PharmaSUG 2025 - Paper ST-207

Clopper Pearson CI? Get your data ready for it!

Ruth Marisol Rivera Barragan, Ephicacy Consulting Group;
Isaac Vazquez, Ephicacy Consulting Group;

ABSTRACT

The Clopper-Pearson method is a widely used approach for calculating exact confidence intervals for binomial proportions, particularly in scenarios requiring high precision. However, incomplete datasets often add challenges to this calculation, since we have data completion before confidence intervals can be accurately derived. This paper presents a way to use SAS and SQL code for completing missing data to enable the calculation of confidence intervals using the Clopper-Pearson method. The proposed approach automates data preparation, identifies and addresses missing values, and integrates confidence interval computation. Through practical examples, we demonstrate the accuracy and efficiency of the methodology, highlighting its applicability in statistical analyses with incomplete data. This work provides a valuable resource for researchers and practitioners working with binomial data in SAS.

INTRODUCTION

Confidence intervals (CIs) are fundamental tools in statistical analysis, providing a range of values within which a population parameter is likely to lie with a given level of confidence. When working with binary or categorical data where outcomes fall into one of two categories, such as "success" or "failure" binomial proportions are used to describe the proportion of successes within a given sample. For example, in a clinical trial, the proportion of patients who respond positively to a treatment can be represented as a binomial proportion. Calculating confidence intervals for these proportions provides valuable insight into the precision of the estimate and the range of plausible values for the true population parameter.

The Clopper-Pearson method is a well-established approach for calculating exact CIs for binomial proportions. Unlike approximate methods, such as the Wald or approximation intervals, the Clopper-Pearson method guarantees coverage at the specified confidence level, even for small sample sizes or extreme proportions close to 0 or 1. This robustness makes it particularly useful in fields like clinical research, quality control, and survey analysis. However, accurate application of this method requires complete datasets, as missing or incomplete data can compromise the validity of the calculated intervals.

In practice, incomplete datasets are a common challenge. Missing observations may occur due to data entry errors, non-responses, incomplete data, or other factors, like in the example shown below, where Adverse Events can occur on a subject case basis, leading to an “Incomplete” dataset, due to the nature that not all AEs are presented for all subjects in your Raw Data; And managing these gaps are essential for maintaining the integrity of statistical analyses. While SAS, a powerful statistical software, offers tools for both data management and CI calculations, there is a lack of systematic approaches specifically tailored to oversee incomplete data in the context of the Clopper-Pearson method.

This paper is primarily intended for programmers and focuses on the practical implementation of data completion and confidence interval calculation using SAS. The goal is not to explore into the statistical theory underlying the Clopper-Pearson method but rather to provide a clear and reproducible programming workflow for handling incomplete data and calculating exact CIs. You can find more specific references for understanding the Clopper-Pearson statistical fundamentals in our references at the end of this paper.

THE PROBLEM - CONTEXT

You are required to work on a program that generates a table for Adverse Events, which includes Clopper-Pearson Exact Confidence Intervals and this is the shell for that table:

Table 1. Shell for Table XX.X.X

Treatment Emergent Adverse Events by System Organ Class and Preferred Term

System Organ Class Preferred Term	**Subjects n(%)** **[Lower Limit, Upper Limit]**
System Organ Class #1	xx (xx.x%) [xx.x, xx.x]
Preferred Term #1	xx (xx.x%) [xx.x, xx.x]
Preferred Term #2	xx (xx.x%) [xx.x, xx.x]
. . .	

All AEs are coded using MedDRA Version XX.X. AEs are presented by System Organ Class and Preferred Term sorted in descending order of frequency of subjects. At each level of summarization (System Organ Class and Preferred Term), subjects who reported more than one adverse event (AE) were only counted once. Clopper-Pearson Exact confidence intervals calculated for percentages.

Table 1. Shell for Table XX.X.X Adverse Events by System Organ Class and Preferred Term

This is the current data you have:

Table 2. Subjects with Treatment Emergent Adverse Events dataset

Subject	**SOC**	**PT**	**TRTEMFL**
001	Gastrointestinal disorders	Abdominal pain	Y
001	Gastrointestinal disorders	Vomiting	Y
002	Infections and infestations	Upper respiratory tract infection	Y
003	Nervous system disorders	Headache	Y
003	Gastrointestinal disorders	Vomiting	Y

Table 2. Subjects with Adverse Events dataset

At first, you might think that using a simple “PROC FREQ” for the current data can be enough, but this will give you incorrect results, since the data is not "complete", because not all subjects have listed all the Adverse Events in a form of a "success" or "failure" response, which is the main concept of a binomial proportion.

