

PROC TABULATE

This procedures print out descriptive simple statistics for numeric variables in a table, each table value is displayed in cell controlled by row and columns.

Syntax:-

```
Proc Tabulate<options>;  
Class variable list;  
Var analysis variable list;  
By variable list;  
Keylabel key=newname  
Classlev Variable<s>/<options>  
Table table specifications/<table options>;
```

Class statement

Identifies class variables for the table. Class variables determine the categories that PROC TABULATE uses to calculate statistics.

Var statement

Identifies analysis variables for the table. Analysis variables contain values for which you want to compute statistics.

By statement

Identifies By variables for the table. By variables determine the groups that PROC TABULATE uses to calculate statistics on each **By** group into separate output.

Table statement

Defines the table that you want PROC TABULATE to produce. **You must specify at least one TABLE statement.** In the TABLE statement you specify page-expressions, row-expressions, and column-expressions.

Keylabel statement

Useful to apply labels for statistical keywords.

Statistical keywords are **RANGE CSS SKEWNESS CV STDDEV KURTOSIS LCLM SUM MAX MEAN MIN USS N VAR NMISS**

Label statement

Applies labels for existing variables.

Classlev statement

Applies styles for classification levels in class variables.

Examples:-

```
Data project;  
Infile datalines;  
Input employee$ week$9. phase$ hours;  
Datalines;  
Chen 11SEP2010 Analysis 8  
Chen 11SEP2010 Analysis 7  
Chen 11SEP2010 Coding 2.5  
Chen 11SEP2010 Testing 8  
Chen 11SEP2010 Coding 8.5  
Chen 11SEP2010 Testing 6  
Chen 11SEP2010 Coding .  
Stewart 11SEP2010 Coding 8  
Stewart 11SEP2010 Testing 4.5  
Stewart 11SEP2010 Coding 4.5  
Stewart 11SEP2010 Coding 10.5  
Stewart 11SEP2010 Testing 10  
;  
Run;
```

Proc Tabulate;

```
Class employee week phase;  
Var hours;  
Table sum*hours;  
Title 'Summary of Hours';  
Run;
```

Options:-

Data= Dataset

Specify the input data set, Otherwise system consider recent created dataset.

```
Proc Tabulate Data=project;  
Class employee week phase;  
Var hours;  
Table sum*hours;  
Title 'Summary of Hours';  
Run;
```

Out= Dataset

Specify the output data set

The number of observations in the output data set depends on the number of categories of data that are used in the tables and the number of subtables that are generated. The output data set contains these variables (in this order):

By variables

Variables that are listed in the BY statement.

Class variables

Variables that are listed in the CLASS statement.

TYPE

A character variable that shows which combination of class variables produced the summary statistics in that observation. Each position in _TYPE_ represents one variable in the CLASS statement. If that variable is in the category that produced the statistic, then the position contains a 1; if it is not, then the position contains a 0. In simple PROC TABULATE steps that do not use the universal class variable ALL, all values of _TYPE_ contain only 1s because the only categories that are being considered involve all class variables. If you use the variable ALL, then your tables will contain data for categories that do not include all the class variables, and positions of _TYPE_ will, therefore, include both 1s and 0s.

PAGE

The logical page that contains the observation.

TABLE

The number of the table that contains the observation.

Statistics

Statistics that are calculated for each observation in the data set.

	employee	week	Type of Observation _Type_	Page for Observation _Page_	Table for Observation _Table_	N	hours_Sum Statistics
1	Chen	11SEP2010	11	1	1	7	.
2	Stewart	11SEP2010	11	1	1	5	.
3			00	1	1	.	77.5

Proc Tabulate Data= project Out=Project2;

Class employee week phase;

Var hours;

Table employee*week sum*hours;

Title 'Summary of Hours';

Run;

Proc Tabulate Data= project Out=Project2;

Class employee week phase;

Var hours;

Table employee*hours week*hours;

Title 'Summary of Hours';

Run;

```
Proc Tabulate Data= project Out=Project2;
Class employee week phase;
Var hours;
Table employee*hours week*hours phase*hours;
Title 'Summary of Hours';
Run;
```

```
Proc Tabulate Data= project Out=Project2;
Class employee week phase;
Var hours;
Table employee= 'employee' week*hours;
Title 'Summary of Hours';
Run;
```

Noseps

Eliminates horizontal separator lines from row titles and body of the printed table.

```
Proc Tabulate Data= project Format=8.2 noseps;
Class employee week phase;
Var hours;
Table week, employee all, sum*hours= ' *(phase all);
Table week, employee all, pctsum*hours= ' *(phase all);
Keylabel sum='Total Hours'
          Pctsum='Percentage of Hours';
Title 'Summary of Hours';
Run;
```

