The Stata Journal (2019)
19, Number 4, pp. 803–819 DOI: 10.1177/1536867X19893624

Permutation tests for stepped-wedge cluster-randomized trials

Jennifer Thompson
London School of Hygiene and Tropical Medicine
London, UK
jennifer.thompson@lshtm.ac.uk

Calum Davey
London School of Hygiene and Tropical Medicine
London, UK
calum.davey@lshtm.ac.uk

Richard Hayes
London School of Hygiene and Tropical Medicine
London, UK
richard.hayes@lshtm.ac.uk

James Hargreaves
London School of Hygiene and Tropical Medicine
London, UK
james.hargreaves@lshtm.ac.uk

Katherine Fielding
London School of Hygiene and Tropical Medicine
London, UK
katherine.fielding@lshtm.ac.uk

Abstract. Permutation tests are useful in stepped-wedge trials to provide robust statistical tests of intervention-effect estimates. However, the `permute` command does not produce valid tests in this setting because individual observations are not exchangeable. We introduce the `swpermute` command, which permutes clusters to sequences to maintain exchangeability. The command provides additional functionality for performing analyses of stepped-wedge trials. In particular, we include the `withinperiod` option, which performs the specified analysis separately in each period of the study with the resulting period-specific intervention-effect estimates combined as a weighted average. We also include functionality to test nonzero null hypotheses to aid in the construction of confidence intervals. Examples of the application of `swpermute` are given using data from a trial testing the impact of a new tuberculosis diagnostic test on bacterial confirmation of a tuberculosis diagnosis.

Keywords: st0577, swpermute, stepped wedge, cluster randomized, permutation test, randomization test

1 Introduction

Permutation tests are a commonly used nonparametric statistical technique for calculating p-values without making distributional assumptions (Pitman 1937; Eden and Yates 1933; Fisher 1971). In individually randomized trials, they are used because they make no distributional assumptions, provide exact p-values and confidence intervals, and do not rely on large-sample approximations (Ernst 2004); the `permute` command provides an intuitive and simple way to perform permutation tests in this scenario.

While the benefits of permutation tests hold for more complex randomized designs, such as stepped-wedge cluster-randomized trials (SW-CRTs), `permute` cannot perform a valid test for these complex designs. In a cluster-randomized trial (CRT), the allocation of clusters of individuals, such as villages or hospital wards, is randomized. An SW-CRT is a CRT run over a number of periods. Clusters are randomized to sequences, where each sequence receives the control condition for a different number of periods and then receives an intervention condition for the remaining periods of the trial (figure 1). Parametric analysis of SW-CRTs requires specification of the correlation structure over time within cluster. This can be difficult to prespecify, and results are sensitive to misspecification (Thompson et al. 2017). Therefore, permutation tests, which do not require specification of correlation structures, are appealing (Hayes and Moulton 2009; Thompson et al. 2018; Ji et al. 2017; Wang and De Gruttola 2017).

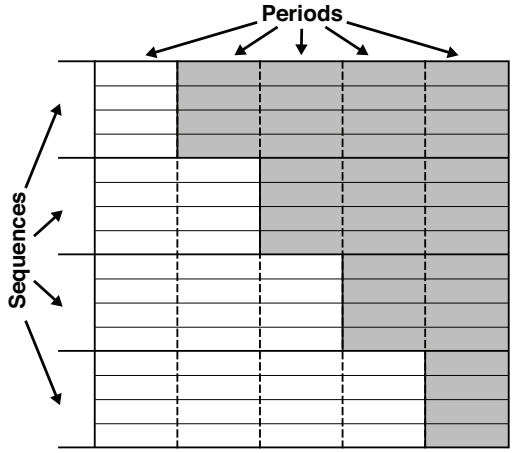


Figure 1. Schematics of an SW-CRT. White = time in control condition. Gray = time in intervention condition.

Here we introduce a new command, `swpermute`, that allows specification of clustering and allocation to a sequence of intervention conditions to enable use with CRTs in general but with a particular focus on SW-CRTs. In the next section, we provide an overview of permutation tests for SW-CRTs. In section 3, we outline the syntax of the `swpermute` command. In section 5, we demonstrate the use of `swpermute` with two examples.

2 Technical details

The `swpermute` command is designed for trials with two treatment conditions; usually, they will be control and intervention conditions, so we will use this terminology throughout this article. In this section, we will summarize the permutation test and show how it is implemented in `swpermute`.

2.1 Permutation tests with individual randomization

Details of the permutation test can be found elsewhere, such as in Good (2005); here, we briefly summarize.