The question is: how can you “complete” this dataset?

THE PROPOSED SOLUTION

Here is the proposed solution path for this problem:

1. Create a dataset with all the unique subjects.

You can achieve this by using a PROC SORT with the NODUPKEY option, for example.:

```
proc sort data=AE out=U_SUBJ (keep=subject) nodupkey;
 by subject;
run;
```

Program 1. Get a dataset with unique subjects.

Table 3. Unique Subjects.

Subject
001
002
003

Table 3. Unique Subjects

2. Create a dataset with all the unique SOC and PT values. Once you have the unique subjects, generate another dataset for the unique SOC and PT, using a similar approach:

```
proc sort data=AE out=U_SOC_PT (keep=soc pt) nodupkey;
 by soc pt;
run;
```

Program 2. Get a dataset with unique SOC and PT.

Table 4. Unique SOC/PT.

SOC	PT
Gastrointestinal disorders	Abdominal pain
Gastrointestinal disorders	Vomiting
Infections and infestations	Upper respiratory tract infection
Nervous system disorders	Headache

Table 4. Unique SOC/PT

3. In this solution, we use a PROC SQL instead of a "merge" data step to avoid the "Merge statement has more than one data set with repeats of BY values" log issue.

```
proc sql;
   create table complete as
   select f1.Subject as Subject ,
          f2.SOC as SOC,
               f2.PT as PT
    from  U_SUBJ  as f1 , U_SOC_PT as f2
   where f1.Subject ne ' '
   order by f1.Subject,  f2.SOC, f2.PT;
quit;
```

Program 3. Use SQL to get the “complete” dataset.

Table 5. Completed Subjects with Adverse Events dataset.

Subject	SOC	PT
001	Gastrointestinal disorders	Abdominal pain
001	Gastrointestinal disorders	Vomiting
001	Infections and infestations	Upper respiratory tract infection
001	Nervous system disorders	Headache
002	Gastrointestinal disorders	Abdominal pain
002	Gastrointestinal disorders	Vomiting
002	Infections and infestations	Upper respiratory tract infection
002	Nervous system disorders	Headache
003	Gastrointestinal disorders	Abdominal pain
003	Gastrointestinal disorders	Vomiting
003	Infections and infestations	Upper respiratory tract infection
003	Nervous system disorders	Headache

Table 5. Completed Subjects with Adverse Events dataset.

4. Merge back with the original AE dataset with this ‘completed’ dataset and create the a “success”/ “failure” variable for the binomial proportion calculation.

   Once you have the "completed" dataset, you merge it back with the original AE dataset and create a variable (let us call it AEYN). This variable indicates whether the subject had the Treatment Emergent Adverse Event, which represents a "success" and should be first in the numerically sorted values (e.g., AEYN=1), or if the subject did not have the Treatment Emergent Adverse Event, which represents a "failure" and should be last in the numeric values (e.g., AEYN=2), in order to correctly calculate the binomial proportion.

   a) First do it by SOC (for SOC rows only).

```
/*keep soc level for CIs*/
proc sort data=ae out=soc_pt2 (keep= subject soc trtemfl) nodupkey;
   by    subject soc;
    where trtemfl='y';
run;

proc sort data=complete out=all_soc (keep= subject soc) nodupkey;
   by subject soc;
run;

data soc_ci;
  merge all_soc (in=a) soc_pt2 (in=b);
   by    subject soc;
    if trtemfl = 'Y' then aeyn=1;
     else aeyn=2;
run;
```

Program 4. Merge back with the original AE dataset with the ‘completed’ dataset only by SOC, then create the a “success”/ “failure” variable for the binomial proportion calculation.

Table 6. SOC_CI dataset result of merging “completed” and unique SUBJECT and SOC.

SUBJECT	SOC	TRTEMFL	AEYN
001	Gastrointestinal disorders	Y	1
002	Gastrointestinal disorders		2
003	Gastrointestinal disorders	Y	1
001	Infections and infestations		2
002	Infections and infestations	Y	1
003	Infections and infestations		2
001	Nervous system disorders		2
002	Nervous system disorders		2
003	Nervous system disorders	Y	1

Table 6. SOC_CI dataset.

b) Apply the “PROC FREQ” for getting the Confidence Limits for the Binomial Proportion

```
/*sort the final dataset before going to the freq*/
proc sort data=soc_ci ;
  by soc ;
run;

/*get the cis for soc level*/
proc freq data=soc_ci ;
 tables aeyn / binomial( exact) alpha=.05;
   by soc;
    output out=binomial_soc binomial;
run;
```

Program 5. Merge back with the original AE dataset with the ‘completed’ dataset by SOC and PT, then create the a “success”/ “failure” variable for the binomial proportion calculation.

