

Paper 275-2010

A Nonparametric Method to Assess Treatment Effects for Unbalanced Designs Using SAS/IML®

Michael Cunningham, University of Pittsburgh, Pittsburgh, PA

ABSTRACT

Two-way randomized block designs with non-normally distributed outcomes may be frequently encountered by researchers. The nonparametric test developed by Friedman can be used to compare more than two treatments when the blocks are completely balanced. Skillings and Mack proposed a general method for comparing treatments for incomplete blocks when the observations are missing arbitrarily. A program to calculate the Skillings-Mack test statistic and the p-value for the hypothesis of no treatment effect was developed using SAS/IML® in SAS version 9.1. Designed for users with intermediate SAS experience, the program reads the results of the unbalanced two-way design directly into IML for analysis. Use of this program will be demonstrated using quality reporting data from a leading healthcare provider.

INTRODUCTION

Randomized block designs are frequently used by researchers to compare treatment effects. Blocks can be individuals, plots of land, or litters of animals (Fleiss, 1986). The “treatment” may be an actual drug treatment, but treatment can be used as a general term for the grouping variable for which a response is measured. Assessments, tests, or methods of performing a task could define the treatment groups. Fleiss (1986) states that the goal of the randomized block design is “to control for the effects of prognostic variables on the variability in response to a treatment.” These “prognostic variables” could be the sex, age, or race of the individuals in the treatment groups being compared. Ideally, by minimizing the variability in the responses by blocking, any observed differences in the treatment group outcomes are likely to be due to the treatment and not some other factor like age, gender, or race.

Methods such as ANOVA can be used to assess the treatment effect in a randomized block design; however, one of the assumptions of the ANOVA is normally distributed errors. Nonparametric methods exist when this normality assumption may not hold. The Wilcoxon signed ranks test (Wilcoxon, 1945) can be used when there are only two treatment groups. With more than two treatments, the Friedman Test (Friedman, 1937) can be used to determine if the responses differ by treatment. One requirement of the Friedman Test is that the design is completely balanced, with no missing responses. Skillings and Mack (1981) proposed their “Friedman-Type” statistic as a method for testing treatment effectiveness in these incomplete block designs. Similar to the Friedman Test, the responses are ranked within the block, but Skillings and Mack weight the blocks to account for the unbalanced design (Skillings and Mack, 1981).

If the randomized block design is completely balanced, SAS procedures exist for determining the treatment effects. An analysis of variance (ANOVA) can be performed using PROC ANOVA. The nonparametric Friedman Test can be performed in PROC FREQ by specifying the “cmh2 scores=rank” option after TABLE. If some data are missing, PROC GLM can be used, again with the assumption of normality. But no SAS procedures or macros exist for calculating the Skillings-Mack statistic. A program to calculate the Skillings-Mack test statistic and the p-value for the hypothesis of no treatment effect was developed using SAS/IML in SAS version 9.1.

APPLICATION OF SKILLINGS-MACK STATISTIC

The National Committee for Quality Assurance (NCQA) is a non-profit organization “dedicated to improving health care quality” (NCQA, 2010a). The NCQA has developed a set of measures, the Healthcare Effectiveness Data and Information Set (HEDIS®) that are used by the majority of health plans across the country as a way to assess and compare health care delivery and quality by product line, state, region, hospital, or physician group (NCQA, 2010b). Measures such as lead screening in children, breast and cervical cancer screening in women, and comprehensive diabetes care are among the 2010 HEDIS measures (NCQA, 2010c).

Highmark, Inc., one of the leading health insurers in Pennsylvania, regularly analyzes these HEDIS rates as an important component of its quality management program. Some comparison populations (hospital or physician) may not meet the minimum sample size requirement for a specific HEDIS measure (for example, a provider may not have enough diabetics in his or her practice to form a “stable” rate), resulting in missing rates. Additionally, the rates in the different treatment groups may have skewed distributions. For these two reasons, the Skillings-Mack statistic

provides an ideal distribution-free method for comparing these HEDIS measure rates by treatment group. Use of the Skillings-Mack statistic to compare these performance rates by state will be demonstrated in a later paper section.

