the file).

## 7.6. Conversion of tab-delimited files into Access

Although rarely done during processing, conversion from tab-delimited format into Access is also possible. However, it should be noted that Access examines the first few cases in any given field and uses these to define the field on importation. For example, the first few cases may all contain entries of less than 256 characters, even if some of the remaining cases contain longer entries. Because it does not assess beyond these cases, Access will import the entire field as a ‘text’ field instead of a memo field. This may lead to truncation of the longer records at 255 characters. However, where Access encounters any errors (including truncation), the import process will be aborted and import errors written to table in the current database. This can then be opened to discover the problem.