Part A. EpiData Entry

Part A: EpiData Entry

Exercise 1 A data documentation sheet for a simple questionnaire

Exercise 2 The `QES-REC-CHK` triplet

Exercise 3 Derived fields and Check file commands unrelated to a specific field

Exercise 4 Data entry and validation

Exercise 5 Using an external file for Labelblocks

Exercise 6 Dealing with incomplete dates

Exercise 7 Keeping track of data entry time

Exercise 8 Safely backing up and encrypting your data

Acknowledgments:

We thank Ajay M V Kumar who has made valuable suggestions to improve the structure and flow of argumentation of Part A.

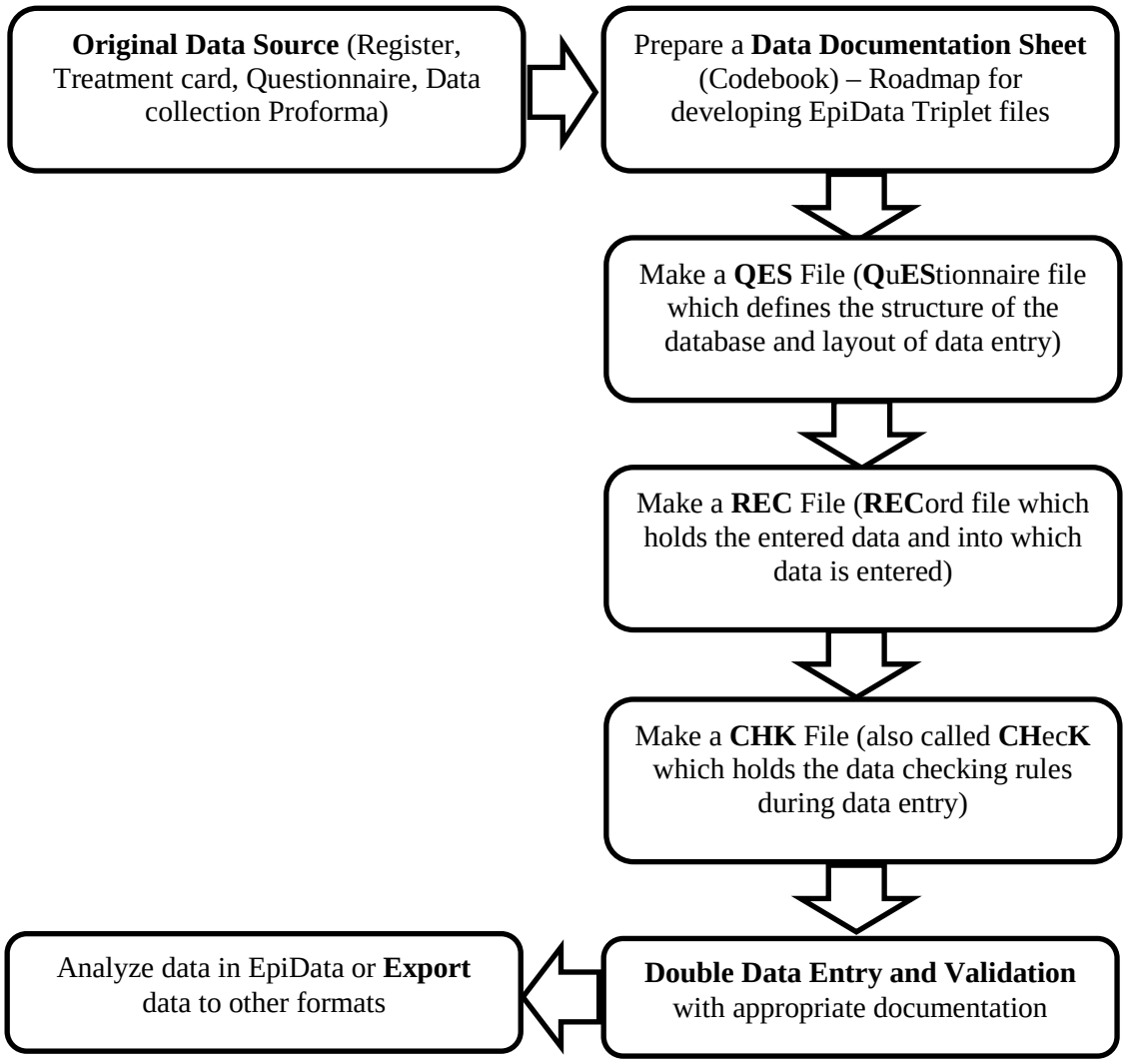
Exercise 1: A data documentation sheet for a simple questionnaire

At the end of this exercise you should be able to:

a. Define the different types of fields/variables (text, numeric, date) and know when to use them.

b. Create a data documentation sheet from a simple questionnaire

Before we proceed to learn the details, let me provide an overview of EpiData entry.



