

Entering data with EpiData

Svend Juul, September 2008

EpiData is an easy-to-use program for entering data. It has the facilities needed, but nothing superfluous. Data entered can be exported to EpiInfo, Excel, DBase, SAS, SPSS and Stata. EpiData with documentation in several languages is available for free from <http://www.epidata.dk>.

I first show the basic tools; with them you should be able to start entering data a few minutes after you installed the program. Detailed documentation is included when you download the program.

1. The basics

EpiData files

If your data set has the name **first**, you will work with three files:

- first.qes** is the definition file where you define variable names and entry fields.
- first.rec** is the data file in EpiInfo 6 format.
- first.chk** is the checkfile defining value labels, legal values and conditional jumps.

Customize EpiData

Before starting for the first time, set general preferences (File ► Options). I recommend:

Show dataform	Create datafile
Font: Courier New bold 10pt. Background: White Field colour: Light blue Active field: Highlighted, yellow Entry field style: Flat with border Line height: 1	IMPORTANT: First word in question is fieldname Lowercase (<i>especially if you use Stata for analysis</i>)

You may also want to change font to improve legibility while entering data (I use a bold Courier font).

The EpiData toolbar

EpiData's toolbar guides you through the process:



Here I describe the [Define data], [Make datafile], [Enter data], and [Export data] steps.

[Define data]: Create a definition file (**.qes** file)

Hit [Define Data], and you get the EpiData editor (a simple text editor) where you can define variable names, labels, and formats. If the name of your data set is **first**, save the definition file as **first.qes**:

```
FIRST.QES   My first try with EpiData.
entrdate   Date entered           <today-dmy>
lbnr       Questionnaire number   #####
init       Initials               ____
sex        Sex                    #      (1 male   2 female)
npreg      Number of pregnancies  ##

=====
                                           Page 2

bdate      Date of birth           <dd/mm/yyyy>
occup      Occupation              ##      (see coding instruction OCCUP)
```

- The first word is the field name (variable name); the following text becomes the variable label.
- **##** indicates a two-digit numeric field;
- **##.#** a four-digit numeric field with one decimal;
- **____** a three character string variable;
- **<dd/mm/yyyy>** a date;
- **<today-dmy>** an automatic variable: the date of entering the observation.

Text not preceding a field definition ("**1 male 2 female**") are instructions etc. while entering data. The purpose of "**=====**" and "**Page 2**" are to make page shifts more visible at the screen while entering data.