SKILLINGS-MACK (SM) STATISTIC

Consider a randomized block design with treatments $j = 1, \dots, k$, blocks $i = 1, \dots, n$, and responses X_{ij} (Figure 1).

		Treatment						
		1	.	.	j	.	.	k
Block	1	X_{11}	.	.	X_{1j}	.	.	X_{1k}
	\vdots							
	i	X_{i1}	.	.	X_{ij}	.	.	X_{ik}
	\vdots							
	n	X_{n1}	.	.	X_{nj}	.	.	X_{nk}

Figure 1. Randomized Block Design Notation

The calculation of the Skillings-Mack statistic is as follows (Hollander and Wolfe, 1999):

1. Calculate the vector of s_i , the number of observed values within block i
2. Determine the within block ranks, r_{ij}
 - For observed values: $r_{ij} = \text{rank of } X_{ij} \text{ in block } i$
 - If observation is missing: $r_{ij} = (s_i + 1)/2$
3. Calculate the weighted sum of centered ranks, A_j , where: $A_j = \sum_{i=1}^n \left(\frac{12}{s_i + 1} \right)^{1/2} \left(r_{ij} - \frac{s_i + 1}{2} \right)$
 - Compose matrix $A = (A_j, \dots, A_{k-1})$
4. Determine the covariance matrix for A :

$$\Sigma_0 = \begin{bmatrix} \sum_{t=2}^k \lambda_{1t} & -\lambda_{12} & -\lambda_{13} & \dots & -\lambda_{1,k-1} \\ -\lambda_{12} & \sum_{t \neq 2}^k \lambda_{2t} & -\lambda_{23} & \dots & -\lambda_{2,k-1} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ -\lambda_{1,k-1} & -\lambda_{2,k-1} & -\lambda_{3,k-1} & \dots & \sum_{t \neq k-1}^k \lambda_{k-1,t} \end{bmatrix}$$

- For $t \neq q = 1, \dots, k$ (k treatments), where $\lambda_{qt} = \lambda_{tq} =$ the number of blocks where treatments q and t are both observed
5. Calculate the Skillings-Mack statistic: $SM = A \Sigma_0^- A'$
 - Where Σ_0^- is a generalized inverse of Σ_0

The null hypothesis for the Skillings-Mack statistic is no difference in the τ_k treatment effects versus the alternative that the treatments are all not equal (Hollander and Wolfe, 1999):

$$H_0 : [\tau_1 = \dots = \tau_k]$$

$$H_A : [\tau_1, \tau_2, \dots, \tau_k \text{ not all equal}]$$

The null hypothesis is rejected if the value of the statistic is greater than or equal to some critical value. However, because there are a large number of combinations of unbalanced randomized block designs, there is no complete set of published tables for these critical values (except for a limited number of special cases (Hollander and Wolfe, 1999)). Skillings and Mack (1981) provide a large sample approximation of the p-value based on the chi squared distribution with $k-1$ degrees of freedom and a significance level α : The null hypothesis is rejected if $SM \geq X_{k-1, \alpha}^2$.

Skillings and Mack (1981) note that the p-values from the chi squared approximation are in most cases too conservative, particularly with type I α -errors of 0.01 or less. They recommend simulation to obtain the critical values for the SM statistic. Hollander and Wolfe (1999) discuss a method of simulating all of the possible rankings for the s_i observed values, while preserving the ranks of the missing observations. For each simulated ranking, the SM statistic is calculated and compared against the actual value of the SM statistic. The simulated p-value is the proportion of simulated SM statistics that are greater than or equal to the actual SM statistic, divided by the number of simulations. Details of this procedure will be shown in the example.

