PhUSE 2014

Paper IS05

Frequency Tables – Aspects to Consider

Katja Glaß, Bayer Pharma AG, Berlin, Germany

INTRODUCTION

Programming table outputs is one of the major tasks of a statistical programmer. A large number of these tables can be classified as frequency tables. Frequencies can be calculated for nearly all data, whether it be demography, adverse events, laboratory and many more. Even most efficacy tables are frequency tables in the end.

Creating frequency tables would appear to be straightforward, but actually there are various questions to consider. This paper demonstrates by example what types of frequency table exist, where the pitfalls are and what you might want to consider when programming them.

DO THE INDEPENDENT DOUBLE PROGRAMMING!

DEMOGRAPHY - JUST A SIMPLE FREQUENCY

Starting with an example, it might be that independent double programming is required for the validation of a frequency table but unfortunately there are no specifications available apart from the table itself. The major advantage of double programming is that the layout is unimportant as only the numbers for comparison matter. The first table to double program might look like this:

Table 1: Demography[1]

	Placebo	Xanomeline High Dose	Xanomeline Low Dose	Total
Number of subjects	86 (100.0%)	84 (100.0%)	84 (100.0%)	254 (100.0%)
Race				
AMERICAN INDIAN OR ALASKA NATIVE	0	1 (1.2%)	0	1 (0.4%)
BLACK OR AFRICAN AMERICAN	8 (9.3%)	9 (10.7%)	6 (7.1%)	23 (9.1%)
WHITE	78 (90.7%)	74 (88.1%)	78 (92.9%)	230 (90.6%)
Sex				
F	53 (61.6%)	40 (47.6%)	50 (59.5%)	143 (56.3%)
M	33 (38.4%)	44 (52.4%)	34 (40.5%)	111 (43.7%)
Ethnicity				
HISPANIC OR LATINO	3 (3.5%)	3 (3.6%)	6 (7.1%)	12 (4.7%)
NOT HISPANIC OR LATINO	83 (96.5%)	81 (96.4%)	78 (92.9%)	242 (95.3%)

As an additional requirement the planned treatment should be used. There are typically planned and actual treatments available in the datasets.

The first step requires investigations into the dataset and variables needed. The ADaM dataset ADSL (Subject Level) contains all required data for this table. The variable names are “SEX”, “RACE” and “ETHNIC”. So all the information needed to recreate the figures in the table is available. The SAS procedure “PROC FREQ” sounds intuitive and that is what is going to be used. One question is how to get the overall treatment counts. An easy solution is to duplicate all observations and use a “Total” treatment for the duplicates. So the source might look like this:

[1] Data used from the Scriptathon of PhUSE (http://code.google.com/p/phuse-scripts/)

```
*duplicate observations to include a total treatment group;
DATA adsl;
    SET ads.adsl;
    OUTPUT;
    trt01p = "TOTAL";
    OUTPUT;
RUN;

*calculate frequencies by treatment;
PROC SORT DATA=adsl OUT=adsl; BY trt01p; RUN;
PROC FREQ DATA=adsl;
    TABLES race            / out=freq_1;
    TABLES sex             / out=freq_2;
    TABLES ethnic          / out=freq_3;
    BY trt01p;
RUN;

*set frequencies together;
DATA freqs; SET freq_1 freq_2 freq_3; RUN;
```

The result is compared to the table output and the numbers match. This approach seems fine so let's move on to the next task.

