SESUG Paper 067-2021

A SAS Macro Program to Calculate the Fragility Index

Tamar Roomian, MS MPH, Department of Orthopaedic Surgery, Atrium Health Musculoskeletal Institute

ABSTRACT

Fisher's Exact Test is a statistical test used to determine the statistical significance of a 2x2 contingency table when the sample size in any of the cells is <5. In comparative research healthcare studies where the outcome of interest is rare, statistical significance can be easily flipped by changing a small number of events. Feinstein first proposed the Fragility Index (FI) as a measure to determine the statistical stability. The Fragility Index is the number of outcomes required to reverse statistical significance and can be used to assess the statistical stability of studies with rare outcomes.

To our knowledge, there is currently no existing SAS macro program that can calculate the FI and FQ. We have created a SAS macro program that calculates the number of event switches required to flip a 2-sided Fisher's exact test for a list of studies. Researchers can then calculate descriptive statistics on the FI to assess statistical stability within a field of study. This presentation will introduce the fragility index, walk through the logic of the macro program, and demonstrate with an example.

INTRODUCTION

Fisher's Exact Test is a statistical test used to determine the statistical significance of a 2x2 contingency table when the sample size in any of the cells is <5. In healthcare comparative research studies where the outcome of interest is rare, statistical significance can be easily flipped by changing a small number of events. Feinstein first proposed the Fragility Index (FI) as a measure to determine the statistical stability (Feinstein, 1990). The Fragility Index is the number of outcomes required to reverse statistical significance and can be used to assess the statistical stability of studies with rare outcomes (Parisien et al., 2019). In addition, the Fragility Quotient is a measure of stability relative to the study sample size and is the FI divided by the total sample size. The larger the FI, the more stable the result. While there is no currently specified threshold for what is considered a robust result, the number of participants required to flip the p-value can be interpreted alongside the number of patients lost to follow-up – results where the number of patients lost to follow-up is higher than the FI should be interpreted cautiously (Dettori & Norvell, 2020).

The FI has several limitations. Because it cannot be adjusted for confounding variables, it should not be calculated for nonrandomized studies.(Andrade, 2020) In addition, it depends on the statistical threshold of 0.05; p-values closer to 0.05 will be less stable that P-values farther from 0.05 (Andrade, 2020).

To our knowledge, there is no SAS macro program that exists that can calculate the FI and FQ. We have created a SAS macro program that calculates the number of event switches required to flip a 2-sided Fisher's exact test for a list of studies.

MACRO REQUIREMENTS

The information for each study can be entered into a spreadsheet, with each row representing one study. Each study must have an ID with a character datatype. In addition, the dataset must already be limited to the studies you wish to include. Finally, the following columns must exist in the dataset and none of the values should be missing (you can use the data step to rename them if necessary).

Denom1: the denominator of group 1

Denom2: the denominator of group 2

Num1: the numerator of group 1

Num2: the numerator of group2

percent1: num1/denom1

percent2: num2/denom2

Study: the ID assignment for each study (character data type)

The macro takes one parameter, the name of the dataset.

The number of patients who do not have the outcome are calculated in a data step prior to calling the macro. Data preparation for an example is as follows:

1. Import your data. Be sure to restrict data to the publications you want to include. In the example, you have 5 studies labeled as "A" – "E." The variables Num1, Denom1, Num2, and Denom2 have already been created and studies not included have been excluded. The dataset can be generated with the following code and produces the following in Output 1:

```
data dataset;
input study $ num1 denom1 num2 denom2 p_value;
datalines ;
A 4 100 8 100 0.37
B 6 100 1 100 0.12
C 2 100 9 100 0.058
D 1 100 3 100 0.62
E 12 100 2 100 0.01
;
run;

proc print data=dataset noobs;
run;
```

study	num1	denom1	num2	denom2	p_value
A	4	100	8	100	0.370
B	6	100	1	100	0.120
C	2	100	9	100	0.058
D	1	100	3	100	0.620
E	12	100	2	100	0.010

