\subsection{Labeling variables}

One great disadvantage of {\color{blue}R} is its lack of labeling of variables in a dataset. One way round it is to name the variables with the label and and then refer to the varaible with its index. This obviously is no where near ideal. This is illustrated below using one of R inate datasets, “women”:

\footnotesize{\color{blue}\begin{verbatim}

> data(women)

> str(women)

'data.frame': 15 obs. of 2 variables:

$ height: num 58 59 60 61 62 63 64 65 66 67 ...

$ weight: num 115 117 120 123 126 129 132 135 139 142 ...

> attributes(women)

$names

[1] "height" "weight"

$row.names

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

$class

[1] "data.frame"

>

> # Renaming the variables to their labels

> attributes(women)$names[1]<-"Height of Women"

> attributes(women)$names[2]<-"Weight of Women"

>

> str(women)

'data.frame': 15 obs. of 2 variables:

$ Height of Women: num 58 59 60 61 62 63 64 65 66 67 ...

$ Weight of Women: num 115 117 120 123 126 129 132 135 139 142 ...

\end{verbatim}}\normalsize

The other option is using the \verb|label.var()| function in the \verb|epicalc| package. This is a very pleasant solution so far as one used the function in \verb|epicalc| but becomes ineffectual when other functions from other packages are used. A typical exmple is given below:

\footnotesize{\color{blue}\begin{verbatim}

> rm(women)

> library(epicalc)

> use(women)

> label.var(height, "Height of women")

> label.var(weight, "Weight of women")

> des()

No. of observations = 15

Variable Class Description

1 height numeric Height of women

2 weight numeric Weight of women

\end{verbatim}}\normalsize

\subsection{Merging and appending datasets}

The simplest way of combing two datasets of the same structure is to use \verb|rbind()|. This situation may arise if two or more dataenty clerks using the same database enter different records. The two files can then be combined as below.

\footnotesize{\color{blue}\begin{verbatim}

> setwd("C:\\Users\\Bal\\Desktop\\ITR")

> df1<-read.table("data2.txt", header=T, sep="\t")

> df2<-read.table("data3.txt", header=T, sep="\t")

> df1 # First file

id age sex

1 1 22 M

2 2 32 M

3 3 12 F

4 4 35 M

5 5 31 F

> df2 # Second file

id age sex

1 6 24 F

2 7 25 M

3 8 27 F

4 9 19 M

5 10 33 F

> df.all<-rbind(df1,df2)

> df.all # combined files

id age sex

1 1 22 M

2 2 32 M

3 3 12 F

4 4 35 M

5 5 31 F

6 6 24 F

7 7 25 M

8 8 27 F

9 9 19 M

10 10 33 F

\end{verbatim}}\normalsize

\footnotesize{\color{blue}\begin{verbatim}

Merging one or more datasets is very common in data management and analysis. This can be done in R using the \verb|merge()| function. Caution has to be taken to ensure that all variables are included in the resulting dataframe. An example is shown below and incolves the derived dataset \verb|df.all| and another whick was used to record only the heights of the study paticipants. Of not eher is that not all the 10 participants in the original file have heights recorded. This is seen in the resulting dataframe to be indicatd as missing.

\footnotesize{\color{blue}\begin{verbatim}

> df3<-read.table("data4.txt", header=T, sep="\t")

> df3 # Height data

id height

1 1 152

2 3 155

3 4 164

4 5 159

5 6 170

6 9 157

7 10 160

> merge(df.all, df3, by.x="id", by.y="id", all.x=T, all.y=T)

id age sex height

1 1 22 M 152

2 2 32 M NA

3 3 12 F 155

4 4 35 M 164

5 5 31 F 159

6 6 24 F 170

7 7 25 M NA

8 8 27 F NA

9 9 19 M 157

10 10 33 F 160

\end{verbatim}}\normalsize

\subsection{Reshaping data}

Often repeated measurements are done on individual subjects and recorded in the long format. However many statisical analysis can only be exclusively be done either on the wide or long format. This means that whatever format one has it should be easily reshaped into the other. This can be achieved in R using the \verb|reshape()| function.

\footnotesize{\color{blue}\begin{verbatim}

\end{verbatim}}\normalsize