ADVERSE EVENTS – MORE FREQUENCIES TO DO

The next table to validate is an adverse event table. Due to the sensitivity of the content it is essential that these tables are sufficiently checked and independent double programming is of course the best way to achieve this – although this typically requires adequate specifications. Adverse events tables are always seen as critical, clearly because of the significance of their content, but these tables are just frequencies and for this reason are simple to double program. The following table should be checked:

Table 2: Subjects with Adverse Events[2]

Body System / Reported Term	Placebo N=86 (100%)	Xanomeline High Dose N=84 (100%)	Xanomeline Low Dose N=84 (100%)	Total N=254 (100%)
ANY EVENT	69 (80.2%)	79 (94.0%)	77 (91.7%)	225 (88.6%)
……				
VASCULAR DISORDERS	3 (3.5%)	2 (2.4%)	3 (3.6%)	8 (3.1%)
HOT FLUSH	0	0	1 (1.2%)	1 (0.4%)
HYPERTENSION	1 (1.2%)	1 (1.2%)	1 (1.2%)	3 (1.2%)
HYPOTENSION	2 (2.3%)	0	1 (1.2%)	3 (1.2%)
ORTHOSTATIC HYPOTENSION	1 (1.2%)	0	0	1 (0.4%)
WOUND HAEMORRHAGE	0	1 (1.2%)	0	1 (0.4%)

The actual treatment should be used in this case. To create the table the ADAE (Adverse Events) dataset together with the variables "AEBODSYS" and "AETERM" will be used. The actual treatment is stored in "TRTA". The same approach as for demography is used, so the same steps are performed and the result is compared. To reduce the complexity the frequencies for the "Reported term" are calculated first. Then a small subgroup like "VASCULAR DISORDERS" and the treatment group "Placebo" is used for crosschecks.

Approach:

1) Duplicate the observations to create a "Total" treatment group.
2) Calculate the frequencies by "Body System" and treatment.
3) Reduce and compare the result.

[2] Data used from the Scriptathon of PhUSE (http://code.google.com/p/phuse-scripts/)

The programmed results look different from the table result:

Term for "VASCULAR DISORDERS"	Table Result	Check Result
HYPERTENSION	1 (1.2%)	2
HYPOTENSION	2 (2.3%)	3
ORTHOSTATIC HYPOTENSION	1 (1.2%)	2

The numbers are far from identical. Now starts the search for reasons. When reviewing the data, there seem to be subjects with several adverse events of the same kind. Whereas the table result displays the number of subjects with an event, the current checked number displays the number of adverse events, in other words the number of observations in the dataset.

There are two types of frequency tables to differentiate:

Observation Count Subject Count

It is important to differentiate the type of frequency required. In clinical study evaluations it is the subject count that is typically required. The SAS procedures create an observation count output. The dataset needs to be modified or a different counting has to be performed. For datasets like ADSL (Subject Level), where we have one observation per subject, the observation and subject count will always come to the same result. But they will not be the same for most other analysis datasets including ADAE (Adverse Events).

The dataset observations need to be reduced so that only one observation per one required count is left. This means reducing the dataset to one observation per subject and term. This can easily be done with the NODUPKEY option of PROC SORT which deletes duplicated observations within the specified BY groups.

```
* Remove duplicated entries per subject for the same AETERM;
PROC SORT DATA=adae OUT= adae_term NODUPKEY;
    BY trta aebodsys aeterm usubjid;
RUN;
```

New process to calculate the frequencies:

1) Duplicate the observations to create a "Total" treatment group.
2) Reduce the observations to have one observation per subject per "to count item" (in this case AETERM).
3) Calculate the frequencies by "Body System" and treatment.
4) Reduce and compare the result.

The result looks much better:

Term for "VASCULAR DISORDERS"	Table Result	Check Result	Check Percent
HYPERTENSION	1 (1.2%)	1	25.0
HYPOTENSION	2 (2.3%)	2	50.0
ORTHOSTATIC HYPOTENSION	1 (1.2%)	1	25.0

The numbered results are as expected, but the percentages do not match at all. SAS performs the calculations on four observations of the dataset, the rest have been removed to crosscheck just for a specific group. For SAS the number of observations is the complete population, meaning 100%. Of course there are many more subjects in our study and not just the four having an observation in the reduced dataset. The percentages should be based on the complete subject population and not just a subset. Whereas the numerator for the percentages is the subject count per event occurrence, the denominator should typically be the complete population, meaning all subjects.

