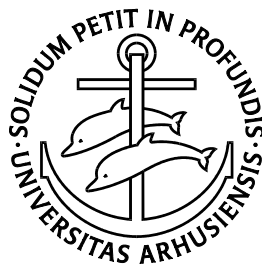# Take good care of your data

by Svend Juul

with contributions from Jens Lauritsen and Annette Jørgensen

Contents

Preface

Imagine:

- that you worked two years collecting data for your project and then discovered that some of the data collected were in a mess. You spend two months trying to reconstruct them, but you need to go back and retrieve 500 medical records and re-enter data from them to make sure you get the information right.

- that you finished a successful research project three years ago. Now you get a very promising research idea, which can be examined by re-analysing the data already collected. You get surprised when you learn that you hardly recall what the data mean and regret that you did not spend more time on documentation during the initial project; now you must spend two months to find out.

- that you finished a successful research project three years ago. Since then you moved to another place of work. Now you get a very promising research idea which can be examined by re-analysing the data already collected. However, you cannot find the data. You believed they were kept at your previous workplace, but nobody there recalls any arrangement, and the person who used to take care of your data has left, tempted by the much higher salaries in business.

- that you cooperate with researchers at three other hospitals on a multi-centre study. Data were coded and entered at each site. When you combine the four files you discover several inconsistencies: At one site your colleague had 'improved' the questionnaire by replacing two questions with three others. Another site used ICD-8 instead if ICD-10 for coding of diagnoses. You spend two months . . .

- that you were conducting a randomised controlled trial. There were 247 candidate patients; 57 did not want to participate, 2 moved out of the region, 3 died before randomisation, and 39 were excluded for various reasons. This should leave 146 patients for the trial, but when you start analysing the data you discover that you have data on 144 patients only. You have a hard time finding out what happened to the last two patients and their data.

- that you spent a lot of time developing, designing and pilot-testing a questionnaire. When the first questionnaires return you discover that you had sent an old, erroneous version of the questionnaire to the printing office.

- that you are a progressive person who finds paper antiquated. You conduct 200 lengthy telephone interviews, entering data directly into the computer during interview. However, due to hardware breakdown data from 25 interviews are lost. When you call the persons, most of them refuse to be interviewed again.

- that you published a research paper in a decent journal. After publication a correspondence correctly points to inconsistencies in the data presented, and the editor asks you to respond. Now you spend two months trying to determine what actually happened in the process between data collection and the results presented in the paper.

- that you published a research paper in a decent journal. However someone (because of jealousy) accuses you of scientific fraud. You know that you did not cheat, but Udvalget Vedrørende Videnskabelig Uredelighed (UVVU) asks you to document how you arrived at your published results. Now you spend two months trying to determine what actually happened in the process between data collection and the results presented in the paper, only to admit that you cannot reconstruct it, but you certainly did not cheat. UVVU believes you (they have seen this so many times) and concludes that it found no evidence of fraud, but criticizes that you could not produce the evidence needed to clean yourself completely.

- that your office burned. You had made backup on CDs, but they were stored next to your computer, and both the computer and your CDs melted down. Fortunately you moved your questionnaires to another building the day before the fire. You spend . . .

These examples are not far-fetched; I have experienced or seen all of the incidents, except the accusation of fraud and the fire – but it has happened, also in Denmark.

The purpose of this booklet is preventive: What can you do to avoid such problems?

Jens Lauritsen, Odense University Hospital, has given several valuable contributions, especially on archiving strategies (chapter 11, appendix 4), and on entering data (chapter 6) and provided a number of other useful suggestions.

Annette Jørgensen at the GCP unit, Aarhus University Hospital wrote appendix 3 on GCP principles and rules.

I welcome any comments and suggestions; my e-mail is: sj@soci.au.dk

Aarhus, November 2004

Svend Juul

1. Structure and notation in this booklet

There are two types of text in this booklet: Principles and examples. Principles apply regardless of the software you use. In the examples I use SPSS[1] and Stata[2] command files; if you use other software the examples hopefully are of help anyway. It is not the intent of this booklet to teach you SPSS or Stata.

SPSS examples are shown in single frames:

```
GET FILE = 'c:\dokumenter\proj1\alfa.sav'.                          SPSS
COMPUTE bmi=weight/(height**2).
SAVE OUTFILE = 'c:\dokumenter\proj1\alfa2.sav'.
```

SPSS words are shown with **UPPERCASE** characters while variable information (file and variable names) are shown with **lowercase** characters.

Stata examples are shown in double frames:

```
use "c:\dokumenter\proj1\alfa.dta", clear                          Stata
generate bmi=weight/(height^2)
save "c:\dokumenter\proj1\alfa2.dta" [ , replace]
```

All Stata text is **lowercase**. Stata words are shown with *italics* – just for clarity. Optional parts of commands are shown with light typeface in square brackets [ ].

The term "command files" is used throughout this note for files including a number of commands to be executed in sequence. In SPSS these files are called "syntax files"; they have the extension **.sps**. In Stata the name is "do-files", and the extension is **.do**.

---

1)   Juul S. *SPSS for Windows 8, 9 and 10*. Århus: Department of Epidemiology and Social Medicine, 2000. Download from www.biostat.au.dk/teaching/software)

2)   Juul S. *Introduction to Stata 8*. Århus: Department of Epidemiology and Social Medicine, 2004. Download from www.biostat.au.dk/teaching/software)

2. The audit trail

When keeping financial accounts e.g. for a company or for an association there are some obvious principles to follow:

It must be possible to go back from the balance sheet to the individual vouchers (bilag). This is done by giving each voucher a unique number. From each item in the balance sheet (regnskabsoversigten) you must be able to identify the component amounts and the vouchers. The term *audit trail* means exactly this: from the final results you must be able to follow the trail backwards to the primary sources of information.

If you are the bookkeeper you need this for yourself, otherwise you will have a hard time tracing errors. And it is an unconditional request for auditing (revision).

The same principles apply when handling information in research, as illustrated in the guidelines from Udvalget vedrørende Videnskabelig Uredelighed (Appendix 1): You should be able to trace each piece of information back to the original document:
- ID (case identifier) included in the original documents and in the data set.
- All corrections must be documented and explained
- All modifications to the data set must be documented by command files
- A command file must document each analysis.

This technique is needed during error checking and correction, it is needed for your own documentation of what you did, and it is needed if your project is exposed to external audit and monitoring.

The purposes are to:
- protect yourself against:
  - mistakes
  - errors
  - waste of time
  - loss of information.
- enable external audit (revision)

Documentation procedures must be included already during project planning, and they should be with you all the time.

3. Overview of the process.

Designing data collection

As an example I use the self-administered questionnaire, but the principles also apply, with modifications, to interviewer-administered questionnaires and to recording forms to be filled in without contact with the persons studied, e.g. when extracting information from medical records. In the following I use 'questionnaires' for all types of forms to record information.

The first consideration is to the respondent, both in terms of the phrasing of questions and response categories, and in layout. The second consideration relates to processing of the information recorded, but this consideration must *never* complicate the questionnaire to the respondent.

Processing of questionnaires before data entry

Questionnaires should be labelled with a unique number (an ID).

A *Codebook* describes the name, meaning, and coding of each variable. Textual information should rarely be entered as is, but rather be coded before data entry. With numerical information, don't make any calculations before data entry; the computer is much better at that. Record dates; do not calculate ages before data entry.

Data entry

Use a professional data entry program, I recommend EpiData.[3] To reduce errors double entry of part or all of the data is advisable.

Checking and correcting errors

Even despite double entry errors may occur, e.g. because of problems with interpreting ambiguous responses or inconsistent coding in the pre-processing of questionnaires. Also, a respondent might have given inconsistent responses. Chapter 7 and 8 concern methods for detecting errors and inconsistencies and advice on methods of corrections, including documentation of corrections.

Modifying your data

Don't modify your original data. But often you will want to derive a number of variables from the original input, e.g. a body mass index from height and weight, an age from two dates, or a quality-of-life score from responses to a number of items. Or you need to combine information from several sources by merging files. Chapter 9 concerns documentation of such modifications.

---

3)   Download – at no cost – the program from www.epidata.dk. Find a short description in *Introduction to Stata 8*; see note 2.

Archiving

Now it is time to do the first archiving of your data. And after finishing your project the data and documentation must be stored safely. Most health researchers do not have a stable affiliation with a research organization, and this complicates archiving. The opportunity to archive data at ERAS at Danish Data Archives is described in chapter 11 and appendix 4.

Analysis

Only a small part of your analyses will be included directly in your final publication, but many analyses will provide a background for your decisions on what results to publish. Chapter 10 gives advice on how to organise and keep the documentation for your analyses.

Safety considerations

There are two main considerations to be covered in chapter 11 and 12:
1. Prevent your data from being lost
2. Prevent your data from being abused by someone else

4. Designing data collection

I will use the self-administered questionnaire as an example, but the principles also apply to interviewer-administered questionnaires and case report forms to be filled in by the investigator or his/her assistants.

It is obvious that in a self-administered questionnaire the phrasing of questions and response categories, the sequence of questions, and the layout are of major importance, while you intuitively give less attention to phrasing and layout in a case report form to be filled in by yourself. There are, however, many examples of sloppy case report forms where even the investigator gets in doubt about how to fill it in consistently, and where an external monitor or auditor gets the (possibly justified) impression that data collection was somewhat haphazard.

Therefore, the advice below on designing self-administered questionnaires also applies to other data collection instruments.

Example 1. A short questionnaire for self-administration

1.	**Questionnaire number:** *123*
2.	**Your sex:**
	Male 1
	Female..................................2
3.	**Which year were you born?**
4.	**At which level did you leave school?**
	Before finishing 9th grade 1
	After 9th grade 2
	After 10th grade 3
	After high school 4
	Other ....5 (*Write below*)
5.	**How many children do you have?**
6.	**Do you have a vocational education?** (*Write below*)

____
Do not write here

____ ____
Do not write here

4.1. Layout

1st consideration: The respondent: The questionnaire should be simple and clear, and there should be no doubt how to fill it in.

2nd consideration: Processing of the information recorded. However, this consideration must *never* complicate the questionnaire to the respondent. The first consideration really is the first.

The layout of the questionnaire in example 1 is simple and it requires only standard word-processing tools. I used the following principles:

1. Each question with response categories is framed by a box, to help the respondent concentrate on one question at a time. Technically it is simple: I created a 6H1 table.
2. Questions are written with **bold** typeface.
3. Response categories are written with ordinary typeface.
4. Instructions *(write)* are written with *italic* typeface
5. For closed questions the response is given by circling a number (the code used when entering data).
6. For open questions responses are written in the box. Do not add lines to write on; they only complicate writing the response.
7. The amount of blank space should be appropriate, both for circling numbers and for writing text.

Example 2. Layout of closed questions

2a.   **Your sex:**
   1   Male
   2   Female

A right-handed person hides the response text, increasing the risk of misplacing the response.
    The response field should be placed to the *right* of the response text. to avoid this problem.

2b.   **Your sex:**

Male 1

Female................ 2

This is good. The dotted lines reduce the risk of misplacing the response.
    It is no more difficult to circle a number than to check a box, and the code is given at once, reducing the risk of errors when entering data.

2c.   **Your sex:**

Male .... □

Female................ □

This is OK for the respondent, but the risk of errors when entering data is higher than in 2b.

2d.   **Your sex:**

Male .... $\square_1$

Female................ $\square_2$

This is OK too and includes the code, reducing the risk of errors when entering data.
    I prefer style 2b myself, but I can't explain exactly why. Perhaps because it looks less pretentious.

4.2. On questions and response categories

There are three major types of variables, defined by what kind of properties they express:

Scale type	Examples	Characteristics
Interval scale	Temperature in °C Weight in grams Age in days Age in 10 year age groups	Measurements reflect continuous properties and are expressed in defined units (°C, grams, years). Interval width can be as narrow as measurements permit, or be grouped in wider categories.
Ordinal scale	always / often / seldom / never rare / medium / well done	There is a natural rank of categories, but no exact relationship to a continuous property.
Nominal scale	Colour Nationality Sex	No natural rank of categories (unless you are a chauvinist).