In an individually randomized trial, we have a sample of observations, half of which were collected under a control condition and half under an intervention condition. We want to know whether the control and intervention conditions result in different distributions of outcomes. If there is truly no difference between the two conditions, then assignment of observations to each condition is arbitrary, and for any set of assignments of the observations to the control and intervention conditions, we can estimate an intervention effect. By repeating this process for each unique assignment of observations to conditions, we obtain the exact distribution of the intervention-effect estimator under the null hypothesis of no effect. The p-value, defined as the probability of the observed data if there is no intervention effect, is then given as the proportion of permuted intervention effects the same as or more extreme than that observed.

Monte Carlo permutations

This process is computationally simplified by randomly sampling a number of permutations from all possible permutations with or without replacement, a process known as Monte Carlo permutations (Good 2005). The p-value calculated may differ when the process is repeated with a different set of permutations.

Constructing confidence intervals

Confidence intervals are created by finding the boundaries of hypothesized intervention effects that lead to two-sided p-values less than the α level. One way to identify the confidence limits is to test several hypothesized intervention effects to see whether the p-value is larger or smaller than α.

A hypothesized intervention of $\theta = \theta_A$ is tested by first subtracting θ_A from observations collected in the intervention condition, and then running the permutation test as described above to get a p-value (Good 2005; Rigdon and Hudgens 2015). The random-number seed should be set to the same seed as the original analysis so that one set of permutations are used throughout the analysis, allowing the confidence intervals and p-value to coincide with one another.

2.2 Extending permutation tests to SW-CRTs

Two assumptions are required for permutation tests to be valid.

First, permutation tests test equivalence of distributions between the conditions. This means they will return a small p-value if either the means or the variances of outcomes differ. The effect of an intervention varying between observations is an example of the latter.

Second, permutation tests assume exchangeability of observations. This means that any assignment of observations to the conditions is equally likely. In the context of SW-CRTs, exchangeability holds for the assignment of clusters to sequences but will not hold at the individual observation level. Clusters should therefore be permuted between sequences. The `permute` command permutes individual observations, so it is not a valid test for SW-CRTs. The `swpermute` command permutes clusters to sequences, so it is valid for SW-CRTs (and CRTs with other designs).

2.3 Selecting an intervention-effect estimator for a stepped-wedge trial

Permutation tests provide a p-value and confidence intervals for a given intervention-effect estimator. A key design feature of all SW-CRTs is that the intervention effect is confounded with time. Therefore, the chosen estimator must account for this confounding either by adjusting for period effects or by conditioning on periods.

To adjust for period effects, generalized linear models or generalized linear mixed models can be used (Bellan et al. 2015; Ji et al. 2017; Wang and De Gruttola 2017).

To condition on the period, the analysis can be conducted within each period with resulting within-period estimates combined as a weighted average. More details of this method, also known as a vertical analysis, are given in Thompson et al. (2018). Any analysis that can be used for a parallel CRT could be used within each period; for example, Thompson et al. (2018) suggested using a cluster-level analysis in each period.

The overall intervention-effect estimate can be estimated more accurately by using appropriate weights for each period (Hayes and Moulton 2009). Periods can be weighted by the imbalance in the number of clusters in the control and intervention conditions (Matthews and Forbes 2017) or by the precision of within-period estimates (Thompson et al. 2018).

3 The swpermute command

The `swpermute` command runs a permutation test for SW-CRTs using any user-specified analysis to estimate the intervention effect. The algorithm identifies sequences in the data and permutes clusters between these sequences. Like the `permute` command, `swpermute` performs Monte Carlo permutations with replacement. The specified anal-

ysis can be run either across all periods in the study or in each period with results combined as a weighted average. Users can specify hypothesized intervention effects to construct confidence intervals.

3.1 Data requirements

The `swpermute` command requires specification of a clustering variable identified in `cluster()`, a period variable identified in `period()`, and an intervention variable identified in `intervention()`. Together, these variables define the design of the trial. The intervention must be specified as a binary variable, where `0` and `1` represent the control and intervention conditions, respectively. All observations within each value of `cluster()` must have the same value of `intervention()` in each `period()`, or the command will return an error. If the intervention variable contains missing values for all observations in a period in a cluster, then this is assumed to be part of the sequence, for example, as a washout period, and the missing value will be permuted. Otherwise, observations with `intervention()` missing will be excluded from the analysis.

The data should be in long format, with observations in each period given in different rows of the data.