SKILLINGS-MACK EXAMPLE

The SAS/IML code developed to calculate the Skillings-Mack statistic will be demonstrated with the following unbalanced block design (SAS Institute, 2004). Skillings and Mack (1981) demonstrated their statistic with the data in Figure 2 (used with permission). Workers (blocks) were asked to assemble the same part using four different methods: A, B, C, and D (treatments). This randomized block design is unbalanced because the number of assembly times is not equal for all blocks. It is assumed that the missing times are unrelated to treatment or block (arbitrary).

		Assembly method (treatment)			
		A	B	C	D
Person (block)	1	3.2	4.1	3.8	4.2
	2	3.1	3.9	3.4	4.0
	3	4.3	3.5	4.6	4.8
	4	3.5	3.6	3.9	4.0
	5	3.6	4.2	3.7	3.9
	6	4.5	4.7	3.7	.
	7	.	4.2	3.4	.
	8	4.3	4.6	4.4	4.9
	9	3.5	.	3.7	3.9

Figure 2. Assembly Time Data (Skillings and Mack, 1981, used with permission)

DATA AND REQUIRED PROGRAM INPUTS

This program requires the input data in wide format, with one observation for each of the n blocks and k variables corresponding to the responses for each of the k treatments. The input DATA step is shown below:

```
data times ;
  input subject method_a method_b method_c method_d ;
  datalines ;
1 3.2 4.1 3.8 4.2
2 3.1 3.9 3.4 4.0
3 4.3 3.5 4.6 4.8
4 3.5 3.6 3.9 4.0
5 3.6 4.2 3.7 3.9
6 4.5 4.7 3.7 .
7 . 4.2 3.4 .
8 4.3 4.6 4.4 4.9
9 3.5 . 3.7 3.9
;
```

Only two additional modifications to the program are required: 1) definition of a macro variable containing the treatment variable names and 2) entering the number of simulations for the simulated p-value:

```
* Assign the list of treatment or group variables to "vars" ;
%let vars = method_a method_b method_c method_d ;

* Set the number of reps for simulated p-values ;
%let sims = 1000 ;
```

The macro variable `&vars` now contains the names of the treatment variables and the number of simulations has been stored in the macro variable `&sims`.

GETTING DATA INTO IML

The next portion of the program reads the treatment group variable data from the data set and stores it in the response matrix X.

```
* Start iml ;
proc iml ;
use times ;
read all var {&vars} into x ;
print x ;
```

The assembly time data are now stored in X:

X			
3.2	4.1	3.8	4.2
3.1	3.9	3.4	4
4.3	3.5	4.6	4.8
3.5	3.6	3.9	4
3.6	4.2	3.7	3.9
4.5	4.7	3.7	.
.	4.2	3.4	.
4.3	4.6	4.4	4.9
3.5	.	3.7	3.9

ASSIGNING RANKS TO DATA

The next portion of code is used to assign the missing observations to an arbitrary maximum value, because the ranking procedures used later will not work with missing values. Indicator matrices are then created to assign values of 0 or 1 to the missing and non-missing observations, respectively. The s_i vector of the number of observed values per block i , which is essential to the subsequent rank calculations, is printed. The final portion of code creates a matrix with missing values set to '99'.

```
* Need to change missings to arbitrary value ;
* All missings will be set to max value of x+100 ;
x_ind = x ;
mx = max(x) ;
indx = loc(x_ind=.) ;
x_ind[indx] =100+mx ;
* Create matrix 'c', 0=missing 1=not missing ;
c=x ;
ind = loc(c^=.|c=0) ;
c[ind] =1 ;
ind = loc(c=.) ;
c[ind] =0 ;
* Create 's', non-missings per row ;
s=c[,+] ;
print s ;

* Create 'c2', change missings to 99 ;
c2 = c ;
ind = loc(c2=0) ;
c2[ind] =99 ;
```

The vector of s_i (transposed from the actual output):

S	4	4	4	4	4	3	2	4	3
---	---	---	---	---	---	---	---	---	---

The value of s for block 7 is two, which can be seen in the original data, where assembly times for Methods B and C are the only observed times for the 7th person.

