PharmaSUG 2019 - Paper BP-115

Freq Out – Proc Freq's Quick Answers to Common Questions

Christine McNichol, Covance

ABSTRACT

Proc freq, true to its name, gives frequency counts, as well as other informative statistics, and those frequency values can be output to a dataset. But proc freq can do more than just count. Its ability to provide a unique list and flexibility to use unsorted data can save both time and keystrokes in a variety of scenarios.

Combining these features with the out= option, provides another method to add to the programming arsenal for a way to grab a list of subjects or parameters, investigate a difference or do a quick comparison.

This paper will look at how proc freq and its functionality can help with a quick response to common questions such as: What subjects were included in this count? What and how many subjects/records are impacted by this data issue? What does the data show for these problem subjects in another dataset? Is there uniqueness within the data by these variables?

Though it might not be the obvious choice, using one proc freq can take the place of multiple steps including procs sorts, data steps and prints to answer these questions. Additionally, the output generated from the proc freq method can very easily be exported by rows, columns, or selections to provide clean and clear responses to ad-hoc requests.

INTRODUCTION

Proc freq may be the first procedure thought of if the result needed is a count. It can be used to find the number of subjects randomized to TRT01P='Placebo', count the number of MALE and FEMALE subjects in Study ABC-XYZ, and produce an Adverse Event table of frequency of subjects within preferred term.

But what if the result needed is not necessarily a count? Proc freq can be a helpful tool for purposes other than counting. The following are just some of the common types of questions that may arise that fit into this category:

- What subjects were included in this count?
- What subjects were impacted by this data issue?
- What tests are involved in this issue?
- What are the impacted values?
- Do any subjects have non-unique records?

Proc freq might not be the first method thought of to answer questions such as these, but it may be a very quick and efficient option to use.

Why might proc freq be a good candidate for these quick responses? First, in these cases, there is less typing than other methods. This equates to both time saved as well as less opportunity for typos. Additionally, there may already be some freqs in the works to check into an issue when further questions arise. So getting the results from the existing frequency counts rather than data processing may be a logical next step. Finally, using the output datasets provided by proc freq, there are readily available lists of unique values, combinations, and counts that can be copied from Enterprise Guide® window by row, column, or selection for quick response or used for further data processing.

PROC FREQ HIGHLIGHTS

Proc freq has several features and uses. But at its core, it is a procedure that will produce counts and statistics related to those counts. It can produce frequency counts and percentages for a single item or

cross tabulation frequencies breaking down counts of combinations of values across multiple variables, such as AEDECOD by TRTA by SEX.

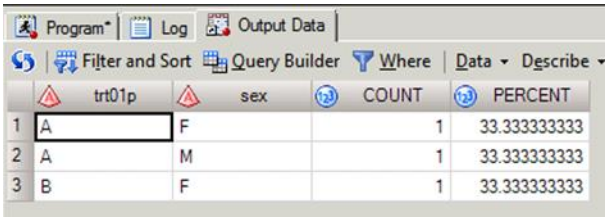
The benefits of proc freq go beyond simple counts. Proc freq is also extremely useful in performing testing and generating various statistics and p-values using several different methods. But for purposes of this paper, the focus will be on the basic counting and lists generated by the tables statement in proc freq.

When proc freq is run, by default, it will display its frequency count result in the output window which can be viewed but may not be useful if the intent is to use that result in a further data step or procedure. Commonly, a denominator other than the one used by default by proc freq is needed in order to calculate the appropriate percent. In this case, the result would need to be further processed, for example, by bringing it into a data step and recalculating percent. To generate a nicely formatted or standardized display, the proc freq output may need to be used by proc report, data _null_, or even passed through a display macro to accomplish this. In order to be able to better utilize the result of the proc freq going forward, this procedure has a couple ways to provide a result in an easy to use format. For simple frequency counts, the OUT= option can be used on the tables statement to generate a dataset containing the result.