3.2 Syntax

The syntax of the `swpermute` command is as follows:

`swpermute` *exp*`, cluster(`*varname*`) period(`*varname*`) intervention(`*varname*`)`
`[reps(#) left|right strata(`*varlist*`) saving(`*filename*`[ , `*suboptions*`])`
`null(`*numlist*`) outcome(`*varname*`) seed(#) withinperiod`
`weightperiod(`*weightperiod*`) nodots level(#)]:` *command*

exp specifies the result to be collected from results stored by the execution of *command*. Examples are `r(mu_1) - r(mu_2)`, the mean difference estimated by `ttest`, and `_b[`*varname*`]`, a coefficient estimate from a regression model.

3.3 Options

Main

`cluster(`*varname*`)` specifies the variable identifying the clusters. `cluster()` is required and must be a numeric variable. All observations within each `cluster()` must have the same value of `intervention()` in each `period()`. Observations with `cluster()` missing will be excluded from the analysis.

`period(`*varname*`)` specifies the variable identifying the periods. `period()` is required and must be a numeric variable. Observations with `period()` missing will be excluded from the analysis.

`intervention(`*varname*`)` specifies the variable identifying the intervention assignment. `intervention()` is required and must be a binary variable. Treatment of missing values is described above.

`reps(`#`)` specifies the number of permutations to perform. The default is `reps(1000)`.

`left` or `right` requests that one-sided p-values be computed. If `left` is specified, the p-value reported is the proportion of permutations where *exp* gives a value less than or equal to the observed value. If `right` is specified, the p-value reported is the proportion of permutations where *exp* gives a value greater than or equal to the observed value. The default is two-sided p-values, where the p-value reported is the proportion of permutations where *exp* is the same as or further from zero than the observed value.

Options

`strata(`*varlist*`)` specifies that the permutations be performed within each stratum defined by the values of *varlist*. This option should be used if randomization of clusters was stratified (Ernst 2004).

`saving(`*filename* [`,` *suboptions*]`)` creates a `.dta` file consisting of a row for each permutation for each value in `null()`. The file consists of three variables containing the `null()` value being tested, the observed value of *exp* for that null value, and values of *exp* for each permutation. A new *filename* is required unless `replace` is specified. The suboption `double` specifies that results should be stored in `double` precision; the default is to store results as `float`. The suboption `every(`#`)` writes results to the file every # permutations; this will allow partial recovery of results should the command not complete running.

`null(`*numlist*`)` specifies a list of values to test as the null hypothesis. For each value specified, the value will be subtracted from the variable specified in `outcome()` if the variable defined in `intervention()` is equal to 1. The permutation test is run on this modified dataset to calculate a p-value. The random-number seed is reset for each value tested. This option should be used only with cluster-period-level or continuous outcomes. The null values are assumed to be on the same scale as the outcome (for example, risk differences if the outcomes are cluster-period risks). Ratios such as risk ratios or odds ratios should be given on the log scale. The default is `null(0)`. When values other than the default are specified, the option `outcome(`*varname*`)` is required.

`outcome(`*varname*`)` specifies the outcome variable from which the null values will be subtracted. The outcome variable is assumed to be on the same scale as the null values. For example, `outcome()` should contain risks if `null()` gives risk differences or log risks if `null()` gives log risk ratios. `outcome()` is required if the option `null()` is given with a value other than 0.

`seed(#)` sets the random-number seed. Specifying this option is equivalent to typing `set seed` # prior to calling `swpermute`. If no seed is specified, then because of the random selection of permutations, `swpermute` will return different results each time it is run.

Within-period analysis

`withinperiod` specifies that a within-period analysis should be performed. *command* is run within each unique value of the variable specified in `period()`, and the resulting values of *exp* are combined as a weighted average using the weights specified in `weightperiod()`. This option allows a "vertical analysis" in stepped-wedge trial literature.

`weightperiod(`*weightperiod*`)` specifies the weights to be used if `withinperiod` is specified. This option is required only when `withinperiod` is specified and is one of the following:

`weightperiod(N)` specifies that periods are weighted by the number of clusters in the control and intervention conditions as

$$w_j = \left(\frac{1}{s_{0j}} + \frac{1}{s_{1j}} \right)^{-1}$$

where s_{0j} and s_{1j} are the numbers of clusters in the control condition and the intervention condition, respectively, in period j. This is the default and is recommended if the total variance is not expected to vary between periods (Matthews and Forbes 2017).

`weightperiod(none)` specifies that each period is given equal weight, so the weight $w_j = 1$ for all periods j.