Consider the population! Is the population in the dataset or defined somewhere else (typically in ADSL)?

Somehow the population information needs to be used within our table. How many subjects are in the study who might have had an adverse event? Typically all subjects in ADSL – sometimes with the restriction of a specific analysis set – should be used as the population. There are two ways to obtain the correct percentages. Either the input dataset for the SAS procedure is inflated or the population is calculated separately and then the percentages

are calculated manually with the counts and the population size. Probably the manual counting approach is easier to comprehend. It might now be feasible to switch to PROC SUMMARY as the percentages are irrelevant.

The process now looks like the following[3]:

1) Duplicate the observations to create a "Total" treatment group.
2) Reduce the observations to have one observation per subject per "to count item" (in this case AETERM).
3) Calculate the frequencies by "Body System" and treatment – ignore percentages.
4) Calculate and merge the population (the denominator).
5) Calculate the percentages.
6) Reduce and compare the result.

The new result:

Term for "VASCULAR DISORDERS"	Table Result	Check Result	Check Percent
HYPERTENSION	1 (1.2%)	1	1.2
HYPOTENSION	2 (2.3%)	2	2.3
ORTHOSTATIC HYPOTENSION	1 (1.2%)	1	1.2

A nice side effect when switching to PROC SUMMARY is that the summary statistics are available per BY group. So the upper group counts get automatically created as well. The comparison of the resulting dataset looks like this:

Term	Table Result	Check Result	Check Percent
VASCULAR DISORDERS	3 (3.5%)	4	4.7
HYPERTENSION	1 (1.2%)	1	1.2
HYPOTENSION	2 (2.3%)	2	2.3
ORTHOSTATIC HYPOTENSION	1 (1.2%)	1	1.2

Here again wrong numbers appear for the upper level, i.e. for "vascular disorders" in this example. This again is a problem due to the subject count. As mentioned, SAS performs an observation count. This is also true when creating the summary statistics for the BY group: the procedure is just counting the observations. Semantically this is not correct, as there was one subject who had different events in the "Vascular Disorders" group.

SAS summary statistics are observation counts! You very likely want a subject count! So count the summery statistics separately.

Finally two calculations have to be performed, one for the "AETERM" per "AEBODSYS" and one per "AEBODSYS". Afterwards these are merged together. Now the independent double program for the validation is done and the compared results are identical. Adverse events are just frequencies … but frequencies are not straightforward. There are yet more things to consider.

LABORATORY FREQUENCIES – WHAT ELSE CAN COME?

After programming adverse events, it might now be easy to create any kind of frequency tables. Just keeping in mind you need subject counts – meaning creating one observation per "to count item" – and you need to calculate each level separately when there is a hierarchy should result in correct programs and results. As all subjects who should be considered for the 100% population might not be in the analysis dataset, the population and percentages should be calculated separately.

The next table is a laboratory table which you might expect to validate quickly without any major issues. Apart from remembering to use the planned treatment, no further information is available for our double programming.

[3] Source available as attachment

Table 3: Number of subjects with laboratory ranges[4]

Analysis Visit: Baseline

Parameter	Reference Range Indicator	Placebo N=86 (100%)	Xanomeline High Dose N=84 (100%)	Xanomeline Low Dose N=82 (100%)	Total N=252 (100%)
Alanine Aminotransferase (U/L)	LOW	0	0	1 (1.2%)	1 (0.4%)
	NORMAL	82 (95.3%)	79 (94.0%)	79 (96.3%)	240 (95.2%)
	HIGH	4 (4.7%)	5 (6.0%)	2 (2.4%)	11 (4.4%)
Alkaline Phosphatase (U/L)	HIGH	4 (4.7%)	1 (1.2%)	3 (3.7%)	8 (3.2%)
	LOW	4 (4.7%)	1 (1.2%)	1 (1.2%)	6 (2.4%)
…	NORMAL	78 (90.7%)	81 (96.4%)	77 (93.9%)	236 (93.7%)

The dataset for this example study is ADLBC and the required variables are “AVISIT” for the visit, “PARAM” for the parameter name and “LBNRIND” as reference indicator. Furthermore the planned treatment “TRTP” and the subject identifier “USUBJIDN” have to be used. The population will be analyzed according the ADSL dataset.