What questions can I ask?

In general any question which the respondent finds reasonable, with his/her understanding of the purpose of the study. In a study of fertility problems the respondents expect you to ask about the frequency of intercourse, but hardly about sexual pleasure. In a study of quality of life some respondents probably would be surprised if you avoided the topic of sexual pleasure. However, with sensitive questions you should in the introduction allow the respondent to skip questions he/she does not want to answer.

The important point is that the respondent must understand the purpose of the study. If the respondent feels that you ask questions that are not justified by the purpose, he/she will probably feel that you have a hidden agenda and stop responding to any of the following questions.

The sequence of questions

Related questions should be neighbours. In example 1 you see the number of children separate the questions on school education and vocational education, and this complicates things to the respondent.

Some recommend to start with 'neutral' questions and leave sensitive questions to the end of the interview. I am not so sure; I believe that the respondent should get the feeling early in a lengthy interview or questionnaire that you ask the important questions.

The phrasing of questions

Use the respondent's own language and avoid medical or bureaucratic jargon. Sentences should be short and easy to read – but not be felt childish. This sometimes conflicts with the demand for unambiguity. In example 1, question 6, we ask about vocational education (erhvervsuddannelse), and this might not be understood by every respondent:

Example 3. Phrasing of questions

3a.	Do you have a vocational education?
3b.	After you left school: Did you have any further education?

The distinction between school education and vocational education (erhvervsuddannelse) is a matter of complex definitions. 3b, although not without problems, may be a better choice.

Response categories

The response categories should be exhaustive and mutually exclusive. Exhaustive means that the list of response options covers any situation. Being exclusive means that you can give one and only one answer.

Example 4. Response categories

4a. **Did your child ever have an itchy skin rash affecting the front of the elbows, behind the knees, front of the ankles, around the neck, or around the eyes?**

Yes 1

No.................................2

This question is several questions in one. The respondent should ask 'Yes' if the child had a rash at at least one of the locations, but the risk is high that a respondent misses a location.

4b. **Did your child ever have an itchy skin rash affecting any of the following locations?**

Front of elbows 1

Behind the knees..........2

Front of ankles 3

etc.

The question is better; you ask about each location, it is easier to read for the respondent and you get more information. However, as the coding demonstrates, you assume that the information can be represented by one variable. It can't, since locations are not mutually exclusive.

4c. **Did your child ever have an itchy skin rash affecting any of the following locations?**
(You may circle more than one)

Front of elbows..............1

Behind the knees...........1

Front of ankles 1

etc.

Here is one variable for each location. But you cannot distinguish between a 'No' and a non-response, and you should use 4d.

4d. **Did your child ever have an itchy skin rash affecting any of the following locations?**
(Please one circle – Yes or No – for each location)

	Yes	No
Front of elbows	1	2
Behind the knees...........	1	2
Front of ankles	1	2
etc.		

This is best.

Collect and record 'raw', not processed information

Avoid to group continuous information already at data collection time. It is as easy for the respondent to state her age in years as to choose between a list of age groups. Even better: the date of birth allows you to calculate the exact age at any other date.

Example 5. Response options

5a. ~~How old are you?~~ 15-25 .... 1 25-35 .... 2 etc.	The response categories are not mutually exclusive: The respondent being 25 would not know how to respond.
5b. ~~How old are you?~~ 15-24 .... 1 25-34 .... 2 etc.	This is formally correct, but grouping data already at data collection prohibits modifying groupings at a later time. Also, you cannot calculate the mean age if desired.
5c. **How old are you?** _____ years old	This is easy for the respondent, and your analyses are not restricted by the *à priori* categories. You may e.g. calculate a mean age for a group.
5d. **When were you born?** _____ _____ month year	This is also easy for the respondent and enables you to calculate the age at any date, e.g. at a date of operation or the date of interview.

Open questions

In example 1 question 2 and 4 are *closed questions*: response categories are defined beforehand. Question 6 is an *open question*: the respondent can fill in any text. Also, in question 4 there is opportunity to enter text if the respondent does not feel to fit into one of the first four response categories. Formally, question 3 and 5 are open questions, but obviously the response options are restricted to years and numbers.

The text information in question 5 should hardly be entered as is in the computer, but you must translate the text into a finite number of categories. The coding should be described in the *codebook* (see later). Add a coding field next to the response field as in example 1, but design it so that it does not confuse the respondent.

In question 6 (vocational education) the number of possible responses is huge, and you need to classify responses into a finite number of categories. Before deciding on a final classification it might be wise to classify a sample of e.g. 100 responses to see if it works. The coding decisions must be included in the codebook; see example 6.

4.3. The codebook

The codebook is the link between the questionnaire and the data entered in the computer, and it should be made early in the process. Example 6 corresponds to the questionnaire in example 1, with an atopic dermatitis question added.

Example 6. A codebook

Variable	Source[§]	Meaning	Codes, valid range	Format[†]
`id`	Q 1	Questionnaire number	1-750	F3.0
`sex`	Q 2	Respondent's sex	1 Male 2 Female 9 No response	F1.0
`byear`	Q 3	Year of birth	1890-1990 –2[*] No response	F4.0
`schooled`	Q 4	Left school, level	1 Before finishing 9th 2 After 9th 3 After 10th 4 After high school 5 Other 9 No response	F1.0
`children`	Q 5	N of children	0-10 –2[*] No response	F2.0
`voced`	Q 6	Vocational education	1 None 2 Manual, < 3 years 3 Manual, 3 years + 4 Non-manual, < 3 years 5 Non-manual, 3-4 years 6 Non-manual, 5 years + 7 Cannot be classified 9 No response	F1.0

`rasha`
`rashb`
`rashc`
etc.
	Q 7a
	Q 7b
	Q 7c
		Rash: elbows
		Rash: knees
		Rash: ankles
		etc.
			1 Yes
			2 No
			9 No response
				F1.0

*)   Missing values for interval scale variables should not be included in calculations.

†)   This notation is frequently used when describing formats. F2.0 means a numerical variable with two digits and no decimals. F5.2 means five digits including decimal period and two decimals. A10 means a 10 character string (text) variable.

§)   You could use this field to identify the source of information, e.g. lab sheet or, as here, a question number.

Variable names

When analysing data you will refer to the variables by their names. In most programs variable names can have up to 8 characters; they must start with a letter. Even if your program allows longer variable names, keep them reasonably short; they may be abbreviated in output. Valid characters are `a-z` (but not æøå) and `0-9`. Avoid special characters.

If you use Stata for analysis, remember that `sex` and `Sex` are different variable names. Always use lowercase variable names in Stata.

With few variables use names that give mnemotechnic meaning, as in the codebook example 6. With many variables rather use names derived from question numbers: `q7a`, `q7b` etc.; a mnemotechnic system will break down for you.

In complex projects where you have data from several sources, use variable names that reflect the source. If you have three interviews with the same persons, use the prefix `a`, `b`, `c`:

1st interview: `a1 a2 a3a a3b a3c` etc.

2nd interview: `b1 b2` etc.

Meaning

A short text describing the meaning of the variable. In most programs this information can be included in the data set (variable labels) and will be displayed in the output.

Codes

Standard recommendation: Always use numerical codes. Not all analyses can handle string codes (e.g. 'M' for male sex), and numerical codes are faster to enter and easier to handle during analysis.

For interval scale variables (year of birth, number of children) just use the value as the code and state the possible range in the codebook. For ordinal and nominal scales (sex, education) state the meaning of each code. In most programs this information can be included in the data set (value labels) and will be displayed in the output.

Codes for no response. Missing values

Missing information may occur in three distinct situations:

1.   The question could not be asked. You hardly ask a male about his last pregnancy nor record complications to an operation that was not performed.
2.   The question was asked, but the respondent did not reply
3.   The respondent replied "Don't know".

In the codebook include your decision on how to record missing data, e.g. –1 for no question, –2 for no answer, and –3 for a "Don't know". Obviously the missing data codes should not be possible valid values.

In SPSS you can define certain values as missing, and Stata 8 has special missing codes, while Stata 7 does not have this facility (see appendix 7). Consistent handling of missing information is especially important for interval scale variables, e.g. to avoid that the code '999' for height is interpreted as almost 10 metres.

Missing values are important for interval scale variables, not so for categorical variables. Often a "Don't know" is as interesting as a "Yes", and should therefore not be considered missing and excluded from analysis.

5. Folders and file names. The log book.

Chapter 6-10 is a short example of the steps from data entry to a final data set for analysis. In the real world there are often many more steps, and you may easily confuse yourself. Most people are too optimistic about their ability to recall past decisions. Here is some advice on decisions that should be taken early.

Decide which folder or folders to work in

My advice is to organise folders by subject, not by file type. For a specific project or sub-project keep all your main text files, data files and command files in the same folder. An illustration is shown in appendix 6. Do not mix files from different (sub)projects in the same folder. Take a copy of final data and command files and put them in a "safe" (see example in appendix 5).

Never put your own files in a program folder. You may never find them again.

Decide a system for naming your data and command files

Command files that make modifications of your data are vital documentation. Both in SPSS and Stata it is easy to issue single commands, one at a time, but it is a lot safer to create a command file with all commands needed and execute them together – in the right sequence. You will make errors while developing a command file, but in the end you will succeed, and you should, of course, only keep the correct command files for documentation. *The vital command files should include everything needed – but nothing else – to reconstruct the data from the original input.*

These command files should have names clearly indicating what they do. I offer the suggestion to let such command files start with `gen.` followed by the name of the result file. Command files not creating new versions of the data should *not* have this prefix. Other systems may be used, but without a system you are at risk to get confused. Example 7 illustrates my suggestion for naming the data and command files used in chapter 6-10:

- All changes to the primary data (errors corrected, labels added, new variables generated, files merged) are documented by command files.
- The names of the command files tell you what they do.
- The final data set can be reproduced from the primary data by executing the modifying command files in the right sequence.

Keep a log book – and keep it updated

For an overview of your past actions keep a log and update it whenever you add modifications to your data. Example 7 shows how this could be done.

Example 7a. A log book. SPSS

Project:	Treatment of disease X		SPSS
Working folder:	c:\dokumenter\disx\data		
Safe folder:	c:\dokumenter\disx\data\safe		

Input data	Syntax file	Output data	Comments
visit1x.sav visit1a.sav			12.10.2001 Final comparison of two corrected data entry files. Agreement documented in: visit1.compare.txt
visit1a.sav	gen.visit1b.sps	visit1b.sav	13.10.2001 Add labels to visit1a.sav (example 8)
visit1b.sav	gen.visit1c.sps	visit1c.sav	15.10.2001 Identified errors corrected (example 11) (see visit1.correct.doc)
visit1c.sav visit2c.sav	gen.visit12.sps	visit12.sav	16.10.2001 Matching data from visit 1 and visit 2 (example 12)
visit12.sav	gen.visit12a.sps	visit12a.sav	16.10.2001 Generate new variables: hrqol, opagr (example 13)

Example 7b. A log book. Stata

Project:	Treatment of disease X		Stata
Working folder:	c:\dokumenter\disx\data		
Safe folder:	c:\dokumenter\disx\data\safe		

Input data	Do-file	Output data	Comments
visit1x.dta visit1a.dta			12.10.2001 Final comparison of two corrected data entry files. Agreement documented in: visit1.compare.txt
visit1a.dta	gen.visit1b.do	visit1b.dta	13.10.2001 Add labels to visit1a.dta (example 8)
visit1b.dta	gen.visit1c.do	visit1c.dta	15.10.2001 Identified errors corrected (example 11) (see visit1.correct.doc)
visit1c.dta visit2c.dta	gen.visit12.do	visit12.dta	16.10.2001 Matching data from visit 1 and visit 2 (example 12)
visit12.dta	gen.visit12a.do	visit12a.dta	16.10.2001 Generate new variables: hrqol, opagr (example 13)

Think ahead!

Keeping track of study participants is important; in the end you must be able to document the flow of participants. The figure is from the CONSORT statement on the requirements for the reporting of randomized trials.[4] The principle, however, is relevant to other studies as well.

Having problems with reconstruting the flow diagram at reporting time is not uncommon. Think ahead and make sure that you record the necessary informations about exclusions and participation *during* the study; it may be quite difficult to reconstruct afterwards.