`weightperiod(variance` exp_2`)` specifies that each period is weighted by the inverse of the statistic exp_2 stored by the execution of *command*. That is,

$$w_j = exp_{2j}^{-1}$$

exp_{2j} is assumed to be the variance of the estimate from the jth period. This specification is suggested by Thompson et al. (2018) when the variance of the outcome is expected to vary between periods.

Reporting

`nodots` suppresses display of the dots at the completion of each permutation. By default, one `.` is displayed for each successful permutation. A red `x` is displayed if *command* returns an error or if the statistic in *exp* is missing for a permutation.

`level(#)` specifies the confidence level, as a percentage, for the confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.8 Specifying the width of confidence intervals**.

3.4 Stored results

`swpermute` stores the following in `r()`:

Scalars

`r(N_cluster)`	number of clusters being permuted
`r(N_strata)`	number of strata if the `strata()` option is specified
`r(obs_value)`	value of *exp* observed in the original data
`r(N_reps)`	number of permutations

Matrices

`r(design)`	matrix of `0` and `1` values showing the design of the SW-CRT
`r(obs_period)`	value of *exp* observed in the original data within each value of `period()` if a within-period analysis is specified
`r(p)`	*p*-values with their confidence intervals for each null value

3.5 The dialog box

The `swpermute` command can be used both as a coded command and through a drop-down dialog box as shown in the appendix. To install the dialog box, run the following commands:

```
. window menu append submenu "stUser" "&Cluster RCTs"
. window menu append item "Cluster RCTs" "Permute for stepped-wedge trials
> (&swpermute)" "db swpermute"
. window menu refresh
```

Running these commands from within Stata will install only the dialog box for the current session of Stata. To install the menus permanently, place the above commands into your `profile.do` file. See [GSW] **B.3 Executing commands every time Stata is started**, [GSM] **B.1 Executing commands every time Stata is started**, or [GSU] **B.1 Executing commands every time Stata is started** for more details on how to do this.

4 Examples

To demonstrate the use of `swpermute`, we will use data from an SW-CRT conducted in Brazil that assessed the impact of switching from the standard tuberculosis (TB) diagnostic test, sputum smear microscopy, to a new diagnostic test called Xpert MTB/RIF.

We will focus on a secondary outcome of the proportion of patients with a bacterial confirmation of their TB diagnosis (Trajman et al. 2015). These examples use real trial data, but the data cannot be provided with the command. Instead, a simulated dataset is included that closely mimics the characteristics of these trial data but will not reproduce these example results.

The trial included 14 laboratories (clusters). At initiation of the study, all laboratories were using sputum smear microscopy to diagnose TB. Following a month of baseline data collection, the Xpert test was rolled out to two randomly assigned laboratories each month (that is, within seven months, all laboratories were using the Xpert test). The dataset contains 3,924 patients; their diagnoses were recorded as either clinical (with a negative test or no test done) or bacterially confirmed. The Xpert test was used to diagnose 2,147 (55%) patients. Across both trial arms, 2,833 (72%) had a confirmed TB diagnosis.

The output below describes the dataset:

```
. describe
Contains data from TBdiagnostic.dta
  obs:         3,924
 vars:             6                          5 Oct 2018 15:33
 size:        27,468
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
lab             byte    %8.0g                 ID of laboratory
patientid       int     %9.0g
study_month     byte    %8.0g                 Study month
arm             byte    %8.0g      armlbl     Intervention status
fav_outcome     byte    %10.0g     favoutlbl
                                              Treatment outcome of individual
confirmed       byte    %11.0g     confirmedlbl
                                              Laboratory confirmation of TB
                                                 diagnosis
-------------------------------------------------------------------------------
Sorted by: lab  study_month
. list in 1/5
```

	lab	patien~d	study_~h	arm	fav_outc~e	confirmed
1.	1	1	1	smear	favourable	unconfirmed
2.	1	2	1	smear	poor	confirmed
3.	1	3	1	smear	favourable	confirmed
4.	1	4	1	smear	poor	confirmed
5.	1	5	1	smear	favourable	confirmed

Each row gives the diagnosis type, `confirmed`, of a patient. `lab` identifies which laboratory they were diagnosed in and so assigns the patient to a cluster. `study_month` identifies which month of the study they were diagnosed in, and `arm` identifies whether the laboratory was using smear microscopy or the Xpert diagnostic at the time of diagnosis.

We will explore two analyses with permutation tests: the first will use a generalized linear mixed model with a permutation test, and the second will demonstrate a within-period analysis.