A basic proc freq to get the number of Male and Female subjects by treatment group and create an output dataset using the OUT= option may look like the following:

```
proc freq data=dm noprint;
  tables trt01p*sex /list out=freq_out;
run;
```

Figure 1 shows an example of output using the OUT= option.



	trt01p	sex	COUNT	PERCENT
1	A	F	1	33.333333333
2	A	M	1	33.333333333
3	B	F	1	33.333333333

Figure 1. Output Using OUT= Option

Alternatively, SAS® Output Delivery System (ODS) can be used to create a dataset containing the results. However, depending on the syntax used, the resulting ODS table may differ. For a one-way frequency, the ODS table created is “OneWayFreqs” and could be generated with this code:

```
ods output OneWayFreqs=freq_out;
proc freq data=dm;
  tables sex;
run;
```

Figure 2 shows an example of output using the ODS table OneWayFreqs.

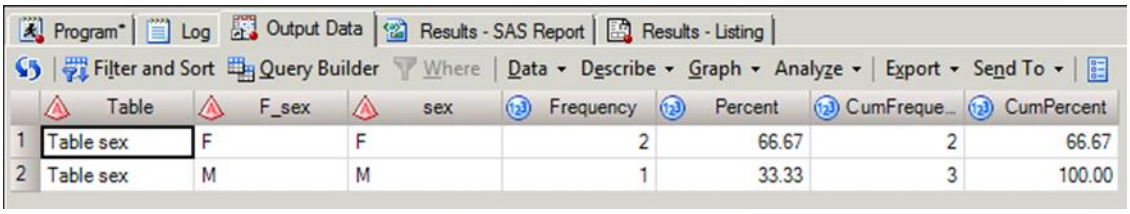


	Table	F_sex	sex	Frequency	Percent	CumFreque...	CumPercent
1	Table sex	F	F	2	66.67	2	66.67
2	Table sex	M	M	1	33.33	3	100.00

Figure 2. Output Using ODS Table OneWayFreqs

For frequencies with more than one variable used, an n-way frequency is calculated using the /list option. The ODS table created from the following code is “List”:

```
ods output List=freq_out;
proc freq data=dm;
```

```
    tables trt01p*sex /list missing;
run;
```

Figure 3 shows an example of output using the ODS table List.

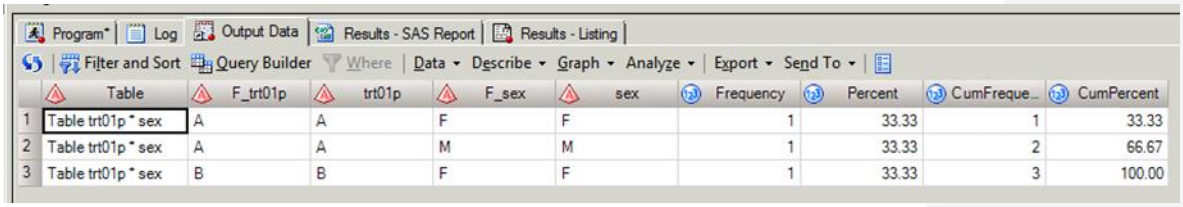


	Table	F_trt01p	trt01p	F_sex	sex	Frequency	Percent	CumFreque...	CumPercent
1	Table trt01p * sex	A	A	F	F	1	33.33	1	33.33
2	Table trt01p * sex	A	A	M	M	1	33.33	2	66.67
3	Table trt01p * sex	B	B	F	F	1	33.33	3	100.00

Figure 3. Output Using ODS Table List

Using either of these methods to output the results to a data step, it may be tempting to use the NOPRINT option to suppress the results in the display window. This can be done with the OUT= option, but caution, using NOPRINT will interfere with the ODS tables, so the display output can be suppressed, but it would need to be done in a different way.

KEY FEATURES FOR OUR QUICK NON-COUNT RESPONSES

There are several features of proc freq that we will use to our advantage to be able to get these quick non-numeric results with minimal coding. The key features for our coding are:

- OUT= option

This option will create a work dataset of the results with a few quick keystrokes. This will allow us to use the results in a variety of ways that differs from just recalculating the percentages. We can use the unique values that proc freq provides to subset other datasets. Or if using Enterprise Guide, we can very easily export or even copy/paste the values in that work dataset into excel for response to a statistician or other individual looking for the results. While similar results can be achieved using ODS, the examples in this paper will use the OUT= option. For our purposes, all necessary information is available in the default output dataset and there are less keystrokes needed – and as a bonus, less possibility for typos – to just use the OUT= option. Additionally, with the OUT= option, the output dataset can be written in one consistent way and there is not a need to change which table is referenced based on the number of variables in the tables statement as there is with ODS.