Flow diagram of the progress through the phases of a randomized trial.

---
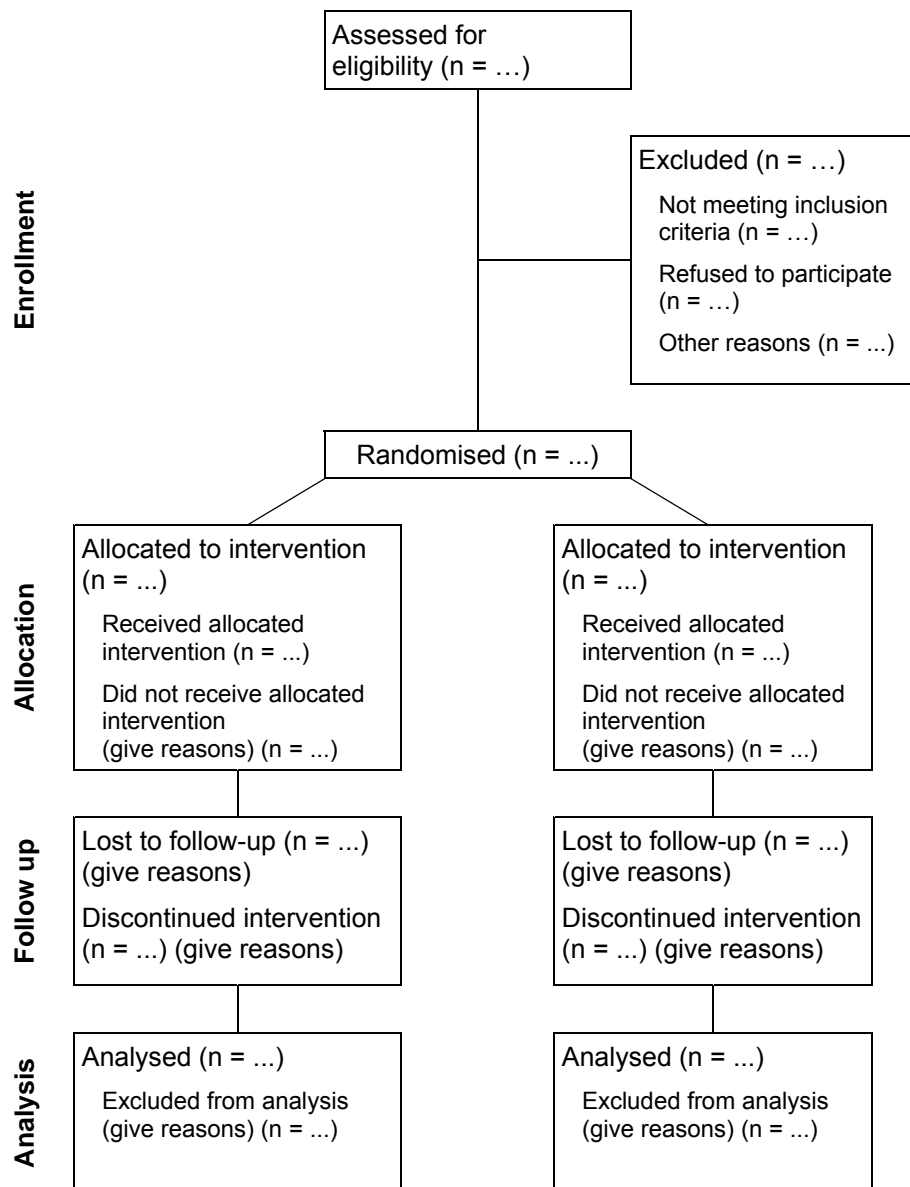
4)   The CONSORT statement: revised recommendations for improving the quality of reports of parallel group randomized trials. http://www.consort-statement.org. Also printed in e.g. Lancet 2001; 357: 1191-4.

6. Entering data

With very small data sets you may enter data in e.g. an Excel spreadsheet or in the SPSS or Stata data window, but with larger data sets this method is cumbersome and prone to give errors. Use a professional data entry program like EpiData. Epidata is free; download it from www.epidata.dk, and find a short description in *Introduction to Stata 8*.

Preparations

Before entering data (yourself or someone you hired) you need a complete codebook. All decisions on coding should be made – and documented in the codebook – before entering data; otherwise the risk of errors increases, and there is a risk that acute coding decisions during data entry are not recorded properly.

Examine the questionnaires for obvious inconsistencies before data entry. *Any decisions, e.g. coding of text information should be made before, not during data entry, and be written in the questionnaire*. Example 1 shows fields for post-coding of text information.

When defining the data set you need the codebook information on variable names and formats (the number of digits needed to represent the information).

Especially when you have data from several sources write down a plan for the process, including folder structure and file names. Also make a plan on how to backup your work to reduce the consequences of human, software or hardware errors.

Error prevention: Set up a data entry form resembling your questionnaire

EpiData enables you to create forms resembling your questionnaire pages; this reduces the risk of misplacing the information during data entry.

"Parallel shifting" during data entry is a frequent source of error: a correct value is entered, but in the wrong place, e.g. by entering the answer to "ankles" in the field for "knees" in example 4, question 4d. You might include a control variable demanding entry of an unusual value (e.g. –9) at the end of each page.

If you enter the wrong ID number it may be very difficult to locate and correct the error. I suggest that you as a safeguard re-enter the ID number as the last field in the form.

Error prevention: Define valid values before entering data

EpiData enables you to specify valid values and value labels for each variable, and to get a warning during data entry if you enter an outlier. While this method identifies illegal values, it does not catch erroneous entries within the legal range. You may also specify extended checks, e.g. that a date of hospital admission is after the date of birth. However, specifying the rules correctly may be time-consuming, and incorrect specifications will interfere with data entry. Also, inconsistent responses – which are not unusual – may interfere with data entry. I prefer to check for illegal values at a later time, as in example 9.

Entering data

For your own health and to reduce the risk of errors: Take a break now and then. Don't enter data for a full day, but do something else in between. If you employ other people to do the work, the same considerations apply.

Error prevention: Double data entry

EpiData enables you to enter data twice – this should be done by two different operators. Next compare the contents of the two files to get a list of discrepancies. Examine the original questionnaires and decide which entry was correct. I recommend to correct errors in *both* files and run a new comparison which should disclose no errors. Print this output for documentation that the data set now is 'clean'.

Error checking: Proof-reading

Proof-reading is a rather inefficient (and boring) way of error-checking, and for large data volumes proof-reading is virtually impossible. At least you need an assistant and a printout of the data entered; proof-reading data on the screen is inefficient and may damage your health.

Errors frequently occur by misplacing the right values during data entry. If you discover an error also proof-read the neighbour variables; they are high risk cases.

Which level of safeguards?

Use cost-benefit thinking. If you made a clinical trial with 50+50 patients the extra cost of double data entry is low compared to the total cost of collecting the information. And a single error might affect the conclusion. No doubt that you should do anything to avoid errors in your data.

If you mailed a questionnaire to 10,000 persons to estimate the prevalence of certain conditions the consequence of a single error is small, and you might decide to avoid double entry of every questionnaire. An example:

In a survey among 10,000 the data quality was checked by a second entering of data from a random sample of 1000 questionnaires. The comparison yielded one or more discrepancies in 5% of questionnaires; this was not satisfactory. Data entry was performed by five different operators, and the operator's ID was recorded in the data set. It turned out that one operator had an error rate of 25% while the other operators had error rates of 0-0.3%. Consequently all questionnaires originally entered by the 'bad' operator were re-entered by one of the 'good' operators.

Optical reading of questionnaires

The polls (Dansk Tipstjeneste) uses optical reading. The one-page 'questionnaire' is simple, and the volume is high. Obviously a good idea in terms of economy and safety.

Few research projects share these properties, and the cost considerations are quite different. The preparations for optical reading of a specific questionnaire are time-consuming, and with

multi-page questionnaires there is a lot of manual work while scanning. And errors do occur, so you need to proof-read anyway. Add to this:

The first layout-consideration is to the respondent, but the layout requirements for optical reading often counteract this consideration. I will not comment further on this issue.

Entering data into the computer during interview

Especially with telephone interviews it might be practical to enter responses directly into the computer during interview, thus avoiding the paper step. No doubt this is a good idea for simple mass surveys and opinion polls. However, for most research purposes I discourage it, because:

- Setting up a correct form for data entry might be quite time consuming
- If an unpredicted situation arises during an interview you might get stuck
- The risk of large scale errors and data loss is higher than with paper questionnaires
- You have no way of documenting the correctness of the data entered.

At least: You need a lot of experience to do that. I will not comment further on this issue.

7. First inspection of data. Error finding

With complex data sets examine and make corrections to each partial data set before merging files. In the following examples there are two partial data sets, visit1 and visit2.

7.1. Add labels to your data

Variable and value labels should be defined before data entry, but you might have received the raw data from another source. In that case you should define labels now, before examining the results, by command files. The program does not need the labels, but *You* need them for legibility of the output.

Example 8a. Adding variable and value labels in SPSS

```
* gen.visit1b.sps generates visit1b.sav - 12.10.2001        SPSS
* Adding labels to primary data.

GET FILE='c:\dokumenter\proj1\visit1a.sav'.

VARIABLE LABELS
  id 'Questionnaire number'
  /sex 'Sex of respondent'
  /byear 'Year of birth'
  etc...

VALUE LABELS
  sex  1 'male'  2 'female'
  /byear 9999 'no response'
  etc...

MISSING VALUES
  byear (9999)
  /children (99).

SORT CASES BY id.

SAVE OUTFILE='c:\dokumenter\proj1\visit1b.sav'.
```

This syntax file includes vital documentation and should be saved and kept in a safe place. It starts with reading the input data file and ends with saving the modified data set. I strongly recommend – for your own sake – to give the syntax file a name that tells what it does: **gen.visit1b.sps** generates the **visit1b.sav** data file.

The sort is made as preparation for later merging the data from visit1 and visit2.

Example 8b. Adding variable and value labels in Stata

```
// gen.visit1b.do generates visit1b.dta - 12.10.2001          Stata
// Adding labels to primary data

use "c:\dokumenter\proj1\visit1a.dta" , clear

label variable id "Questionnaire number"
label variable sex "Sex of respondent"
label variable byear "Year of birth"
etc...

label define sexlbl  1 "male"  2 "female"
label values sex sexlbl
etc...

recode byear (9999=.b)
label define byear  .b "no response"
label values byear byear

numlabel , add

sort id

save "c:\dokumenter\proj1\visit1b.dta" [ , replace]
```

This do-file includes vital documentation and should be saved and kept in a safe place.
It starts with reading the input data file and ends with saving the modified data set. I strongly recommend – for your own sake – to give the do-file a name that tells what it does: **gen.visit1b.do** generates the **visit1b.dta** data file.

9999 was entered for missing year of birth; this code is converted to Stata's user-defined missing code **.b** and furnished with a value label.

Stata does not display the codes and the value labels simultaneously in output, but the command *numabel* incorporates the numeric codes in the value labels.

The sort is made as preparation for later merging the data from visit1 and visit2.

7.2. Searching for errors

Next make printouts of:

1. a codebook from your data
2. an overview of your variables
3. simple frequency tables of appropriate variables

Example 9a. Initial printouts. SPSS

```
* Overview of the visit1b data set.                        SPSS
GET FILE='c:\dokumenter\proj1\visit1b.sav'.
DISPLAY DICTIONARY.
DESCRIPTIVES ALL.
FREQUENCIES ALL
  /FORMAT=LIMIT(20).
CROSSTABS sex BY pregnant.
```

DISPLAY DICTIONARY shows the codebook.

DESCRIPTIVES shows summary information for all numerical variables; look for minimum, maximum, and number of valid values.

FREQUENCIES shows tables for all variables. Tables with more than 20 different values are not displayed.

CROSSTABS can disclose inconsistent information (pregnant males).