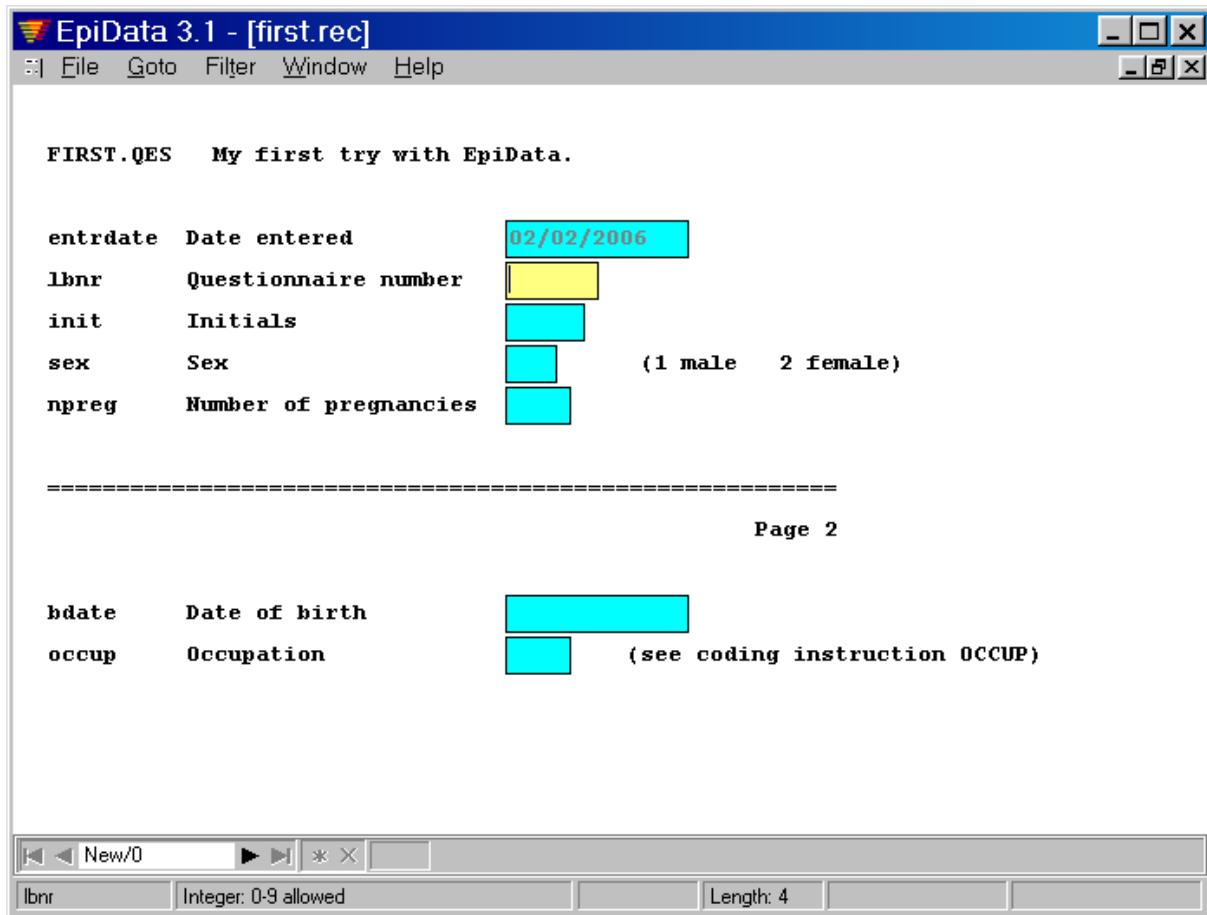
Variable names can have up to 8 characters **a-z** (but not æøå) and **0-9**; they must start with a letter. Avoid special characters; also avoid **_** (underscore). If you use Stata for analysis, remember that Stata is case-sensitive, so you will typically use lowercase variable names.

[Make datafile]: Create an empty data file (**.rec** file)

Now hit [Make Datafile], using the definition file **first.qes** and create the empty data file **first.rec**.

[Enter data]: Enter data

Open the newly created **first.rec**. You now see a data entry form as you defined it; it is straightforward. With the options suggested the active field shifts color to yellow, making it easy for you to see where you are. When you have finished entering data for an observation, you are asked whether you want to save it to disk. The answer should be **Yes**, of course (just hit *Enter*).



The screenshot shows the EpiData 3.1 interface for the file 'first.rec'. The window title is 'EpiData 3.1 - [first.rec]'. The menu bar includes 'File', 'Goto', 'Filter', 'Window', and 'Help'. The main display area shows the questionnaire definition for 'FIRST.QES' with the title 'My first try with EpiData.'. The form contains several fields: 'entrdate' (Date entered) with the value '02/02/2006', 'lbnr' (Questionnaire number) which is highlighted in yellow, 'init' (Initials), 'sex' (Sex) with options '(1 male 2 female)', and 'npreg' (Number of pregnancies). A dashed line separates the top section from the bottom section, which is labeled 'Page 2'. The bottom section includes 'bdate' (Date of birth) and 'occup' (Occupation) with a note '(see coding instruction OCCUP)'. At the bottom of the window, there is a status bar with a variable name 'lbnr', a description 'Integer: 0-9 allowed', and a length indicator 'Length: 4'.

