

Tabulating Fractional Observations

Phil Busby, Live Data Systems, Inc. Apex, NC

Abstract:

This paper presents an advanced coding technique to make Proc TABULATE display cross-tabulations of "partial" observations. Each observation has a numeric variable with a value between zero and one representing an employee's full-time equivalent (FTE) as a university employee. The HR user sees the total FTE count for all of the employees extracted by his selection criteria.

The Application

The Human Resources division of a major university in Chapel Hill, NC has implemented a tool to quickly generate and display cross-tabulations of students and/or employees. Using a SAS/INTRNET-generated web page, the user selects a maximum of two row and two column class variables from a predefined set of selection criteria, such as age range, sex, race, school, department, etc. The selection fields correspond to variables in the university's database that are used in a dynamically-generated WHERE clause.

Here is what the user sees, with a Department selection off to the right:



HR Table / Extract Builder

Select subset of organization - may select multiple values from one drop-down

<u>Vice Chancellor</u>	<u>School</u>	<u>Level3</u>
not selected All University Chancellor Exec VC & Provost	not selected Chancellor Athletics Provost	not selected 0000-Other 2201-Office of the Chancellor 2202-UNC-CH Fdn Reporting

Options

&dept_option home appt	&table_option head_count fte
Dept for table: ☒ Home ☐ Appointing (EPA only)	Generate table for: ☒ Head Count ☐ FTE
Report Option: ☒ Table ☐ Extract	Output Format: ☒ HTML ☐ PDF ☐ Excel
&rept_option table extract	&output_option html pdf excel

And there are other selection screens to refine the request. The selection values can be as complex as the user wants. The resulting subset is described in the TITLE statements. Here is a complex WHERE clause result:

UNC Data Warehouse Project

HR Table/Extract Builder

Full/Part

Current Data

((Citizenship = Non-resident) or (Citizenship = Resident)) and ((FOC =
 Apprentice) or (FOC = Clerical)
 or (FOC = Executive) or (FOC = Faculty) or (FOC = Professional)) and
 (Perm/Temp = Permanent) and ((Race
 = African-American) or (Race = American Indian) or (Race = Asian) or
 (Race = Hispanic)) and Payroll Status
 = Active and ((Age Range = 0-19) or (Age Range = 20-29) or (Age Range =
 30-39) or (Age Range = 40-49) or
 (Age Range = 50-59))

Full/Part	Total
Full-Time	
Count	Count
53	53

The FTE variable's fractional value chops the part-timers down to a "fractional" observation.

For each table requested, the user also specifies the measure for each cell of the cross-tabulation: head count or FTE. For example, a professor who works only 3 days out of a 5-day week will have a value of .6 for his FTE. When the FTE option is chosen, the system adds the FTE variable to the list of variables extracted from the database. FTE is a numeric value between 0 and 1. It is known that the values are all expressed as 2-decimal fractions. But Proc TABULATE simply counts observations. When FTEs are wanted, the system must tell Proc TABULATE to count each observation as a partial observation depending on the observation's FTE value.

In the building of the list of just the needed variables, the FTE variable is appended to the string:

```
%* build &hrvarlst HR variable list for tables, separated by commas;
%if "&hrvarlst" = " " or "&hrvarlst" = "" %then
    %let hrvarlst = payroll_status_desc, curr_ind_hr,
                    new_status, ssn, spec_payroll_ind, &dept_var;
%else
    %let hrvarlst = &hrvarlst., payroll_status_desc, curr_ind_hr,
                    new_status, ssn, spec_payroll_ind, &dept_var;
%if %upcase(&measure_option) = FTE %then %do;
    %let hrvarlst = &hrvarlst., fte;
%end;
```

The selected observations are extracted from the database into the Wanted data set containing just the needed variables. The first step in counting FTEs is to multiply the FTE value by 10000 to create a temporary variable FTE100. Next, we create a format to display the result:

```
Proc Format;
    Picture hundred
        LOW-HIGH = '000,000,009.99'
        (mult=.01 fill=' ');
Run;
```

As you can see in the code below, we must check for the special case where all FTE values are zero, which Proc TABULATE does not handle well. Then, on the Proc TABULATE

statement, the custom format Hundred is specified when FTE is desired. Finally, the FREQ statement is generated for FTEs using our temporary variable FTE100 that contains the fractional FTE value for the observation multiplied by 10,000. Proc TABULATE still works only with integers, but when the results are displayed for each cell, the value is divided by 100 by the format, and then displayed with an explicit decimal point to the left of the units and tens digits, making it look like a fractional value of hundredths.

```
%let somefte=Maybe;
%if %upcase(&measure_option) = FTE %then %do;
  proc format;
    picture hundred
      LOW-HIGH = '000,000,009.99' (mult=.01 fill=' ');
  run;
  data hr_extract;
    retain some 0;
    set hr_extract end=eof;
    head = 1;
    if fte > 0 then do;
      call symput('somefte','YES');
      some = some + 1;
      fte100 = fte * 10000;
    end;
    if eof and some = 0 then call symput('somefte', 'No');
    ...

proc tabulate data=hr_extract missing
  %if %upcase(&measure_option) = FTE %then %do;
    format=hundred.
  %end;
  %else %do;
    format=comma.
  %end;
  %if &measure_option = head_count %then %do;
    %* just count observations (Tabulate default);
  %end;
  %else %if %upcase(&measure_option) = FTE %then %do;
    %* Multiply by full-time equivalent value times 10000;
    freq fte100;
  %end;
  ...
```

SAS® is a registered trademark of SAS Institute Inc., Cary, NC USA.

Contact:

Phil Busby
Live Data Systems, Inc.
342 Calico Drive
Apex, North Carolina 27523 USA
919/523-0289
fransioli@hotmail.com
www.cdbaby.com/fransioli