4.1 Example 1: Generalized linear mixed model

A mixed-effects logistic regression, adjusting for period effects as a fixed categorical variable and with a random intercept for cluster, can be used in combination with a permutation test to analyze this trial, as shown below.

```
. swpermute _b[arm], cluster(lab) period(study_month) intervention(arm)
> reps(1000) seed(20255) nodots:
> melogit confirmed i.study_month arm || lab:

Monte Carlo permutation results
     command:  melogit confirmed i.study_month arm || lab :
   statistic:  _b[arm]
      design:
```

freq	1	2	3	4	5	6	7	8
2	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
2	0	0	0	0	0	1	1	1
2	0	0	0	0	1	1	1	1
2	0	0	0	1	1	1	1	1
2	0	0	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1

statistic	obs_value	null	c	n	p	[95% Conf.	Interval]
_b[arm]	.4155579	0	2	1000	0.0020	.0002423	.0072058

```
Note: confidence interval is with respect to p
p-value is two-sided
```

`swpermute` shows the design-pattern matrix of the trial to allow users to check that sequences have been correctly identified. Each row represents a unique sequence of allocations observed within the data, and each column represents a period. For each sequence and period, a 0 or 1 is shown to represent the intervention condition of clusters in that sequence in that period. The leftmost column shows the number of clusters assigned to each sequence.

The table below this matrix gives the results of the permutation test. First, we see the intervention-effect estimate observed in the data (`obs_value`), and then the null hypothesis being tested (`null`). The third column (`c`) gives the number of permutations with a value of *exp* the same as or more extreme than the observed value, and the fourth column (`n`) gives the total number of permutations successfully completed.

The intervention-effect estimate is the value estimated by the `melogit` command [odds ratio = exp(0.42) = 1.52].

Only 2/1000 permutations gave a result the same as or more extreme than that observed, giving the p-value 2/1000 = 0.002 shown in column 5 (`p`). This analysis suggests there is strong evidence that the Xpert test increases the odds of a confirmed diagnosis.

The last two columns give a two-sided 95% confidence interval for the p-value that indicates the level of uncertainty around the p-value from the random selection of permutations. In this example, the interpretation of the p-value does not substantively change for values within this interval. Where interpretation would be altered for different values within the interval, the analysis should be rerun with more permutations.

4.2 Example 2: Within-period analysis and generating confidence intervals

In our second example, we will use a within-period analysis to calculate the difference in the risk (the proportion) of a confirmed diagnosis using a cluster-level analysis within each period, and we show how to construct confidence intervals.

First, we calculate the proportion of confirmed diagnoses in each cluster period by collapsing the data. We run `swpermute` with `regress` as the *command* to calculate a risk difference and its variance. We select the `withinperiod` option to run the regression within each period and set the period weights as variance weights.

```
. collapse (mean) risk_confirmed = confirmed , by(lab study_month arm)
. swpermute _b[arm], cluster(lab) period(study_month) intervention(arm)
> seed(9845) withinperiod weightperiod(variance _se[arm]^2) nodots
> reps(1000): regress risk_confirmed arm

Warning: study_month = 1 not included in analysis. Clusters all in one condition
Warning: study_month = 8 not included in analysis. Clusters all in one condition
Monte Carlo permutation results
     command:  regress risk_confirmed arm
   statistic:  _b[arm]
      design:
```

freq	1	2	3	4	5	6	7	8
2	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
2	0	0	0	0	0	1	1	1
2	0	0	0	0	1	1	1	1
2	0	0	0	1	1	1	1	1
2	0	0	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1

```
Within period Estimates and Weights:
                Estimate    Weight
study_month:2     0.0401    0.0581
study_month:3     0.0998    0.1787
study_month:4     0.0334    0.1329
study_month:5     0.1512    0.1835
study_month:6     0.1362    0.2714
study_month:7     0.0909    0.1753
```

statistic	obs_value	null	c	n	p	[95% Conf.	Interval]
_b[arm]	.1052611	0	50	1000	0.0500	.0373354	.0653905

```
Note: confidence interval is with respect to p
p-value is two-sided
```

We are warned that study month 1 and study month 8 are not included in this analysis. All clusters are in the same condition during these periods, so an intervention effect cannot be calculated.

The command displays a list of effect estimates and weights for each period in the study. The greatest weight is given to study month 6 despite the imbalance in clusters in the control and intervention conditions. This is because there was less variability in the cluster-level outcomes during this period, leading to a lower variance for the estimated intervention effect.

