What does the keyword “to” indicate in SPSS?

Authored by

stats writer

June 30, 2024

The keyword "to" in SPSS is used to specify a range of values in syntax commands. It is commonly used to select a range of variables or cases, such as in the command "SELECT VARIABLES=income to age," which will select all variables from income to age. It can also be used in conditions, such as in "SELECT IF=income to age > 5000," which will select all cases where the value of income to age falls within the specified range.

In SPSS the keyword to as two different meanings that depend on the context in which the word is used.

If existing variables are being manipulated, to means positionally consecutive. "Positionally consecutive" means that all of the variables in between the two listed in the command will have the operation performed on them, regardless of the numeric suffixes of the variables. For example, if you have the variables a1 a2 b1 a3 in that order in your data set and issue the recode command using "to", recode a1 to a3, all four variables will be recoded.

The keyword to can also be used to indicate numeric consecutiveness. "Numerically consecutive" means that variables with the same prefix and numeric suffixes will be considered, regardless of their position in the data set. In SPSS, if the command will create new variables, then

the keyword to will act in a numerically consecutive manner. For example,
if you issue the do repeat command to create a series of new variables, SPSS will create the new variables with whatever
prefix you specify (e.g., newvar) and consecutive numeric suffixes (e.g., 1, 2, 3 etc.).

We will use the following hypothetical data set to illustrate how some
of the commands use the keyword to.

```
data list list / num1 num2 num5 pos1 pos2 snorf pos3 x z.
begin data
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 2 1
1 1 1 1 1 2 1 3 1
1 1 1 1 1 2 1 4 1
end data.
execute.
save outfile 'c:to.sav'.
```

Examples of the keyword "to" acting in a positionally consecutive

manner

recode pos1 to pos3 (1 = 0).
list.

NUM1 NUM2 NUM5 POS1 POS2 SNORF POS3 X Z

1.00 1.00 1.00 .00 .00 .00 .00 1.00 1.00
1.00 1.00 1.00 .00 .00 .00 .00 2.00 1.00
1.00 1.00 1.00 .00 .00 2.00 .00 3.00 1.00
1.00 1.00 1.00 .00 .00 2.00 .00 4.00 1.00

Number of cases read: 4 Number of cases listed: 4

missing values pos1 to pos3 (8,9).
display dictionary.

List of variables on the working file

Name Position

NUM1 1
Measurement Level: Scale
Column Width: 8 Alignment: Right
Print Format: F8.2

Write Format: F8.2

NUM2 2

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

NUM5 3

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

POS1 4

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

Missing Values: 8.00, 9.00

POS2 5

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

Missing Values: 8.00, 9.00

SNORF 6

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

Missing Values: 8.00, 9.00

POS3 7

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

Missing Values: 8.00, 9.00

X 8

Measurement Level: Scale

Column Width: 8 Alignment: Right

Print Format: F8.2

Write Format: F8.2

Z 9

Measurement Level: Scale

Column Width: 8 Alignment: Right
Print Format: F8.2
Write Format: F8.2
As you can see, all of the variables between pos1 and pos3, including snorf, were recoded and had missing values assigned to them.

If you need to reorder the variables in your data set to make them positionally consecutive, you can use either the save or the xsave command with the /keep subcommand. The variables will be saved in the order in which they are listed on the keep subcommand. Note that any variables not listed on the keep subcommand will be dropped. To avoid potential problems, you can list the variables that you want first in the data set in the order that you need, and then use the SPSS keyword all. This will include all of the variables not already listed, in the order in which they are in the current data set. For example,

save outfile 'c:test1.sav'

/keep pos1 to pos3 all
/map.

OUTPUT FILE MAP

Result Input1
---- ----
POS1 POS1
POS2 POS2
SNORF SNORF
POS3 POS3
NUM1 NUM1
NUM2 NUM2
NUM5 NUM5
X X
Z Z

get file 'c:test1.sav'.
list.

POS1 POS2 SNORF POS3 NUM1 NUM2 NUM5 X Z

.00 .00 .00 .00 1.00 1.00 1.00 1.00 1.00
.00 .00 .00 .00 1.00 1.00 1.00 2.00 1.00
.00 .00 2.00 .00 1.00 1.00 1.00 3.00 1.00

.00 .00 2.00 .00 1.00 1.00 1.00 4.00 1.00

Number of cases read: 4 Number of cases listed: 4

Also note that the keyword "to" is used in the keep subcommand. Again, because it refers to variables that already exist in the data set,
to acts positionally.

