4. More about SPSS syntax

Karl B Christensen
http://publicifsv.sund.ku.dk/~kach/SPSS

Reproducible research

Different steps in making the research reproducible

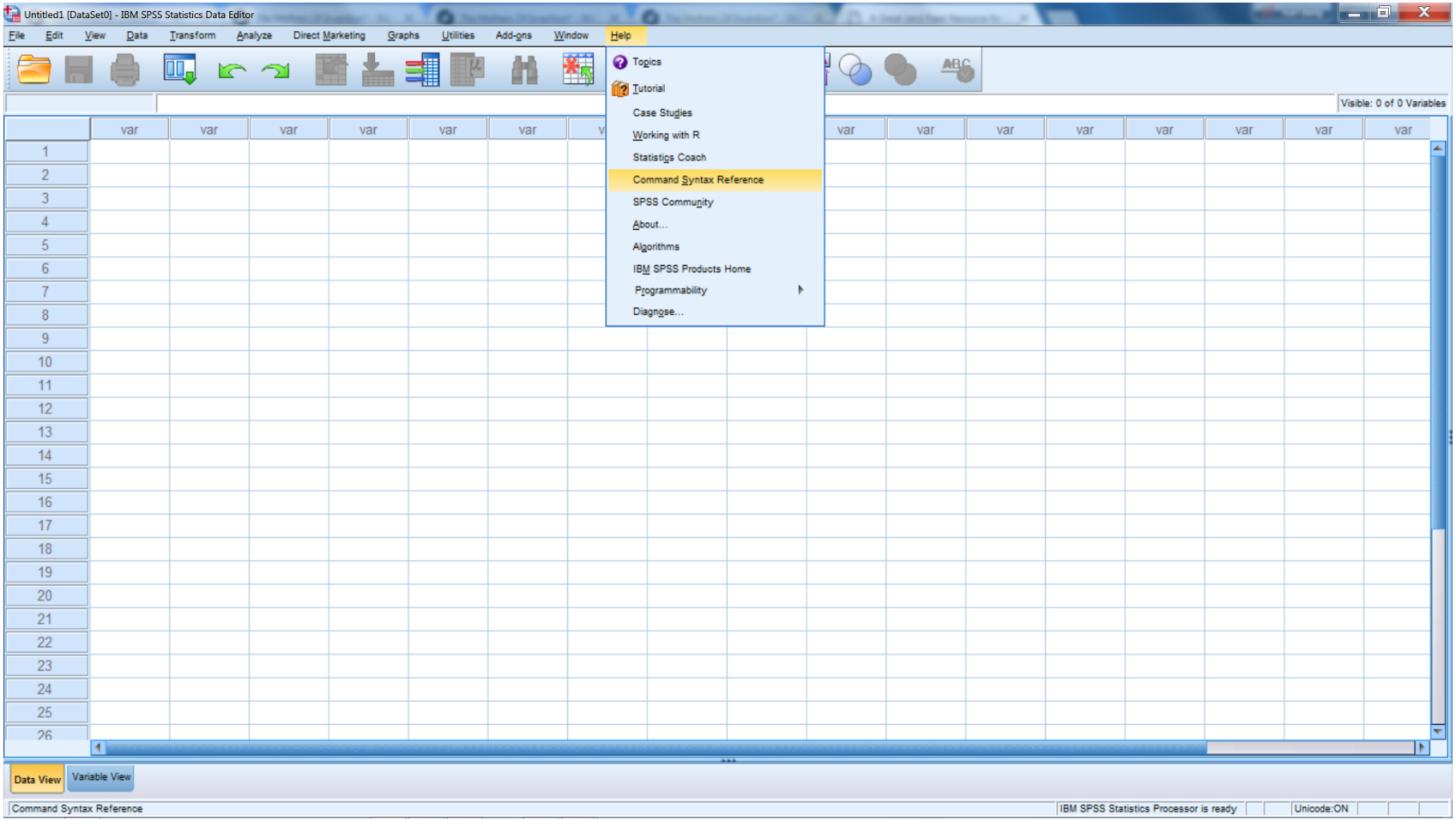
- Data handling, data bases, merging
- manipulating the data: setting up definitions and codes.
- statistical analyses (also involves data manipulation).

steps typically takes place in statistical software and can thus be

- documented
- reproduced
- shared

in collaborative research - need syntax for this.