The program now determines the within block ranks, r_{ij} :

```

* Create 'rank' with ranks of 'x_ind', missings will be ranked highest ;
rank=c2 ;
do i=1 to nrow(x) by 1 ;
rank[i,]=ranktie(x_ind[i,]) ;
end ;
* Change rank of the missings to average of non-missing ranks ;
ranks2 =rank ;
do i=1 to nrow(x) by 1 ;
do j = 1 to ncol(x) by 1 ;
if c2[i,j]=99 then ranks2[i,j] = (s[i,1]+1)/2 ;
else ranks2[i,j] = rank[i,j] ;
end ;
end ;
* Print the final ranks ;
print ranks2 ;

```

The first four lines of the previous code assign the ranks within a block using the RANKTIE function. The missing values will be assigned the highest ranks (because they were arbitrarily set to a maximum value), and will therefore not interfere with the ranking of the other observed values. The correct rankings for the missing observations are then performed by assigning the value of $(s_i + 1)/2$. The final ranking of X, stored as `ranks2`, is printed:

RANKS2			
1	3	2	4
1	3	2	4
2	1	3	4
1	2	3	4
1	4	2	3
2	3	1	2
1.5	2	1	1.5
1	3	2	4
1	2	2	3

Note the position of the missing observations and the assignment of an average rank to those responses.

CALCULATE THE SKILLINGS-MACK STATISTIC

The next code uses the ranks in `ranks2` to calculate the weighted sum of centered ranks, A_j , and the matrix $A = (A_j, \dots, A_{k-1})$.

```

* Create matrix 'a1' ;
a1 = J(nrow(x),ncol(x),0) ;
do i=1 to nrow(x) by 1 ;
do j=1 to ncol(x) by 1 ;
a1[i,j] = sqrt(12/(s[i,1]+1))*(ranks2[i,j]-((s[i,1]+1)/2)) ;
end ;
end ;
* Sum the elements of 'a' ;
a = a1[+,] ;
print a ;
r = 1:(ncol(x)-1) ;
* 'a_final' is the sub-matrix used for the inverse calculation ;
a_final = a[{1},(r)] ;
print a_final ;

```

The previous code will output the following matrices:

A			
-14.1256	4.2812441	-4.281244	14.125598
A_FINAL			
-14.1256	4.2812441	-4.281244	

The covariance matrix must now be calculated. Recall that the off diagonal elements of the covariance matrix are the negative of the number of times a treatment group pair is observed. The following section is lengthy and relatively complex, but it is designed to do two things: 1) count the observed treatment pairs forming the covariance off diagonal elements (zees matrix) and 2) calculate the diagonal elements for the covariance matrix (z matrix):

```

* Create matrix 'y' for indicator of missings ;
y = J(nrow(x),ncol(x),0) ;
do i = 1 to ncol(x) by 1 ;
y[,i] = 1#(x[,i]^=.) ;
end ;
* Create matrix made of y ;
yall = J(nrow(x)*ncol(x),ncol(x)*2,0) ;
do i = 1 to ((ncol(x)-1)*(nrow(x))+1) by nrow(x) ;
do j = 1 to (ncol(x)*2) by 1 ;

* if j is even ;
if j/2 = round(j/2) then yall[i:i+(nrow(x)-1),j] = y[, (1/2)*j] ;
* else if j is odd ;
else yall[i:i+(nrow(x)-1),j] = y[, ((i+(nrow(x)-1))/nrow(x))] ;
end ;
end ;
* Create 'zall' and 'zees' ;
zall = J(nrow(x)*ncol(x),ncol(x),0) ;
zees = J(ncol(x),ncol(x),0) ;
do i = 1 to ((ncol(x)-1)*(nrow(x))+1) by nrow(x) ;
do j = 1 to ((ncol(x)*2)-1) by 2 ;
zall[i:i+(nrow(x)-1),(j+1)/2] = 1#(yall[i:i+(nrow(x)-1),j]=yall[i:i+(nrow(x)-1),j+1]& yall[i:i+(nrow(x)-1),j]^=0) ;
zees[((i+(nrow(x)-1))/nrow(x)),(j+1)/2] = zall[+,(j+1)/2] ;
* Reset 'zall' to zero ;
zall = J(nrow(x)*ncol(x),ncol(x),0) ;
end ;
end ;
do i = 1 to ncol(x) by 1 ;
do j = 1 to ncol(x) by 1 ;
if i = j then zeess[i,j]=0 ;
else zeess[i,j] = zeess[i,j] ;
end ;
end ;
* Sum the rows of 'zee' ;
zrow = zeess[+,] ;
* Create diagonal elements, z ;
z =zrow[, (r)] ;