The first step in the process is to prepare a plan for data entry. This plan is called the **data documentation sheet**. This should not be confused with *data collection form* which is the proforma used for collecting the data from study participants or extracted out of the programme records. Data documentation sheet is a codebook containing the details of all the variables (like names, labels, type, length, possible values and value labels) to be entered and the check rules to be applied during the process of data entry. Like Epi Info 6, EpiData Entry uses the same principle of what we call the `QES-REC-CHK` (pronounced "Ques-Rec-Check")

files principle. First we create a `QES` file. This file defines the structure of the database and the layout for data entry including field names, field labels, field type, and field length. From the `QES` file we then create a `REC` file (data entry file which will contain all the data), and finally we create a so-called `CHK` file (which contains and applies the rules of data entry) linked to the data entry file to control data entry. These are referred to as *EpiData triplet files* and are identified by their file extensions (`.qes`, `.rec` and `.chk`). Double data entry and validation is considered a benchmark in assuring data quality and we fully subscribe to this idea. Once `QES-REC-CHK` files are created, data are entered twice, independently by two persons and compared with each other to identify discrepancies. These are then corrected by referral to the original data source and saved as a final file which is used for analysis. The rationale of double data entry is that the probability of committing the same error in the same field twice, when entered by two independent data entry operators is small and hence data entry errors remaining after validation will be minimal.

Note: Please do not worry if you are not able to understand all the terms at this point in time. Be assured that you will appreciate these as you go along.

But let us proceed step by step and say that we have the following questionnaire:

Laboratory serial number: ___

Date specimen received (dd/mm/yyyy): ___/___/_____

Sex: ___

Age in years: ____

Reason for examination: ___

Result of specimen 1: ___

Result of specimen 2: ___

Result of specimen 3: ___

This might present a typical simple questionnaire as used by an interviewer. Often such questionnaires are first completed on paper. This is actually an excerpt from the Tuberculosis Laboratory Register proposed by The Union:

Tuberculosis Programme Form 2

Tuberculosis laboratory register Year_____

Lab Serial No.	Date specimen received	Name	Sex M/F	Age	Name of referring facility	Address - patient for diagnosis	Reason for examination*		Results of specimen			Only for SS+ for diagnosis: TB Number or treatment centre**	Remarks
							Diagnosis (tick)	Month of follow up	1	2	3		

We will use this register as the basis for this course. For the time being, you plan to write a short and concise electronic data capture form, retaining only variables that are easy to capture and are likely to be useful for the analysis. *Please note this as a first principle in being efficient – capture only those variables which you will use for analysis!*

Each of the questions can be conceived of as a variable and the answer to the question as the value that the variable takes for a particular individual. **Variables** are also referred to as **'Fields'** in EpiData – both mean the same and will be used interchangeably. We will give

each variable a unique name. A completely entered data form for one study subject is called a '**Record'**. A set of such records is called a **'File'** (REC file). The REC file thus contains several records and each record contains information about one individual with respect to several variables. We are going to describe each variable with respect to several attributes in the data documentation sheet. Let us now understand some terminology we are going to use.

- **Field name:** This is the name of the variable and in EpiData, there are certain rules to be followed in arriving at this name. We will come to these rules in a short while.
- **Field Label:** This is the descriptive name for the variable and contains a more detailed description than the variable name can convey.
- **Field Type:** This describes the type of the variable – text, numeric or date being the major types.
- **Field length:** This describes the number of characters that a value can take.
- **Field values:** This describes the possible values that a variable can take.
- **Value labels:** These are descriptive names for the values. For categorical variables which are numerically coded, it is always useful to label them so that it is easier to read and understand what each of the codes mean.

"**Labels**" are also called "**metadata**" or "**data about data**". They play a key role in data files. We may have entered a value "9" for a given field, but this number remains meaningless for everyone without clearly specifying for what this value stands. It is important to get acquainted to these terms and understand them clearly since we will be using them frequently. We will be using several examples later in this chapter to clarify these terms.

Field name: Now, let us understand some rules in naming a variable.

First, it has to be **single word** that has **not more than ten characters**. This means that you cannot use a space in the name as a space makes it more than a word. Also, you cannot use any special characters like comma, semicolon, full-stop or underscore.

> Note that some other analysis software may accept only a field length of eight characters. If you later plan to export your EpiData files for analysis to such a software package and you had used the full field length of ten, then your field names get truncated.

Second, use a name which is **intuitive** to understand what it means instead of generic field names like v1, v2 and so on.

Third, it may be a good practice to keep the field names in **lower case**. This can be forced by setting up an option. We have already done it and you may verify the set-up in "File" | "Options" | "Create Data File". Though EpiData is not case-sensitive, some other software is. So, a field name of '`sex`' (lower case), '`SEX`' (Upper case) and '`Sex`' (sentence case) are understood differently. If you later plan to export your EpiData files for analysis to such a software package (two examples are Stata and R, both of which are "case-sensitive"), it may be a problem. Hence, the recommendation to keep it uniformly, "lower case".