Command Syntax Reference



Take control of your SPSS data set

Three variable definitions

- Variable Labels
- Value Labels
- Missing Data codes.

syntax is simple

Take control of your SPSS data set

For a data set with the variables Gender, Smoke, and Exercise, with the following definitions:

Gender	0=Male, 1=Female
Smoking	1=Never 2=Sometimes 3=Daily
Alcohol	1=Never 2=Sometimes 3=Daily

(for all variables: 999 = a user-defined missing value)

Take control of your SPSS data set

```
VARIABLE LABELS
GENDER 'Participant Gender'
SMOKE 'Does Participant Smoke Cigarettes?'
EXERCISE 'How Often Does Participant drink alcohol'.
```

Notice: all three in the same `VARIABLE LABELS` statement.
Remember a period at the end of the statement. This is required.

Take control of your SPSS data set

```
VALUE LABELS
GENDER 0 'Male' 1 'Female'
/SMOKE EXERCISE
1 'Never'
2 'Sometimes'
3 'Daily'.

MISSING VALUES
GENDER SMOKE EXERCISE (999).
```

Notice: all three have the same missing data code - include them all in the same statement.

Take control of your SPSS data set

Using point-and-click: 'Variable View' (define type, missing, measure (ordinal, nominal, scale). Can change these by double clicking (name, missing, type)

```
Data -> Define variable properties
```

Can define labels for ordinal and nominal data. Can define missing values

Reshaping data

Transposing data sets. Two types of transposing

- long to wide
- wide to long

typically used with longitudinal where different analyses requires either long or wide format.

Reshaping data

Wide format

id	age1	age2	age3	height1	height2	height3
1	9	10	11	130	140	148
2	9.5	10.23	10.78	120	125	130

Long format

id	age	height
1	9	130
1	10	140
1	11	148
2	9.5	120
2	10.23	125
2	10.78	130

Reshaping data

Wide to long

```
data -> restructure data -> 'restructure selected
                variables into cases'
```

- 'more than one' (2) (age and height)
- variables to be transposed has two levels, start with first
- `age -> age1,age2,age3`
- `height -> height1,height2,height3`
- use the arrow on the 'trans1' to change the name and to select 'trans2' for height.
- continue to 'index variables'
- continue to 'create one index variables'
- OK
- OK

Reshaping data

long to wide

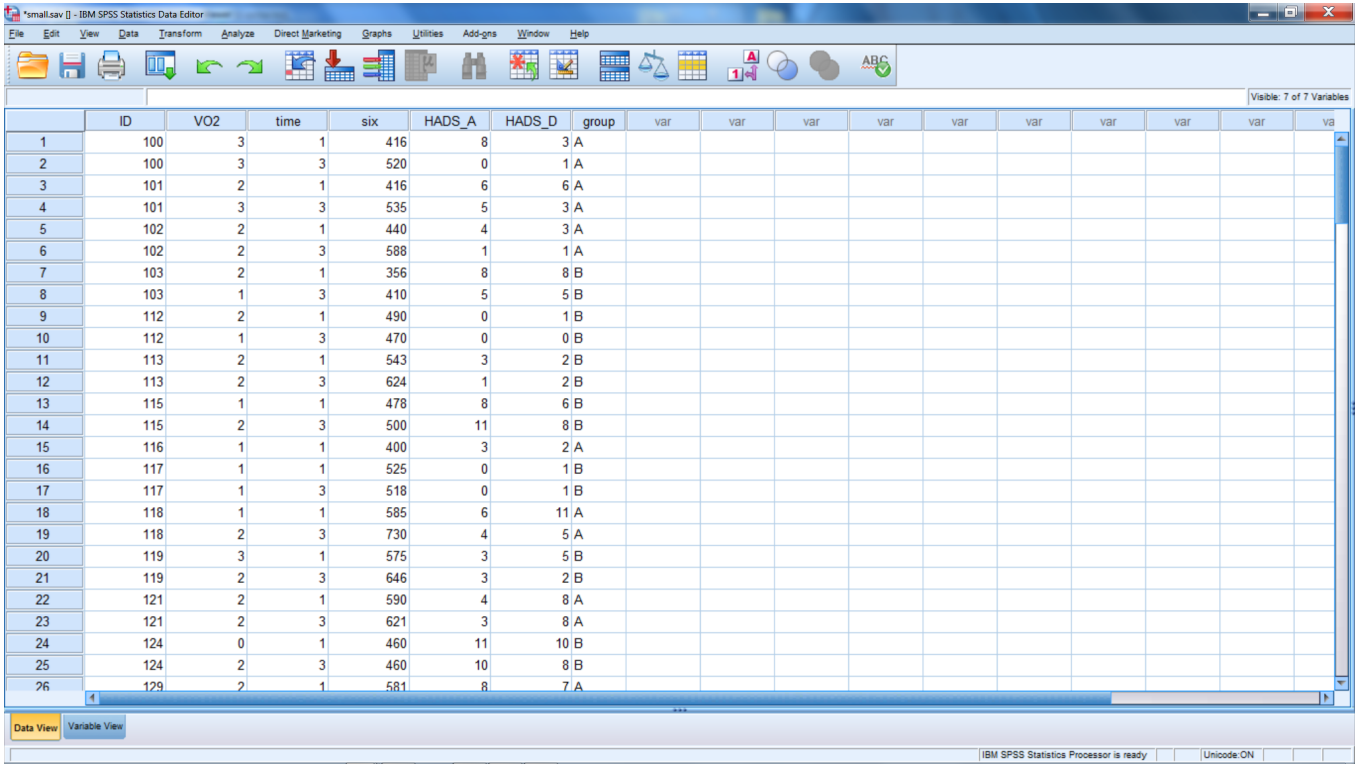
```
data -> restructure data -> 'restructure cases
            selected into variable'
```

