

A SAS/IML algorithm for an exact permutation test

Ein SAS/IML-Algorithmus für einen exakten Permutationstest

Abstract

An algorithm written in SAS/IML is presented that can perform an exact permutation test for a two-sample comparison. All possible permutations are considered. The Baumgartner-Weiß-Schindler statistic is exemplarily used as the test statistic for the permutation test.

Zusammenfassung

Ein in SAS/IML geschriebener Algorithmus wird präsentiert, mit dem ein exakter Permutationstest für das Zweistichproben-Problem durchgeführt werden kann. Dabei werden alle möglichen Permutationen berücksichtigt. Exemplarisch wird die Baumgartner-Weiß-Schindler-Statistik als Teststatistik für den Permutationstest genutzt.

Introduction

For many nonparametric test statistics there are two alternative ways to carry out the test. One possibility is to use the asymptotic distribution of the statistic. For instance, the Wilcoxon rank-sum statistic asymptotically follows a normal distribution (under the null hypothesis). Alternatively, the exact permutation null distribution can be used. In order to perform an exact permutation test, all possible permutations have to be generated, so that the test statistic can be calculated for each permutation. The p-value is the probability of the permutations giving a value of the test statistic as or more supportive of the alternative than the observed value.

In the population model ([1], p. 56) the inference is referred to a defined population that has been randomly sampled. Using this classical model-based inference, the asymptotic distribution can be used if the sample sizes were not too small. The randomization model of inference does not require that populations have been randomly sampled. However, the experimental units should be randomized to the groups or treatments ([1], p. 5; see also [2]). In this design-based framework, the p-value should be that from the permutation distribution (see also [3]).

Ludbrook and Dudley [4] surveyed prospective, comparative studies reported in several biomedical journals. They found that randomization rather than random sampling is the norm in biomedical research. Very few studies used random sampling of defined populations to construct experimental groups. Furthermore, sample sizes were often small. This suggests that exact permutation tests are useful for many biomedical studies. This is also true for some other areas, e.g. psychology [5].

As mentioned above, all possible permutations under the null hypothesis have to be generated in order to perform an exact permutation test. For some commonly used tests such as Wilcoxon's rank-sum test the exact permutation test is implemented in software packages. In SAS, for example, the Wilcoxon permutation test can be applied using the procedure NPAR1WAY. However, permutation tests with several other, especially newer statistics are not implemented in procedures. To apply these tests, one has to generate the permutations.

Some programs were recommended in the past, the shuffle algorithm of Chen and Dunlap ([6], p. 409) is one example. This algorithm considers a random sample of permutations only. Note that it is useful to use a simple random sample of all possible permutations if the sample sizes were large because, in that case, the number of permutations can be enormous ([7], p. 185). However, when sample sizes are small, an exact permutation test can be performed based on all possible permutations.

Here, we propose a SAS/IML program that considers all possible permutations for a two-sample comparison. This program performs an exact permutation test, the Baumgartner-Weiß-Schindler statistics [8] is exemplarily used as the test statistic for the permutation test. The exact permutation test based on the Baumgartner-Weiß-Schindler statistic was proposed by Neuhäuser [9]. The proposed algorithm is very fast for sample sizes up to 10 per group. For larger samples one may use the shuffle algorithm mentioned above.

Our algorithm combines depth-first search and backtracking [10]. Depth-first search is an algorithm for searching a tree. At each node, it is checked whether or not progress to a valid solution is possible. If not, or if a goal node is

Markus Neuhäuser¹
Andreas Schulz¹
Daniel Czech¹

¹ Department of Mathematics and Technique, RheinAhrCampus, Koblenz University of Applied Sciences, Remagen, Germany

found, the search backtracks, i.e. it returns to the most recent node that was not finished.

Note that a few parts of the original Baumgartner-Weiß-Schindler statistic are identical for every permutation. These parts are redundant for a permutation test and were omitted in the code given below.

