Supplemental Material

SAS code to perform the re-randomization test for treatment differences in mRS scores (or other ordinal outcomes)

```
/* this program calculates the indices for treatment difference for the mRS and performs a
   permutation test of treatment differences.   It assumes that data will include two variables:
   1)  TRT which is a 0/1 indicator variable for treatment, and 2) MRS which is the modified
    Rankin Score.  It would be wise to create a dataset that only contains these two variables
    (reduces the chance that a variable    name in the dataset would be the same as a variable
    that was used in the program.   */



/* the data step calculates the percent that is better */
data perm;
        set tmp end=last;  /* data set with the data is inserted here */

        retain
                a0 a1 a2 a3 a4 a5
                b0 b1 b2 b3 b4 b5 num_g0 ;

        array a a0 a1 a2 a3 a4 a5;
        array b b0 b1 b2 b3 b4 b5;

        if _n_=1 then do;
                a0=0; a1=0; a2=0; a3=0; a4=0; a5=0;
                b0=0; b1=0; b2=0; b3=0; b4=0; b5=0;
                num_g0=0;
        end;

        if trt=0 then num_g0=num_g0+1;

        if trt=0 then a[mrs+1]=a[mrs+1]+1;
                else b[mrs+1]=b[mrs+1]+1;


        if last=1 then do;
                atot=a0+a1+a2+a3+a4+a5;
                btot=b0+b1+b2+b3+b4+b5;
                better =
                        a0*b1 + a0*b2 + a0*b3 + a0*b4 + a0*b5 +
                        a1*b2 + a1*b3 + a1*b4 + a1*b5 +
                        a2*b3 + a2*b4 + a2*b5 +
                        a3*b4 + a3*b5 +
                        a4*b5 ;
                worse =
                        b0*a1 + b0*a2 + b0*a3 + b0*a4 + b0*a5 +
                        b1*a2 + b1*a3 + b1*a4 + b1*a5 +
                        b2*a3 + b2*a4 + b2*a5 +
                        b3*a4 + b3*a5 +
                        b4*a5 ;
                same = a0*b0 + a1*b1 + a2*b2 + a3*b3 + a4*b4 + a5*b5;

                ratio=better/(better+worse);
                dif_ratio=abs(ratio-.5);

                m=1;

                pct_better=better/(better+same+worse);
                pct_same=same/(better+same+worse);
                pct_worse=worse/(better+same+worse);

                format pct_better pct_same pct_worse nlpct5.2;

                output;
        end;
        keep a0-a5 b0-b5 atot btot better same worse pct_better pct_same pct_worse ratio
dif_ratio num_g0 m;
        run;

proc print data=perm;
        var a0-a5 b0-b5 better same worse pct_better pct_same pct_worse ratio num_g0;
        title 'Listing of the number of people in each mRS category for treatment 0 (A0-A5),';
        title2 'treatment 1 (B0-B5), the number of pairs that are better (BETTER) and worse
(WORSE),';
```

```
        title3 'the ratio that are better as a function of the non-ties (BETTER/(BETTER+WORSE))'
        title4 'and the number in TRT=0 (NUM_G0)';
;
        run;
/* now re-randomize */

/* assign a random number for ordering */
data reran;
        set tmp;
        length trt mrs reran_rep t 4;
        keep trt mrs reran_rep t m;

        m=1;


        /* this is the "random" seed.   This can be any large number.   If it is set
        to zero, the program will create a randomly selected number.   The advantage of
        this is runs of the program are independent.   The disadvantage is that
        running the program on the same data set will provide slightly different answers
        (not the case if a large number is entered, which will provide completely identical
        estimates */

        retain seed 8382732;

        do reran_rep=1 to 10000;  /* number of replications of the re-randomzation.   Should be
                                                         at least 5000 */
                t=ranuni(seed);
                output;
        end;
        run;

/* randomly order the observations */
proc sort data=reran;
        by reran_rep t;
        run;

/* now merge in the PERM file to get the number treated with TRT=0 */
data reran2;
        merge reran perm;
        by m;

        keep trt mrs reran_rep num_g0;

        run;


/* now randomly assign treatments */
data reran3;
        set reran2;
        by reran_rep;

        keep reran_rep trt mrs ;
        length reran_rep trt mrs 4;

        retain nobs;

        if first.reran_rep then nobs=0;

        nobs=nobs+1;

        /* now randomly assign treatment */
        if nobs<=num_g0 then trt=0; else trt=1;

run;
/* now get ratios from each of the reran_reps */
data perm2;
        set reran3;
        by reran_rep;

        retain
                a0 a1 a2 a3 a4 a5
                b0 b1 b2 b3 b4 b5 ;

        array a a0 a1 a2 a3 a4 a5;
        array b b0 b1 b2 b3 b4 b5;

        if first.reran_rep then do;
                a0=0; a1=0; a2=0; a3=0; a4=0; a5=0;
                b0=0; b1=0; b2=0; b3=0; b4=0; b5=0;
        end;

        if trt=0 then a[mrs+1]=a[mrs+1]+1;
                else b[mrs+1]=b[mrs+1]+1;


        if last.reran_rep then do;
                atot=a0+a1+a2+a3+a4+a5;
                btot=b0+b1+b2+b3+b4+b5;
                better =
                        a0*b1 + a0*b2 + a0*b3 + a0*b4 + a0*b5 +
                        a1*b2 + a1*b3 + a1*b4 + a1*b5 +
                        a2*b3 + a2*b4 + a2*b5 +
                        a3*b4 + a3*b5 +
                        a4*b5 ;
                worse =
                        b0*a1 + b0*a2 + b0*a3 + b0*a4 + b0*a5 +
                        b1*a2 + b1*a3 + b1*a4 + b1*a5 +
                        b2*a3 + b2*a4 + b2*a5 +
                        b3*a4 + b3*a5 +
                        b4*a5 ;

                reran_ratio=better/(better+worse);
                diff_rr_ratio = abs(reran_ratio - .5);

                m=1;

                output;
        end;
        length reran_rep reran_ratio diff_rr_ratio 4;
        keep reran_rep reran_ratio diff_rr_ratio m;
        run;
proc chart data=perm_both;
        hbar reran_ratio diff_rr_ratio;
        title 'Distribution of the score under the null';
        run;
/*  now get the number with bigger deviations */
data perm_both;
        merge perm perm2;
        by m;
        if dif_ratio>diff_rr_ratio then bigger=0; else bigger=1;
        run;
/* now get p-value */
proc means data=perm_both mean;
        var bigger;
        title 'the p-value is the proportion with bigger';
        output mean=pr_perm out=perm_test;
        run;
```