- 'indentifier variables'='id' (next)
- yes or no to sorting
- 'order of new variables groups' check 'group by index'

Reshaping data using syntax

syntax

```
GET FILE='p:\small.sav'.
```



Reshaping data using syntax

syntax

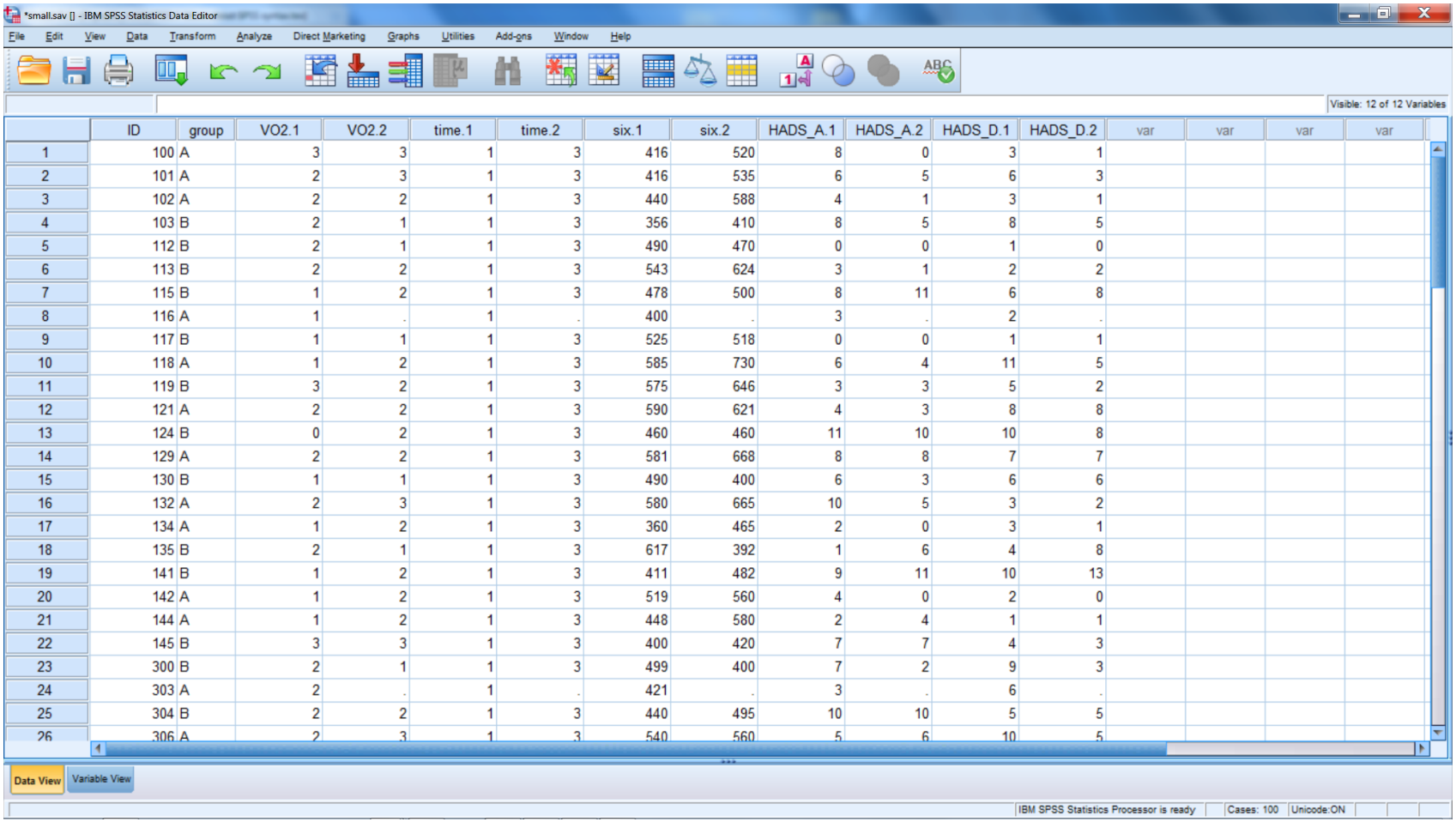
```
SORT CASES BY id .
CASESTOVARS
  /ID=id
  /GROUPBY=VARIABLE.
```