```

The covariance matrix calculation continues with a module that loops through the elements of the matrix and inserts the diagonal elements from z or the off diagonal elements from zeess. The final covariance matrix, sigma, is printed to the output window in the last line:

```

* Initialize values for 'sigma' ;
sigma = J(ncol(x)-1,ncol(x)-1,0) ;
start ;
k = 0 ;
do i = 1 to ncol(x)-1 by 1 ;
k = k+1 ;

```

```

l = 0 ;
do j = 1 to ncol(x)-1 by 1 ;
l = l+1 ;
if i^=j then do ;
sigma[k,l]=--zees[k,l] ;
sigma[l,k]=--zees[l,k] ;
end ;
else do ;
sigma[l,l] = z[l,j] ;
end ;
end ;
end ;
end ;
finish ;
run ;
print sigma ;

```

SIGMA		
22	-7	-8
-7	21	-8
-8	-8	23

All of the elements required to calculate the Skillings-Mack statistic and the approximate p-value have now been calculated. The following code calculates SM and the p-value, using the PROBCHI function with $k-1$ degrees of freedom.

```

* Calculate Skillings' Mack (sm) from 'a_final' and inverse of 'sigma' ;
sm = a_final*inv(sigma)*a_final` ;
* Calculate the approximate p-value ;
p_value = 1-probchi(sm,ncol(x)-1) ;
print sm p_value ;

```

The SM statistic and p-value are output:

SM	P_VALUE
15.49	0.0014

The Skillings-Mack statistic is 15.49 and the approximate p-value is 0.0014, significant at the $\alpha = 0.05$ level. The null hypothesis of equal treatments is rejected, indicating a difference in the times for four assembly methods.

Looking at the weighted sum of centered ranks (the A_j) for each assembly method can show which have the higher or lower ranked observations. The final short section of code creates a SAS data set `sm_out` from these values, which can then be displayed by the user (graphically, tabular, etc.).

```

groups = J(ncol(x),1,0) ;
do i=1 to ncol(x) by 1 ;
groups[i,1] = i ;
end ;
output = groups||a` ;
create sm_out from output[colname={method weighted_sum}] ;
append from output ;

```

SAS data set `sm_out` :

Obs	METHOD	WEIGHTED_ SUM
1	1	-14.1256
2	2	4.2812
3	3	-4.2812
4	4	14.1256

It appears as if Method 4 (D) has the longest assembly times, while Method 1 (A) has the shortest times.

SIMULATED P-VALUE

As discussed earlier, the approximate p-value may be too conservative. The basic method for determining a simulated p-value is as follows. For each simulation, the ranks are randomly ordered within a block. The ranks and order of the missing observations are preserved. The Skillings-Mack test statistic is then calculated. The number of times the simulated SM statistic is greater than or equal to the observed statistic is counted over all simulations. The simulated p-value is this count divided by the number of simulations (Hollander and Wolfe, 1999).