The Baumgartner-Weiß-Schindler test was proposed as an omnibus test [8]. However, permutation tests are also useful in a location-shift model, and also in case of potential differences in variability [11]. In these cases, it is useful to compute a confidence interval for the location shift in addition to the p-value. A possible way to obtain a confidence interval was proposed by Bauer ([12], see also [13]).

Program

The following SAS macro includes the exact permutation test:

```
%MACRO Permtest(indata);
proc iml;

/* Reading the data */
USE &indata;
READ ALL INTO currdata;

/* Computation of ranks */
ranks=RANKTIE(currdata[,2]);

/* Calculation of the sample sizes per group */
N_total=Nrow(currdata[,2]);
n2=currdata[+,1];
n1=N_total-n2;
print N_total n1 n2;

/* Creation of all possible permutations */
start perm(n,n_1);
matrix = shape(0,(gamma(n+1)/(gamma(n_1+1))*
gamma(n-n_1+1)),n);
index = 1;
vektor=shape(-1,1,n);
pos = 1;
ok = 1;
do while(ok=1);
  if pos > n then do;

    if vektor[,+] = n_1 then do;
      matrix[index,]=vektor;
      index = index + 1;
    end;

    pos = pos-1;
  end;
else do;
  if vektor[,pos] < 1 then do;
    vektor[,pos] = vektor[,pos]+1;
```

```
    pos = pos+1;
  end;

  else do;
    vektor[,pos]=-1;
    pos = pos-1;
  end;
end;

if pos < 1 then ok = 0;
end;

return (matrix);
finish;

permutations = perm(N_total,n1);
P=Nrow(permutations);

/* Calculation of test statistic */
start test_sta(R1, R2, N_total, n1, n2);

b=R1;
R1[,rank(R1)]=b;
b=R2;
R2[,rank(R2)]=b;

i=1:n1;
j=1:n2;
Bx=sum( (R1-(N_total/n1)#i)##2/( (i/(n1+1))#(1-
(i/(n1+1))#n2 ) ));
By=sum( (R2-(N_total/n2)#j)##2/( (j/(n2+1))#(1-
(j/(n2+1))#n1 ) ));
B=Bx+By;

return (B);
finish;

/* Carrying out the test */
Tab=REPEAT(T(ranks),P,1);

R1=choose(permutations=0,..,Tab);
R2=choose(permutations=1,..,Tab);

R1g=R1[loc(R1^=.)];
R2g=R2[loc(R2^=.)];

R1z=shape(R1g,P, n1);
R2z=shape(R2g,P, n2);

test_st0=
  test_sta(T(ranks[1:n1]),T(ranks[(n1+1):N_total]),
  N_total, n1, n2);
Pval=0;

do i=1 to P by 1;
  B = test_sta(R1z[ i , ], R2z[ i , ], N_total, n1, n2);
```

```

if B >= test_st0 then Pval=Pval+1;
end;

Pval=Pval/P;

/* Definition of output */
x=(Pval || test_st0 || P);
cols={P_value test_statistic total_Pperms};
print x[colname=cols];

```

```

/* optional: Creation of an output dataset called results
*/
CREATE results FROM x[colname=cols];
APPEND FROM x;
CLOSE results;
/*****/

quit;
%MEND Permtest;

```

In order to use this macro the SAS dataset needs the variables GROUP and VALUE (in this order). The variable GROUP should have the values 0 and 1 as codes for the two groups. Thus, the data step is as follows:

```

DATA example1;
  INPUT group value @@;
CARDS;
0 154 0 155 0 158 0 159 0 161 0 163 0 177 0 183 0
192 0 219
1 171 1 172 1 178 1 179 1 184 1 185 1 186 1 194 1
196 1 223
;
RUN;

```

Here we use a placebo-controlled trial as an example. The raw data in the data step given above are reaction times (in msec) and were presented by Sedlmeier and Renkewitz ([14], p. 583). The placebo group is coded as 0, the active group as 1. In this example there are no ties. However, our program also works in the presence of ties. The macro can be invoked with the following statement:

```
%Permtest(example1);
```

Output

The first part of the output lists (Figure 1) the total sample size N_TOTAL as well as the groups' sample sizes N1 and N2. In the second part the p-value, the value of the test statistic for the observed data and the number of permutations (TOTAL_PERMS) are given.