The first step would be to duplicate the observations to create the “Total” treatment group. But the laboratory dataset is huge and the observation duplication takes some time. It might be a better idea to perform the total treatment calculations not on duplicated records, but to run the calculation procedures additionally without the treatment group. Working with duplicated records would then be avoided and the same program can then also be used for crossover studies.

Do not duplicate observations to create a “Total” treatment group,
but run the calculation procedures twice, with and without treatment!
➔ Better performance
➔ Crossover handling support

The next step would be the reduction of observations. For laboratory evaluations it is typically the individual values which are of importance, in contrast to the adverse events where it is the presence of an event that counts. For this, attention has to be paid to the individual values. The reduction might already be tricky as duplicated observations might have different analysis values.

Do not reduce the observations when they might have different analysis values!
Check for duplicates and investigate which observation to use for evaluations!

Typically there should only be one subject per parameter and time point used for the analysis. A little test program will check for duplicates:

```
1 PROC SORT DATA=ads.adlbc OUT=adlbc DUPOUT=err NODUPKEY;
2 BY avisitn avisit param usubjid;
3 RUN;
NOTE: The data set WORK.ADLBC has 74138 observations and 46 variables.
NOTE: The data set WORK.ERR has 126 observations and 46 variables.
```

There are 126 duplicates present. There is no obvious analysis flag to select on. Further specifications have to be provided. Finally only observations having a defined “AVISIT” should be considered for the evaluation of this table. All duplicates have no defined “AVISIT”, so the duplicates can be ignored.

The next step is to perform the counting with PROC SUMMARY (or PROC FREQ – just ignore the percentages). Afterwards the percentages with the number of subjects according to the complete population are calculated. The ADSL dataset is used again for the population.

[4] Data used from the Scriptathon of PhUSE (http://code.google.com/p/phuse-scripts/)

The results are the following:

Visit = “Baseline” and Param = “Alanine Aminotransferase (U/L)” – Treatment Total N=252 (100%)				
Reference Range Indicator	Table Result:	Check Result	Check Percent	Check Population
LOW	1 (0.4%)	1	0.4	254
NORMAL	240 (95.2%)	240	94.5	254
HIGH	11 (4.4%)	11	4.3	254

The numbers are fine, but there seems to be an issue with the percentages. The used population is “252” in the provided table whereas “254” is used according to the ADSL dataset. Further specifications for the population are required. This laboratory table is evaluated per time point. Due to this, only subjects which had been available for the specific time point should be considered for the calculations.

Typically subjects available at specific time points could have been stored in the ADSV (Subject Visits) dataset. The example study does not contain this dataset, so after further discussions it has been agreed to use subjects with at least one entry for a specific time point in the ADLBC dataset. This specification is written down and after a comparison of the newly programmed table, all numbers and percentages looks fine.

Which population should be used? All according to the study (ADSL), all available per time point or even others?

SUMMARY – POINTS TO CONSIDER

So finally, frequency table creation is not as simple as it may seem at first glance. There are various aspects to consider for the calculations. This paper addressed the following:

- Subject count vs. observation count
- Duplicating observations to obtain “Total” counts
- Population / Denominator
- Reduction of duplicates
- Total treatment handing

But there are even more options to be considered. One example is the definition of missing value handling. Should subjects with missing values be excluded or included? How is missing defined? There might be a missing because a laboratory value has not been collected or there might be a missing as no laboratory values have been collected for a particular subject.

Another example is multidimensional frequencies. The most common form is where the denominator still displays the complete population but sometimes the subgroup should sum up to 100% for the upper group, for example when you want to know how many treatment deviations are available per age group.

Also for shift tables it must be known whether 100% should be the complete “block” or per-row or per-column.