The remaining program code is used to simulate the p-value for the given unbalanced randomized block design:

```
* Generate simulated p-value ;
simulated = J(&sims,4,0) ;
simulated[,2] = sm ;
```

The simulated matrix is initialized, with rows equal to the number of simulations specified in the macro variable &sims. The second column will contain the observed SM statistic values. The next section is repeated for each simulation. It is fairly involved, but is designed to assign ranks to the observed values and then randomly order them within each block, while keeping the missing ranks the same.

```
* Start the simulations ;
do w = 1 to &sims by 1 ;

rank=c2 ;
do i=1 to nrow(x) by 1 ;
rank[i,]=ranktie(x_ind[i,]) ;
end ;
* put missings back in ;
do i=1 to nrow(x) by 1 ;
do j = 1 to ncol(x) by 1 ;
if c2[i,j]=99 then rank[i,j] = . ;
else rank[i,j] = rank[i,j] ;
end ;
end ;
sim_ranks1 = J(nrow(x)*ncol(x),2,0) ;
do i=1 to nrow(x) by 1 ;
sim_ranks1[((ncol(x))*i)-(ncol(x)-1):(ncol(x))*i,1] = (rank[i,])^ ;
sim_ranks1[((ncol(x))*i)-(ncol(x)-1):(ncol(x))*i,2] = i ;
end ;
do i=1 to (nrow(x)*ncol(x)) by 1 ;
if sim_ranks1[i,1]=. then sim_ranks1[i,2] = . ;
end ;
```

The next code uses the UNIFORM function to assign random numbers to each element within a block. The CALL SORT then sorts the uniform random numbers, which will then randomly order the ranks within a block.

```
* Create uniform random numbers 'rand' ;
indicator = J(nrow(x)*ncol(x),1,0) ;
rand = uniform(indicator) ;
* Sort on random numbers ;
sim_out = sim_ranks1||rand ;
call sort(sim_out,{2 3}) ;
sim_out = sim_out[(ncol(ind)+1):(nrow(x)*ncol(x)),1:2] ;
* Insert missings back into 'sim_out' ;
* Create the missing matrix ;
missings= J(ncol(ind),2,.) ;
* Put missings back in ;
do i = 1 to ncol(ind) by 1 ;
sim_out = insert(sim_out,missings[i,],ind[i]) ;
end ;
sim_ranks2 = J(nrow(x),ncol(x),0) ;
do i = 1 to nrow(x) by 1 ;
sim_ranks2[i,] = (sim_out[((ncol(x))*i)-(ncol(x)-1):(ncol(x))*i,1])^ ;
end ;
ranks3 = sim_ranks2 ;
```

The original ranks of the missing observations are put back into the rank matrix:

```

* Put original 'missing' ranks back in ;
do i=1 to nrow(x) by 1 ;
do j = 1 to ncol(x) by 1 ;
if c2[i,j]=99 then ranks3[i,j] = ranks2[i,j] ;
else ranks3[i,j] = ranks3[i,j] ;
end ;
end ;
print ranks3 ;

```

Two of the `ranks3` matrices, the randomly ordered ranks, are shown below to illustrate how the ranks are randomly ordered:

RANKS3				RANKS3			
2	4	3	1	4	2	3	1
4	1	2	3	3	2	4	1
3	2	4	1	2	1	4	3
1	2	4	3	3	2	4	1
1	2	4	3	1	2	4	3
2	3	1	2	3	2	1	2
1.5	1	2	1.5	1.5	2	1	1.5
3	2	1	4	3	2	1	4
1	2	3	2	2	2	3	1

If the `ranks3` matrices are compared, it can be seen that the ranks have been randomly ordered within a block for the observed responses, while the rank and position of the missing responses are the same (1.5 for the 7th block).

The program continues with the calculation of the SM statistic for each of the `ranks3` matrices. The entire section of code will not be shown here, because it is identical to the earlier portion of the program, except that the SM statistic for each simulation is output to the first column of the `simulated` matrix along with the simulation iteration, `w`. The “end” statement is the end of the simulation loop.

```

simulated[w,1] = a_final*inv(sigma)*a_final` ;
simulated[w,4] = w ;
end ;

```

The simulated p-values are then compared to the observed SM test statistic. If the simulated SM statistic is greater than or equal to the observed SM statistic, an indicator variable is set equal to 1. The next few lines of code do this:

```

* Compare the simulated to the actual ;
do i = 1 to &sims by 1 ;
* Indicator = 1 if the simulated value is equal to or greater than actual S-M ;
if (simulated[i,1] - simulated[i,2]) > -0.000000001 then simulated[i,3] = 1 ;
else simulated[i,3] = 0 ;
end ;
print simulated ;