The observed value in the table of results is the weighted average of these period-specific estimates. The percentage of patients with a confirmed diagnosis was 10.5% higher in patients diagnosed with the Xpert test compared with patients diagnosed with smear microscopy, and there is some evidence against the intervention having no effect ($p = 0.05$).

Next, we demonstrate the construction of 95% confidence intervals. The initial estimate of the confidence interval boundaries can be found by assuming that the intervention-effect estimate follows a normal distribution and using the p-value to estimate a standard error as follows:

```
. * Estimate standard error
. display .1052611 / invnorm( 1 - 0.0500 / 2 )
.05370563
. * Initial lower bound of 95% CI
. display .1052611 - 1.96 * .0537
9.100e-06
. * Initial upper bound of 95% CI
. display .1052611 + 1.96 * .0537
.2105131
```

Permutation tests are conducted to test these initial values. An example is shown below for the initial proposed upper boundary; the dialog boxes shown in the appendix replicate this example.

```
. swpermute _b[arm], cluster(lab) period(study_month) intervention(arm)
> seed(9845) withinperiod weightperiod(variance _se[arm]^2) nodots
> reps(1000) null(0.211) outcome(risk_confirmed):
> regress risk_confirmed arm

Warning: study_month = 1 not included in analysis. Clusters all in one condition
Warning: study_month = 8 not included in analysis. Clusters all in one condition
Monte Carlo permutation results
     command:  regress risk_confirmed arm
   statistic:  _b[arm]
      design:
```

freq	1	2	3	4	5	6	7	8
2	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
2	0	0	0	0	0	1	1	1
2	0	0	0	0	1	1	1	1
2	0	0	0	1	1	1	1	1
2	0	0	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1

```
Within period Estimates and Weights:
                Estimate    Weight
study_month:2     0.0401    0.0581
study_month:3     0.0998    0.1787
study_month:4     0.0334    0.1329
study_month:5     0.1512    0.1835
study_month:6     0.1362    0.2714
study_month:7     0.0909    0.1753
```

statistic	obs_value	null	c	n	p	[95% Conf.	Interval]
_b[arm]	.1052611	.211	26	1000	0.0260	.0170528	.0378651

```
Note: confidence interval is with respect to p
p-value is two-sided
```

Depending on the p-value, the proposed boundary value is either increased or decreased until the boundary value with $p > 0.05$ is identified. To identify the lower boundary in our example, the initial estimate was 0.00, which gives $p = 0.0500$, so we tested a null value of -0.001 to see if a smaller value also fell with the 95% confidence interval. This has $p = 0.048$, so the lower boundary is 0.00. For the upper boundary, the initial estimate of 0.211 gave $p = 0.026$, well outside the 95% confidence interval. A null of 0.20 gave $p = 0.045$, null = 0.197 gave $p = 0.054$, null = 0.198 gave $p = 0.051$, and lastly null = 0.199 gave $p = 0.047$. The largest value within the 95% confidence interval is 19.8%. Therefore, our 95% confidence interval is [0.0%, 19.8%].

5 Concluding remarks

`swpermute`, an extension of the command `permute`, permutes clusters between sequences and can perform within-period analyses. We also incorporated functionality to test nonzero null hypotheses to facilitate the construction of confidence intervals. Although this command has been designed for use with SW-CRTs, it can also be used with other trial designs such as parallel and crossover CRTs.

However, `swpermute` has limitations. Testing nonzero null hypothesis values is only available for continuous outcomes (including cluster–period-level summaries). For other outcome types, the process involves manipulating the dataset to such a degree that we felt it was safer for the user to perform this themselves. For example, with a binary outcome, a risk difference cannot simply be subtracted from the outcome of 0 or 1. While we have incorporated stratification of randomization by a list of variables, some randomization strategies, such as restricted randomization, cannot be captured this way (Moulton 2004). In general, it is a limitation of permutation tests that confidence interval construction is computationally intensive.

`swpermute` facilitates the use of robust analysis methods for an SW-CRT, simplifying complex analysis.

6 Acknowledgments

This work was cofunded by the UK Medical Research Council (MRC) (MR/L004933/1-P27) and jointly funded by the MRC and the UK Department for International Development (DFID) under the MRC/DFID Concordat agreement and is also part of the EDCTP2 program supported by the European Union (MR/K012126/1) and (MR/R010161/1).

We thank Professor Anete Trajman, Dr. Betina Durovni, Dr. Valeria Saraceni, Professor Frank Cobelens, and Dr. Susan van den Hof for allowing us to use data from their study to demonstrate the command.