Reshaping data using syntax

Generated Variables

Name	Aerobic capacity (VO2)	1	VO2.1
		2	VO2.2
	time	1	time.1
		2	time.2
	6 minute walk distance (6MWD)	1	six.1
		2	six.2
	HADS_A	1	HADS_A.1
		2	HADS_A.2
	HADS_D	1	HADS_D.1
		2	HADS_D.2
Label	Aerobic capacity (VO2)	1	VO2.1: Aerobic capacity (VO2)
		2	VO2.2: Aerobic capacity (VO2)
	6 minute walk distance (6MWD)	1	six.1: 6 minute walk distance (6MWD)
		2	six.2: 6 minute walk distance (6MWD)

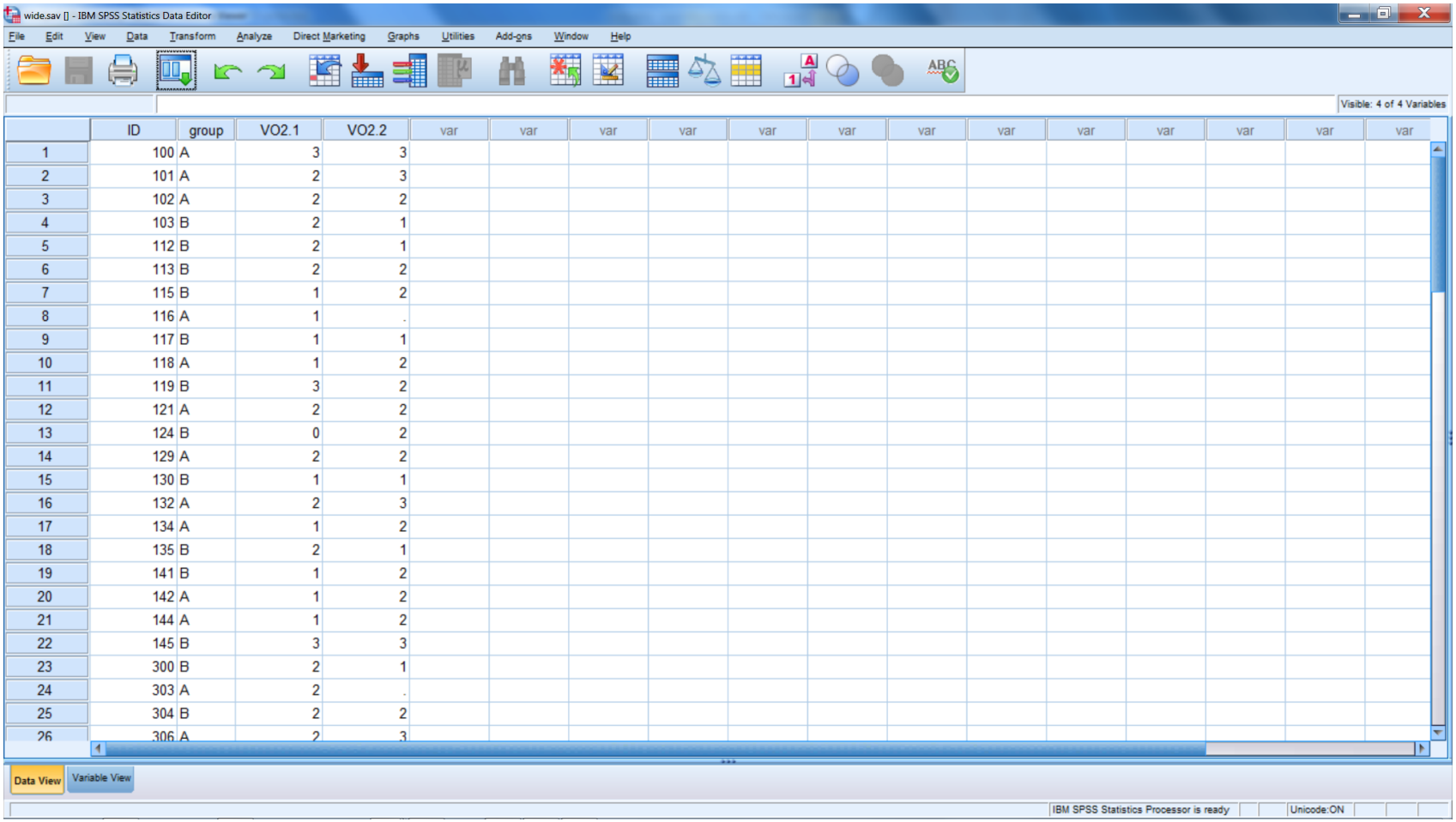
Reshaping data using syntax



	ID	group	VO2.1	VO2.2	time.1	time.2	six.1	six.2	HADS_A.1	HADS_A.2	HADS_D.1	HADS_D.2	var	var	var	var
1	100	A	3	3	1	3	416	520	8	0	3	1				
2	101	A	2	3	1	3	416	535	6	5	6	3				
3	102	A	2	2	1	3	440	588	4	1	3	1				
4	103	B	2	1	1	3	356	410	8	5	8	5				
5	112	B	2	1	1	3	490	470	0	0	1	0				