```

The first 10 rows of the `simulated` matrix are shown here:

SIMULATED			
1.5638472	15.493049	0	1
3.626647	15.493049	0	2
1.043264	15.493049	0	3
1.8586492	15.493049	0	4
2.5055418	15.493049	0	5
2.8180394	15.493049	0	6
4.4134246	15.493049	0	7
3.6312743	15.493049	0	8
0.3107932	15.493049	0	9
0.1883529	15.493049	0	10

Column one shows the simulated SM statistics for each random ordering of the ranks. The second column is the observed SM value and the third column is the indicator that equals "1" if the simulated SM is greater than or equal to the observed value. The iteration is shown in the final column.

The `simulated` matrix will typically not be shown (this would print too many lines for a large number of simulations), so the results are summarized in one final matrix, `simulated_p`, which contains three columns: the sum of the indicators, the number of simulations, and the p-value. The final line quits the PROC IML session:

```
* Sum the ones and calculate the simulated p-value ;
simulated_p = simulated[+,3] || &sims || (simulated[+,3]/&sims) ;
print simulated_p ;
quit ;
```

The `simulated_p` output:

SIMULATED_P		
0	1000	0

Of the 1000 simulated randomly ordered rank matrices, none produced a SM test statistic as extreme as the observed SM test statistic, indicating that there does appear to be a difference in assembly methods (p -value = 0.000).

Skillings and Mack (1981) present some selected critical values for the SM statistic and probabilities, α , for balanced, incomplete block designs. In order to validate the simulated p -values obtained using this program, a set of data was created to match one of the entries in their table. This hypothetical dataset contains 9 blocks with 3 treatments, but only two treatments were observed within each block. This corresponds to the $n=9$, $p=6$, $\lambda=3$ incomplete block design in their table. The hypothetical data were manipulated to obtain a SM statistic equal to 6.222. From the table in Skillings and Mack (1981), this configuration and critical value would result in an alpha of 0.0824.

With 5000 simulations of the SM statistic with this dataset, the following output is obtained:

SM		
6.222		
SIMULATED_P		
409	5000	0.0818

The p -value of 0.0818 from this program is very close to the p -value of 0.0824 in Skillings and Mack (1981).

SKILLINGS-MACK COMPARED TO ANOVA

One potential benefit of using the SM statistic to assess treatment effects is the potential increased sensitivity in detecting differences in treatments. Using their assembly time data from Figure 2, Skillings and Mack (1981) noted that the p -value was larger for the F test comparing assembly times than the p -value from the SM statistic. To reproduce this result, the original data was first reshaped into long format. Using PROC GLM with the `model time = method person` statement, the output for the F test of a treatment effect (method) is obtained:

Source	DF	Type III SS	Mean Square	F Value	Pr > F
method	3	1.32490880	0.44163627	3.75	0.0274
person	8	3.40677387	0.42584673	3.62	0.0093

The p -value for the effect of method (treatment) is 0.0274, which is greater than the p -value of 0.0014 obtained from the SM statistic.

HEDIS RATE COMPARISON USING THE SKILLINGS-MACK STATISTIC

A typical analysis of HEDIS rates will be used as an additional example of the Skillings-Mack statistic. Consider the hypothetical data in Figure 3. Five HEDIS measure rates were collected for five different states; rates closer to 1.00 indicate better performance on a measure. Some of the measures are missing, which could be due to an insufficient sample size for the reporting of the individual rate or for some other unknown reason. In either case, it is assumed that the causes for the missing measure data are arbitrary.