Output 1. The dataset generated from the DATA step

2. Create the variables required by the macro (Output 2):

```
/*macro requirements*/
/*variables must be renamed as denom1, denom2, num1, and num2*/
/*there should be an id for each record called "study" that is a
character format*/
data dataset2;
```

```
set dataset;

absent1=denom1-num1;
absent2=denom2-num2;
percent1=num1/denom1;
percent2=num2/denom2;

if denom1=. or denom2=. or num1=. or num2=. then delete;

run;

proc print data=dataset2 noobs;
run;
```

study	num1	denom1	num2	denom2	p_value	absent1	absent2	percent1	percent2
A	4	100	8	100	0.370	96	92	0.04	0.08
B	6	100	1	100	0.120	94	99	0.06	0.01
C	2	100	9	100	0.058	98	91	0.02	0.09
D	1	100	3	100	0.620	99	97	0.01	0.03
E	12	100	2	100	0.010	88	98	0.12	0.02

Output 2. Additional variables added by the DATA step

MACRO PROGRAM

After the data is prepared, the macro program can be run. The steps of the macro program are detailed below:

1. Create a macro variable that contains a list of the studies. Create a macro variable that counts the number of studies:

```
%macro loop (dataset);
proc sql noprint;
select study into: study_list separated by ' '
from &dataset.
;
quit;

%let study_list_count=%sysfunc (countw(&study_list));
```

2. Dataset needs to be in “long” format. This is required for the proc freq weight statement further in the program:

```
/*dataset needs to be in long format*/
proc sort data=&dataset.;
by study;
run;

proc transpose data=&dataset. out=&dataset._long;
```

```
by study;
var num1 absent1 num2 absent2;
run;

/*additional variables needed for proc freq weight statement */
data &dataset._longb;
set &dataset._long;
group=substr(_name_, length(_name_), 1);
complication=substr(_name_, 1, length(_name_)-1);
run;
```

3. Start the first loop. The following steps will repeat for each study:

```
/*loop through this process for each study*/
%do j=1 %to &study_list_count;

/*limit dataset to 1 study*/
      data &dataset._long&j;
      set &dataset._longb;
      where study="%scan (&study_list, &j)";
      run;

      data test&j;
      set &dataset.;
      where study="%scan (&study_list, &j)";
      run;

/*calculate fisher's exact test*/
      proc freq data=&dataset._long&j;
      title "Study %scan (&study_list, &j)";
      weight col1;
      tables group*complication /fisher ;
      ods output fishersexact= fisher;
      /*ods output crosstabfreqs=freqs;*/
      run;

/*save p value as a macro variable*/
      data _null_;
      set fisher;
      where name1="XP2_FISH" ;
      keep nvalue1;
      call symputx ('pvalue', nvalue1);
      run;

/*save outcome rates as macro variables*/
      data _null_;
      set test&j;
      keep percent1;
      call symputx('percent1', percent1);
      run;

      data _null_;
      set test&j;
      keep percent2;
      call symputx('percent2', percent2);
      run;
```

4. Evaluate the p-value. If the p-value greater than or equal to .05, make the difference more extreme depending on which group is already higher. Repeat this loop until the p-value flips. Count the number of events required and save in the macro variable called i. If the p-value is less than .05, make the difference less extreme depending on which group is already higher. Repeat this loop until the p-value flips. Count the number of events required and save in the macro variable called i.

```
/*if p value is greater than .05*/
      %if (%sysevalf(&pvalue >=.05)) %then %do;
            %let i=0;
                  %do %until (%sysevalf(&pvalue <.05)); /*do loop until
p value flips*/
                        %let i=%eval(&i+1);

                        %if %sysevalf(&percent2>&percent1) %then %do;
/*make difference more extreme depending on which percent is already
higher*/
                              data test&j._&i;
                              set test&j;
                              num2=num2+&i;
                              absent2=absent2-&i;
                              run;
                        %end;

                        %else %do;
                              data test&j._&i;
                              set test&j;
                              num1=num1+&i;
                              absent1=absent1-&i;
                              run;
                        %end;

                        proc sort data=test&j._&i;
                        by study;
                        run;

                        proc transpose data=test&j._&i
                        out=test&j._&i._long;
                        by study;
                        var num1 absent1 num2 absent2;
                        run;


                        data test&j._&i._long2;
                        set test&j._&i._long;
                        group=substr(_name_, length(_name_), 1);
                        complication=substr(_name_, 1,
                        length(_name_)-1);
                        run;



                        proc freq data=test&j._&i._long2;
                        title "Study %scan (&study_list, &j)
                        iteration &i";
```

```
                  weight col1;
                  tables group*complication /fisher ;
                  ods output fishersexact= fisher&j._&i;
            /*ods output crosstabfreqs=freqs&i;*/
                  run;
                  title;

                  data _null_;
                  set fisher&j._&i;
                  where name1="XP2_FISH";
                  keep nvalue1;
                  call symputx ('pvalue', nvalue1);
                  run;


            %end;
%end;


%else %if (%sysevalf(&pvalue <.05)) %then %do;
      %let i=0;
      %do %until (%sysevalf(&pvalue >=.05));
            %let i=%eval(&i+1);


            %if %sysevalf(&percent2>&percent1) %then %do; /*make
            difference less extreme*/
                  data test&j._&i;
                  set test&j;
                  num1=num1+&i;
                  absent1=absent1-&i;
                  run;
            %end;

            %else %do;
                  data test&j._&i;
                  set test&j;
                  num2=num2+&i;
                  absent2=absent2-&i;
                  run;
            %end;

            proc sort data=test&j._&i;
            by study;
            run;

            proc transpose data=test&j._&i
            out=test&j._&i._long;
            by study;
            var num1 absent1 num2 absent2;
            run;


            data test&j._&i._long2;
            set test&j._&i._long;
            group=substr(_name_, length(_name_), 1);
            complication=substr(_name_, 1, length(_name_)-1);
```

```
                run;



                proc freq data=test&j._&i._long2;
                weight col1;
                title "Study %scan (&study_list, &j) iteration
                &i";
                tables group*complication /fisher ;
                ods output fishersexact= fisher&j._&i;
/*              ods output crosstabfreqs=freqs&i;*/
                run;
                title;

                data _null_;
                set fisher&j._&i;
                where name1="XP2_FISH";
                keep nvalue1;
                call symputx ('pvalue', nvalue1);
                run;

                /*%end;*/
        %end;
%end;
```

