The Stata Journal (2017)
17, Number 2, pp. 503–510

Covariate-constrained randomization routine for achieving baseline balance in cluster-randomized trials

Eva Lorenz
Institute of Public Health
University of Heidelberg
Heidelberg, Germany
and
Institute of Medical Biostatistics
Epidemiology and Informatics
University Medical Center
Mainz, Germany
eva.lorenz@uni-mainz.de

Sabine Gabrysch
Institute of Public Health
University of Heidelberg
Heidelberg, Germany
sabine.gabrysch@uni-heidelberg.de

Abstract. In cluster-randomized trials, groups or clusters of individuals, rather than individuals themselves, are randomly allocated to intervention or control. In this article, we describe a new command, `ccrand`, that implements a covariate-constrained randomization procedure for cluster-randomized trials. It can ensure balance of one or more baseline covariates between trial arms by restriction to allocations that meet specified balance criteria. We provide a brief overview of the theoretical background, describe `ccrand` and its options, and illustrate it using an example.

Keywords: st0484, ccrand, covariate-constrained randomization, cluster-randomized trials

1 Introduction

In cluster-randomized trials, groups or clusters of individuals, rather than individuals themselves, are randomly allocated to intervention or control (Donner and Klar 2000). Provided the sample size is large, randomization will ensure comparable arms in terms of the distribution of known and, more importantly, unknown factors that may influence the outcome (Dos Santos Silva 1999). The number of clusters in cluster-randomized trials is often limited. Therefore, one cannot rely on chance alone to ensure balance of important covariates (including the baseline value of the outcome) and sample size between arms (Moulton 2004). Achieving baseline balance of important covariates not only avoids the need to adjust for them in the final analysis but also is important for a trial's credibility. Furthermore, it increases statistical power and precision (Ivers et al. 2012).

In a recent methodological review, Ivers et al. (2012) discussed several techniques to balance baseline covariates in cluster-randomized trials along with their advantages and limitations, including stratification, matching, minimization, and covariate-constrained randomization. The last is often the method of choice when baseline data are available but has rarely been used because of the need for statistical support and specialized computer software to implement it (Ivers et al. 2012). Moulton (2004) proposed a method for implementing covariate-constrained randomization, and Chaudhary and Moulton (2006) wrote a SAS command and tutorial. Carter and Hood (2008) implemented a covariate-constrained randomization tool in R.

Our aim is to make covariate-constrained randomization more accessible to nonstatisticians by providing Stata code and demonstrating its use in a simple example. In this article, we give a brief overview of the covariate-constrained randomization procedure, including notation, key definitions, and concepts. Then, we describe the `ccrand` command and its implementation in Stata. Finally, we illustrate the command using an example dataset. We provide the code for the example in the supporting information (available from the journal's web page).

Stratified randomization can be useful when there is substantial variability in the outcome of interest between clusters. This means that strata are formed and clusters are randomized separately in each relatively homogeneous stratum. While stratification can also help balance on a limited number of covariates, it is recommended mainly for improving precision of estimation by reducing intracluster correlation (Hayes and Moulton 2009). Our procedure allows users to combine restricted randomization with stratification to both ensure balance and improve efficiency.

2 The covariate-constrained randomization procedure

2.1 The basic idea

The purpose of covariate-constrained randomization is to achieve balance between arms on one or more important baseline covariates that are thought to be predictive of the outcome of the trial, often including the baseline values of the primary endpoint itself (Moulton 2004). This requires that covariate data be available for all clusters prior to allocation. The main method for covariate-constrained randomization implies that all possible allocations of dividing the clusters into two arms are simulated, and differences in covariates between arms are calculated for each and checked against prespecified balance criteria. For example, for the continuous covariate education, one could specify the difference in group means between arms to be no greater than one year's difference in mean school attendance. The final set of allocations is then limited to those that meet the prespecified balance criteria, and the actual allocation is chosen randomly from this acceptable set.