		State (treatment)				
		1	2	3	4	5
HEDIS® Measure (block)	1	.	.	0.836	0.937	0.839
	2	0.708	0.638	.	0.707	.
	3	0.577	.	0.594	0.678	0.634
	4	0.700	0.694	0.660	0.775	0.649
	5	0.429	0.333	0.418	0.852	0.353

Figure 3. Hypothetical HEDIS Measure Rates by State

The goal of the analysis is to test whether or not there are differences in the states, which can be done using the SAS PROC/IML program. The output of the analysis is shown below:

SM	P_VALUE
8.390	0.078
SIMULATED_P	
278	5000 0.056

The SM statistic (SM) equals 8.39, with an approximate P_VALUE of 0.078, indicating there is no difference in the measure rates by state. With 5000 simulations, the p-value (SIMULATED_P) is 0.056, which is still not significant, but just marginally greater than the typical 0.05 level. It appears that there could be some difference in the measure rates by state. The weighted sum of centered ranks can be compared to get a sense of which states are performing better or worse on the five measures:

State	Weighted_Sum
1	2.23669
2	-4.56048
3	-3.92086
4	9.71270
5	-3.46804

It appears as if State 4 performed best on the five measures, while State 2 performed the worst (WEIGHTED_SUM values are 9.71 and -4.56, respectively). These weighted sums are consistent with the data in Figure 3, where State 4 appears to have the highest rates. In this example, while the SM test did not indicate a statistically significant difference in the measure rates by state, the weighted sums provide a useful descriptive comparison tool.

CONCLUSION

Many methods exist for analyzing treatment effects in a completely balanced, randomized block design. If the design is unbalanced, there are fewer options for analyzing the data, particularly when the normality assumption may not hold. The Skillings-Mack statistic provides researchers with a distribution-free test for treatment effects for these unbalanced randomized block designs, and in certain situations, the statistic may even be more powerful than traditional parametric methods. Using the PROC/IML code demonstrated in this paper, researchers will be able to calculate the Skillings-Mack test statistic, the approximate and simulated p-values, and determine if there is a difference in the treatment groups.

REFERENCES

- Fleiss JL. (1986), *Design and Analysis of Clinical Experiments*, New York, NY: John Wiley & Sons.
- Friedman M. (1937), "The use of ranks to avoid the assumption implicit in the analysis of variance," *Journal of the American Statistical Association* 32:675-701.
- Hollander M and Wolfe DA. (1999), *Nonparametric Statistical Methods*, New York, NY: John Wiley & Sons.
- National Committee for Quality Assurance. (2010a), HEDIS® & Quality Measurement. Available at: <http://www.ncqa.org/tabid/675/Default.aspx>. Last accessed February 25, 2010.
- National Committee for Quality Assurance. (2010b), HEDIS® & Quality Measurement. Available at: <http://www.ncqa.org/tabid/59/Default.aspx>. Last accessed February 25, 2010.
- National Committee for Quality Assurance. (2010c), *HEDIS® 2010 Volume 2: Technical Specifications*, Washington, DC: National Committee for Quality Assurance.
- SAS Institute Inc. (2004), *SAS/IML® 9.1 User's Guide*, Cary, NC: SAS Institute Inc.
- Skillings JH and Mack GA. (1981), "On the use of a Friedman-type statistic in balanced and unbalanced block designs," *Technometrics* 23:171-177.
- Wilcoxon F. (1945), "Individual comparisons by ranking methods," *Biometrics* 1:80-83.

ACKNOWLEDGMENTS

Development of this program was supported through an agreement between Highmark, Inc. and the University of Pittsburgh, Department of Biostatistics. I would like to thank Linda Ruhl of Highmark, Inc. and Dr. Gary Marsh of the University of Pittsburgh for guidance and assistance with this project. I would also like to thank Dr. John H. Skillings for permission to use data from his original article in this paper.

CONTACT INFORMATION

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

Michael Cunningham
University of Pittsburgh, Department of Biostatistics, Graduate School of Public Health
A440 Crabtree Hall, 130 DeSoto Street
Pittsburgh, PA 15261
Phone: (412) 624-3039
Fax: (412) 624-9969
E-mail: mac20@pitt.edu
Web: <http://www.biostat.pitt.edu/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.