



SAS® Tip: Producing Distinct Variable Levels

Determining the number of and distinct values a categorical variable contains is critical knowledge that all data analysts seek an answer to. Acquiring this information helps everyone involved to better understand the number of distinct variable levels, the unique values and their associated counts, and the necessary details for developing data-driven programming constructs and elements. This tip explores three SAS® Base software programming techniques to help anyone gain insight into the distinct variable levels for any variable in a data set.

Approach #1:

Using PROC SORT, DATA Step, and PROC PRINT to Produce Distinct Variable Levels

DATA step programmers with FIRST. (first dot) knowledge are able to produce distinct variable levels easily using a three-step approach with PROC SORT, a DATA step, and PROC PRINT. As the following example shows, a PROC SORT is first specified to arrange the SASHELP.CARS data set in ascending order by the MAKE variable. The resulting sorted data set is then processed in a user-written DATA step using FIRST. logic to initialize and increment counters, and the creation of two data sets: Unique_Groups and NLevels_Total. At the end-of-file, the NLevels_Total data set is produced. Finally, the two data set results are then produced with two PROC PRINTs produce the results in the NLevels_Total and Unique_Groups data sets.

PROC SORT, DATA Step, and PROC PRINT Code:

```
proc sort data=sashelp.cars
            out=work.cars_sorted ;
    by make ;
run ;

data work.Unique_Groups (keep=make ctr_make)
    work.NLevels_Total(keep=ctr_nlevels) ;
    set work.cars_sorted end=eof ;
    by make ;
    if first.make then
    do ;
        ctr_make = 0 ; /* Initialize Categorical Variable Counter */
        ctr_nlevels + 1 ; /* Increment NLevels Counter */
    end ;
    ctr_make + 1 ; /* Increment Categorical Variable Counter */
    if last.make then output work.Unique_Groups ;
    if eof then output work.NLevels_Total ; /* At End-of-File, Output NLevels Counter */
run ;

title "Display Total NLevels for Make in SASHELP.CARS with PROC PRINT" ;
proc print data=work.NLevels_Total(keep=ctr_nlevels) noobs ;
run ;

title "Display Unique Groups and Counts for Make in SASHELP.CARS with PROC PRINT" ;
proc print data=work.Unique_Groups(keep=make ctr_make) noobs ;
run ;
```

Results:

Display Total NLevels for Make in SASHELP.CARS with PROC PRINT

ctr_nlevels
38

Display Unique Groups and Counts for Make in SASHELP.CARS with PROC PRINT

Make	ctr_make
Acura	7
Audi	19
BMW	20
Buick	9
Cadillac	8
Chevrolet	27
Chrysler	15
Dodge	13

.

Porsche	7
Saab	7
Saturn	8
Scion	2
Subaru	11
Suzuki	8
Toyota	28
Volkswagen	15
Volvo	12

Approach #2:

Using PROC SQL to Produce Distinct Variable Levels

Approach #2 uses PROC SQL's COUNT(DISTINCT column-name) and COUNT(column-name) functions to produce distinct variable levels and their unique values.

PROC SQL with COUNT(DISTINCT make) and COUNT(make) Code:

```
proc sql ;
  title "Display NLevels for Make in SASHELP.CARS with PROC SQL" ;
  select COUNT(DISTINCT make) "Unique Levels"
  from sashelp.cars ;

  title "Display NLevels and Counts for Make in SASHELP.CARS with PROC SQL" ;
  select make,
  COUNT(make) AS ctr_make
```

```

from sashelp.cars
  group by make ;
quit ;

```

Results:

Display NLevels for Make in SASHELP.CARS with PROC SQL

Unique Levels
38

Display NLevels and Counts for Make in SASHELP.CARS with PROC SQL

Make	ctr_make
Acura	7
Audi	19
BMW	20
Buick	9
Cadillac	8
Chevrolet	27
Chrysler	15
Dodge	13

... ..

Porsche	7
Saab	7
Saturn	8
Scion	2
Subaru	11
Suzuki	8
Toyota	28
Volkswagen	15
Volvo	12

Approach #3:

Using PROC FREQ with the NLEVELS Option

Approach #3 uses PROC FREQ with the NLEVELS option to produce distinct variable levels and their unique values.

PROC FREQ with the NLEVELS Option Code:

```

title "NLevels for Make in SASHELP.CARS with Details" ;
proc freq data=sashelp.cars nlevels ;
  tables make / nopct nocum ;
run ;

```

Results:

NLevels for Make in SASHELP.CARS with Details

The FREQ Procedure

Number of Variable Levels	
Variable	Levels
Make	38

Make	Frequency
Acura	7
Audi	19
BMW	20
Buick	9
Cadillac	8
Chevrolet	27
Chrysler	15
Dodge	13

.....

Porsche	7
Saab	7
Saturn	8
Scion	2
Subaru	11
Suzuki	8
Toyota	28
Volkswagen	15
Volvo	12

About The Author



Kirk Paul Lafler is an entrepreneur, consultant and founder of Software Intelligence Corporation, and has been using SAS since 1979. Kirk is a SAS application developer, programmer, certified professional, provider of IT consulting services, mentor, advisor and professor at UC San Diego Extension, educator to SAS users around the world, and emeritus sasCommunity.org Advisory Board member. As the author of six books including Google® Search Complete (Odyssey Press. 2014) and PROC SQL: Beyond the Basics Using SAS, Second Edition (SAS Press. 2013); Kirk has written hundreds of papers and articles; been an Invited speaker and trainer at hundreds of SAS International, regional, special-interest, local, and in-house user group conferences and meetings; and is the recipient of 25 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Kirk can be reached for questions at:

Kirk Paul Lafler
SAS Consultant, Application Developer, Programmer, Entrepreneur, Educator and Author
Software Intelligence Corporation

E-mail: KirkLafler@cs.com

LinkedIn: <http://www.linkedin.com/in/KirkPaulLafler>

Twitter: @sasNerd

[Google Search Complete!](#)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.