

## Lecture 21

1. Longitudinal data: long form and wide form
2. Reshaping data from wide to long
3. Reshaping data from long to wide
4. Proc Transpose
5. Writing SAS datasets out to CSV files or Excel spreadsheets
6. Reference SAS code

1

### Long form and wide form

Multiple measurements per subject can be displayed in 2 ways:

**Long form:** multiple observations per subject,  
one measurement (income) in each observation,  
with variables identifying subject ID and observation number (year)

Obs	family_ id	income	year
1	1	66483	1990
2	1	69146	1991
3	1	74643	1992
4	1	79783	1993
5	1	81710	1994
6	1	86143	1995
7	2	17510	1990
8	2	19484	1992
9	2	20979	1993
10	2	21268	1994

(Example data adapted from UCLA Academic Technology Services, [www.ats.ucla.edu/stat/](http://www.ats.ucla.edu/stat/))

2

**Wide form:** one observation per subject,  
 all measurements (income) in one observation,  
 with variables identifying subject ID,  
 and observation number (year) encoded in variable name

Obs	family_ id	income_ 1990	income_ 1991	income_ 1992	income_ 1993	income_ 1994	income_ 1995
1	1	66483	69146	74643	79783	81710	86143
2	2	17510	.	19484	20979	21268	22998
3	3	57947	62964	68717	70957	75198	75722
4	4	64831	71060	71918	72514	73100	74379
5	5	18904	19949	21335	22237	23829	23913
6	6	32057	34770	35834	37387	40899	42372
7	7	60551	64869	67983	70498	71253	75177
8	8	16553	18189	18349	19815	21739	22980
9	9	32611	33465	35961	36416	37183	40627
10	10	61379	66002	67936	70513	74405	76009

3

**Long form.** Required for graphing and analysis procedures

**Wide form.** All measurements from a subject are available because they are all in the same row.

Easy to find subject's maximum value, correlations between measurements, area under the curve (AUC), etc.

Usual format in spreadsheets.

Often necessary to reshape data: wide  $\rightarrow$  long, or long  $\rightarrow$  wide

4

## Reshaping data from wide to long

Obs	family_ id	income_ 1990	income_ 1991	income_ 1992	income_ 1993	income_ 1994	income_ 1995
1	1	66483	69146	74643	79783	81710	86143
2	2	17510	.	19484	20979	21268	22998
3	3	57947	62964	68717	70957	75198	75722

Make several observations from each one (*LSB §6.10*)

```
data long;
  set wide;
  array inc[6] income_1990 - income_1995;
  do j = 1 to 6;
    income = inc[j];
    year = j + 1989;
  output;
  end;
  keep family_id income year;
```

5

Result:

Obs	family_ id	income	year
1	1	66483	1990
2	1	69146	1991
3	1	74643	1992
4	1	79783	1993
5	1	81710	1994
6	1	86143	1995
7	2	17510	1990
8	2	.	1991
9	2	19484	1992
10	2	20979	1993

With additional arrays, add annual measurements of expenditure, debt status.

6

## Reshaping data from long to wide

Obs	family_ id	income	year
1	1	66483	1990
2	1	69146	1991
3	1	74643	1992
4	1	79783	1993
5	1	81710	1994
6	1	86143	1995

- Need to RETAIN values of income
- make income\_199x variables
- start over with each family
- output one observation per family

7

```
proc sort data=long;
  by family_id year;

data wide2;
  set long;
  by family_id ; to get first.family-ID and last.family-ID

  array inc[6] income_1990 - income_1995;
  retain income_1990 - income_1995;

  if (first.family_id=1) then do j = 1 to 6;
    inc[j] = . ; start all income variables at missing
  end;
```

8

```

if year=1990 then income_1990=income;
if year=1991 then income_1991=income;
if year=1992 then income_1992=income;
if year=1993 then income_1993=income;
if year=1994 then income_1994=income;
if year=1995 then income_1995=income;

if (last.family_id=1) then output;
keep family_id income_1990 - income_1995;

```

Obs	family_ id	income_ 1990	income_ 1991	income_ 1992	income_ 1993	income_ 1994	income_ 1995
1	1	66483	69146	74643	79783	81710	86143
2	2	17510	.	19484	20979	21268	22998
3	3	57947	62964	68717	70957	75198	75722

9

### Proc Transpose: reshaping data from long to wide (LSB §6.14)

#### Proc Transpose: reshaping data from long to wide

Obs	family_ id	income	year
1	1	66483	1990
2	1	69146	1991
3	1	74643	1992
4	1	79783	1993
5	1	81710	1994
6	1	86143	1995

Proc Transpose performs these steps:

- make income\_199x variables
- output one observation per family

This is the goal:

Obs	family_ id	income_ 1990	income_ 1991	income_ 1992	income_ 1993	income_ 1994	income_ 1995
1	1	66483	69146	74643	79783	81710	86143
2	2	17510	.	19484	20979	21268	22998

```
Proc Transpose data=long out=wide3 prefix =income_ ;
  ID year;
  VAR income;
  BY family_id;
```

**ID** variable values become names of variables in output data (wide)

**prefix** option makes variable-name “stem”—no quotes

**VAR** repeated measurements (will be transposed)

**BY** subject identifier: one observation for each level of BY variable(s)

11

Obs	family_ id	income_ _NAME_	income_ 1990	income_ 1991	income_ 1992	income_ 1993	income_ 1994	income_ 1995
1	1	income	66483	69146	74643	79783	81710	86143
2	2	income	17510	.	19484	20979	21268	22998
3	3	income	57947	62964	68717	70957	75198	75722
4	4	income	64831	71060	71918	72514	73100	74379
5	5	income	18904	19949	21335	22237	23829	23913
6	6	income	32057	34770	35834	37387	40899	42372
7	7	income	60551	64869	67983	70498	71253	75177
8	8	income	16553	18189	18349	19815	21739	22980
9	9	income	32611	33465	35961	36416	37183	40627
10	10	income	61379	66002	67936	70513	74405	76009

All the other variables in the data set are gone.