2.2 Validity

If the constraints are too strict, it may be that two clusters are always or never in the same arm in all the acceptable allocations because the balance criteria can be fulfilled only this way. This challenges the validity of the design because then it would not hold true that “every pair of clusters has the same probability of being allocated to the same treatment”, which “violates the assumption of independence between clusters in each arm” and could potentially change the type I error (Hayes and Moulton 2009). These situations can be identified by counting the number of times any given pair of clusters has the same treatment allocation. Examining under- or overrepresented pairs can then reveal the balance constraints responsible, which need to be relaxed (Moulton 2004). However, relaxing the balancing constraints implies a reduction of covariate balance between study arms.

3 The ccrand command

Our approach is similar to the steps described by Chaudhary and Moulton (2006) for the SAS command.

1. Separately for each stratum, we first generate all possible allocations dividing the clusters into two arms.
2. For each allocation (in each stratum), we calculate the means of relevant covariates in both arms and retain only the allocations meeting the balance criteria. If too few allocations are retained in some strata, the criteria need to be relaxed. If too many allocations are retained (which may make the subsequent computation too heavy), the criteria should be tightened.
3. By combining the acceptable stratum-level allocations, we generate all possible overall allocations.
4. For each overall allocation, we calculate the means of relevant covariates in both arms and again retain only the allocations that meet the overall balance criteria.
5. Finally, we perform a validity check by calculating how often each pair of clusters is allocated to the same arm. If n is the number of acceptable allocations, it should occur in approximately $n/2$ of these. The exact value is $m! \times \binom{n}{k}$, where m is the number of clusters in a pair of clusters (2), n is the number of clusters per stratum, and k is the number of strata.
6. We select the final allocation at random from all acceptable overall allocations.

3.1 Syntax

The syntax of `ccrand` is as follows:

`ccrand` *mainvarlist*`, ibc("`*string*`") obc("`*string*`")` [`seed(`*#*`) cluster(`*varname*`)`
`stratum(`*varname*`) validitycheck(`*string*`) selectfinal(`*string*`)`]

where *mainvarlist* contains all the covariates to be balanced. Arguments in squared brackets are optional and have default values assigned as described below.

3.2 Options

`ibc("`*string*`")` (initial balancing criteria) specifies a string of numeric values that is equal to the number of covariates separated by a blank (`" "`) and that represents the maximum allowable differences between arms for each covariate for allocations within strata. All covariates must satisfy the criteria individually for an allowable allocation. Values must be positive. `ibc()` is required.

`obc("`*string*`")` (overall balancing criteria) specifies a string of numeric values that is equal to the number of covariates separated by a blank (`" "`) and that represents the maximum allowable differences between arms for each covariate for overall allocations. All covariates must satisfy the balancing criteria individually for an allowable allocation. Values must be positive. `obc()` is required.

`seed(`*#*`)` sets the reproducible random-number seed to # for selecting one final overall allocation from all acceptable overall allocations. The seed needs to be set to reproduce the results.

`cluster(`*varname*`)` specifies the name of the covariate containing cluster IDs. The default is `cluster(cluster)`.

`stratum(`*varname*`)` specifies the name of the covariate containing stratum IDs. The default is `stratum(stratum)`.

`validitycheck(`*string*`)` specifies whether the validity check should be performed and details displayed after randomization (`no`; default is `validitycheck(yes)`).

`selectfinal(`*string*`)` specifies whether one final overall allocation should be selected at random from all acceptable overall allocations (`no`; default is `selectfinal(yes)`).

3.3 The data structure

`ccrand` requires an input Stata dataset that contains the stratum and cluster IDs. It also requires the cluster-level covariates to be balanced on.

4 Illustration using a data example

4.1 Example: A cluster-randomized trial of a complex intervention to reduce child undernutrition

The aim of the Food and Agricultural Approaches to Reducing Malnutrition (FAARM) cluster-randomized controlled trial (http://www.clinicaltrials.gov/ct2/show/NCT02505711/) is to evaluate the impact of an integrated home gardening, nutrition, and hygiene intervention on chronic undernutrition in young children in a low-income setting. The intervention is delivered to women's groups. The trial includes 2,700 young women and their children from 96 settlements within 2 subdistricts of Habiganj District in Bangladesh. Following a baseline survey in 2015, settlements were allocated to either intervention or control arms in a 1:1 ratio using covariate-constrained randomization. Women in the intervention arm will receive training and support over three years. The primary endpoint is linear growth (length for age) in children under three years old, which will be collected in 2019.