Block wise

	LOW	NORMAL	HIGH	TOTAL
LOW	4 (1.7%)	2 (0.8%)	0	6 (2.5%)
NORMAL	2 (0.8%)	222 (92.5%)	1 (0.4%)	225 (93.8%)
HIGH	0	2 (0.8%)	7 (2.9%)	9 (3.8%)
TOTAL	6 (2.5%)	226 (94.2%)	8 (3.3%)	**240 (100.0%)**

Row wise

	LOW	NORMAL	HIGH	TOTAL
LOW	4 (66.7%)	2 (33.3%)	0	**6 (100.0%)**
NORMAL	2 (0.9%)	222 (98.7%)	1 (0.4%)	**225 (100.0%)**
HIGH	0	2 (22.2%)	7 (77.8%)	**9 (100.0%)**
TOTAL	6 (2.5%)	226 (94.2%)	8 (3.3%)	**240 (100.0%)**

Column wise

	LOW	NORMAL	HIGH	TOTAL
LOW	4 (66.7%)	2 (0.9%)	0	6 (2.5%)
NORMAL	2 (33.3%)	222 (98.2%)	1 (12.5%)	225 (93.8%)
HIGH	0	2 (0.9%)	7 (87.5%)	9 (3.8%)
TOTAL	**6 (100.0%)**	**226 (100.0%)**	**8 (100.0%)**	**240 (100.0%)**

There might be special “At-Risk” calculations, where only subjects who meet special pre-requisites should be included. As an example abnormal laboratory values should be evaluated, but only subjects with a normal value at baseline should be considered. And this is probably required per laboratory parameter. This means that the numerator and denominator per laboratory parameter change continuously.

Finally it is of crucial importance that the specifications are available at the start of programming. And the specifications of course need to contain all necessary details. Quite often the specifications are not described sufficiently. If specifications require assumptions, than those need to be documented as well. Even for frequency tables there are various aspects to consider, and they too need to be specified in an appropriate way.

CONTACT INFORMATION

Author: Katja Glaß
Institution: Bayer Pharma AG
Address: Sellerstr. 32, 13353 Berlin, Germany
E-mail: Katja.Glass@Bayer.com

