Tweaking your Tables

Suppressing superfluous subtotals in PROC TABULATE
**Tweaking your Tables**

**Steve Cavill**

**Infoclarity**

---

**Introduction**

PROC TABULATE is a great tool for generating cross tab style reports. It’s very flexible but has a few little limitations. One is suppressing superfluous subtotals. The ALL keyword creates a total or subtotal for the categories in one dimension. However, if there is only one category in the dimension, the subtotal is still shown, which is really just repeating the detail line again. This can look a bit confusing in the final output. This talk demonstrates a method to suppress those superfluous totals by saving the output from PROC TABULATE using the OUT= option. That data set is then reprocessed to remove the undesirable totals using the _TYPE_ variable which identifies the total rows. PROC TABULATE is then run again against the reprocessed data set to create the final table.

---

The technique highlights the flexibility of the SAS® programming language to get exactly the output you want.

---

**Example PROC TABULATE output**

Example uses subset of SASHELP.CARS data set

```sas
proc tabulate data=data format=6.;
class origin make type ;
var msrp;
tables origin *(make*(type all) all) all='Grand Total' ,n msrp*mean*f=dollar7.;
run;
```

---

**Method – Step 1 send output to a table**

- Send output to a table:
  ```sas
  proc tabulate data=data
  out=table…
  _TYPE_;
  • 111 = detail
  • 110 = Make total
  • 100 = Origin total
  • 000 = Grand total
  • We want to delete the marked rows
  ```

---

**Method – Step 2 remove the subtotals we don’t want**

- Which rows to delete?
  - _type_=110 (Make total) and one type per make, OR
  - _type_=100 (Origin total) and one make per origin

- Identify rows rows to delete
  ```sas
  proc sql;
  create table make_suppress as
  select make,count (distinct type) as freq
  from data
  group by make
  having count (distinct type)=1 /* only one type within the make*/
  create table origin_suppress as
  select origin,count (distinct make) as freq
  from data
  group by origin
  having count (distinct make)=1 /* only one make in the origin */
  /* now delete the rows */
  proc sql;
  delete from table where _type_='110'
  and make in (select make from make_suppress);
  /* will delete Jeep and Isuzu totals */
  delete from table where _type_='100'
  and origin in (select origin from origin_suppress);
  /* will delete Asia subtotal */
  ```

---

**Desired output:**

- The extra total lines in Figure 1 are superfluous
- Suppress them and add data specific subtotal headings, as in Fig 2

---

**Figure 1: Standard output**

---

**Figure 2: Modified output**

---

**Example PROC TABULATE output**

```sas
proc tabulate data=data format=6.;
class origin make type ;
var msrp;
tables origin *(make*(type all) all) all='Grand Total' ,n msrp*mean*f=dollar7.;
run;
```
Tweaking your Tables
Steve Cavill
Infoclarity

**Method – Step 3 Make data specific headings**

```
data table;
  length origin make type $20; /* increase label width*/
  set table;
  /* first change the total labels (which are blank in the output data set) */
  if _type_='000' then origin='Grand Total'; /*grand total*/
  /*origin total - include the value of Origin on the total line */
  if _type_='100' then make=catx(' ',origin,'Total') ;
  /*make total - include the value of Make on the total line */
  if _type_='110' then type=catx(' ',make,'Total') ;
run;
```

**Desired output:**

```
proc tabulate data=table format=6.;
class origin make type /
order=data missing;
var N msrp_mean;
/* N and msrp_mean were calculated 
in the previous tabulate so just show it */
tables origin*make*type,
      N*sum
      msrp_mean*sum*f=dollar7.
;
Run;
```

- **Input data set is our summary table**
- **Order=data** – don’t re-sort data, it’s in the order we want with totals at the bottom
- **Missing keyword** is required because subtotal subcategories are blank
- **Msrp is now msrp_mean**
- **All keywords removed – table already has subtotals**
- `<var>*sum` – data has already been summarised, so just display the value (sum of itself)

**Conclusion**

- Most reporting procedures in SAS have OUT= option
- If you don’t like the standard output, you can change it!
- Easy to generalize the example above as a macro
- Source code is in the conference proceedings
SAS® GLOBAL FORUM 2018

April 8 - 11 | Denver, CO
Colorado Convention Center

#SASGF