- No pre-sorting

While using the BY statement in proc freq does require the data to be presorted, it is not a requirement in using the TABLES statement. In the long run in a full program and with a large dataset or in cases where the dataset may be used multiple times, it will likely be more efficient to use proc sort and then use the dataset presorted for each use. But here, we have one-off quick looks at the data and doing this quick look, we can have proc freq 'sort' our data for us while it processes the TABLES statement instead of writing a separate proc sort.

- Generation of unique lists – one-way and cross tabs

In addition to providing the counts of one-way and n-way frequencies, proc freq also provides the unique list of values from the tables statement that go along with those counts in order to identify. But since these values are each provided as a variable with the same name as in the source dataset, we can use those variables for a simple view of the unique values list or even to further subset other datasets by those values using the original variable names.

QUESTION 1: WHAT SUBJECTS WERE INCLUDED IN THIS COUNT?

This question arises frequently, in a variety of cases from working through a difference in table validation to answering a question about an unexpected result. A specific example of looking for subjects included in a count is the question: What subjects were counted in this AE table cell? The result could be found in any number of ways in SAS. Sorts and data steps could be used or proc SQL is another option. A data

step might be the first thought for a solution. First the data needs to be sorted (proc sort). Then records of interest and unique subjects need to be selected (data step). Then the records output (proc print).

But proc freq can answer this question in one block of code without a data step! A proc freq can be used with a WHERE statement to subset to the treatment group and AE term of interest. A TABLES statement of USUBJID can then be used to get a unique list of the subjects included in that subset. Finally, the OUT= option can be used to create a dataset of the resulting USUBJID list. The code would look like the following:

```
proc freq data=adae;
  where trta='Placebo' and aedecod='Headache';
  tables usubjid /out=adae_pbo_headache;
run;
```

Figure 4 shows the proc freq output dataset containing the list of unique USUBJID values from the AE table cell.

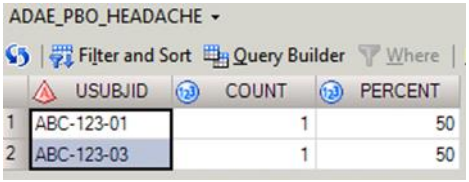


Figure 4. Output Showing Unique USUBJID from the AE Table Cell

The highlighted USUBJID values from the resulting output dataset can be easily copied and pasted into email, Excel, or some other format for a quick reply.

QUESTION 2: WHAT SUBJECTS WERE IMPACTED BY THIS DATA ISSUE?

Imagine that it has been found that there are some lab records with LBDTC prior to informed consent date. The first question is likely: Which subjects have LBDTC prior to RFICDTC? But this is the type of question where the results may lead to follow-up questions and the scope of the investigation may expand requiring additional coding to get to the bottom of the issue. Regardless of the method followed, the investigation will start with a dataset containing both the lab data and the informed consent date for comparison. This question may be answered by working within a data step or by using the proc freq method.

Data step method:

Working within a data step, if the dataset is not already sorted, a proc sort would come first and then a data step to select the unique usubjid values of interest.

```
proc sort data=lb_ic;
  by usubjid;
run;

data unique_usubjid(keep=usubjid);
  set lb_ic(where=(lbdtc<rficdtc));
  by usubjid;
  if first.usubjid;
run;
```

Figure 5 shows the resulting dataset from data step processing for subjects impacted by the issue.

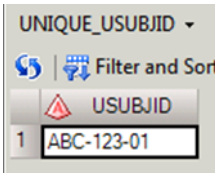


Figure 5. Output Showing Unique USUBJID Impacted by the Issue as Generated by Data Step Processing

Proc freq method:

Working with a proc freq, we could simply do a freq with USUBJID in the TABLES statement in addition to using the WHERE statement to subset to the records with issue:

```
proc freq data=lb_ic;
  where lbdtc<rficdtc;
  tables usubjid /list missing out=frqout_usubjid;
run;
```

Figure 6 shows the resulting dataset from proc freq for subjects impacted by the issue.

