

Converting a dataset of “stacked” form into “split” form

Consider the following dataset, summarizing the 3-year height-growth of 5 trees that were applied some treatment:

tree_no	treatment	year	ht
1	A	1	23
1	A	2	33
1	A	3	40
2	A	1	25
2	A	2	34
3	A	1	31
3	A	2	38
3	A	3	42
4	B	1	19
4	B	2	28
4	B	3	38
5	B	1	21
5	B	2	26
5	B	3	35

But for subsequent analysis, the data needs to be rearranged to look like this:

tree_no	treatment	ht_year1	ht_year2	ht_year3
1	A	23	33	40
2	A	25	34	.
3	A	31	38	42
4	B	19	28	38
5	B	21	26	35

The arrangement of first dataset is sometimes called stacked, long or univariate (with respect to height). The second is called split, wide or multivariate (with respect to height).

Write a SAS program to convert the first stacked dataset into the second split one. Bonus marks if you can write additional code to convert it back again.

Solution #1 by Gord Nigh (BC Ministry of Forests and Range)

```
data trees;
input tree_no treatment $ year ht;
cards;
1 A 1 23
1 A 2 33
1 A 3 40
2 A 1 25
2 A 2 34
3 A 1 31
3 A 2 38
3 A 3 42
4 B 1 19
4 B 2 28
4 B 3 38
5 B 1 21
5 B 2 26
5 B 3 35
;
```

```
proc print; run;
```

```
data trees1;
set trees;
if year ne 1 then delete;
ht_year1 = ht;
```

```
data trees2;
set trees;
if year ne 2 then delete;
ht_year2 = ht;
```

```
data trees3;
set trees;
if year ne 3 then delete;
ht_year3 = ht;
```

```
data trees;
merge trees1 trees2 trees3;
by tree_no;
drop ht;
```

```
proc print; run;
```

```
data trees;
set trees;
year = 1; ht = ht_year1; output;
year = 2; ht = ht_year2; output;
year = 3; ht = ht_year3; output;
```

```
data trees;
set trees;
if ht = . then delete;
drop ht_year1 ht_year2 ht_year3;
```

Solutions to the Open Problem. From the SUAVe meeting of October 7, 2008

```
proc print; run;
```

```
quit;
```

Solution #2 by Gary Koett (Government of BC)

```
/* Converting from Stacked to Split */

/* Set print line length to 80 */
options linesize=80 nodate;

/* Read in stacked data */
data STACKED;
  input TREE_NO TREATMENT $ YEAR HT;
  cards;
1 A 1 23
1 A 2 33
1 A 3 40
2 A 1 25
2 A 2 34
3 A 1 31
3 A 2 38
3 A 3 42
4 B 1 19
4 B 2 28
4 B 3 38
5 B 1 21
5 B 2 26
5 B 3 35
;

/* Sort for summary */
proc sort data=STACKED;
  by TREE_NO TREATMENT;

/* Split by year (3 max) */
data SPLIT (drop=YEAR HT);
  set STACKED;
  by TREE_NO TREATMENT;

  retain HT_YEAR1 HT_YEAR2 HT_YEAR3;

  /* Set values to missing initially */
  if first.TREE_NO & first.TREATMENT
  then do;
    HT_YEAR1 = .;
    HT_YEAR2 = .;
    HT_YEAR3 = .;
  end;

/* Split up years */
select (YEAR);
  when (1) HT_YEAR1 = HT;
  when (2) HT_YEAR2 = HT;
  when (3) HT_YEAR3 = HT;
otherwise do;
  file print; /* Output to SASLIST */
  put 'ERROR: Invalid year value ' YEAR 'encountered';
  abort abend 8; /* Terminate program (U0008 abend) */
end;
```

Solutions to the Open Problem. From the SUAVE meeting of October 7, 2008

```
end; /* SELECT (YEAR) */

if last.TREE_NO & last.TREATMENT then output;

/* Print out results */
title Tree Data (Split by Year);
proc print data=SPLIT noobs; run;

/* Converting from Split to Stacked */

/* Convert tree data from split to stacked using a macro */
data STACKED2 (drop=HT_YEAR1 HT_YEAR2 HT_YEAR3);
set SPLIT;

/* Define macro */
%macro STACK;

/* Stack data (loop through years) */
%do I = 1 %to 3;
  if HT_YEAR&I ^= .
  then do;
    YEAR = &I;
    HT = HT_YEAR&I;
    output;
  end;
%end;
%mend STACK;

/* Execute macro */
%STACK;

/* Print out results */
title Tree Data (Stacked);
proc print data=STACKED2 noobs;run;

quit;
```

Solution #3 by Aijun Yang (BC Ministry of Health)

```
data test;
input tree_no treatment $1. year ht;
datalines;
1 A 1 23
1 A 2 33
1 A 3 40
2 A 1 25
2 A 2 34
3 A 1 31
3 A 2 38
3 A 3 42
4 B 1 19
4 B 2 28
4 B 3 38
5 B 1 21
5 B 2 26
5 B 3 35
;
run;

proc sort data=test;by tree_no treatment year;run;

proc transpose data=test out=tran_test (drop=_name_) prefix=ht_year;
by tree_no treatment;
var ht;
id year;
run;

*covert back;
proc sql noprint;
select max(year) into:maxyear from test;
quit;

%let maxyear=%eval(&maxyear+0);

data test_back;
retain tree_no treatment;
array old[&maxyear.] ht_year1 - ht_year&maxyear.;
set tran_test;
do i=1 to &maxyear.;
  if old[i]^=. then do
    year=i;
    ht=old[i];
    output;
  end;
end;
drop i;
keep tree_no treatment year ht;

proc sort;by tree_no treatment year;
run;

quit;
```

Solution #4 by Mike Atkinson (Acko Systems)

* Create stacked dataset as starting point;

```
data stacked;
  infile cards4;
  input tree_no treatment $ year ht;
cards4;
1 A 1 23
1 A 2 33
1 A 3 40
2 A 1 25
2 A 2 34
3 A 1 31
3 A 2 38
3 A 3 42
4 B 1 19
4 B 2 28
4 B 3 38
5 B 1 21
5 B 2 26
5 B 3 35
;
run;
```

* Transpose to split format;

* Note: variable `_name_` is not dropped, since it facilitates going back the other way;

```
proc transpose data=stacked
  out=split prefix=ht_year;
  by tree_no treatment;
  id year;
  var ht;
run;
```

* Transpose back into stacked format;

```
proc transpose data=split
  out=stacked_2;
  by tree_no treatment;
  var ht_year1-ht_year3;
run;
```

* Get back the year variable, and drop observation(s) with missing values;

```
data stacked_3 (drop=_name_);
  set stacked_2;
  where not missing(ht);
  year = 0 + substr(_name_, 8, 1);
run;
```

Solutions to the Open Problem. From the SUAVe meeting of October 7, 2008

```
* Confirm that it matches the original dataset (it does);
```

```
proc compare data=stacked compare=stacked_3;
```

```
run;
```

```
quit;
```