Table 7. Binomial_SOC dataset.

SOC	N	XL_BIN	XU_BIN
Gastrointestinal disorders	3	0.094299	0.991596
Infections and infestations	3	0.008404	0.905701
Nervous system disorders	3	0.008404	0.905701

Table 7. Binomial_SOC dataset.

5. Once you get the confidence intervals by SOC, you repeat steps 4.a) and b) by SOC and PT, then apply formatting to "Exact Lower CL, Binomial Proportion" (XL_BIN) and "Exact Upper CL, Binomial Proportion" (XU_BIN) for your final dataset as following:

```
data all_stats;
  set  binomial_soc ;
   limits = "["||strip(put(xl_bin*100,8.1))||"%, "||strip(put(xu_bin*100,8.1))||"%]";
run;
```

Program 6. Apply formatting to the Lower and Upper Clopper Pearson Confidence Intervals

Table 8. Formatted Confidence Intervals limits.

SOC	LIMITS
Gastrointestinal disorders	[9.4%, 99.2%]
Infections and infestations	[0.8%, 90.6%]
Nervous system disorders	[0.8%, 90.6%]

Table 8. Formatted Confidence Intervals limits.

FORMATTING AND FINAL OUTPUT

After calculating your frequencies and confidence intervals by SOC and SOC/PT, add the confidence interval limits to your final table and make any necessary adjustments to prepare it for the final output.

Table 9. Final table output

Treatment Emergent Adverse Events by System Organ Class and Preferred Term

	Subjects n(%) [Lower Limit, Upper Limit]
Gastrointestinal disorders	2 (66.67%) [9.4%, 99.2%]
Vomiting	2 (66.67%) [9.4%, 99.2%]
Abdominal pain	1 (33.33%) [0.8%, 90.6%]
Infections and infestations	1 (33.33%) [0.8%, 90.6%]
Upper respiratory tract infection	1 (33.33%) [0.8%, 90.6%]
Nervous system disorders	1 (33.33%) [0.8%, 90.6%]
Headache	1 (33.33%) [0.8%, 90.6%]

All AEs are coded using MedDRA Version XX.X. AEs are presented by System Organ Class and Preferred Term sorted in descending order of frequency of subjects. At each level of summarization (System Organ Class and Preferred Term), subjects who reported more than one adverse event (AE) were only counted once. Clopper-Pearson Exact confidence intervals calculated for percentages.

Table 9. Final table output

Note: In this case, the CI limits for SOC and PT are similar due to the simplicity of the sample. However, when applying this approach to a larger dataset, the results may vary.

CONCLUSION

The Clopper Pearson Confidence Intervals are a valuable tool in Clinical Trial analysis, offering a reliable way to assess the precision of binomial proportions. These intervals are particularly useful when estimating the probability of rare events, such as Treatment Emergent Adverse Events (AEs), in clinical trials. The application of Clopper Pearson Confidence Intervals can help researchers and statisticians make more informed decisions, as they provide a range of values within which the true population parameter is likely to lie, offering a clearer understanding of the underlying variability in the data.

In clinical trials, accurate statistical analysis is critical for making data-driven decisions, particularly when evaluating the safety and efficacy of treatments. Therefore, the Clopper Pearson Confidence Interval provides a critical layer of validation, allowing for more precise conclusions regarding the relationships between treatment and outcomes. When applied correctly, this method helps ensure the robustness and reliability of clinical trial findings, making it a cornerstone of sound statistical practice in medical research.

The Clopper Pearson Confidence Intervals are widely applicable in various situations and contexts within Clinical Trial models. When the data is complete, calculating these intervals using the "PROC FREQ" procedure is relatively straightforward. However, the success of this process depends on following the correct steps to properly prepare the dataset. It is essential to ensure that the full dataset is used, including all combinations of subjects, conditions (such as Adverse Events), and a correctly defined success/failure variable, before calculating the Binomial proportion in PROC FREQ. Neglecting these preparatory steps can lead to inaccurate results and misleading conclusions, undermining the reliability of the statistical analysis.

REFERENCES

Jose Abraham, 2013. “Computation of CIs for Binomial proportions in SAS and its practical difficulties” PhUSE 2013. Available at https://www.lexjansen.com/phuse/2013/sp/SP05.pdf

SAS Documentation, Example 44.4 Binomial Proportions: SAS Documentation, example for Binomial Proportions.

CONTACT INFORMATION

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

Marisol Rivera
Ephicacy Consulting Group
marisol.rivera@ephicacy.com

Isaac Vazquez
Ephicacy Consulting Group
isaac.vazquez@ephicacy.com