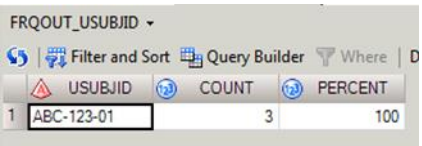


Figure 6. Output Showing Unique USUBJID Impacted by the Issue as Generated by Proc Freq

So far, we have saved a little bit of typing. Proc freq has also nicely kept in the resulting dataset only the USUBJID variable from the large LB dataset and has not carried forward the extraneous and bulky list of variables. But we did need to use a keep statement in the data step to remove the unnecessary variables. But as investigations deepen and follow-up questions continue to arise, the difference in lines of necessary code between the data step method and freq method will increase.

FOLLOWUP QUESTION 1: WHAT TESTS ARE INVOLVED IN THIS ISSUE?

Perhaps there is a test that is protocol specified that may be included from historic data. The LBTESTCD values impacted by this issue would need to be identified to determine if they are limited to this allowed test.

Data step method:

Another proc sort would be needed to ensure that the data is sorted by LBTESTCD. Then another data step would be needed to select the additional unique level by LBTESTCD.

```
proc sort data=lb_ic;
  by lbtestcd;
run;

data unique_lbtestcd(keep=lbtestcd);
  set lb_ic(where=(lbdtc<rficdtc));
  by lbtestcd;
  if first.lbtestcd;
run;
```

Figure 7 shows the resulting dataset from data step processing for lab tests impacted by the issue.

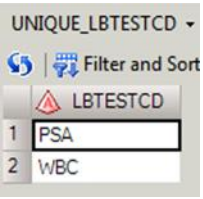


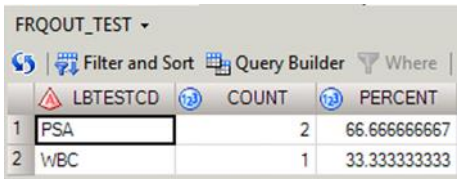
Figure 7. Output Showing Unique LBTESTCD Impacted by the Issue as Generated by Data Step Processing

Proc freq method:

Additional code required in the proc freq method is just another TABLES statement using LBTEST added to the same proc freq:

```
  tables lbtestcd  /list missing out=frqout_test;
```

Figure 8 shows the resulting dataset from proc freq for lab tests impacted by the issue.


FRQOUT_TEST

Filter and Sort | Query Builder | Where

	LBTESTCD	COUNT	PERCENT
1	PSA	2	66.666666667
2	WBC	1	33.333333333

Figure 8. Output Showing Unique LBTESTCD Impacted by the Issue as Generated by Proc Freq

FOLLOWUP QUESTION 2: WHAT ARE THE IMPACTED VALUES?

What is the difference between the LBDTC and RFICDTC? This may help to identify whether the issue is likely a collection issue or if there is potentially a problem such as the prior year being entered in an early January date or other data issue.

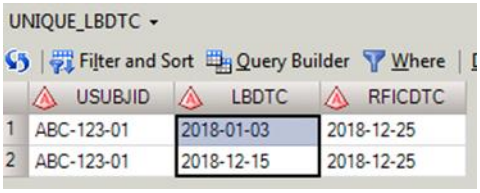
Data step method:

Again, the data would need to be resorted with proc sort to change the sorting to include LBDTC. Then another data step would be needed to select the additional unique level including LBDTC.

```
proc sort data=lb_ic;
  by usubjid lbdtc rficdtc;
run;

data unique_lbdtc(keep=usubjid lbdtc rficdtc);
  set lb_ic(where=(lbdtc<rficdtc));
  by usubjid lbdtc rficdtc;
  if first.lbdtc;
run;
```

Figure 9 shows the resulting dataset from data step processing of LBDTC and RFICDTC involved in the issue.


UNIQUE_LBDTC

Filter and Sort | Query Builder | Where

	USUBJID	LBDTC	RFICDTC
1	ABC-123-01	2018-01-03	2018-12-25
2	ABC-123-01	2018-12-15	2018-12-25

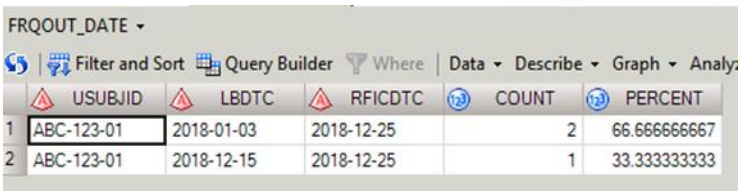
Figure 9. Output Showing Unique USUBJID and LBDTC Involved in the Issue as Generated by Data Step Processing

Proc freq method:

Like the prior addition, the proc freq method just requires another TABLES statement using LBDTC and RFICDTC added to the same proc freq:

```
    tables usubjid*lbdtc*rficdtc  /list missing out=frqout_date;
```

Figure 10 shows the resulting dataset from proc freq of LBDTC and RFICDTC involved in the issue.