Format=

Specify a format for each cell in the table

```
Proc Tabulate Data= project Format=8.1;
Class employee week phase;
Var hours;
Table week, employee all, sum*hours= ' *(phase all);
Table week, employee all, pctsum*hours= ' *(phase all);
Keylabel sum='Total Hours'
          Pctsum='Percentage of Hours';
Title 'Summary of Hours';
Run;
```

```
Proc Tabulate Data= project Format=8.1;
Class employee week phase;
Var hours;
Table week, employee all, sum*hours= ' *(phase all);
Table week, employee all, pctsum*hours= ' *(phase all);
Table week, employee all, n*hours= ' *(phase all);
```

```
Table week, employee all, nmiss*hours=' *(phase all);
Table week, employee all, range*hours=' *(phase all);
Table week, employee all, stddev*hours=' *(phase all);
Table week, employee all, css*hours=' *(phase all);
Table week, employee all, skewness*hours=' *(phase all);
Table week, employee all, max*hours=' *(phase all);
Table week, employee all, min*hours=' *(phase all);
Table week, employee all, kurtosis*hours=' *(phase all);
Table week, employee all, lclm*hours=' *(phase all);
Table week, employee all, uss*hours=' *(phase all);
Table week, employee all, var*hours=' *(phase all);
Title 'Summary of Hours';
Run;
```

Missing

Considers missing values as valid values to create the combinations of class variables.

Special missing values that are used to represent numeric values (The letters A through Z and the underscore (_) character) are each considered as a separate value. A heading for each missing value appears in the table.

If you omit MISSING, then PROC TABULATE does not include observations with a missing value for any class variable in the report

```
Proc Tabulate Data= project Format=8.2 noseps missing;
Class employee week phase;
Var hours;
Table week, employee all, sum*hours=' *(phase all);
Table week, employee all, pctsum*hours=' *(phase all);
Keylabel sum='Total Hours'
          Pctsum='Percentage of Hours';
Title 'Summary of Hours';
Run;
```

ORDER=DATA | FORMATTED | FREQ | UNFORMATTED

Specifies the sort order to create the unique combinations of the values of the class variables, which form the headings of the table, according to the specified order.

DATA - orders values according to their order in the input data set.

FORMATTED - orders values by their ascending formatted values. If no format has been assigned to a numeric class variable, then the default format, BEST12., is used. This order depends on your operating environment.

FREQ - orders values by descending frequency count.

UNFORMATTED - orders values by their unformatted values, which yields the same order as PROC SORT. This order depends on your operating environment. This sort sequence is particularly useful for displaying dates chronologically.

```
Proc sort data=Project;
By descending employee;
Run;
Proc Tabulate Data=Project Order=data;
Class employee week phase;
Var hours;
Table employee*week sum*hours;
Title 'Summary of Hours';
Run;
```

FORMCHAR <(position(s))>='formatting-character(s)'

Defines the characters to use for constructing the table outlines and dividers.

Position

Identifies the position of one or more characters in the SAS formatting-character string. A space or a comma separates the positions.

```
Proc Tabulate Data=Project Order=data formchar(3,7)='*#';
Class employee week phase;
Var hours;
Table employee*week sum*hours;
Title 'Summary of Hours';
Run;
```

STYLE=<style-element-name|<PARENT>>[style-attribute-name=style-attribute-value<... style-attribute-name=style-attribute-value>]

Specifies the style element to use for the data cells of a table when it is used in the PROC TABULATE statement.

```
Proc Tabulate Data= PROJECT format=8.2 Noseps style={background=ywh};
Class employee week phase;
Var hours;
Table week, employee all, sum*hours='*(phase all);
Table week, employee all, pctsum*hours='*(phase all);
Keylabel sum='Total Hours'
          Pctsum='Percentage of Hours';
Title 'Summary of Hours';
Run;
```

```
Proc Tabulate Data= PROJECT format=8.2 Noseps Style={ background=ywh foreground=red};
Class employee week phase;
Var hours;
Table week, employee all, sum*hours='*(phase all);
Table week, employee all, pctsum*hours='*(phase all);
Keylabel sum='Total Hours' Pctsum='Percentage of Hours';
```

Title 'Summary of Hours';

Run;

Proc Tabulate Data= PROJECT format=8.2 Noseps

Style={background=ywh foreground=red font_weight=bold
font_face=Algerian}; Class employee week phase;

Var hours;

Table week, employee all, sum*hours=' *(phase all);

Table week, employee all, pctsum*hours=' *(phase all);

Keylabel sum='Total Hours'

Pctsum='Percentage of Hours';

Title 'Summary of Hours';

Run;

Proc Tabulate Data= PROJECT format=8.2 Noseps

Style={background=yellow foreground=red font_weight=bold font_face=Algerian};

Class employee/Style={background=magenta} ;

Class week/Style={background=cyan};

Class phase/Style={background=orange};

Var hours;

Table week, employee all, sum*hours=' *(phase all);

Table week, employee all, pctsum*hours=' *(phase all);

Keylabel sum='Total Hours'

Pctsum='Percentage of Hours';

Title 'Summary of Hours';

Run;

Proc Tabulate Data= PROJECT format=8.2 Noseps

Style={background=yellow foreground=red font_weight=bold font_face=Algerian};

Class employee/Style={background=magenta} ;

Class week/Style={background=cyan};

Class phase/Style={background=orange};

Classlev employee phase/Style={background=lime};

Var hours;

Table week, employee all, sum*hours=' *(phase all);

Table week, employee all, pctsum*hours=' *(phase all);

Keylabel sum='Total Hours' Pctsum='Percentage of Hours';

Title 'Summary of Hours';

Run;

Observe below programs and output.