5. Add the FI and FQ for each dataset:

```
/*count number of iterations for each study, calculate Fragility
Quotient*/
      data FI&j;
      set test&j;
      FI=&i;
      FQ=&i/(denom1+denom2);
      run;
%end;
```

6. Set all the rows that now have the FI and FQ back together:

```
/*set the data back together*/
data final;
set FI1-FI&study_list_count;
run;

/*Print final results*/
proc print data=final;
run;
title;

/*Print descriptive statistics*/
proc means data=final median q1 q3 min max;
var FI FQ;
run;

%mend;
```

7. Finally, call the macro:

```
%loop (dataset2);
```

MACRO OUTPUT

After calling the macro program, the final dataset is called “final” and printed with the FI and FQ (Output 3). Descriptive statistics on the FI and FQ are also output.

Obs	study	num1	denom1	num2	denom2	p_value	absent1	absent2	percent1	percent2	FI	FQ
1	A	4	100	8	100	0.370	96	92	0.04	0.08	5	0.025
2	B	6	100	1	100	0.120	94	99	0.06	0.01	2	0.010
3	C	2	100	9	100	0.058	98	91	0.02	0.09	1	0.005
4	D	1	100	3	100	0.620	99	97	0.01	0.03	5	0.025
5	E	12	100	2	100	0.010	88	98	0.12	0.02	2	0.010

Output 3. The final output with the FI and FQ

CONCLUSION

The Fragility Index is a measure to determine the statistical stability of 2x2 contingency tables when the outcome of interest is rare. This macro program calculates the Fragility Index and Fragility Quotient for a list of studies. Then, you can calculate descriptive statistics on the FI and FQ to assess statistical stability within a field of study.

REFERENCES

Andrade, C. (2020). The Use and Limitations of the Fragility Index in the Interpretation of Clinical Trial Findings. *The Journal of Clinical Psychiatry*, *81*(2). https://doi.org/10.4088/JCP.20f13334

Dettori, J. R., & Norvell, D. C. (2020). How Fragile Are the Results of a Trial? The Fragility Index. *Global Spine Journal*, *10*(7), 940–942. https://doi.org/10.1177/2192568220941684

Feinstein, A. R. (1990). The unit fragility index: An additional appraisal of “statistical significance” for a contrast of two proportions. *Journal of Clinical Epidemiology*, *43*(2), 201–209. https://doi.org/10.1016/0895-4356(90)90186-s

Parisien, R. L., Dashe, J., Cronin, P. K., Bhandari, M., & Tornetta, P. (2019). Statistical Significance in Trauma Research: Too Unstable to Trust? *Journal of Orthopaedic Trauma*, *33*(12), e466–e470. https://doi.org/10.1097/BOT.0000000000001595

CONTACT INFORMATION

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

Tamar Roomian
tamar.roomian@atriumhealth.org