Proc Transpose drops all variables not created by VAR and ID,  
and not in the BY statement

12

## Proc Transpose: keeping additional variables

We would like to keep family ID *and study group* in the wide form for the income data.

Obs	family_ id	income	year	expenses	debt	group
1	1	66483	1990	49804	no	A
2	1	69146	1991	65634	no	A
3	1	74643	1992	61820	no	A
4	1	79783	1993	68387	no	A
5	1	81710	1994	85504	yes	A
6	1	86143	1995	75640	no	A
7	2	17510	1990	21609	yes	B
8	2	19484	1992	18180	no	B
9	2	20979	1993	22985	yes	B
10	2	21268	1994	11097	no	B
11	2	22998	1995	21768	no	B

13

Include `group` in the `BY` statement of the `sort` and `Transpose`:

```
proc sort data=a;
```

```
  BY family_id group;
```

```
Proc Transpose data=A out=B prefix=income_;
```

```
  ID year;
```

```
  VAR income;
```

```
  BY family_id group;
```

14

Obs	family_ id	group	_NAME_	income_ 1990	income_ 1991	income_ 1992	income_ 1993	income_ 1994	income_ 1995
1	1	A	income	66483	69146	74643	79783	81710	86143
2	2	B	income	17510	.	19484	20979	21268	22998
3	3	A	income	57947	62964	68717	70957	75198	75722
4	4	A	income	64831	71060	71918	72514	73100	74379
5	5	A	income	18904	19949	21335	22237	23829	23913
6	6	A	income	32057	34770	35834	37387	40899	42372
7	7	A	income	60551	64869	67983	70498	71253	75177
8	8	A	income	16553	18189	18349	19815	21739	22980
9	9	A	income	32611	33465	35961	36416	37183	40627
10	10	A	income	61379	66002	67936	70513	74405	76009

We have kept group; expenses and debt are gone.

### Proc Transpose with more than one outcome

Suppose we want income, expenses, and debt all in the wide form.

Obs	family_ id	income	year	expenses	debt	group
1	1	66483	1990	49804	no	A
2	1	69146	1991	65634	no	A
3	1	74643	1992	61820	no	A
4	1	79783	1993	68387	no	A
5	1	81710	1994	85504	yes	A
6	1	86143	1995	75640	no	A

To transpose values of income, expenses, and debt:

Transpose just one variable in each Proc Transpose.

Make 3 long datasets, one for each variable, and merge.

```
Proc Transpose data=a out=income prefix=income_ ;
  ID year;
  VAR income ;
  BY family_id;
```

```
Proc Transpose data=a out=expenses prefix=expense_ ;
  ID year;
  VAR expenses ;
  BY family_id;
```

```
Proc Transpose data=a out=debt prefix=debt_ ;
  ID year;
  VAR debt ;
  BY family_id;
```

17

```
data wide4;
  merge income expenses debt;
  by family_id;
```

Obs	family_		income_	income_	income_	income_	income_
	id	_NAME_	1990	1991	1992	1993	1994
1	1	debt	66483	69146	74643	79783	81710
2	2	debt	17510	.	19484	20979	21268

  

Obs	income_	expense_	expense_	expense_	expense_	expense_
	1995	1990	1991	1992	1993	1994
1	86143	49804	65634	61820	68387	85504
2	22998	21609	.	18180	22985	11097

  

Obs	expense_	debt_	debt_	debt_	debt_	debt_	debt_
	1995	1990	1991	1992	1993	1994	1995
1	75640	no	no	no	no	yes	no
2	21768	yes		no	yes	no	no

18

Proc Transpose is complicated and things can go wrong in unexpected ways.

Use Proc Print to check carefully that numbers are in the right places.

References (course website):

Zirbel (2009) *Learn the Basics of Proc Transpose*

Tilanus (2007) *Turning the data around: Proc Transpose*

### **Writing data sets to CSV or Excel spreadsheets**

Proc Export will write a SAS dataset out to a CSV file or Excel spreadsheet.

To write out a date as a date (instead of number of days since 1/1/1960), format the variable in the SAS dataset.

```
Proc Export data = A  outfile =  
    "C:\Documents and Settings\Administrator\Desktop\To_R.csv"  
    DBMS=CSV replace;
```

```
Proc Export data=A  outfile =  
    "C:\Documents and Settings\Administrator\Desktop\To_excel.xls"  
    DBMS=excel replace;
```

## **SAS reference program**

It's hard to remember all the details of SAS procedures and functions, and it takes time to look them up.

Keep a reference program with examples and pieces of code, with notes on how things work. My reference program (on course website) includes:

- path to my desktop
- SGplot code with options
- back-transformation code for Proc GLM and Proc Mixed
- Code to read CSV, with informants
- data step options