Do not start the variable name with a number. It cannot be '`1v`', but it can be '`v1`'

The following words '`date`', '`month`', and '`year`' are functions in EpiData and are reserved names. Hence they cannot be used as variable names.

Note: If the Field label begins with a word that is identical to the Field name, you will note later in EpiData Analysis, that this word will be truncated from the Field label. For instance, if your Field name was SEX, and you used 'SEX OF EXAMINEE' as your Field label, this would be truncated to 'OF EXAMINEE'. While this can be fixed easily in EpiData Analysis, it is preferable to prevent it by choosing an alternative Field label during questionnaire design.

Field label: This is the full description of the variable and can be more than a word. Anything that is written between the Field name and the field definition in the QES file is considered as field label.

Field Type: There are different types of entry fields for the variables (we will follow the EpiData Entry notation and call them "Fields"):

- **Text fields**: These fields take letters or numbers or a combination of these as possible values, like PETER, KOCH1882, giraffe, 45677 etc. You can type anything on the keyboard into this field. If you enter a number into such a field, it is accepted but you will not be able to make any calculation with it. These fields are also sometimes designated as character or alphanumeric fields, or most simply "**string**" (denoted by **S**) fields as they take any string of characters.
- **Numeric fields**: These are numbers. The numbers might be integers (denoted by **I**) like 885, 33, 1235 or real numbers like 3.4, 6.88, and 66.5 (also called **floats** and denoted by **'F'**). You can make calculations with such fields.
- **Date fields**: (denoted by **'D'**) In different countries, different ways of writing dates are used and this can be confusing for people from another culture. Some write *5 March 2005*, others *March 5 2005*, and again others *2005 March 5*. EpiData Entry lets you choose the type of date you wish to take. In this course we will use European dates, i.e. dates of the format *5 March 2005* or symbolized with DD/MM/YYYY.
- One other type of variables is called **"logic" or "Boolean"** variables. This is sometimes used in food-borne outbreak investigations. There, answers to questions on food items eaten is limited to "yes" and "no" and "missing". In EpiData Entry, this type of field accepts only the values Y, N, 0, 1, and space. There is no need for using this field type, and we actually discourage its use as it might pose problems in analysis. The alternative is a numeric field with a label block.

While you are asked to limit the length of the field name, you have much more flexibility with the length of the value a field can take (up to a field length of 80), but we will try to use it as efficiently as possible, that is we will limit the value length to the minimum needed.

Data Documentation Sheet

It is good practice to write what we call a **data documentation sheet** before you make your actual EpiData Entry QES file. As mentioned earlier, EpiData refers to this as **Codebook**.

In the past, fields like SEX were commonly made text field with values "F" or "M" denoting Females and Males. It is efficient as it uses only a length of 1. Things would get less efficient, if we would have to code treatment outcome with the possible values "cured", "completed", "died", "failed", "lost from follow-up", "transferred out", and "outcome not recorded".

Numeric coding is much simpler as there are up to ten possible values with a field length of just 1:

```
1  Cured
2  Treatment completed
3  Died of any cause
4  Failed bacteriologically
5  Lost from follow-up
6  Transferred out
9  Outcome not recorded
```

You will also realize later in the analysis that this will be very convenient to apply the selection criteria when you want to select a subset of data and undertake analysis only on the subset. Of course, a prerequisite is that the link between the numeric value and the text label is unambiguously clear. The role of labels is of enormous importance, they are also called meta-data or "data about data". We are going to make full use of numeric coding of field values and using explicit text as value labels.

Let us now go through a few examples of the variables (from the tuberculosis laboratory register example) and describe the various attributes of the variables in the data documentation sheet. As you go through, you will note that preparation of data documentation sheet requires thinking and knowledge of study data.

This is how we would write such a data documentation sheet:

Field name	Field label	Field type	Field length	Field values	Value labels	Comment
serno	Laboratory serial number *	I	4	1-9000, 9001, 9002,		Serial number starting with 1 each year Enter 9001, 9002,… if serial number is *not unique* or *missing*, and write a data entry note (use F5 to open a note file)
regdate	Registration date	D	10	01/01/2000-31/12/2005, 01/01/1800		Range of legal registration dates Enter 01/011800 if no date recorded
sex	Examinee's sex	I	1	1 2 9	Female sex Male sex Sex not recorded	

**** Note***: Commonly, it will be preferable to make the identifier a text field. If it is a number, as in this case here with the laboratory serial number, precautions must be taken to distinguish e.g. "`0001`" from "`1`", requiring that the numeric value is entered into one field, and another field, the actual identifier field, is automatically correctly calculated to add leading zeros where appropriate. This will actually be done in a later exercise.

Task:

- ***Complete the data documentation sheet for all fields in the questionnaire. Note that you should always define a value if no answer was provided to a question.***
- ***Think of the most efficient ways to code reason for examination and results of microscopic examination***