Example 9b. Initial printouts. Stata

```
// Overview of the visit1b data set.                       Stata
use "c:\dokumenter\proj1\visit1b.dta", clear
describe
codebook
label list icd10
summarize
tab1 sex nation-educ
tab2 sex pregnant
```

describe and *codebook* give a short and a long codebook.

label list shows long value label lists (a max. of 9 are shown by *codebook*).

summarize shows summary information for all variables; look for minimum, maximum, and number of valid values.

tab1 shows tables for all variables mentioned (avoid tables for variables with many values).

tab2 can disclose inconsistent information (pregnant males).

Since this job did not modify your data, you need not save the command-file (syntax file; do-file) for documentation.

You *must* make a printout of the tables produced; don't use the screen. You can easily miss an error – and strain your eyes. Examine the following:

1. Compare the codebook created with your original codebook (chapter 4) and see if you made the label information correctly.
2. Inspect the overview table (descriptives/summarize), especially for illegal or improbable minimum and maximum values of variables. Also see if the number of valid observations for each variable is as expected.
3. Inspect the frequency tables (frequencies/tab1) for strange values and for values that should have labels, but haven't.
4. Examine tables that could disclose inconsistencies (pregnant males).

If you identified any suspicious values, list them with the id-number (also written at the questionnaire's front page) and control their correctness.

Example 10a. Listing of suspect values. SPSS

```
* Listing of suspect values.                                    SPSS

GET FILE='c:\dokumenter\proj1\visit1b.sav'.

TEMPORARY.
SELECT IF (sex>2).
LIST id byear TO diag.

TEMPORARY.
SELECT IF (sex=1 AND pregnant=1).
LIST id byear TO diag age1 TO educ.
```

Errors frequently occur by misplacing values during data entry. If you discover an error, also proof-read neighbour variables; they are high risk cases. **byear TO diag** represents **sex** and its neighbours; **age1 TO educ** represents **pregnant** and its neighbours.

Example 10b. Listing of suspect values. Stata

```
// Listing of suspect values.                                   Stata

use "c:\dokumenter\proj1\visit1b.dta", clear

list id byear-diag if sex>2

list id byear-diag age1-educ if sex==1 & pregnant==1
```

Errors frequently occur by misplacing the right values during data entry. If you discover an error, also proof-read the neighbour variables; they are high risk cases. **byear-diag** represents **sex** and its neighbours; **age1-educ** represents **pregnant** and its neighbours.

You *must* make a printout of the lists created in example 10; don't use the screen. Next go back to the original documents, identify the errors and write the corrections on the lists. Again: examine the neighbour variables carefully; they are at high risk to be erroneous as well.

How to make corrections is described in chapter 8.

Handling inconsistent information

A frequent situation is that data were entered correctly, but that the respondent filled in the questionnaire in an inconsistent way. A respondent might claim to be male & pregnant or to be 23 with her oldest son 19 years old.

One principle is that all inconsistent data should be recoded to missing, without further considerations. Another is that the investigator should examine other information available and judge which piece of information is most likely to be correct. My opinion is that the latter principle should be followed – with caution. But no matter which principle you followed, you made a decision on how to interpret data, and *such decisions must be documented in writing*.

The missing data problem

A non-response is a non-response, but in certain cases missing data have a high cost. In a regression analysis with 10 predictors, a case in which just one predictor is missing is omitted from the analysis. If many cases have one or more missing predictors this leads to a heavy loss of information – and the result may be bias'ed if non-response is related to one of the factors of interest.

There are formal remedies for this situation: missing value imputation, where the most likely response is estimated from the characteristics of respondents with non-missing information. I have no experience with the method, and I am sceptical. At least a precondition for significance testing (independence of observations) is violated.

But under certain circumstances you are in a position to give a sound judgement. A woman whose children are 1, 9 and 20 years old is likely to be close to 40 herself. What is most correct: to consider her age unknown or to use 40 as a pretty close, but imperfect judgement? If a respondent did not answer a question on a rare symptom, what is most correct: to treat it as no knowledge or to consider it a 'no'?

No matter what you decide, *it is a decision which must be documented in writing*.

8. Correction of errors. Documentation

If you discovered an error you might intuitively go to the data window and correct it there. This method is strongly discouraged. Firstly the risk of 'correcting' the wrong variable or case is high. Secondly the change is undocumented, and the audit trail is broken.

Therefore: Make corrections in a command file. Note the **gen.** prefix indicating that this command file generates a new version of the data set.

Example 11a. Correction of errors using syntax file. SPSS

```
* gen.visit1c.sps.                                        SPSS
* Corrections 14.10.2001. See project log page 27.

GET FILE='c:\dokumenter\proj1\visit1b.sav'.

IF (id=2473) sex=2.
...
SAVE OUTFILE='c:\dokumenter\proj1\visit1c.sav'.
```

Example 11b. Correction of errors using do-file. Stata

```
// gen.visit1c.do                                          Stata
// Corrections 14.10.2001. See project log page 27.

use "c:\dokumenter\proj1\visit1b.dta", clear

replace sex=2 if id==2473
...
save "c:\dokumenter\proj1\visit1c.dta" [ , replace]
```

In this way you have a full documentation of changes made to the data set. The audit trail was not broken.

On the other hand: If you discover errors when comparing files after double data entry you can make corrections directly in the data entered, provided you end this step with a comparison of the two files entered and corrected, demonstrating that there now are no disagreements.

The point is that you split the process in distinct and well-defined steps and that your documentation from one step to the next is consistent. But you should not bother with documenting that you made and corrected errors during the data entry step. No more than I should document which spelling errors I made and corrected while developing the text in front of you.

Archive now

Once you have a 'clean' and documented version of your primary data, save one copy in a safe place and do your work using another copy. Also archive a copy of the clean primary data at Danish Data Archives. See section 11.2 and appendix 4 on archiving.

9. Modifications of data. Documentation

9.1. Merging partial data sets

Don't merge data sets before you are sure that each of them is OK; error checking and corrections should take place before merging. Both data sets have previously been sorted according to the matching key (**id**).

Example 12a. Matching data from two sources. SPSS

```
* gen.visit12.sps.                                              SPSS
* Merge corrected VISIT1C and VISIT2C data sets.

MATCH FILES
  FILE='c:\dokumenter\proj1\visit1c.sav'  /IN=in1
  /FILE='c:\dokumenter\proj1\visit2c.sav' /IN=in2
  /BY id.
SAVE OUTFILE='c:\dokumenter\proj1\visit12.sav'.

CROSSTABS in1 BY in2.

TEMPORARY.
SELECT IF (in1=0 or in2=0).
LIST id in1 in2.
```

The **in1** and **in2** variables created are **1** if the file contributed with a case, **0** if not. A perfect match will show as an **in1 BY in2** crosstable with one cell: all cases should have **in1=1** and **in2=1**.

If there is an unexpected mismatch one reason might be an error in the **id** entered in one of the data sets. Such errors can be quite difficult to disentangle.

Example 12b. Merging data from two sources. Stata

```
// gen.visit12.do                                               Stata
// Merge corrected VISIT1C and VISIT2C data sets.

use "c:\dokumenter\proj1\visit1c.dta", clear
merge id using "c:\dokumenter\proj1\visit2c.dta"
save "c:\dokumenter\proj1\visit12.dta"

tab1 _merge
list id _merge if _merge<3
```

The **_merge** variable is: **1** if only data set 1 (visit1c) contributes; **2** if only data set 2 (visit2c) contributes; **3** if both data sets contribute. A perfect match will show as a **_merge** table where all observations have the value **3**.

If there is an unexpected mismatch, it might be due to an error in the **id** entered in one of the data sets. Use **duplicate report** and **duplicate list** to check *before* merging whether the values of the id variable are unique.

The command file and the output from the merge operation should be kept as documentation. The output should verify that there were no mismatches. If there were mismatches they should be corrected before a new merge attempt. Of course you only need to keep the documentation for the correct merge operation; keeping documentation of your now corrected errors is no more than noise.

9.2. Adding derived variables to your data

You should certainly not modify your original data, but you often want to derive new variables from the original information. You might e.g. combine the information from several questions on well-being in one new variable; you might calculate an age from two dates; and you might group the age in five intervals. This leads to the following rules:

1.   If you add modifications to your data the result should be saved as a file with a *new* name. The name should tell which version this is (include a, b, c or 1, 2, 3 in the filename).

2.   The command file making the modifications must be saved with an appropriate name. My firm recommendation is to use the **gen.** prefix to illustrate that this file generated modified data.

Example 13a. Generating derived variables. SPSS

```
* gen.visit12a.sps.                                              SPSS
* generates visit12a.sav with new variables.

GET FILE='c:\dokumenter\proj1\visit12.sav'.

* Calculate hrqol: quality of life score.
COMPUTE hrqol=SUM(q1 TO q10).
VARIABLE LABELS hrqol 'Quality of life score'.

* Calculate opage: age at operation.
COMPUTE opage=(opdate-bdate)/(365.25*24*60*60).
VARIABLE LABELS opage 'Age at operation'.

* Calculate opagr: age groups at operation.
RECODE opage (0 THRU 14.999=1)(14.999 THRU 34.999=2)
  (34.999 THRU 54.999=3)(54.999 THRU 120=4) INTO opagr.
VARIABLE LABELS opagr 'Age at operation, 4 groups'.
VALUE LABELS opagr  1 '-14'  2 '15-34'  3 '35-54'  4 '55+'.
FORMATS opagr (F1).

DOCUMENT Visit12a.sav created by gen.visit12a.sps, 02.01.2001.
SAVE OUTFILE='c:\dokumenter\proj1\visit12a.sav'.
```

gen.visit12a.sps is the syntax file generating **visit12a.sav**. It starts with the command reading the data (**GET FILE**) and ends with the command saving the modified data (**SAVE OUTFILE**).

Three new variables are created from the original information. The **RECODE... INTO** keeps the original **opage** unchanged while creating a new variable **opagr**.

Define variable and value labels immediately after creating a new variable. It will never be easier than now, and they will make it much easier for you to read the syntax file at a later time.

Comments are helpful explaining the purpose of complex operations.

Notes can be included in the data set (**DOCUMENT**). The notes can be displayed by **DISPLAY DOCUMENTS**.

WARNING!
At exit SPSS offers you to save your data if the working file was modified. The response should be **NO!** If you modified the data the documentation should be explicit in a syntax file. See appendix 7.

3. The command file must start with the command indicating the input data and end with the command indicating the output data. Commands not relevant to the modifications should be omitted from the command file.
4. Include comments to explain (to yourself or others) the purpose of complex operations.
5. You may include further documenting information in the data file (**DOCUMENT/***label, note*).

Example 13b. Generating derived variables. Stata

```
// gen.visit12a.do                                                    Stata
// generates visit12a.dta with new variables.

use "c:\dokumenter\proj1\visit12.dta", clear

// Calculate hrqol: quality of life score.
egen hrqol=rsum(q1-q10)
label variable hrqol "Quality of life score"

// Calculate opage: age at operation.
generate opage=(opdate-bdate)/365.25
label variable opage "Age at operation"

// Calculate opagr: age groups at operation.
recode opage (55/max=4 "55+")(35/55=3 "35-54")(15/35=2 "15-34") ///
  (min/15=1 "-14") , generate(opagr)
label variable opagr "Age at operation, 4 groups"
numlabel opagr , add

label data "Visit12a.dta created by gen.visit12a.do, 02.01.2001"

save "c:\dokumenter\proj1\visit12a.dta" [ , replace]
```

gen.visit12a.do is the do-file generating **visit12a.dta**. It starts with the command reading the data (**use**) and ends with the command saving the modified data (**save**).

Three new variables are created from the original information. To keep the original **opage** I used the **generate(opagr)** option in **recode**.

Define variable and value labels immediately after creating a new variable. It will never be easier than now, and they will make it much easier for you to read the do-file at a later time.

Comments are helpful explaining the purpose of complex operations.

label data attaches a label to the data set to be displayed each time it is opened.

Notes can be included in the data set (**note**). The notes included in a data set are displayed by the command **notes**.

WARNING! Don't overwrite good data with bad data. Only use **save**'s **replace** option if you really want to overwrite an existing data set. See appendix 7.