6	113	B	2	2	1	3	543	624	3	1	2	2				
7	115	B	1	2	1	3	478	500	8	11	6	8				
8	116	A	1	.	1	.	400	.	3	.	2	.				
9	117	B	1	1	1	3	525	518	0	0	1	1				
10	118	A	1	2	1	3	585	730	6	4	11	5				
11	119	B	3	2	1	3	575	646	3	3	5	2				
12	121	A	2	2	1	3	590	621	4	3	8	8				
13	124	B	0	2	1	3	460	460	11	10	10	8				
14	129	A	2	2	1	3	581	668	8	8	7	7				
15	130	B	1	1	1	3	490	400	6	3	6	6				
16	132	A	2	3	1	3	580	665	10	5	3	2				
17	134	A	1	2	1	3	360	465	2	0	3	1				
18	135	B	2	1	1	3	617	392	1	6	4	8				
19	141	B	1	2	1	3	411	482	9	11	10	13				
20	142	A	1	2	1	3	519	560	4	0	2	0				
21	144	A	1	2	1	3	448	580	2	4	1	1				
22	145	B	3	3	1	3	400	420	7	7	4	3				
23	300	B	2	1	1	3	499	400	7	2	9	3				
24	303	A	2	.	1	.	421	.	3	.	6	.				
25	304	B	2	2	1	3	440	495	10	10	5	5				
26	306	A	2	3	1	3	540	560	5	6	10	5				