Examples of the keyword "to" acting in a numerically consecutive manner

The do repeat and vector commands create new variables and act
in a numerically consecutive manner.

```
do repeat v = nu1 to nu5.
compute v = 0.
end repeat.
execute.
list.
```

The variables are listed in the following order:

LINE 1: POS1 POS2 SNORF POS3 NUM1 NUM2 NUM5

LINE 2: X Z NU1 NU2 NU3 NU4 NU5

POS1: .00 .00 .00 .00 1.00 1.00 1.00
X: 1.00 1.00 .00 .00 .00 .00 .00

POS1: .00 .00 .00 .00 1.00 1.00 1.00
X: 2.00 1.00 .00 .00 .00 .00 .00

POS1: .00 .00 2.00 .00 1.00 1.00 1.00
X: 3.00 1.00 .00 .00 .00 .00 .00

POS1: .00 .00 2.00 .00 1.00 1.00 1.00
X: 4.00 1.00 .00 .00 .00 .00 .00

Number of cases read: 4 Number of cases listed: 4

```
vector Anum = num1 to num5.
loop x = 1 to 3.
compute nm = Anum(x).
end loop.
execute.
list.
```

The variables are listed in the following order:

LINE 1: POS1 POS2 SNORF POS3 NUM1 NUM2 NUM5

LINE 2: X Z NU1 NU2 NU3 NU4 NU5

LINE 3: NM

POS1: .00 .00 .00 .00 1.00 1.00 1.00
X: 4.00 1.00 .00 .00 .00 .00 .00
NM: 1.00

POS1: .00 .00 .00 .00 1.00 1.00 1.00
X: 4.00 1.00 .00 .00 .00 .00 .00
NM: 1.00

POS1: .00 .00 2.00 .00 1.00 1.00 1.00
X: 4.00 1.00 .00 .00 .00 .00 .00
NM: 1.00

POS1: .00 .00 2.00 .00 1.00 1.00 1.00
X: 4.00 1.00 .00 .00 .00 .00 .00
NM: 1.00

Number of cases read: 4 Number of cases listed: 4

Examples of the keyword "to" acting in both a positionally and a numerically consecutive manner

There are some commands in SPSS that act in both a positionally and

a numerically consecutive manner. These commands make modifications to existing variables and create new variables with the modified values. Hence, the keyword to acts positionally in these commands when variables that already exist in the data set are being referenced. When new variables are being created, to acts numerically.

Example 1

```
autorecode pos1 to pos3 /into ppos1 to ppos4.
list.
```

The variables are listed in the following order:

LINE 1: POS1 POS2 SNORF POS3 NUM1 NUM2 NUM5

LINE 2: X Z NU1 NU2 NU3 NU4 NU5

LINE 3: NM PPOS1 PPOS2 PPOS3 PPOS4

POS1: .00 .00 .00 .00 1.00 1.00 1.00
X: 4.00 1.00 .00 .00 .00 .00 .00
NM: 1.00 1 1 1 1

POS1: .00 .00 .00 .00 1.00 1.00 1.00

X: 4.00 1.00 .00 .00 .00 .00 .00

NM: 1.00 1 1 1 1

POS1: .00 .00 2.00 .00 1.00 1.00 1.00

X: 4.00 1.00 .00 .00 .00 .00 .00

NM: 1.00 1 1 2 1

POS1: .00 .00 2.00 .00 1.00 1.00 1.00

X: 4.00 1.00 .00 .00 .00 .00 .00

NM: 1.00 1 1 2 1

Number of cases read: 4 Number of cases listed: 4

As you can see, all of the variables between pos1 and pos3 (including
snorf) were autorecoded. We needed to specify four names because
there were four variables between pos1 and pos3 (including pos1 and pos3).

Example 2

```
aggregate outfile 'c:test.sav'
/break snorf
/maxpos1 to maxpos2 = max(pos1 to pos2).
get file 'c:test.sav'.
```

list.

SNORF MAXPOS1 MAXPOS2

.00 .00 .00
2.00 .00 .00

Number of cases read: 2 Number of cases listed: 2
In the example above, maxpos1 and maxpos2 are variables that are created
by the aggregate command; hence, the keyword to is numerically consecutive. However, the
max function is operating on variables that already exist in the data set; hence, they must by positionally consecutive.

Example 3

get file 'c:to.sav'.
rename variables (pos1 to pos3 = p1 to p4).
list.

NUM1 NUM2 NUM5 P1 P2 P3 P4 X Z

1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 2.00 1.00
1.00 1.00 1.00 1.00 1.00 2.00 1.00 3.00 1.00
1.00 1.00 1.00 1.00 1.00 2.00 1.00 4.00 1.00

Number of cases read: 4 Number of cases listed: 4
In this example, we see again that the keyword to is acting positionally: we needed to give four names after the equals sign (p1,
p2, p3 and p4) because there are four variable between pos1 and
pos3 (including these variables).