9.3. Checking correctness of modifications

Things may go wrong, and your modifications might not do what you intended. The modifying command file should demonstrate by your comments what you intended to do, and by the commands what you actually did. Look at the distributions of the new variables and list a sample of cases with both the source and target variables to check the correctness of calculations. *Print the lists before inspection!* Reading the screen is unpleasant, unhealthy, and unreliable.

Example 14a. Checking for correctness of new variables. SPSS

```
GET FILE='c:\dokumenter\proj1\visit12a.sav'.                    SPSS

DESCRIPTIVES hrqol opage opagr.
FREQUENCIES opagr.

SAMPLE 0.01.
LIST q1 TO q10 hrqol.
SORT CASES BY opage.
LIST bdate opdate opage opagr.
```

In the **DESCRIPTIVES** output look for the number of valid cases and for minimum and maximum values.

FREQUENCIES is useful for variables with few categories, like **opagr**.

LIST enables you, for a 1% sample, to compare **hrqol** with the original values of **q1** to **q10** and the derived age variables with the original dates.

Sorting by **opage** makes the comparisons between **opage** and **opagr** easier.

If you identify inconsistencies you must go back, modify **gen.visit12a.sps** and check again.

Example 14b. Checking for correctness of new variables. Stata

```
use "c:\dokumenter\proj1\visit12a.dta" , clear                   Stata

summarize hrqol opage opagr
tab1 opagr

sample 1
slist q1-q10 hrqol
sort opage
list bdate opdate opage opagr
```

In the *summarize* output look for the number of valid observations and for minimum and maximum values.

tab1 is useful for variables with few categories, like **opagr**.

list enables you, for a 1% sample, to compare **hrqol** with the original values of **q1** to **q10** and the derived age variables with the original dates. *slist* workes better than *list* with many variables; find it by *findit* **slist**.

Sorting by **opage** makes the comparisons between **opage** and **opagr** easier.

If you identify inconsistencies you must go back, modify **gen.visit12a.do** and check again.

10. Analysis

10.1. Make sure you use the right data set

Also for analyses I strongly recommend to create command files, *starting with the command reading the data*. There are two reasons for this recommendation:

1.	You might during a session have made non-permanent modifications to your data (e.g. a selection of cases, or a recoding of a variable) – and forgot that you did that.
2.	Documenting an analysis includes documenting which data you used.

With simple analyses you need not save the command files, but the commands issued should be part of the printed output.

Example 15a. Syntax file for analysis. SPSS

```
GET FILE='c:\dokumenter\proj1\visit12a.sav'.                    SPSS


REGRESSION
  /STATISTICS COEFF CI R ANOVA CHANGE
  /DEPENDENT sbp
  /METHOD=ENTER sex height weight.
```
Including the filename with full path documents which data were used for the analysis.

Example 15b. Do-file for analysis. Stata

```
use "c:\dokumenter\proj1\visit12a.dta" , clear               Stata
regress sbp sex height weight
```
Including the filename with full path documents which data were used for the analysis.

10.2. Late discovery of errors and inconsistencies

Despite your efforts to secure data quality you may during analysis discover errors and inconsistencies. This must lead to a comparison between the source documents (e.g. questionnaires) and the data set; the explanation may be a data entry error. This should be corrected.

It is an obvious advantage to have discovered and corrected errors before starting analysis. But if you have a good organization of your command files corrections are not that difficult: Go back and modify the correction command file (example 11). Run this and the subsequent command files (examples 12-13), and you have a corrected analysis file.

If you followed the recommendations in chapter 5 (log book, naming of command files), this is easy. If not, the procedure will be time consuming – and risky. I recommend letting a master command file call the individual command-files; with Stata the master do-file for the preceding activities could look like:

```
// master.do                                              Stata

// The cd command defines the path for all of the following files.
cd "c:\dokumenter\proj1"

do gen.visit1b.do   // Add labels; example 8
do gen.visit1c.do   // Corrections; example 11
do gen.visit12.do   // Merge; example 12
do gen.visit12a.do  // Derived variables; example 13
```

Now, if you made a late discovery of an error, modify the correction do-file (**gen.visit1c.do**) and run this and the following do-files; the master do-file makes this easy and safe.

With SPSS you can make a similar master command file, using **INCLUDE** to call the individual syntax files.

11. Backing up. Archiving.

By Jens M. Lauritsen and Svend Juul

The distinction between backing up and archiving may seem subtle, but the purposes are different. *Backing up* is an everyday activity, and the purpose is to be able to restore your data and documents in case of destruction or loss of data. Destruction may be physical, but most cases of data loss are due to human errors, e.g. unintentional deletion or overwriting of files.

Archiving takes place once or a few times during the life of a project. The purpose is to preserve your data and documents for a more distant future, maybe even to allow other researchers access to the information.

11.1. Backing up

Try to answer the following two questions:

1. **How much time are you willing to spend on making your computer work again after a serious breakdown?**	
1-2 hours	1
About one day	2
Less than one week	3
1-4 weeks	4
One month (or more)	5

While you are reading this text your computer has a major breakdown and it can not be used any more. All data are lost. You buy a new computer and continue with question 2:

2. **How long time will it take you to install programs and reload data such that you can continue where you left off?**	
1-2 hours	1
About one day	2
Less than one week	3
1-4 weeks	4
One month (or more)	5
I can't – my data are lost	6

If there is any discrepancy between your answer in question one and two, you have a problem.

Computer professionals tend to be trustworthy persons, but their occupational mobility is high. Ask your supervisor or department if there is a local policy on issues of data protection, ownership to data and responsibility for proper management, backing up and archiving of data. Unfortunately in many departments these issues are not clearly stated as they should be – in writing. If the department has no written instructions, take the full responsibility yourself. If it has: evaluate the instructions and decide if you need any further safeguards.

Strategy considerations

The way you handle your own data have strong implications for your possibilities to back up, archive, and restore your data. We will especially point to the following issues:

- A logical and transparent folder structure is good for your everyday work – and it facilitates backing up and recovery of your data. Appendix 5 suggests a folder structure where your own data and documents are kept separately from program folders.
- File names are important. Chapter 5 gives suggestions for naming strategies.
- Back up not only data sets, but also command files modifying your data (chapter 8 and 9). And of course also written documents: protocol, log book (chapter 5) and other documenting information.

By far the safest strategy is to take regular back up of all of your files, using a semi-automated procedure.

Take a *full backup* with regular intervals, e.g. every 6 months. This backup includes all of your data and documents – but not program files.

Take an *incremental backup* regularly, e.g. at the end of each working day. Include all new files and files modified since the last backup.

File names are important. We recommend that you give your zip-files names that enable you to sort them by age: `2001.12.09.zip` for a zip-file created 9th December 2001. If you need to restore your data, start with the oldest to avoid overwriting new files with older versions.

Make sure that someone else than yourself knows where to find your data, documentation and encryption passwords. You might be incapacitated, but your valuable work should not be lost.

Software considerations

There are several possibilities. One of them is the backup program accompanying Windows; it is rather clumsy. WinZip is useful for backing up – and for other purposes as well. Appendix 6 is a short description of WinZip.

Media considerations

It is important that your backup media are stored in a different building than your computer; in case of fire you may otherwise loose all data.

Diskettes are cheap, but not very stable. If you rely on diskettes you should make two copies to be stored separately.

CD-ROMs require that you have the equipment to write them. They are probably quite stable, but the experience on long term stability is limited.

Magnetic tape is probably history by now. Stability varies.

Sending to a remote computer is our preferred method. If you have access to a server (located in a different building) you might use it for storing your backup files. Or you might have a mutual agreement with a friend living elsewhere that you exchange backup files by e-mail. Security and privacy considerations might lead you to encrypt the backup files (appendix 6).

Test your backup copy

Make sure that you actually can restore the data from your backup. If you use diskettes, CDs or tapes, you must test this on a *different* computer with different equipment; tape, diskette, and CD drives may differ in calibration.

11.2 Archiving

What to archive?

While backing up is an everyday activity, archiving takes place once or a few times during the life of a project. If you worked with your data using consistent documentation procedures, archiving is easy. If not, it is difficult. The final archiving of a project should include:

a.   Study protocol
b.   Applications to and permissions from Ethical committees etc.
c.   Data collection instruments (questionnaires, case report forms etc.)
d.   Coding instructions and other technical descriptions
e.   The log book (see chapter 5) and other written documentation on the processing of data. This documentation serves to facilitate the understanding of (f) and (g).
f.   At least the first and the final version of your data
g.   All command files modifying data. The command files should enable to reconstruct the final version from the first version of your data
h.   Publications

Where?

If you are affiliated with a research department it should take a responsibility for archiving of data in general – ask for written guidelines. Danish Data Archives (see appendix 4) is useful for archiving electronic data with documentation, both as part as the department's policy, and for individual researchers. If your affiliation is temporary you strongly need Danish Data Archives. In practice, regardless of your affiliation, assure yourself that archiving actually has taken place.

At Danish Data Archives you may, without any costs to yourself, archive data at any stage of a project. There are several levels of access to be decided by you yourself, e.g.

•   Only you yourself have access
•   Others have access only with your agreement
•   Anybody has access

The data protection authorities (Datatilsynet, see appendix 2) have decided that archiving of information with personal identification at Danish Data Archives is permitted even if it was required that the personal identification should be deleted after a certain time span. This gives opportunity for follow-up studies not previously possible.

When?

There is no point in archiving data before they are in a documented state. Remember that archiving means storing for future use – in contrast to backup which ensures viability of data on a short term. On the other hand, waiting until completion and publication of the study can give you trouble. At that time it can be difficult to collect the documents describing the process and to remember on which computers the different versions of the data and command files are kept.

The appropriate time of archiving depends on the project, but in general a three step strategy is advisable:

1. Archive the key file (see example 16) with a copy of the study protocol, questionnaires etc. at onset of data collection. At this time contact Danish Data Archives to have your study registered.
2. Archive the raw data set when you have finished entering data, corrected errors and documented the data (see chapters 6-8). Include all documents written since the first archiving, and archive copies of both data and command files.
3. After analysis archive updated data files and the corresponding command files (see chapter 9). Include copies of publications and other major documents written since the last archiving.

12. Protection against abuse

12.1. Motives and opportunities

Obviously sensitive information about people must be kept confidentially without giving other persons the opportunity to see the information.

The *motives* to intrude could be curiosity (My neighbour's mystery disease), money (Should we insure Mr. NN?), creation of newspaper headlines ("Researchers fool around with confidential data"), and other indecent motives (If it becomes known that XX handles his data carelessly he will get in trouble – and I would love that).

It is the obligation of the researcher to give no other person the *opportunity* to see confidential information. Below are shown the two key methods to prevent unauthorized access to confidential electronic information.

Securing the electronic information is not enough, also information on paper must be protected against unauthorized access. Information on paper is probably more vulnerable to accidental access than the information in a computer file.

12.2. Separate external identification from information

As internal identifier linking the questionnaires with the data set you usually give each questionnaire a unique number also to be recorded in the data set. The internal identifier has no meaning outside your project.

The data protection authority (Datatilsynet; see Appendix 2) requires that you remove external identification (name, CPR number) from your analysis file as soon as possible or that you keep the data encrypted when you are not analysing. Usually you don't need the external identifier, except when linking your data with information from other sources.

In example 16 I show a technique to remove an external identifier (cpr). I recommend that you archive the key file at Danish Data Archives (see section 11.2 and appendix 4).

12.3. Encryption

There are several possibilities for encryption; I use the facility in WinZip, at the same time saving disk space. It is easy and fast – but of course you must neither forget your encryption password nor enable others to read it.

See appendix 6 on encryption using WinZip.

Example 16a. Move external identification to key file. SPSS

```
* gen.safe.keyfile.sps
* removing external identification

GET FILE='c:\dokumenter\proj1\unsafe.sav'.

SORT CASES BY id.

SAVE OUTFILE='a:\keyfile.sav'
  /KEEP=id cpr.

SAVE OUTFILE='c:\dokumenter\proj1\safe.sav'
  /DROP=cpr.
```

The key file (**keyfile.sav**) linking the internal identifier (**id**) with the external identifier (**cpr**) should be stored separately, i.e. not at the same computer as the information. Here I used a diskette, but beware: diskettes are not very stable, so make an extra backup copy.