FRQOUT_DATE

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analy

	USUBJID	LBDTC	RFICDTC	COUNT	PERCENT
1	ABC-123-01	2018-01-03	2018-12-25	2	66.666666667
2	ABC-123-01	2018-12-15	2018-12-25	1	33.333333333

Figure 10. Output Showing Unique USUBJID and LBDTC Involved in the Issue as Generated by Proc Freq

By the time the code is in place to answer the initial question and just the first two follow-up questions, about 24 lines of code have been written using proc sorts and data steps. However, with the proc freq method, all the answers to the questions so far can be provided with one proc and 6 lines of code.

```
proc freq data=lb_ic;
  where lbdtc<rficdtc;
  tables usubjid /list missing out=frqout_usubjid;
  tables lbtestcd  /list missing out=frqout_test;
  tables usubjid*lbdtc*rficdtc  /list missing out=frqout_date;
run;
```

Additionally, with the proc freq method, the counts that go along with the unique USUBJID, LBTEST or USUBJID by LBDTC values are already provided, so the number of impacted records is also readily available as that result will also likely be of interest.

FOLLOWUP QUESTION 3: WERE THE IMPACTED SUBJECTS SCREEN FAILURES?

The next thought may be to check if these are subjects that are known to have some issue already and did not pass screening. The unique USUBJID list from the proc freq output dataset can then be used easily going forward in a merge to DS to subset the Disposition data to then see study status for each of the subjects.

```
data lb_ds;
  merge frqout_usubjid(keep=usubjid in=ina) ds;
  by usubjid;
  if ina;
run;
```

QUESTION 3: DO ANY SUBJECTS HAVE NON-UNIQUE RECORDS?

Specifically, the question may arise: Are there any subjects with multiple records per parameter per visit in ADVS? With proc freq this can be seen in a very quick freq using USUBJID*PARAMCD*AVISITN in the TABLES statement to get a count of how many records meet each combination of the three variables:

```
proc freq data=advs noprint;
  tables usubjid*paramcd*avisitn /out=advs_usubjid_frq;
run;
```

Figure 11 shows an example of output showing non-unique records in ADVS by USUBJID, PARAMCD, AVISITN.

ADVS_USUBJID_FRQ

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze

	USUBJID	PARAMCD	AVISITN	COUNT	PERCENT
1	ABC-123-01	HEIGHT	1	1	10
2	ABC-123-01	WEIGHT	1	1	10
3	ABC-123-01	WEIGHT	2	1	10
4	ABC-123-02	HEIGHT	1	1	10
5	ABC-123-02	WEIGHT	1	1	10
6	ABC-123-02	WEIGHT	2	2	20
7	ABC-123-03	HEIGHT	1	1	10
8	ABC-123-03	WEIGHT	1	1	10
9	ABC-123-03	WEIGHT	2	1	10

Figure 11. Output Showing Uniqueness of Records in ADVS

From this result, on a short list of values, a simple visual scan of the count column for anything other than a one can quickly give the answer about uniqueness. For a longer result list, a quick limit to where count>1 will show us if there are any duplicates by USUBJID PARAMCD AVISITN.

CONCLUSION

Using proc freq instead of data step processing may not always be the method selected, but it is another valid option to be considered. It may be of most benefit in small to medium datasets and data may not

already be sorted as needed. It can be a nice quick way with minimal coding to get unique lists as well as counts to answer questions or aid in investigations. As with many things in SAS, there are numerous ways of coding to get to the result needed. While this may not always be the first solution to come to mind, it can be a very quick and clean option.

REFERENCES

https://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#freq_toc.htm

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Christine McNichol
Covance, Inc.
Christine.McNichol@Covance.com