Reshaping data using syntax - the other way around

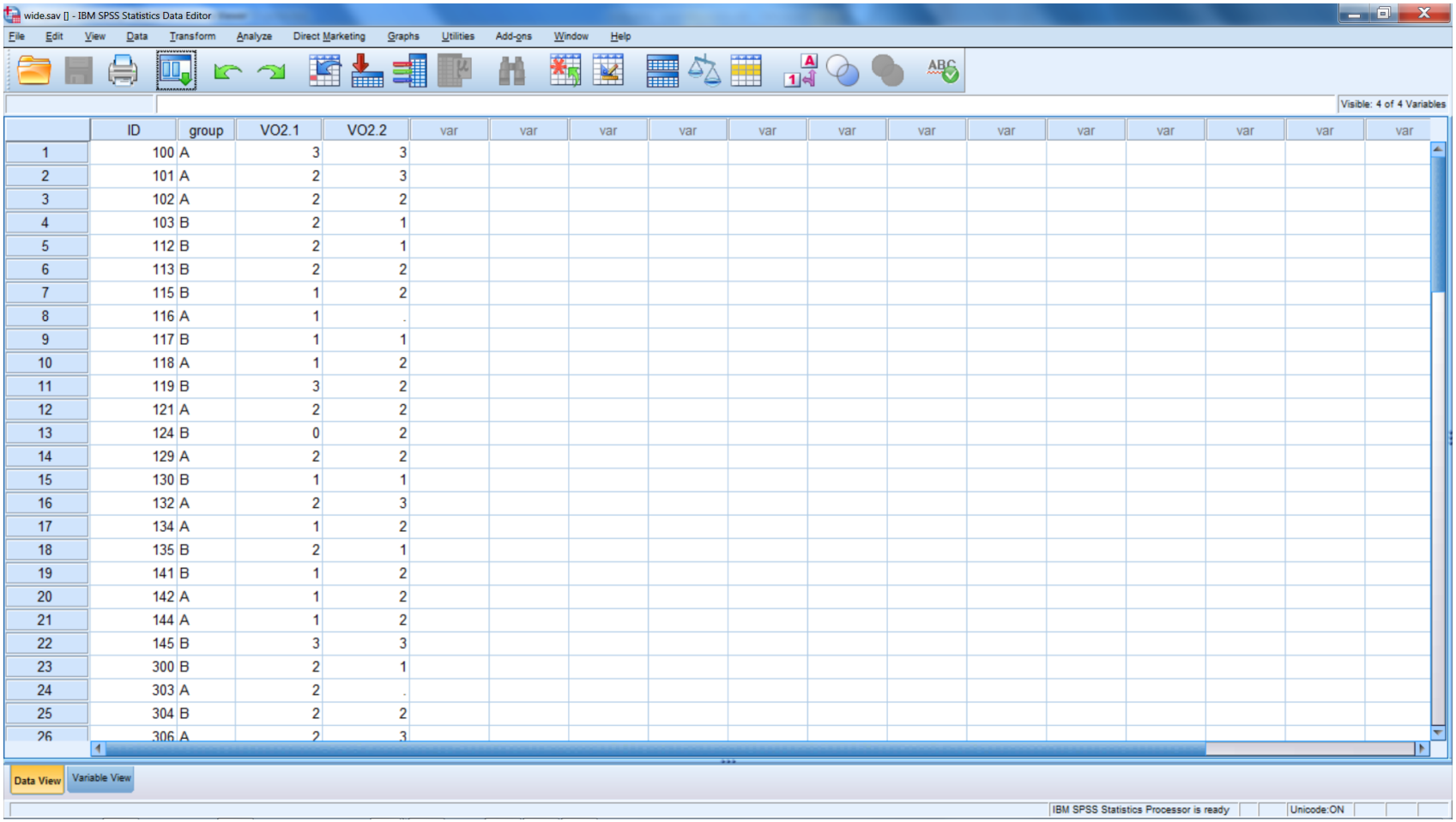
```
GET FILE='p:\wide.sav'.
```



	ID	group	VO2.1	VO2.2	var	var	var	var	var	var	var	var	var	var	var	var
1	100	A	3	3												
2	101	A	2	3												
3	102	A	2	2												
4	103	B	2	1												
5	112	B	2	1												
6	113	B	2	2												
7	115	B	1	2												
8	116	A	1	.												
9	117	B	1	1												
10	118	A	1	2												
11	119	B	3	2												
12	121	A	2	2												
13	124	B	0	2												
14	129	A	2	2												
15	130	B	1	1												
16	132	A	2	3												
17	134	A	1	2												
18	135	B	2	1												
19	141	B	1	2												
20	142	A	1	2												
21	144	A	1	2												
22	145	B	3	3												
23	300	B	2	1												
24	303	A	2	.												
25	304	B	2	2												
26	306	A	2	3												