I sorted by **id** before the separation, to facilitate later matching, if needed.

If you later need to include **cpr** – e.g. for matching with external data:

```
MATCH FILES FILE='a:\keyfile.sav'
  /FILE='c:\dokumenter\proj1\safe.sav'
  /BY id.
```

Example 16b. Move external identification to key file. Stata

```
// gen.safe.keyfile.do
// removing external identification

use "c:\dokumenter\proj1\unsafe.dta", clear
sort id
keep id cpr
save "a:\keyfile.dta" [ , replace]

use "c:\dokumenter\proj1\unsafe.dta" , clear
sort id
drop cpr
save "c:\dokumenter\proj1\safe.dta" [ , replace]
```

The key file (**keyfile.dta**) linking the internal identifier (**id**) with the external identifier (**cpr**) should be stored separately, i.e. not at the same computer as the information. Here I used a diskette, but beware: diskettes are not very stable, so make an extra backup copy.

I sorted by **id** before the separation, to facilitate later merging, if needed.

If you later need to include **cpr** – e.g. for matching with external data:

```
use "c:\dokumenter\proj1\safe.dta", clear
merge id using "a:\keyfile.dta"
```

Appendix 1

Vejledning for udformning af undersøgelsesplaner, datadokumentation og opbevaring af data inden for klinisk og klinisk-epidemiologisk forskning

Vejledningen er en del af UVVU's Vejledning i God Videnskabelig Praksis.
http://www.forsk.dk/uvvu/publ/vejled_vid_praksis

Det er væsentligt for ethvert projekts gennemførelse, at forsker, vejleder og evt. andre medvirkende har gensidig informationsforpligtelse vedrørende de originale forsøgsresultater, deres bearbejdning og fortolkning. En hensigtsmæssig udformning og opbevaring af undersøgelsesplaner m.m. er derfor af afgørende betydning.

- Undersøgelsesplaner, spørgeskemaer, personbilag (case report forms) og andre bilag skal være overskuelige og utvetydige for alle implicerede parter, ikke alene for dem der planlægger og udfører forskningen, men også for dem, der eventuelt senere skal vurdere resultaterne. Brug derfor skrivemaskine eller PC og en standardiseret opstilling af undersøgelsens titel, formål, materialer, procedurer, rådata og beregninger, som disposition for hver undersøgelsesplan.

- Undersøgelsesplaner skal udformes i så god tid, at der bliver tid til at teste personbilagenes praktiske anvendelighed, indhente videnskabsetisk accept, m.m.

- Undersøgelsesplaner og -bilag skal være så detaljerede, at det bliver muligt at vurdere, hvorvidt en stikprøve er repræsentativ i forhold til den population, den stammer fra. Der skal derfor være præcise inklusionskriterier for undersøgte personer og beskrivelse af indgang til undersøgelsen, fx om den er planlagt at være konsekutiv, ligesom betingelser for udgang af undersøgelsen før tiden (drop outs) skal omtales.

- Personrelaterede data fra klinisk-videnskabelige undersøgelser skal kunne identificeres sikkert, og de skal dateres og signeres. Rubrikker, der ikke udfyldes, udstreges.

- Personbilagenes data skal i så stor udstrækning som muligt omfatte originale rådata i letlæselig form, fx som papirdokumentation af elektroniske data, opklæbede strimler fra printere, kurveskrivere, automatiske vægte, tællere, autoanalysatorer og regnemaskiner. Kopi af væsentlige elektroniske originaldata bør snarest muligt efter undersøgelsens afslutning arkiveres i en fælles database, som skal bero i institutionen. De deltagende forskere kan disponere over egne kopier.

- Bilag til undersøgelsesrapporten skal indeholde udførte beregninger, herunder observationsberegninger, korrektioner og disses forudsætninger, som nødvendig dokumentation og for at lette forståelsen af de opnåede resultater.

- Der skal foreligge oplysninger om kvalitetskontrol af væsentlige data, og det skal anføres hvilke statistiske metoder og edb-programmer, der er anvendt.

- Det skal være muligt ud fra undersøgelsesbilag og spørgeskemaer at identificere de originale observationer, som indgår i publicerede tabeller og figurer.

- Indhentede tilladelser fra det videnskabsetiske komitésystem, Registertilsynet, Strålehygiejnisk Laboratorium, Sundhedsstyrelsen og eventuelt andre berørte instanser, samt samtykkeerklæringer fra personer, der indgår i undersøgelsen, skal gemmes i henhold til de forordninger, der fremgår af gældende lovgivning, og under hensyn til eventuelle opfølgende undersøgelser. Det samme gælder interview- eller spørgeskemaer og personbilag. Alt sådant materiale skal opbevares separat, ikke i journaler, og gemmes i mindst 10 år. Dansk Data Arkiv er velegnet til opbevaring af data, specielt fra samfundsmedicinsk forskning. Anvendte koder til anonymisering skal ligeledes gemmes, i den udstrækning lovgivningen tillader det.

Det påhviler forskningsinstitutionens ledelse og projektvejledere at gøre ovenstående retningslinier bekendt for alle involverede parter, enten i den foreliggende form eller som en af institutionen udarbejdet vejledning, baseret på lignende principper

Udvalgene vedr. Videnskabelig Uredelighed
Forskningsstyrelsen
Randersgade 60
2100 København Ø
Tlf. 3544 6318
E-mail: skj@forsk.dk
http://www.forsk.dk/uvvu/

Appendix 2

Datatilsynets vilkår for private forsknings- og statistikprojekter

Når Datatilsynet giver tilladelse til behandling af følsomme personoplysninger i et privat projekt til videnskabelige eller statistiske formål, fastsættes samtidigt en række vilkår for projektet. Vilkårene skal først og fremmest sikre, at den dataansvarlige (projektlederen) overholder reglerne i persondataloven og at databehandlingen lever op til lovens krav om datasikkerhed.

Af lovens § 41, stk. 3, fremgår det, at den dataansvarlige skal træffe de fornødne tekniske og organisatoriske sikkerhedsforanstaltninger mod, at oplysningerne hændeligt eller ulovligt tilintetgøres, fortabes eller forringes, samt mod at de kommer til uvedkommendes kendskab, misbruges eller i øvrigt behandles i strid med loven.

På denne baggrund har Datatilsynet udforme et sæt standardvilkår, som alle projekter skal efterleve. Hermed sikres også, at alle deltagere (de registrerede) sikres den samme beskyttelse.

Det er disse vilkår Datatilsynet fastsætter, når man giver tilladelse til et forskningsprojekt i medfør af lovens § 50. Datatilsynets tilladelse til projektet er betinget af, at man vil følge Datatilsynets vilkår. Dette angives ved at angive (sætte kryds) i anmeldelsesblankettens pkt. 7 vedrørende sikkerhed, at der vil blive truffet sikkerhedsforanstaltninger i overensstemmelse med Datatilsynets standardvilkår.

Standardvilkårene er gengivet nedenfor. Vilkår markeret specialvilkår fastsættes efter behov, øvrige vilkår fastsættes for alle projekter. I særlige tilfælde vil Datatilsynet herudover også kunne fastsætte individuelle vilkår for et projekt. *(Specialvilkår som ikke er medtaget her: Biobanker og biologisk materiale. Behandling ved ekstern databehandler. Overførsel af oplysninger til tredjelande).*

Datatilsynets standardvilkår for private forsknings- og statistikprojekter

Generelle vilkår

- <Den dataansvarliges navn> er ansvarlig for overholdelsen af de fastsatte vilkår.

- Oplysningerne må kun anvendes til brug for projektets gennemførelse.

- Behandling af personoplysninger må kun foretages af den dataansvarlige eller på foranledning af den dataansvarlige og på dennes ansvar.

- Enhver, der foretager behandling af projektets oplysninger, skal være bekendt med de fastsatte vilkår.

- De fastsatte vilkår skal tillige iagttages ved behandling, der foretages af databehandler.

- Lokaler, der benyttes til opbevaring og behandling af projektets oplysninger, skal være indrettet med henblik på at forhindre uvedkommende adgang.

- Behandling af oplysninger skal tilrettelægges således, at der foretages den fornødne kontrol for at sikre, at der ikke behandles urigtige eller vildledende oplysninger. Urigtige eller vildledende oplysninger eller oplysninger, som er behandlet i strid med loven eller disse vilkår, skal berigtiges eller slettes.

- Oplysninger må ikke opbevares på en måde, der giver mulighed for at identificere de registrerede i et længere tidsrum end det, der er nødvendigt af hensyn til projektets gennemførelse.

- En eventuel offentliggørelse af undersøgelsens resultater må ikke ske på en sådan måde, at det er muligt at identificere enkeltpersoner.

- Eventuelle vilkår, der fastsættes efter anden lovgivning, forudsættes overholdt.

Elektroniske oplysninger

- Identifikationsoplysninger skal krypteres eller erstattes af et kodenummer el. lign. Alternativt kan alle oplysninger lagres krypteret. Krypteringsnøgle, kodenøgle m.v. skal opbevares forsvarligt og adskilt fra personoplysningerne.

- Adgangen til projektdata må kun finde sted ved benyttelse af et fortroligt password. Password skal udskiftes mindst én gang om året, og når forholdene tilsiger det.

- Ved overførsel af personhenførbare oplysninger via Internet eller andet eksternt netværk skal der træffes de fornødne sikkerhedsforanstaltninger mod, at oplysningerne kommer til uvedkommendes kendskab. Oplysningerne skal som minimum være forsvarligt krypteret under hele transmissionen. Ved anvendelse af interne net skal det sikres, at uvedkommende ikke kan få adgang til oplysningerne.

- Udtagelige lagringsmedier, sikkerhedskopier af data m.v. skal opbevares forsvarligt aflåst og således, at uvedkommende ikke kan få adgang til oplysningerne.

Manuelle oplysninger

- Manuelt projektmateriale, udskrifter, fejl- og kontrollister, m.v., der direkte eller indirekte kan henføres til bestemte personer, skal opbevares forsvarligt aflåst og på en sådan måde, at uvedkommende ikke kan gøre sig bekendt med indholdet.

Oplysningspligt over for den registrerede

- Hvis der skal indsamles oplysninger hos den registrerede (ved interview, spørgeskema, klinisk eller paraklinisk undersøgelse, behandling, observation m.v.) skal der uddeles/fremsendes nærmere information om projektet. Den registrerede skal heri oplyses om den dataansvarliges navn, formålet med projektet, at det er frivilligt at deltage, og at et samtykke til deltagelse til enhver tid kan trækkes tilbage. Hvis oplysningerne skal videregives til brug i anden videnskabelig eller statistisk sammenhæng, skal der også oplyses om formålet med videregivelsen samt modtagerens identitet.

- Den registrerede bør endvidere oplyses om, at projektet er anmeldt til Datatilsynet efter lov om behandling af personoplysninger, samt at Datatilsynet har fastsat nærmere vilkår for projektet til beskyttelse af den registreredes privatliv.

41

Indsigtsret

- Den registrerede har ikke krav på indsigt i de oplysninger, der behandles om den pågældende.

Videregivelse

- Videregivelse af personhenførbare oplysninger til tredjepart må kun ske til brug i andet statistisk eller videnskabeligt øjemed.

- Videregivelse må kun ske efter forudgående tilladelse fra Datatilsynet. Datatilsynet kan stille nærmere vilkår for videregivelsen samt for modtagerens behandling af oplysningerne.

- (Specialvilkår) Oplysninger kan herudover videregives, hvis det fremgår af anden lovgivning, at oplysningerne skal videregives.

Ændringer i projektet

- Væsentlige ændringer i projektet skal anmeldes til Datatilsynet (som ændring af eksisterende anmeldelse). Ændringer af mindre væsentlig betydning kan meddeles Datatilsynet.

- Ændring af tidspunktet for projektets afslutning skal altid anmeldes.

Ved projektets afslutning

- Senest ved projektets afslutning skal oplysningerne slettes, anonymiseres eller tilintetgøres, således at det efterfølgende ikke er muligt at identificere enkeltpersoner, der indgår i undersøgelsen.

- Alternativt kan oplysningerne overføres til videre opbevaring i Statens Arkiver (herunder Dansk Dataarkiv) efter arkivlovens regler.