The dataset

The example dataset is a subset of the FAARM baseline survey containing 24 clusters in 3 strata and 5 covariates to balance on. A partial output of the data is shown in table 1 for illustration. Each of the three strata contains eight clusters, of which four clusters are to be allocated to the intervention arm. The five cluster-level baseline covariates (some converted into z scores) are the number of included women in the cluster (`women`), women's height (`z_wht`), child length for age at baseline (`z_lfa`), child age in months at baseline (`z_age`), and child diarrhea prevalence at baseline (`diarrhea`).

Table 1. Listing of the first five lines of the input Stata dataset sorted by cluster

obs	cluster	s	`women`	`z_wht`	`z_lfa`	`z_age`	`diarrhea`
1	1	3	37	−0.24	0.70	−0.22	0.10
2	2	1	14	−0.32	−0.05	−0.06	0.10
3	3	1	17	0.21	1.86	−0.26	0.11
4	4	3	51	−0.07	−0.61	0.52	0.07
5	5	2	29	−0.54	−1.17	0.01	0.16

As in the example, covariates of interest may be standardized before applying the `ccrand` command by calculating the respective z scores. The z score measures how many standard deviations above the mean (positive values) or below the mean (negative values) a data point is. z scores can be calculated using the formula $Z = (X - \mu)/\sigma$, where X is the covariate of interest, μ is the mean, and σ is the standard deviation. Constraints can then be set in terms of standard deviations instead of the original covariate units.

The command

```
ccrand women diarrhea z_wht z_lfa z_age, ibc("4 0.2 0.5 0.5 0.5")  ///
  obc("3 0.1 0.5 0.5 0.5") seed(89574) cluster(cluster) stratum(stratum)
```

The output

After processing the input dataset, the program generates all possible arm 1 and arm 2 allocations separately along with the relevant covariate data. In our example, we want to choose 4 clusters from 8 in each stratum; thus the number of combinations per stratum is 8 choose 4 $= \binom{8}{4} = 70$. The total number of arm 1 allocations calculated by the program for all 3 strata is the sum of the combinations in the strata $\binom{8}{4} + \binom{8}{4} + \binom{8}{4} =$

70+70+70 = 210, which yields 210×4 = 840 data rows. The number of combinations is multiplied by 4 because there are 4 clusters stored in a separate row in each combination. For each stratum, we compute covariate means for each allocation in each arm separately and merge the results to compute the differences in means between the two arms. These differences are compared with the values of the within-stratum initial balancing criteria (`ibc()`). We then select the allocations that satisfy these initial criteria. In this example, 62 allocations out of a total of 210 qualify: 18 in stratum 1, 22 in stratum 2, and 22 in stratum 3. Based on these acceptable allocations in each stratum, all possible overall allocations are generated by selecting one allocation from each stratum. In our example, this results in a total of $(18 \times 22 \times 22) \times 3 = 8712 \times 3 = 26136$ possible overall allocations. Again, we calculate covariate means in both arms and check the difference in means against the overall balancing criteria (`obc()`). In the example, 8,328 allocations fulfilled these criteria. A listing of these allocations with the respective stratum IDs and within-stratum allocation IDs, called `rno`, is saved in a dataset. Finally, we compute the number of times a cluster appears with another cluster in the same arm as a check for the validity of allocations. In the end, one allocation—to be used to implement the randomization for the trial—is selected at random from those allocations qualifying overall balancing criteria and displayed by the program. With the seed chosen in the example, this results in allocation 498 with the "keep it simple, stupid" (KISS) random-number generator (default until Stata 13) and allocation 283 with the Mersenne Twister generator (new default introduced in Stata 14).