7 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 19-4
. net install st0577     (to install program files, if available)
. net get st0577         (to install ancillary files, if available)
```

8 References

Bellan, S. E., J. R. C. Pulliam, C. A. B. Pearson, D. Champredon, S. J. Fox, L. Skrip, A. P. Galvani, M. Gambhir, B. A. Lopman, T. C. Porco, L. A. Meyers, and J. Dushoff. 2015. Statistical power and validity of Ebola vaccine trials in Sierra Leone: A simulation study of trial design and analysis. *Lancet Infectious Disease* 15: 703–710.

Eden, T., and F. Yates. 1933. On the validity of Fisher's z test when applied to an actual example of non-normal data. *Journal of Agricultural Science* 23: 6–17.

Ernst, M. D. 2004. Permutation methods: A basis for exact inference. *Statistical Science* 19: 676–685.

Fisher, R. A. 1971. *The Design of Experiments*. 9th ed. New York: Macmillan.

Good, P. I. 2005. *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. 3rd ed. New York: Springer.

Hayes, R. J., and L. H. Moulton. 2009. *Cluster Randomised Trials*. Boca Raton, FL: Chapman & Hall/CRC.

Ji, X., G. Fink, P. J. Robyn, and D. S. Small. 2017. Randomization inference for stepped-wedge cluster-randomized trials: An application to community-based health insurance. *Annals of Applied Statistics* 11: 1–20.

Matthews, J. N. S., and A. B. Forbes. 2017. Stepped wedge designs: Insights from a design of experiments perspective. *Statistics in Medicine* 36: 3772–3790.

Moulton, L. H. 2004. Covariate-based constrained randomization of group-randomized trials. *Clinical Trials* 1: 297–305.

Pitman, E. J. G. 1937. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society* 4: 119–130.

Rigdon, J., and M. G. Hudgens. 2015. Randomization inference for treatment effects on a binary outcome. *Statistics in Medicine* 34: 924–935.

Thompson, J. A., C. Davey, K. Fielding, J. R. Hargreaves, and R. J. Hayes. 2018. Robust analysis of stepped wedge trials using cluster-level summaries within periods. *Statistics in Medicine* 37: 2487–2500.