- Den dataansvarlige skal meddele Datatilsynet, når projektet er afsluttet, og oplysningerne slettet, anonymiseret, tilintetgjort eller overført til Statens Arkiver.

- Sletning af oplysninger fra elektroniske medier skal ske på en sådan måde, at oplysningerne ikke kan genetableres.

- (Specialvilkår for kliniske forsøg med lægemidler.) Oplysninger vedrørende den enkelte person kan dog, hvis en myndighed stiller krav herom eller GCP-reglerne skal overholdes, opbevares i indtil 15 år efter projektets ophør. En liste over deltagende personer må tillige opbevares i samme tidsrum.

Principles and rules of Good Clinical Practice (GCP)

By Annette Jørgensen, the GCP-unit at Aarhus University Hospital

GCP is an international quality standard for designing, conducting, recording and reporting trials that involve the participation of human subjects. Compliance with this standard provides public assurance that:

- the trial is ethically and scientifically sound
- the rights, safety and well-being of trial subject are being protected
- the clinical trial data are credible

The principles of GCP are described in the ICH-GCP guideline (http://www.eudra.org/humandocs/ PDFs/ICH/013595en.pdf). This document also describes the specific terminology used.

GCP - when?

The principles of GCP must be followed when generating clinical trial data that are intended to be submitted to the regulatory authorities. Until now the principles have therefore primarily been followed by the pharmaceutical industry. Where drug research is concerned compliance with GCP principles is expected to be laid down by law in a very few years. This means that all trials, including trials with approved drugs, have to be performed according to the GCP principles.

In this field the County Council of Aarhus has pioneered as it in 2000 decided that the GCP principles should be followed for all clinical drug trials at the hospitals of the county.

Although developed with the object of quality assurance of clinical drug trials, the principles of GCP also apply to other clinical investigations that may have an impact on the safety and well-being of human subjects.

GCP – how?

The GCP-principles as described in the GCP-guideline are intended as a guide. Compliance with the principles therefore implies a quality assurance system with written standard operating procedures of how to handle the specific trial. An important procedure is monitoring.

Monitoring

The act of overseeing the progress of a clinical trial to verify that:

- the subjects are protected
- the trail data are accurate, complete, and verifiable from source documents
- the conduct of the trial is in compliance with the protocol, GCP, and applicable regulatory requirements

When a trial is sponsored by a pharmaceutical company implementation of a quality assurance system is the responsibility of the sponsor. But when a trial is initiated by an independent investigator, for instance a physician at a hospital, his/her obligations include both those of a sponsor and those of an investigator. An independent investigator who wants to claim compliance with GCP therefore at least has to ensure that the trial is monitored.

The GCP-unit at Aarhus University Hospital

The GCP-unit at Aarhus University Hospital was established to perform quality assurance procedures on trials. The GCP-unit addresses primarily independent investigators making it possible for them to perform a trial according to the GCP principles independently of the pharmaceutical industry. To the investigator this might lead to some extra work, but also to the benefit of being able to document that the trial was independently monitored.

In brief the monitoring procedure of the GCP-unit is aimed at ensuring that the *audit trail* (see chapter 2 and elsewhere in this booklet) is kept intact, and that the final results are consistent with the data collected. On the other hand the GCP unit has no responsibility for the scientific interpretation of the results. The monitoring includes the following:

- Guidance in:
    - o design of the protocol and the case report form (CRF)
    - o notifications to the regulatory authorities
    - o Initiation visit to check and document:
    - o approvals and agreements in writing
    - o that facilities are adequate
    - o that a trial file is established
- Monitoring visits to check and document:
    - o compliance with the protocol
    - o that written informed consent was obtained form each subject
    - o that data are accurate, complete, and verifiable from source documents
    - o that all deviations are documented
- Final monitoring visit to check an document:
    - o that the trial file is complete
    - o consistency between the source documents and the database, possibly on a sample of data

The GCP-unit is funded by the University and by the County of Aarhus. Monitoring of a trial initiated by a Ph.D-student or a small trial within the University Hospital is free of charge. Costs of the GCP-unit should be paid when it concerns monitoring of multicenter trials and other large trials.

GCP-enheden ved Århus Universitetshospital
Bygning 1A, 1. sal
Århus Kommunehospital
Nørrebrogade 44
8000 Århus C
Telefon: 8949 2196
http://www.auh.dk/gcp/dk

Appendix 4

DDA/Sundhed: Arkivering af sundhedvidenskabelige data.

Arkivering og formidling af sundhedsvidenskabelige forskningsdata har tidligere været overladt til enkeltpersoner, sygehusafdelinger, institutioner osv. ERAS (Enheden for Registrering og Arkivering af Sundhedsvidenskabelige data ved Dansk Data Arkiv) varetog gennem nogle år opgaven som en forsøgsordning; der er nu etableret en særlig afdeling (DDA/Sundhed) ved Dansk Data Arkiv under Statens Arkiver, med henblik på at samle alle sundhedsvidenskabelige forskningsdata i ét landsdækkende arkiv.

Formål

DDA/Sundhed's opgave er at øge registrering og arkivering af sundhedsvidenskabelige data samt at skabe en professionel arkivfunktion, der givet primærundersøger tillader, også formidler adgang til eksisterende sundhedsvidenskabelige forskningsdata for andre forskere.

Tilbud til forskningsmiljøet

- Optimal opbevaring af undersøgelser (data og dokumentation)
- Grundig bearbejdning af data og dokumentation i forbindelse med arkivering
- Standardiseret arkiveringsformat uafhængig af skiftende programmer og styresystemer
- Mulighed for opbevaring og genudlevering af personfølsomme undersøgelsesdata ved senere follow-up
- Opbevaring og backup af forskningsdata med adgangsrestriktioner for tredjepart hvis ønsket
- Videreformidling af undersøgelsesdata til andre forskere og studerende hvis ønsket
- Mulighed for opbevaring af forskningsregistre med personfølsomt indhold, idet den projektansvarlige kan opnå dispensation for Datatilsynets krav om sletning og anonymisering af data efter opbevaringsperiodens udløb.

DDA/Sundhed er finansieret via Statens Arkiver. Det er gratis at lade data arkivere i arkivet og at rekvirere data til sekundæranalyse.

Generelt gælder, at forskningsmateriale skal være på elektronisk form for at kunne arkiveres i DDA/Sundhed. Arkivet omfatter således ikke biobanker og samlinger af parakliniske materialer.

Registerforskning

Ofte er sundhedsvidenskabelige undersøgelser baseret på dataudtræk fra registre. DDA/Sundhed kan opbevare administrative registerdata, som normalt vil blive slettet, når den registerforvaltende myndighed ikke længere har brug for dem. DDA/Sundhed's opbevaring af denne type data er godkendt af Datatilsynet.

Hvem ejer data i DDA/Sundhed?

Forskere, der lader deres data opbevare og evt. videreformidle af DDA/Sundhed beholder ophavsretten til undersøgelsesmaterialet. Ingen rettigheder overdrages til DDA/Sundhed som følge deraf.

Hvem har adgang til data?

Forskeren bestemmer ved deponeringen, hvilken adgangsklausul datamaterialet skal pålægges, lige fra fri afbenyttelse for tredjepart til krav om primærundersøgers skriftlige accept ved enhver videreformidling af data.

En af tankerne med DDA/Sundhed er at øge mulighederne for genanvendelse af data til sekundæranalyse, derfor opfordrer DDA/Sundhed som udgangspunkt til så fri tilgængelighed som muligt. Uanset graden af adgangsklausulering informeres primærundersøger i forbindelse med enhver udlevering.

Hvordan indleveres data?

Indlevering foretages ved, at forskeren sender DDA/Sundhed en kopi af undersøgelsens data og dokumentation. Sammen med data og dokumentation vedlægges et udfyldt lokaliseringsskema, hvilket kan rekvireres ved henvendelse til DDA/Sundhed i Dansk Data Arkiv.

Data

Der stilles ikke krav til bestemte afleveringsformater - men de store statistikpakker SAS, SPSS og Stata foretrækkes.

- Så vidt muligt skal der være tale om de oprindelige variable. Rekodede/konstruerede variable ønskes kun, hvis de er af væsentlig betydning for undersøgelsen.
- Opgiv venligst hvilket program, version og format, der er anvendt ved dannelse af datasættet.
- Ved data fra registerundersøgelser bedes hele den aktuelle kørsel vedlagt, hvis udtrækket ikke kan skabes ved at køre programmet igen.
- Er der personfølsomme data i materialet, træffes der en særlig aftale omkring de praktiske omstændigheder ved indlevering til DDA/Sundhed og forhold omkring registertilladelser mm.

Dokumentation

Alle former for undersøgelsesdokumentation er som udgangspunkt relevante for DDA/Sundhed og kan indleveres sammen data:

- Al skriftligt og elektronisk dokumentation fra undersøgelsen – eksempelvis spørgeskemaer (u-udfyldte), instrukser til deltagere og undersøgere, variabellister, kodebeskrivelser mm.
- Publikationer/referencer til publikationer udgået fra undersøgelsen.
- Beskrivelse af evt. rekodede/konstruerede variable.

Hvori består en arkivering i DDA/Sundhed?

DDA/Sundhed foretager en såkaldt "oparbejdning" af det indleverede datasæt. Ved oparbejdning standardiseres data til et fælles elektroniske format, arkivet anvender. Arkivering i netop dette format sikrer, at undersøgelsens data og dokumentation vil kunne

genkaldes uafhængigt af de med tiden skiftende formater. Derudover er et omfattende back up-system med til at sikre data og dokumentation.

Personfølsomme oplysninger

DDA/Sundhed har – som an afdeling af Statens Arkiver – mulighed for at opbevare personfølsomme data udover den tidsperiode, der er afsat til det enkelte forskningsprojekts gennemførelse. En arkivering i DDA/Sundhed sidestilles af Datatilsynet med sletning, som det er krævet ved den enkelte registertilladelses udløb. Den afleverende forsker har endvidere mulighed for at søge Datatilsynet om genudlevering af originaldata fra DDA/Sundhed på et senere tidspunkt med henblik på follow-up-studier. Datasæt med personfølsomme oplysninger opbevares selvstændigt under maksimal beskyttelse på en separat og afsondret server. Ved oparbejdningen af datasæt med personfølsomme data fremstilles også et anonymiseret datasæt. Hermed kan tredjepart få adgang til datamaterialet til sekundæranalyse - naturligvis forudsat at donor/primærundersøger måtte tillade dette.

Lokaliseringsskema til brug ved indlevering af data kan downloades fra DDA/Sundheds hjemmeside; se nedenfor.

DDA/Sundhed
Islandsgade 10
5000 Odense C
Tlf. 6611 3010
Fax 6611 3060
http://www.sa.dk/dda/ddasundhed/om/

Appendix 5

Some advice on using Windows

It is rather unsafe to use any program without mastering the fundamental structure and facilities in Windows. Some of the kiosk booklets give good descriptions and advice, I find the booklets written by Michael Karbo excellent (forlaget IDG).

My main comments and recommendations apply to handling of the folder (directory, bibliotek, mappe) structure. There are several ways to move and copy files; I only show one technique.

Create a smart folder structure

Don't mix your own data and documents with program files; this is risky and will inevitably lead to confusion. Create a main folder for all of your own files (data, command files, text documents), e.g. `c:\dokumenter`, with all of your own files in subfolders under your main folder.

Organize your folder structure by subject, not by file type.

Example of folder structure.

Folder structure	Description
C:\	C:\ is the root folder

Programs
        EpiData
        SPSS
        Stata
        WordPerfect
        WinZip
        Games
            Solitaire
            Doom
    Windows
	Program folders should include programs only, *never* data nor documents created by yourself.

Dokumenter
        Personal
            CV
            Secrets (encrypted)
        Project 1
            Protocol
            Administration
            Data
               Safe
            Manuscripts
        Project 2
            Protocol
            Administration
            Data
               Safe
            Manuscripts
	C:\Dokumenter is your own main folder.
	
	All of your own data and documents should be placed in subfolders under your main folder. Organize the folders by subject, not by file type.
	