Data ds;

Set sashelp.class;

Run;


```
Proc tabulate data=ds;
Class name sex;
Var height weight age;
Table name*height;
Run;
```

```
Proc tabulate data=ds;
Class name sex;
Var height weight age;
Table name, height;
Run;
```

```
Proc tabulate data= ds;
Class name sex;
Var height weight age;
Table name*height, sex;
Run;
```

```
Proc tabulate data=ds;
Class name sex;
Var height weight age;
Table name*(height weight), sex;
Run;
```

```
Proc tabulate data=ds;
Class name sex;
Var height weight age;
Table name*(height*(MEAN N) weight*(STD)), sex;
Run;
```

```
Proc tabulate data=ds;
Class name sex;
Var height weight age;
Table name*(height*(MEAN N) weight*(STD)), sex;
LABEL name='Student name';
Keylabel std='standard deviation';
Run;
```

```
Proc format;
Value $gen
'M'='C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\m.jpg'
'F'='C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\f.jpg';
Run;
```

```
Proc tabulate data=a noseps missing;
Class name sex;
Var height weight age;
```



```
Table name*(height*(MEAN N) weight*(STD)), sex;
LABEL name='Student name';
Keylabel STD='standard deviation';
CLASSLEV SEX/STYLE={BACKGROUND=pink PREIMAGE=$gen.};
Run;
```

Realtime example:-

**** INPUT SAMPLE DEMOGRAPHICS DATA ****;

Data demog;

Label subjid = "Subject Number"

trt = "Treatment"

gender = "Gender"

race = "Race"

age = "Age";

Input subjid trt gender race age @@;

Datalines;

```
101 0 1 3 37 301 0 1 1 70 501 0 1 2 33 601 0 1 1 50 701 1 1 1 60
102 1 2 1 65 302 0 1 2 55 502 1 2 1 44 602 0 2 2 30 702 0 1 1 28
103 1 1 2 32 303 1 1 1 65 503 1 1 1 64 603 1 2 1 33 703 1 1 2 44
104 0 2 1 23 304 0 1 1 45 504 0 1 3 56 604 0 1 1 65 704 0 2 1 66
105 1 1 3 44 305 1 1 1 36 505 1 1 2 73 605 1 2 1 57 705 1 1 2 46
106 0 2 1 49 306 0 1 2 46 506 0 1 1 46 606 0 1 2 56 706 1 1 1 75
201 1 1 3 35 401 1 2 1 44 507 1 1 2 44 607 1 1 1 67 707 1 1 1 46
202 0 2 1 50 402 0 2 2 77 508 0 2 1 53 608 0 2 2 46 708 0 2 1 55
203 1 1 2 49 403 1 1 1 45 509 0 1 1 45 609 1 2 1 72 709 0 2 2 57
204 0 2 1 60 404 1 1 1 59 510 0 1 3 65 610 0 1 1 29 710 0 1 1 63
205 1 1 3 39 405 0 2 1 49 511 1 2 2 43 611 1 2 1 65 711 1 1 2 61
206 1 2 1 67 406 1 1 2 33 512 1 1 1 39 612 1 1 2 46 712 0 . 1 49
```

;

Run;

**** DEFINE VARIABLE FORMATS NEEDED FOR TABLE ****

Proc format;

Value trt 1 = "Active"

0 = "Placebo";

Value gend 1 = "Male"

2 = "Female";

Value race 1 = "White"

2 = "Black"

3 = "Other*";

Run;

**** DEFINE OPTIONS FOR ASCII TEXT OUTPUT ****

```
options nodate ls = 80 ps = 38 formchar = "|----|+|---+=|-\<>*";
```

**** CREATE SUMMARY OF DEMOGRAPHICS WITH PROC TABULATE ****

```
Proc tabulate data = demog;
  Class trt gender race;
  Var age;
  Table Age = 'Age' *
    (n = 'n' * f = 8.
    Mean = 'Mean' * f = 5.1
    STD = 'Standard Deviation' * f = 5.1
    Min = 'Min' * f = 3.
    Max = 'Max' * f = 3.)
  Gender = 'Gender' *
    (n='n' * f = 3.
    colpctn = '%' * f = 4.1)
  Race = 'Race' *
    (n = 'n' * f = 3.
    colpctn = '%' * f = 4.1),
    (trt = " ") (all = 'Overall');
  Format trt trt. race race. gender gender.;
  title1 'Table 1';
  title2 'Demographics and Baseline Characteristics';
  footnote1 "* Other includes Asian, Native American, and other races.";
  footnote2 "Created by %sysfunc(getoption(sysin)) on &sysdate9.";
```

Run;