Reshaping data using syntax - the other way around

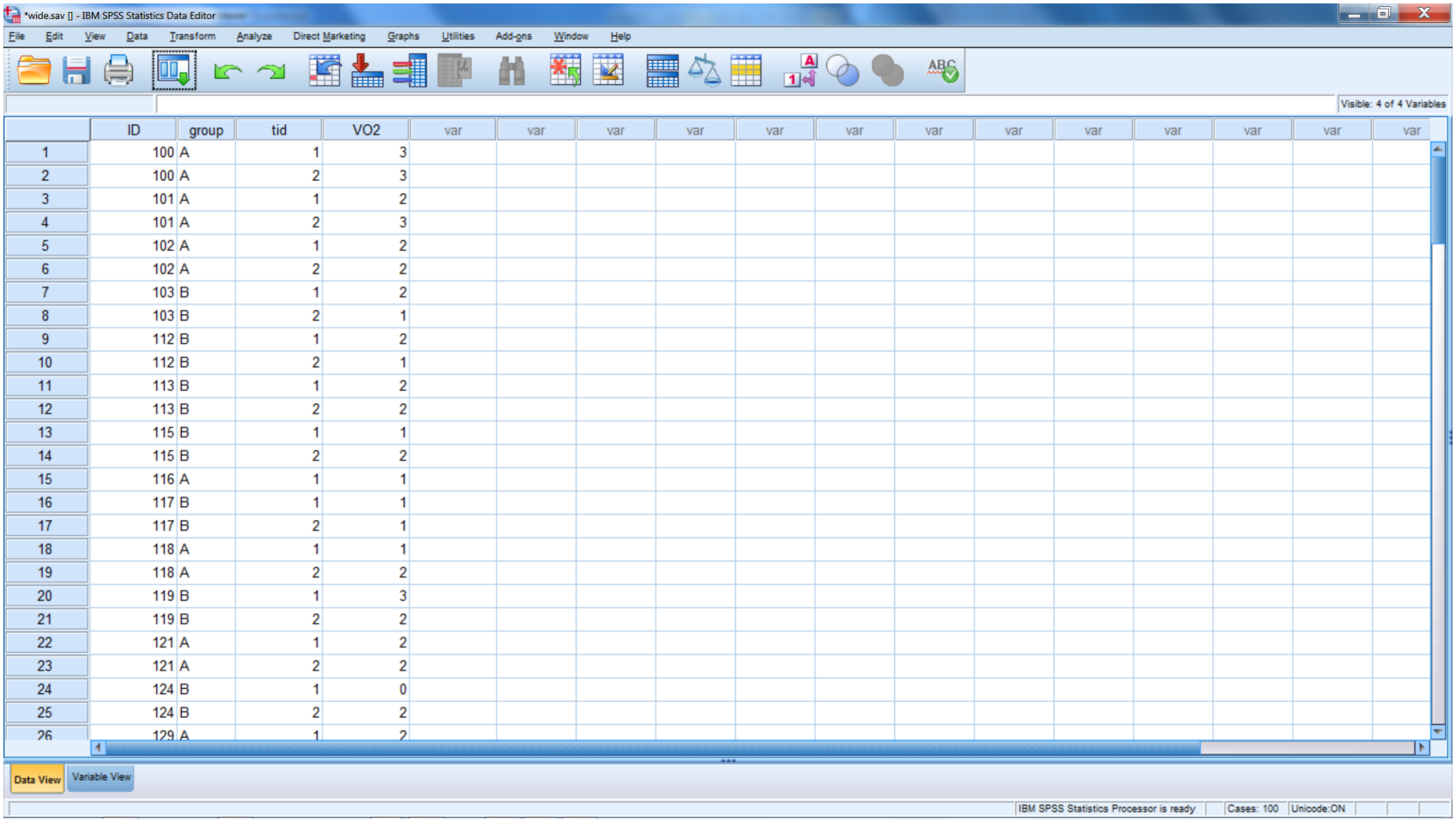
```
GET FILE='p:\wide.sav'.
```



	ID	group	VO2.1	VO2.2	var	var	var	var	var	var	var	var	var	var	var	var
1	100	A	3	3												
2	101	A	2	3												
3	102	A	2	2												
4	103	B	2	1												
5	112	B	2	1												
6	113	B	2	2												
7	115	B	1	2												
8	116	A	1	.												
9	117	B	1	1												
10	118	A	1	2												
11	119	B	3	2												
12	121	A	2	2												
13	124	B	0	2												
14	129	A	2	2												
15	130	B	1	1												
16	132	A	2	3												
17	134	A	1	2												
18	135	B	2	1												
19	141	B	1	2												
20	142	A	1	2												
21	144	A	1	2												
22	145	B	3	3												
23	300	B	2	1												
24	303	A	2	.												
25	304	B	2	2												
26	306	A	2	3												