	This structure:
	• makes it easy for you to locate your own files.
	• facilitates the selection of files to be backed up (C:\Dokumenter and its subfolders).

This structure has several advantages:

- You avoid mixing own files with program files
- You can select your main folder (`c:\dokumenter`) as the default root folder for all of your own folders (see below), so that when opening or saving files, you see only your own folders, not the program folders.
- It is much easier to set up a consistent backup procedure (section 11.1 and appendix 6).

How to select a default folder for a program

The installation default working folder for many programs, e.g. SPSS, is the program folder itself. **This is an extremely poor choice**. You should never mix your own documents and data files with program files. You might never find your own files again; you might accidentally delete your data e.g. when installing a new version of the program; or you might accidentally delete program files.

Later I show how to select a default working folder for Explorer (Stifnder); for most other programs the technique is different. To make `c:\dokumenter` the default SPSS folder (the folder suggested when you want to open and save files):

- Right-click the SPSS shortcut icon
- Properties ► Shortcut ► Start in ► `c:\dokumenter`
  (Egenskaber ► Genvej ► Start i ►)

Use Windows Explorer (Stifinder)

I recommend to use Explorer rather than My Computer, and to put a shortcut at your desktop:

- Right-click the [Start] button and select Open
- Open the Programs folder
- *Right*-click the Explorer shortcut icon and drag it to the desktop; select Copy Here.

Make your main folder default when opening Explorer

- *Right*-click the Explorer shortcut icon
- Properties ► Shortcut ► Path ►
  (Egenskaber ► Genvej ► Sti ►)
  `C:\WINDOWS\EXPLORER.EXE /n, /e, c:\dokumenter`

Make Explorer display file name extensions

For reasons not understood by me, Microsoft decided not to display file name extensions by default. This is inconvenient (you can not distinguish the syntax file `alpha.sps` from the data file `alpha.sav`), and you should set Explorer to display file name extensions.

- Open Explorer
- View ► Options
  (Vis ► Indstillinger)
- • Uncheck: "Hide extensions for known file types"
            ("Skjul filtypenavne for kendte filtyper")

How to create a new folder

The example is to create the folder **project3** under **c:\dokumenter**

- Double-click the Explorer (Stifinder) icon at the desktop
- Click **c:\dokumenter** (root folder for own files)
- Files ► New ► Folder
  (Filer ► Ny ► Mappe)
- Rename 'New Folder' (Ny Mappe) to **'project3'**

How to rename a folder or file

- In Explorer, *right*-click the folder or file and select Rename
- Write the name desired and press [Enter]

How to copy a file or a folder to another folder or to a diskette

- In Explorer, highlight the source file or folder; press [Ctrl]+[C] (copy to clipboard)
- Highlight the target folder (or A:); press [Ctrl]+[V] (paste from clipboard)

How to move a file or a folder to another folder

- In Explorer, highlight the source file or folder icon; press [Ctrl]+[X] (copy to clipboard and delete source file)
- Highlight the target folder; press [Ctrl]+[V] (paste from clipboard)

You may also copy or move files and folders using the mouse to drag and drop. But the result may be a bit confusing (the effect is different whether you drag and drop within the same disk or between disks). The [Ctrl]+[C], [Ctrl]+[X], [Ctrl]+[V] method works consistently, and it works much the same as when editing text in a wordprocessor.

How to write-protect a file

To prevent a file from accidental deletion or overwriting you may write-protect it. To see the write-protection attribute for a file, *right*-click the file in Explorer and select Properties (Egenskaber). A checkbox indicates whether the write-protection attribute is off or on; you may change it manually.

Smart users write-protect their vital command and data files once they are OK.

Appendix 6

WinZip: a compression program for backup etc.

Download the program from <u>www.winzip.com</u> for a cost of $29 (you may test the program without any charge). The following description applies to WinZip 8.0.

A WinZip file (extension `.zip`) is termed an *archive*; it includes compressed copies of one or more files. To *archive* or *zip* files means to add compressed copies of files to an archive (.zip) file. To *extract* or *unzip* files means to restore uncompressed versions of the files.

Understanding the archive attribute

The archive attribute is used by backup programs to determine whether a file is new or modified since the last backup. When you create or modify a file, Windows sets the archive attribute on. To see the archive attribute for a file, *right*-click the file in Explorer and select Properties (Egenskaber). G or Y indicate whether the archive attribute is off or on; you may change it manually.

You may instruct WinZip and other backup programs to backup only files with the archive attribute on – and then turn it off. It remains off until you modify the file.

Main operations of WinZip

- *Create new zip-files*: In WinZip click the [New] toolbar button. Decide the name and location of the zip-file.

- *Open existing zip-file*: In WinZip click the [Open] toolbar button and locate the zip-file to be opened.

- *Add files to zip-file*: Click the [Add] toolbar button. You now see a dialogue box (next page).
  - o specify names of the files to be added. `*.*` indicates all files in the folder.
  - o If you want to add also files in subfolders check:
    Include subfolders   Save full path info
  - o If you want to add only files whose archive attribute is on, check:
    Include only if archive attribute is set   Reset archive attribute
                                                         (turn archive attribute off)
- Click the [Add] menu button

- *Extract (unzip) files from zip-file*: Open the zip-file with the [Open] toolbar button.
  - o Select the files to unzip and click the [Extract] button
  - o Choose target folder or check:
         Use folder names   to keep the original folder structure
  - o In the Extract dialogue box click [Extract].

Use WinZip to back up

On principles for backing up see section 11.1. You may use diskettes or CD-Rom, or you may send zip-files by E-mail to another computer. I recommend that you give your zip-files names that include the creation date in a sortable way: `2001.12.09.zip` for a zipfile created 9th December 2001. In this way it is easy for you to determine the sequence of restoring (unzipping) the backup files.

For a *total backup* select all files (`*.*`) in your own root folder (e.g. `c:\dokumenter`), and the following options:

Check:	Include subfolders	Check:	Save extra folder info
Uncheck:	Include only if archive attribute is set	Check:	Reset archive attribute

For an *incremental backup* select all files (`*.*`) in your own root folder, and the options:

Check:	Include subfolders	Check:	Save extra folder info
Check:	Include only if archive attribute is set	Check:	Reset archive attribute

Use WinZip to save disk space

This is the original purpose of zip programs. You might decide to remove older versions of your data set from your working folder and keep them in a zip-file.

Send large or multiple files by E-mail

The advantage is obvious with large files. But also with many smaller files it is an advantage to pack them together into one before attaching to the E-mail. The recipient should have a Zip program as well to be able to unzip. However, the purchased version (not the free download version) of WinZip can create 'self-extracting' zip-files.

Encryption

You may encrypt files using a password of your own choice. The password must be defined *before* archiving files. Use the [Password] button in the Add dialogue box.
If you loose your password you cannot access your data. So beware:

- **Do** select a password that you can remember but nobody else can guess. Long passwords with a mix of numbers, upper- and lowercase letters are recommended.
- **Do** use the same password for all your encrypted files
- Do **not** use your spouse's name, your server logon password nor your credit card pin code
- Do **not** write your password on a yellow sticker at your noticeboard

In the file display encrypted files are shown with a **+** after the filename

WinZip's degree of security will suffice for most health research data. WinZip says:

> "Password protecting files in a Zip provides a measure of protection against casual users who don't have the password and are trying to determine the contents of your files . . . . but does not provide absolute protection against determined individuals with advanced cryptographic tools."

Pitfalls and advice, SPSS and Stata.

Both programs have some pitfalls; here I address a few items especially relevant to documentation and safe handling of data.

SPSS

Default working folder

By default SPSS suggests that you use the program folder as the default working folder. **Don't do that!** See appendix 5 on how to define a default working folder for a program. Choose your own root folder (e.g. `c:\dokumenter`) as the default working folder; select the appropriate subfolder from there when opening or saving files.

Setup recommendations

Use Edit < Options to set your default preferences. In my SPSS booklet section 6 you see my recommendations in detail. Especially these choices are important:

[Viewer]:	Y Display commands in the log		
[Output labels]:	Pivot table labelling:	Variables:	Names and Labels
		Values:	Values and Labels

Risk to destroy good data

At exit from SPSS you may get this question:

> Save contents of data editor to <filename>?

Your response should be **NO!** If you made changes in the data set, eg. by a `SELECT IF` or a `RECODE` command, your good original data will be overwritten with bad or undocumented data.

- If you did not modify your data, you will not be asked this question.
- If you made modifications intended to be temporary, you should obviously not overwrite your original data. Respond **NO**.
- If you want to make permanent modifications, do it in syntax (as in examples 8, 11, 13):
  - The syntax file starts with a `GET FILE` and ends with a `SAVE OUTFILE` command, with the transformation commands in between.
  - The modified data should be saved with a **new** name (example: `visit1c.sav`).
  - Also save the syntax file with a name that reflects what it did (`gen.visit1c.sps`).

As a safeguard, keep a copy of your data and syntax files in a safe folder (see chapter 5 and appendix 5). It is also a good idea to write-protect your data files (see appendix 5).

Syntax files (`.sps`) and output files (`.spo`) can normally be saved at exit without data loss. It is the data set (`.sav`) you can damage if you are not careful.

Mouse and menus: a good servant, but an evil master

SPSS has a highly developed menu system that enables you do create almost any command without looking it up in the manual. For frequently used commands, including documentation commands, it is, however, much faster to type the commands than to use the menus. And sometimes the command developed by the menu system is less than transparent. I want to analyze males only (sex=1). Look here:

```
USE ALL.
COMPUTE filter_$=(sex=1).
VARIABLE LABEL filter_$
  'sex=1 (FILTER)'.
VALUE LABELS filter_$
  0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
```

These commands were created by mouse and menus (Data ► Select cases). They work as desired: only males are included in the following analysis. But I had a hard time finding out what was actually going on.

```
TEMPORARY.
SELECT IF (sex=1).
```

These commands I can easily type – and understand afterwards.

Stata

Default working folder

By default Stata suggests that you use `c:\data` as the default working folder. However, I stand by the advice in chapter 5 to organize folders by subject, not by file type. Choose your own root folder (e.g. `c:\dokumenter`) as the default working folder (see appendix 5 how to do it). Select the appropriate subfolder when opening or saving files.

Risk to destroy good data

If you want to make permanent modifications, do it with a do-file (as in examples 8, 11, 13):

- The do-file starts with a *use* and ends with a *save* command, with the transformation commands in between.
- The modified data should be saved with a **new** name (example: `visit1c.dta`).
- Also save the do-file with a name that reflects what it did (`gen.visit1c.do`).

When saving a data set your request will be rejected if a file with that name already exists. This is a safeguard against unintentionally overwriting good data. If you really want to overwrite existing data, use the *replace* option:

save `"c:\dokumenter\proj1\visit1b.dta"` *, replace*

To avoid accidents, only use the *replace* option if you really want to overwrite existing data. The typical situation is after correction of an error in the do-file.

As a safeguard, keep a copy of your data in a safe folder (see chapter 5 and appendix 5).

Value labels

Stata has some shortcomings of a rather trivial kind: the display of labels in tables etc. is less than optimal. While SPSS can display both the code and the value labels (see SPSS recommendation above), Stata displays *either* the code *or* the value label, but the command *mumlabel* ensures that both codes and value labels are displayed:

numlabel _all, add
save "c:\dokumenter\proj1\visit1b.dta" [, *replace*]

Although you can define long value labels, Stata in some tables only displays the first few characters, so value labels should be kept short.

Missing values

There are two types of missing values:

The *system missing value* is shown as a **.** (period). It is created in input when a numeric field is empty, by invalid calculations, e.g. division by 0, or calculations involving a missing value.

User-defined missing values are **.a, .b, .c, ...z** . It is a good idea to use a general principle consistently, e.g.:

.a   Question not asked (complications to an operation not performed)
.b   Question asked, no response
.c   Response: Don't know

Unfortunately no data entry program accepts **.a** in a numeric field. In EpiData you might choose the codes **-1** to **-3** (provided, of course, that they could not be valid codes) and let Stata recode them:

recode _all (-1=.a)(-2=.b)(-3=.c)

In the primary data set you should definitely keep the originally entered codes.