Block 3: Analysing two variables (and sometimes three)

3.4.1 Conditional transformations

[Draft only: 25 May 2013]

Although it is possible to derive or modify variables with **COMPUTE** and **RECODE** (and often much cheaper) there are occasions when it is necessary to use conditional statements. Examples might be generation of a variable with a "pensionable age" category in which men are 65+ but women women 60+. Another example would be **height in metres** from the pre-course questionnaire, where some people have only given it in **feet** and **inches**. Although **COMPUTE** can do the conversions, it can't easily get round the problem of missing values. In such cases we have to use the **IF** or **DO IF** command.

1: IF [1]

For example, to generate a new variable **PENS** with value 1 for male pensioners, 2 for female pensioners, 3 for male non-pensioners and 4 for female non-pensioners, where the pensionable age for men is 65, but is 60 for women:

```
if ((sex eq 1) and (agegroup ge 65)) pens=1 .
if ((sex eq 2) and (agegroup ge 60)) pens=2 .
if ((sex eq 1) and (agegroup lt 65)) pens=3 .
if ((sex eq 2) and (agegroup lt 60)) pens=4 .
```

IF will work without brackets, but it is useful, and sometimes essential, to use brackets to make it absolutely clear what logical steps you wish to be followed.

Logical expressions have to be **TRUE** for the command to be executed. If they are **FALSE**, or if the expression cannot be evaluated because one or more of the arguments is **MISSING**, then the command will not be executed.

Commands are executed in the order in which they are encountered, so be careful not to override an earlier command with a later one.

IF is an expensive procedure to run, as it makes a pass through the data for each **IF** command: it is usually cheaper to use combinations of **COMPUTE** and **RECODE** to achieve the same effect, as in the following "trick of the trade":

```
compute    pens = sex*100+age.
recode     pens (100 thru 164 =1) ( 200 thru 259 =2) (165 thru 190 = 3) (260 thru
290=4).
format     pens   (f1.0).
var lab    pens 'Gender/pensioner combination'.
val lab    pens   1 'Male under 65'       2 'Female under 60'
                  3 'Male pensioner'      4 'Female pensioner'.
freq pens.
```

[1] **Format:**

```
IF  ( <logical expression> )  <target var> = <expression>
```

Logical operators: (as for **SELECT IF**)

Equal to	=	**EQ**	Not equal to	<>	**NE**
Greater than	>	**GT**	Less than	<	**LT**
Greater than or equal	>=	**GE**	Less than or equal	<=	**LE**

together with one or more keywords **MISSING AND OR NOT**

pens

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Male under 65	312	33.5	34.0	34.0
	Female under 60	374	40.1	40.7	74.7
	Male pensioner	68	7.3	7.4	82.1
	Female pensioner	164	17.6	17.9	100.0
	Total	918	98.5	100.0	
Missing	System	14	1.5		
Total		932	100.0		

[Source: Quality of Life in Urban Britain 1975]

To avoid writing multiple **IF** commands (each of which makes a pass through the data) several commands can be executed within a **loop**. This is done with **DO IF ~ ~ ~ ~ END IF**

This saves processing time (and money!).

2: DO IF [2]

The above example could also be written:

```
do if ((sex = 1) and (age >= 65))
compute pens = 1 .
else if ((sex = 2) and (age >= 60))
compute pens = 2 .
else if ((sex = 1) and (age < 65))
compute pens = 3 .
else if ((sex = 2) and (age < 60))
compute pens = 4 .
else
compute pens = 0 .
end if .
missing values pens (0) .
```

For both **IF** and **DO IF ~ ~ ~ ~ END IF**, whenever SPSS finds a missing value for any variable in a logical expression, the target variable will be set to system-missing if it is newly specified, or will retain its previous value if it is an existing variable).

End of session

Next session: *[Still to be written, but will be at least one exercise and one homework exercise for* **IF** *and* **DO IF***]*

[Back to Block 3 menu]

[2] **Format:**

```
DO IF    ( <logical expression> )
<transformation commands> .
[ ELSE IF ( <logical expression> ) ]
<transformation commands> .
~ ~ ~ ~
[ ELSE ]
<transformation commands> .
END IF .
```