SOURCE

```
/************************************************************************
* Name             : 2014_is05_phuse_frequencies
*
* Purpose          : Source for the PhUSE presentation IS05 2014
*                    "Frequency Tables - Aspects to Consider"
*
* Validation Level : not applicable / not validated
* SAS Version      : HP-UX 9.2
* Pre-Requirements : Data from "http://code.google.com/p/phuse-
scripts/source/browse/#svn%2Ftrunk%2Fscriptathon2014%2Fdata%253Fstate%253Dclosed"
*                    as SAS datasets in library ADS
*
* License           : MIT (http://opensource.org/licenses/MIT)
*    Copyright (c) 2014 Katja Glaß
*    Permission is hereby granted, free of charge, to any person obtaining a copy of this software and
associated documentation files (the "Software"),
*    to deal in the Software without restriction, including without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense,
*    and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
*
*    The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.
*
*    THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT
LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
*    FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
*    WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
************************************************************************
* Author(s)        : Katja Glaß / date: 27AUG2014
************************************************************************/


* Initialize ADS library;
LIBNAME ads "<path to scriptathon data SAS datasets>";


************ DEMOGRAPHY - JUST A SIMPLE FREQUENCY ************************************************************;
* example source for double programming the first table;
*************************************************************************************************************************;
*duplicate observations to include a total treatment group;
DATA adsl;
    SET ads.adsl;
    OUTPUT;
    trt01p = "TOTAL";
    OUTPUT;
RUN;
*calculate frequencies by treatment;
PROC SORT DATA=adsl OUT=adsl; BY trt01p; RUN;
PROC FREQ DATA=adsl;
    TABLES race            / out=freq_1;
    TABLES sex             / out=freq_2;
    TABLES ethnic          / out=freq_3;
    BY trt01p;
RUN;
*set frequencies together;
DATA freqs; SET freq_1 freq_2 freq_3; RUN;
*RESULT:;
* The numbers fit the numbers of the table;
* double programming is fine;




************ ADVERSE EVENTS - MORE FREQUENCIES TO DO ***********************************************************;
* example source for double programming the second table - Adverse Events;
*************************************************************************************************************************;

************************************************;
********** first, try the approach from above *****;
************************************************;
*duplicate observations to include a total treatment group;
DATA adae;
    SET ads.adae;
    OUTPUT;
    trta = "TOTAL";
    OUTPUT;
RUN;
*calculate frequencies by treatment;
PROC SORT DATA=adae OUT=adae; BY trta aebodsys; RUN;
PROC FREQ DATA=adae;
```

```
    TABLES aeterm / out=ae_freq;
    BY trta aebodsys;
RUN;
*compare just a few result values;
DATA ae_freq;
    SET ae_freq (WHERE=(aebodsys = "VASCULAR DISORDERS" AND trta = "Placebo"));
    FORMAT percent 4.1;
RUN;
*RESULT:;
* PROBLEM:;
* The numbers do NOT fit the number of the table;
* Procedure performs OBSERVATION count whereas we would need a SUBJECT count;
* SOLUTION:;
* To count each subject just once, eliminate the duplicates;


************************************************;
********** second, eliminate duplicates **********;
************************************************;
*duplicate observations to include a total treatment group;
DATA adae;
    SET ads.adae;
    OUTPUT;
    trta = "TOTAL";
    OUTPUT;
RUN;
* Remove duplicated entries per subject for the same AETERM;
PROC SORT DATA=adae OUT=adae_term NODUPKEY;
    BY trta aebodsys aeterm usubjid;
RUN;
*calculate frequencies by treatment;
PROC FREQ DATA=adae_term;
    TABLES aeterm / out=ae_freq;
    BY trta aebodsys;
RUN;
*compare just a few result values;
DATA ae_freq;
    SET ae_freq (WHERE=(aebodsys = "VASCULAR DISORDERS" AND trta = "Placebo"));
    FORMAT percent 4.1;
RUN;
*RESULT:;
* PROBLEM:;
* The nominator of the frequency is as expected, but the denominator does not fit.;
* SAS procedure only knows available subjects of the given dataset.;
* The population to be used is not in the input dataset.;
* SOLUTION:;
* Determine the population in a separate step and calculate the percentages separately;