Reshaping data using syntax - the other way around

```
VARSTOCASES
 /MAKE VO2 FROM VO2.1 VO2.2
 /INDEX   = tid.
LIST id group.
```



	ID	group	tid	VO2	var	var	var	var	var	var	var	var	var	var	var	var
1	100	A	1	3												
2	100	A	2	3												
3	101	A	1	2												
4	101	A	2	3												
5	102	A	1	2												
6	102	A	2	2												
7	103	B	1	2												
8	103	B	2	1												
9	112	B	1	2												
10	112	B	2	1												
11	113	B	1	2												
12	113	B	2	2												
13	115	B	1	1												
14	115	B	2	2												
15	116	A	1	1												
16	117	B	1	1												
17	117	B	2	1												
18	118	A	1	1												
19	118	A	2	2												
20	119	B	1	3												
21	119	B	2	2												
22	121	A	1	2												
23	121	A	2	2												
24	124	B	1	0												
25	124	B	2	2												
26	129	A	1	2												

Sorting of data (`split file`)- revisited

Obs	age	csex	fev1	pemax	bmi
:	:	m	:	:	:
:	:	m	:	:	:
:	:	m	:	:	:
:	:	f	:	:	:
:	:	f	:	:	:
:	:	f	:	:	:

Obs	age	csex	fev1	pemax	bmi
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:

Today: combining smaller data sets into larger ones

Appending data sets: ADD

More cases, same variables. Two data sets `hosp1.sav`

Obs	id	sex	bp	weight	bmi	treatment	hospital
1	1	1	110	75.8	24.5	0	herlev
2	2	2	125	89.8	26.8	0	herlev

and `hosp2.sav`

Obs	id	day	sex	bp	weight	teatment	hospital
1	3	11/11/97	2	131	57.8	1	gentofte
2	4	12/11/97	2	121	98.8	1	gentofte

put together using `ADD FILES`

```
ADD FILES FILE="p:\hosp1.sav"
/FILE="p:\hosp2.sav".
SAVE OUTFILE="p:\hosp.sav".
```

Result

Obs	id	sex	bp	weight	bmi	treatment	hospital	day	teatment
1	1	1	110	75.8	24.5	0	herlev	.	
2	2	2	125	89.8	26.8	0	herlev	.	
3	3	2	131	57.8	.		gentofte	11/11/97	1
4	4	2	121	98.8	.		gentofte	12/11/97	1

Note that missing values are generated for variables not present in all data sets.

Merging data set: MERGE

new variables, same cases. Often there is a key, say `id`, and all data sets must be merged by `id`. Data `quest.sav` concerns the same sample of patients

Obs	id	income	employed
1	3	35000	1
2	2	44000	1
3	5	25500	1

We want to merge this with the data set `hosp.sav`

Code for merging data sets

```
GET FILE = "p:\hosp.sav".
SORT CASE BY id.
SAVE OUTFILE = "p:\merge1.sav".

GET FILE = "p:\quest.sav".
SORT CASE BY id.
SAVE OUTFILE = "p:\merge2.sav".

MATCH FILES
/FILE="p:\merge1.sav"
/FILE="p:\merge2.sav"
/BY id.

SAVE OUTFILE = "p:\data.sav".
```

Result

Obs	id	sex	bp	weight	bmi	treat -ment	hospital	day	teat -ment	income	employed
1	1	1	110	75.8	24.5	0	herlev	.	.	.	.
2	2	2	125	89.8	26.8	0	herlev	.	.	44000	1
3	3	2	131	57.8	.	.	gentofte	11/11/97	1	35000	1
4	4	2	121	98.8	.	.	gentofte	12/11/97	1	.	.
5	5	.	.	.	.	.		.	.	25500	1

Note, that subject with `id=5` who is present only in `quest.sav` has missing values for variables not in that data set.

MERGE exercise

Look at data set `RCT.sav`. Compute the change scores

$$\Delta = (\text{VO}_2\text{max after intervention}) - (\text{baseline VO}_2\text{max})$$

for each subject. One (complicated) way of doing this:

1. Make two data sets: a baseline data set and a follow-up data set.
2. In the baseline data set with make a new variable `VO2base`.
3. In the follow-up data set with make a new variable `VO2foll`.
4. Merge these data sets so that information from the same patient are in the same row.
5. Calculate the change scores

This can also be done using `CASESTOVARS`