Thompson, J. A., K. L. Fielding, C. Davey, A. M. Aiken, J. R. Hargreaves, and R. J. Hayes. 2017. Bias and inference from misspecified mixed-effect models in stepped wedge trial analysis. *Statistics in Medicine* 36: 3670–3682.

Trajman, A., B. Durovni, V. Saraceni, A. Menezes, M. Cordeiro-Santos, F. Cobelens, and S. Van den Hof. 2015. Impact on patients' treatment outcomes of XpertMTB/RIF implementation for the diagnosis of tuberculosis: Follow-up of a stepped-wedge randomized clinical trial. *PLOS ONE* 10: e0123252.

Wang, R., and V. De Gruttola. 2017. The use of permutation tests for the analysis of parallel and stepped-wedge cluster-randomized trials. *Statistics in Medicine* 36: 2831–2843.

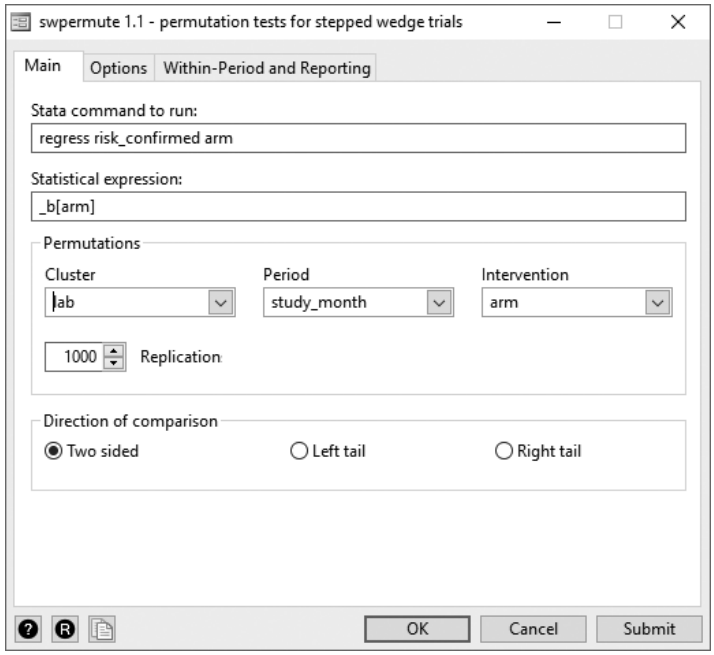
About the authors

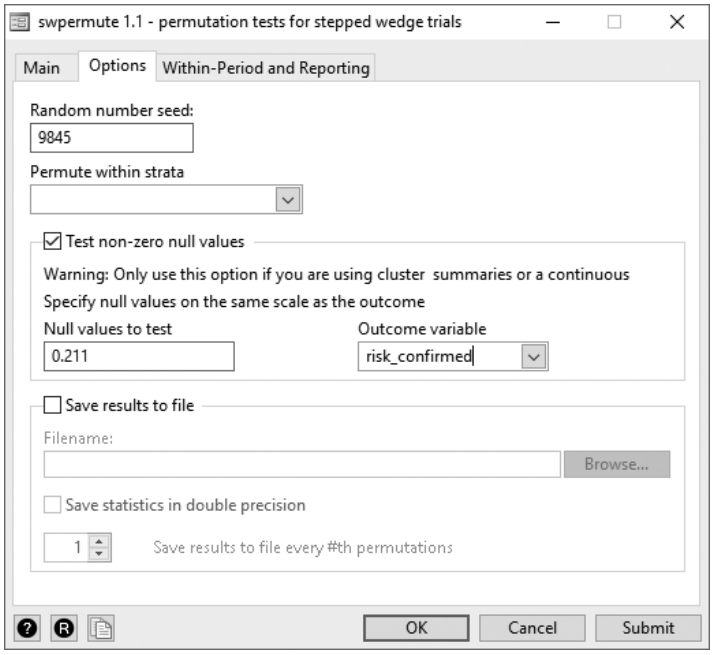
Jennifer Thompson is a statistician at the London School of Hygiene and Tropical Medicine in the UK. She has a PhD in the design and analysis of stepped-wedge trials.

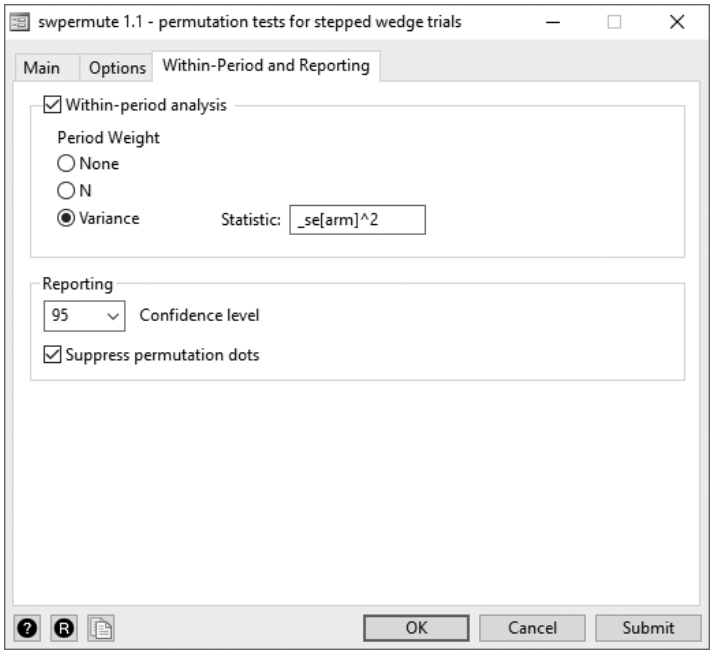
Calum Davey is an assistant professor at the London School of Hygiene and Tropical Medicine in the UK.

Richard Hayes, James Hargreaves, and Katherine Fielding are professors at the London School of Hygiene and Tropical Medicine in the UK.

A Appendix: Dialog boxes to run the example in section 4.2




swpermute 1.1 - permutation tests for stepped wedge trials
Main
Options
Within-Period and Reporting
Random number seed:
9845
Permute within strata
Test non-zero null values
Warning: Only use this option if you are using cluster summaries or a continuous
Specify null values on the same scale as the outcome
Null values to test
0.211
Outcome variable
risk_confirmed
Save results to file
Filename:
Browse...
Save statistics in double precision
1
Save results to file every #th permutations
OK
Cancel
Submit


swpermute 1.1 - permutation tests for stepped wedge trials
Main
Options
Within-Period and Reporting
Within-period analysis
Period Weight
None
N
Variance
Statistic:
_se[arm]^2
Reporting
95
Confidence level
Suppress permutation dots
OK
Cancel
Submit