	N_TOTAL	N1	N2
	20	10	10
	X		
P_VALUE	TEST_STATISTIC	TOTAL_PERMS	
0.0301479	121.86429	184756	

Figure 1: Output lists

Notes

Acknowledgement

The first author gratefully acknowledges support of this work by the Ministry for Education, Science, Youth and Culture of Rhineland-Palatinate for a Competence Center in Biomathematics.

Conflict of interest

None declared.

References

1. Lehmann EL. Nonparametrics: Statistical methods based on ranks. San Francisco, CA: Holden-Day; 1975. DOI: 10.1002/zamm.19770570922
2. Ludbrook J, Dudley H. Issues in biomedical statistics: statistical inference. Aust N Z J Surg. 1994;64:630-6. DOI: 10.1111/j.1445-2197.1994.tb02308.x
3. Neuhäuser M. Exact tests based on the Baumgartner-Weiß-Schindler statistic - a survey. Stat Papers. 2005;46:1-30. DOI: 10.1007/BF02762032
4. Ludbrook J, Dudley H. Why permutation tests are superior to t and F tests in biomedical research. Am Stat. 1998;52:127-32. DOI: 10.2307/2685470
5. Hunter MA, May RB. Some myths concerning parametric and nonparametric tests. Can Psychol. 1993;34(4):384-9. DOI: 10.1037/h0078860
6. Chen RS, Dunlap WP. SAS procedures for approximate randomization tests. Behav Res Methods Instrum Comput. 1993;25:406-9.
7. Good PI. Permutation tests. 2nd ed. New York: Springer; 2000.
8. Baumgartner W, Weiß P, Schindler H. A nonparametric test for the general two-sample problem. Biometrics. 1998;54:1129-35. DOI: 10.2307/2533862
9. Neuhäuser M. An exact two-sample test based on the Baumgartner-Weiß-Schindler statistic and a modification of Lepage's test. Communications in Statistics - Theory and Methods. 2000;29(1):67-78. DOI: 10.1080/03610920008832469
10. Bothner PP, Kähler WM. Programmieren in PROLOG: eine umfassende und praxisorientierte Einführung. Braunschweig: Vieweg; 1991
11. Neubert K, Brunner E. A studentized permutation test for the non-parametric Behrens-Fisher problem. Comput Stat Data Anal. 2007;51(10):5192-204. DOI: 10.1016/j.csda.2006.05.024

12. Bauer DF. Constructing confidence sets using rank statistics. *J Am Stat Assoc.* 1972;67(339):687-90. DOI: 10.2307/2284469
13. Neubert K. Das nichtparametrische Behrens-Fisher-Problem: ein studentisierter Permutationstest und robuste Konfidenzintervalle für den Shift-Effekt [PhD thesis]. Göttingen: University; 2006.
14. Sedlmeier P, Renkewitz F. *Forschungsmethoden und Statistik in der Psychologie*. München: Pearson Studium; 2008

Please cite as

Neuhäuser M, Schulz A, Czech D. A SAS/IML algorithm for an exact permutation test. *GMS Med Inform Biom Epidemiol.* 2009;5(2):Doc13.

This article is freely available from

<http://www.egms.de/en/journals/mibe/2009-5/mibe000092.shtml>

Published: 2009-03-16

Copyright

©2009 Neuhäuser et al. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en>). You are free: to Share – to copy, distribute and transmit the work, provided the original author and source are credited.

Corresponding author:

Markus Neuhäuser
Department of Mathematics and Technique,
RheinAhrCampus, Koblenz University of Applied Sciences,
Südallee 2, 53424 Remagen, Germany
neuhaeuser@rheinahrcampus.de