Recommendations

A prerequisite of the covariate-constrained randomization method is that recruitment of clusters and collection of covariates for balancing must be completed prior to the cluster allocation.

Implementing the randomization procedure supports a large number of strata with a moderate number of clusters within strata. The initial balancing criteria should not result in more than 25 allowable allocations per stratum, because all possible combinations of clusters per strata are generated in the next step, which increases exponentially and is computationally very intense. There are no restrictions on the maximum number of covariates to balance on.

The program can also handle uneven numbers of clusters per stratum and will assign the higher number of clusters to the first study arm, that is, for a stratum size of 7, 4 clusters will be assigned to the first study arm and 3 clusters will be assigned to the second study arm.

5 Conclusions

Covariate-constrained randomization is a valuable tool for cluster-randomized trials. However, the method was used in only 2% of 300 trials published from 2000 to 2008 (Ivers et al. 2011). This is partly because of its apparent complexity and the lack of software routines until the recent programs in SAS and R were available (Chaudhary and Moulton 2006; Carter and Hood 2008). Because our program is inspired by the SAS macro written by Chaudhary and Moulton, it covers the same functionality. Additionally, the number of covariates is not limited in the Stata implementation and the performance (run time) is better. The implementation in R does not provide the validity check but can be extended with knowledge of programming in R. We hope that this routine will make the procedure more accessible to a wider audience of applied researchers.

6 Acknowledgments

The program is inspired by the CCR_V1.0 macro in SAS, which was written by Chaudhary and Moulton (2006). We thank Larry Moulton for helpful discussions via email.

We are very grateful to Anja Schoeps, Robin C. Nesbitt, and especially Andreas Deckert and an anonymous reviewer for their valuable suggestions on how to improve this command.

7 Funding sources

Sabine Gabrysch is funded by a grant (award number 01ER1201) of the German Federal Ministry of Education and Research. The content of this publication is solely the responsibility of the authors.

8 Contributions

Eva Lorenz implemented `ccrand` and wrote the first draft of this article. Sabine Gabrysch had the idea for `ccrand`, supervised its implementation, and contributed substantially to the writing of this article. Both authors read and approved the final version of the manuscript.

9 References

Carter, B. R., and K. Hood. 2008. Balance algorithm for cluster randomized trials. *BMC Medical Research Methodology* 8: 65.

Chaudhary, M. A., and L. H. Moulton. 2006. A SAS macro for constrained randomization of group-randomized designs. *Computer Methods and Programs in Biomedicine* 83: 205–210.

Donner, A., and N. Klar. 2000. *Design and Analysis of Cluster Randomization Trials in Health Research.* London: Arnold.

Dos Santos Silva, I., ed. 1999. *Cancer Epidemiology: Principles and Methods.* Geneva: World Health Organization.

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

Ivers, N. M., I. J. Halperin, J. Barnsley, J. M. Grimshaw, B. R. Shah, K. Tu, R. Upshur, and M. Zwarenstein. 2012. Allocation techniques for balance at baseline in cluster randomized trials: A methodological review. *Trials* 13: 120.

Ivers, N. M., M. Taljaard, S. Dixon, C. Bennett, A. McRae, J. Taleban, Z. Skea, J. C. Brehaut, R. F. Boruch, M. P. Eccles, J. M. Grimshaw, C. Weijer, M. Zwarenstein, and A. Donner. 2011. Impact of CONSORT extension for cluster randomised trials on quality of reporting and study methodology: Review of random sample of 300 trials, 2000-8. *British Medical Journal* 343(d5886): 1–14.

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

About the authors

Eva Lorenz is a research associate at the Institute of Medical Biostatistics, Epidemiology and Informatics in Mainz, Germany. Her research interests include development and application of epidemiological methodology and statistical programming.

Sabine Gabrysch is head of the Unit of Epidemiology and Biostatistics and deputy head of the Institute of Public Health at Heidelberg University, Germany. Her main research interest is maternal and child health in low-income settings. She is leading the FAARM trial on malnutrition in Bangladesh.