[Export]: Export data to a format of your own choice

Finally you can export your data to a statistical analysis program. The **.rec** file is in EpiInfo 6 format, and EpiData creates dBase, Excel, Stata, SPSS and SAS files. Variable and value labels (if any) are transferred to Stata, SAS and SPSS files, but not to spreadsheets.

2. Slightly more advanced:

[Add checks]: Add value labels and valid values

Hit [Add checks], and select the appropriate .rec file. You now see a dialog box; its meaning is not obvious at first sight, and I will explain a little:

first.chk		Checkfile name
sex	▼	Select the variable
Sex of respondent	Number	Variable label and data type displayed
Range, Legal	1, 2, 9	Define possible values. A range e.g. as: 0-10, 99
Jumps	1>bdate	Jump to bdate if sex is 1
Must enter	No ▼	Skipping the field may be prevented
Repeat	No ▼	Same value in all records (e.g. operator ID)
Value label	sexlbl ▼ +	[▼] Select among existing label definitions [+] Define new value labels
Save Edit Close		Save Save variable definitions Edit Edit variable definitions

The results of your actions are stored in a checkfile (**first.chk**) which is structured as below.

```
* FIRST.CHK
LABELBLOCK
  LABEL sexlbl
    1 Male
    2 Female
  END
END
sex
  COMMENT LEGAL USE sexlbl
  JUMPS
    1 bdate
  END
END
```

Good idea to include the checkfile name as a comment.

Create the label definition **sexlbl**. You might have used the name **sex**. A label definition, e.g., **n0y1** (0 No; 1 Yes) may define a common label for many variables.

Use the **sexlbl** label definition for **sex**. Other entries than 1, 2, and nothing will be rejected.

If you enter 1 for **sex**, you will jump to the variable **bdate**.

Experienced users enter the specifications directly in a checkfile; less experienced use the dialog. Specifying legal ranges before entering data may, however, not be worth the effort. I find it easier to check for invalid values as the first step in the analysis, and if a questionnaire was filled in inconsistently, it may give difficulties while entering data.

[Document]: Documentation procedures

You may create a list of variables or a codebook including variable and value labels and checking rules.

Entering data twice

As an assurance against typing errors you may enter the data a second time in a second file and compare the contents of the two files. From the primary data set you can create an empty copy (the secondary data set):

Tools ► Prepare Double Entry Verification

Next, enter the data in the secondary data set. Compare the two data sets by:

Document ► Validate Duplicate Files

In the dialog that appears, specify the location and names of the primary and secondary data sets. In **Select Key Fields**, specify the id variable or variables that uniquely define each observation.

This creates a report on the discrepancies between the two data sets. If there are differences, find the correct entries in the original questionnaires. I recommend making corrections in both data sets and repeating the comparison. This comparison should be without errors, and you should save the report for documentation.

3. Stata users only: Two problems – and the solutions.

Problem 1: Abbreviation of long variable names

When exporting from EpiData to Stata, you may experience that long variable names are abbreviated to 8 characters. This occurs if EpiData saves the data set in Stata 6 format (Prior to Stata 7, variable names could be no longer than 8 characters).

Solution: When exporting, click the Options tab and select Stata 8 format.

Problem 2: Abbreviation of value labels

When exporting from EpiData to Stata, a variable that in EpiData was defined as two-digit (##), gets the `%2.0f` display format in Stata, meaning fixed format with two digits. If a value label was defined in EpiData, many Stata commands, e.g., `list`, will display only the first two characters of the value label.

Solution: Redefine the display format in Stata for all variables to general format:

```
format _all %9.0g
```

New problem: If the data set includes string variables, the above command will give an error message. Solution: Identify the numeric variables and apply the **format** command to them only:

```
sysuse auto.dta
describe
format price-foreign %9.0g
```

Slightly more advanced: This can be automated. The following **ds** command creates a macro, **r(varlist)**, with the names of all numeric variables, and the format specification is used on these variables:

```
ds, has(type numeric)
format `r(varlist)' %9.0g
```