************************************************;
********** third, use self-determined population **;
************************************************;
*duplicate observations to include a total treatment group;
DATA adae;
    SET ads.adae;
    OUTPUT;
    trta = "TOTAL";
    OUTPUT;
RUN;
* Remove duplicated entries per subject for the same AETERM;
PROC SORT DATA=adae  OUT=adae_term NODUPKEY;
    BY trta aebodsys aeterm usubjid;
RUN;
*calculate frequency count (no percentages) by treatment;
PROC SUMMARY DATA = adae_term MISSING;
   CLASS aeterm;
   BY trta aebodsys;
   OUTPUT OUT = ae_freq;
RUN;
*calculate and merge population from ADSL;
PROC SUMMARY DATA = ads.adsl MISSING;
   CLASS trt01a;
   OUTPUT OUT = pop;
RUN;
DATA pop (DROP=_type_);
    SET pop (RENAME=(trt01A=trta _freq_ = n_pop));
    IF MISSING(trta) AND _TYPE_ = 0 THEN trta = "TOTAL";
RUN;
PROC SORT DATA=pop;     BY trta; RUN;
PROC SORT DATA=ae_freq; BY trta; RUN;
DATA ae_freq;
    MERGE ae_freq pop;
    BY trta;
    _perc_ = 100 * _freq_ / n_pop;
    FORMAT _perc_ 4.1;
RUN;
*compare just a few result values;
DATA ae_freq;
    SET ae_freq (WHERE=(aebodsys = "VASCULAR DISORDERS" AND trta = "Placebo"));
RUN;
```

```
*RESULT:;
* The numbers fit the numbers of the table;
* double programming is fine;



************ LABORATORY FREQUENCY - WHAT ELSE CAN COME? ************************************************;
* example source for double programming the third table - Laboratory Analysis;
*******************************************************************************************************;
* comment: observation duplication for the "Total" treatment column is not recommended;
*          would cause performance issues;
*          would be problematic for cross-over studies;
*          -> perform two calculations, one by treatment and one without treatment;
* Check for duplicates of subjects per timepoint and parameter - there should be none;
PROC SORT DATA=ads.adlbc OUT=adlbc DUPOUT=err NODUPKEY;
    BY avisitn avisit param usubjid;
RUN;
* RESULT:;
* There are 126 duplicates, check what to do with them;
* -> only use observations having a valid "AVISIT";
* Check for duplicates of subjects per timepoint and parameter - there should be none;
PROC SORT DATA=ads.adlbc (WHERE=(avisitn NE .)) OUT=adlbc DUPOUT=err NODUPKEY;
    BY avisitn avisit param usubjid;
RUN;
* RESULT: No duplicates -> no issues;
*calculate frequency count (no percentages) over all treatments (Total);
*make just a subselection to check just a few values for now;
PROC SUMMARY DATA = adlbc (WHERE=(SUBSTR(paramcd,1,1) NE "_" AND
                                  SUBSTR(paramcd,1,1) < "B" AND
                                  avisitn IN (0))) MISSING;
   CLASS lbnrind;
   BY avisitn avisit param;
   OUTPUT OUT = lb_freq;
RUN;
*calculate and merge population from ADSL over all treatments (Total);
PROC SUMMARY DATA = ads.adsl MISSING;
   OUTPUT OUT = pop;
RUN;
DATA pop;     SET pop;     const=1; RUN;
DATA lb_freq; SET lb_freq; const=1; RUN;
PROC SORT DATA=pop;     BY const; RUN;
PROC SORT DATA=lb_freq; BY const; RUN;
DATA lb_freq;
    MERGE lb_freq pop(RENAME=(_freq_ =n_pop));
    BY const;
    _perc_ = 100 * _freq_ / n_pop;
    FORMAT _perc_ 4.1;
RUN;
*RESULT:;
* PROBLEM:;
* The nominator of the frequency is as expected, but the denominator does not fit.;
* After investigations it has been figures out that the population should be calculated per availability per
timepoint;
* SOLUTION:;
* Calculate and merge the subjects available at visits;
*calculate frequency count (no percentages) over all treatments (Total);
*make just a subselection to check just a few values for now;
PROC SUMMARY DATA = adlbc (WHERE=(SUBSTR(paramcd,1,1) NE "_" AND
                                  SUBSTR(paramcd,1,1) < "B" AND
                                  avisitn IN (0))) MISSING;
   CLASS lbnrind;
   BY avisitn avisit param;
   OUTPUT OUT = lb_freq;
RUN;
*calculate and merge population from ADLBC over all treatments (Total);
*get correct specification where to take the population from and document this -;
*probably there is a ADSV (Subject Visits) or similar dataset available to be used;
PROC SORT DATA=ads.adlbc(KEEP=avisitn avisit usubjid) OUT=pop_l NODUPKEY;
    BY avisitn avisit usubjid;
RUN;
PROC SUMMARY DATA = pop_l MISSING;
   OUTPUT OUT = pop;
   BY avisitn avisit;
RUN;
DATA lb_freq;
    MERGE lb_freq (IN=lab) pop(RENAME=(_freq_ =n_pop));
    BY avisitn avisit;
    _perc_ = 100 * _freq_ / n_pop;
    FORMAT _perc_ 4.1;
    IF lab;
RUN;
*RESULT:;
* The numbers fit the numbers of the table;
* Now the remaining tests and timepoints and per-treatment analysis need to be performed for a complete check;
```