Blocking in Randomized Evaluations

Motivation

Most OES projects involve randomized evaluations, where our researchers work with agency partners to assign participants to different conditions using lottery-like mechanisms. These mechanisms give us the best chance of knowing that any differences we see between groups are due to our intervention, not confounding factors.

Whenever possible, we incorporate background information about participants directly into our designs. This helps minimize our estimation error and decrease the uncertainty around our estimates of treatment effects. By blocking on prognostic pre-treatment covariates, we ensure that our experiments are as precise as possible, and that their results are indicative of actual causal effects – not just differences at baseline.

What to Block On

Blocking involves creating homogeneous subsets of the experimental units, then randomly assigning treatments within those subsets, called blocks. The blocks should be created from quantities that are pre-treatment and prognostic as to the outcomes of interest. Often, the single best quantity on which to block is the outcome measure at baseline, or in a prior period. By blocking on quantities that are strongly prognostic of the outcome, we are, by proxy, helping ensure that our treatment and control conditions are good substitutes for what would have happened had each unit received the other condition. Additionally, if we plan to estimate treatment effects within subgroups, blocking on the subgroup variable can promote high precision and low estimation error for the subgroup analysis.

Depending on the nature of the prognostic variables, we might block on (a) a small number of discrete covariates, or (b) on any number of discrete or continuous covariates (up to limits imposed by the sample size).

Two Types of Blocking

Blocking on a Small Number of Discrete Variables

To illustrate the case of a small number of discrete covariates, we suppose we have 100 doctors, 50 in hospital A and 50 in hospital B; the two hospitals have very different patient populations.[1] We will assign half the doctors to receive reminder letters, and the other half to a business-as-usual, no-reminder condition. If we randomly assign half the doctors to reminders, we

[1] We show similar examples at https://gsa-oes.github.io/sop/, §4.4.

might get an assignment that is unbalanced in its counts. For example, below, 28 of the B doctors are assigned to letters, but only 22 of the A doctors are.

Hospital	Letter	No Letter
A	22	28
B	28	22

By blocking on hospital, then randomly assigning half the doctors within each hospital to letters, we ensure that our treatment conditions have the same distributions of A and B doctors. Below, the letter and no letter conditions each have 25 doctors from each hospital:

Hospital	Letter	No Letter
A	25	25
B	25	25

By ensuring that the two conditions have the same distributions of the important prognostic factor of hospital, we minimize the estimation error and maximize the precision of our treatment effect estimates.

For examples from OES work, consider project 1903 on wildfire risk in Montana, in which we exact-blocked on county and risk category. Similarly, in project 1808 on residential energy use, we exact-blocked on apartment size and the presence or absence of baseline usage data.

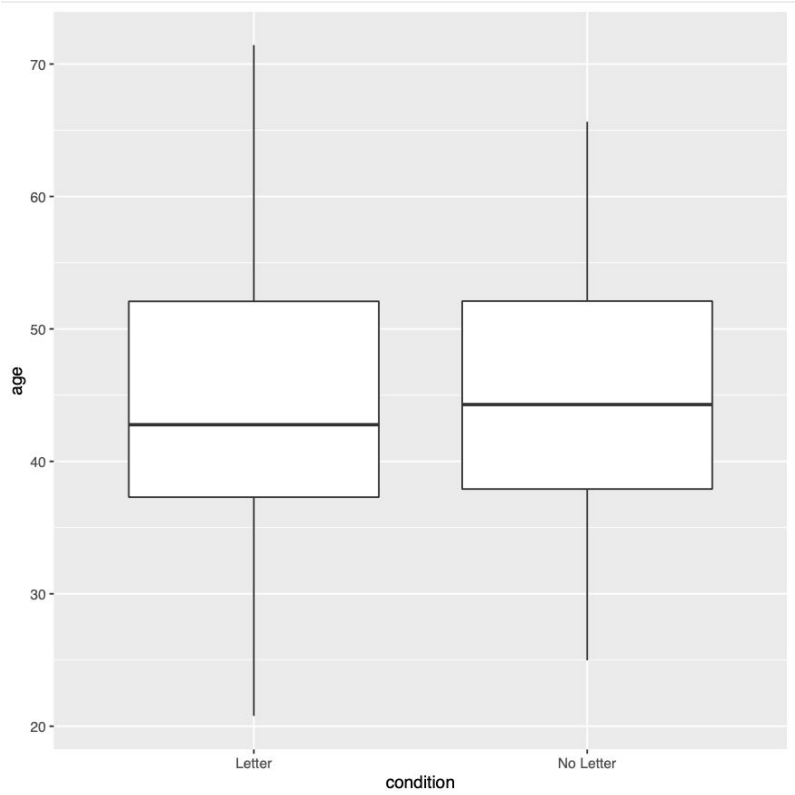
Blocking on Several (Possibly Continuous) Variables Using a Distance Metric

Often we have many prognostic factors, both discrete and continuous, that we want to balance between our treatment and control groups. In these cases, we will create blocks of units that look like each other across the set of covariates. This procedure involves summarizing many covariate differences into a single "distance" metric, creating blocks using that metric, then randomizing within the blocks. In this case, blocks may not be perfectly homogeneous.

Suppose that our hospitals vary in the average age of patients, which then impacts whether reminder letters work. We can block doctors on both average patient age and an indicator for hospital, creating pairs of units that are similar on both, then randomizing within pairs.[2] This requires quantifying tradeoffs between patient age and hospital into a single metric. We prefer blocks of units that are like each other in both attributes, but in some cases, we may be willing to tolerate slight heterogeneity in one variable for more homogeneity on the other. Below we incorporate both predictors in a blocking algorithm before assigning doctors to treatments. After

[2] We typically block on the Mahalanobis distance between units, or use a similar dimension-reduction technique.

doing so, we can see that both factors, age and hospital, are well-balanced across the two conditions. The figure shows the distributions of average patient ages for the two conditions, and the table shows that the hospital counts are balanced.



Hospital	Letter	No Letter
A	25	25
B	25	25

What Not to Block On

If you have quantities that are unrelated to the outcomes of interest, blocking on them will not improve the design. However, even if you block on random noise, the random assignment within blocks preserves features like the unbiasedness of the difference in means estimator.

If you have a prognostic covariate, and you create blocks that tend to be *dissimilar* on that covariate, the blocked design can have more variance around the treatment effect than an unblocked design. Avoid making blocks that are strongly heterogeneous.

Attrition

In some cases, we are concerned about attrition in small blocks, particularly when the treatment probability varies across blocks. Estimators like the blocked difference-in-means estimator require a treatment effect within each block; if there are no treated or no control units in a block, then the block-level treatment effect cannot be estimated. For such estimators, blocks should be large enough so that the treatment effect and its variance can be estimated.

However, if the treatment probability is constant across blocks, the treatment effect can still be estimated, for example, with the Lin estimator. The attrition itself may introduce bias, but the blocked design still retains a precision advantage over the unblocked one.

Analysis & Reporting

When the probability of treatment is the same within each block, three common estimators perform similarly: the simple difference in means, regression with block indicators, and adjusting for block membership using the Lin (2013) estimator. The Lin estimator tends to outperform the others in power.

However, if the probability of treatment varies across blocks, we account for this variation. For example, suppose we had 70 doctors from A hospitals and 30 from B hospitals, but we will select 20 doctors from each for treatment. Or, suppose we have 50 doctors from each hospital type, but only 10 A doctors will be selected, while 20 B doctors will be selected.

In such cases, we weight block-level estimates of the treatment effect by the sizes of the blocks. Then, the difference in means estimator (or a regression equivalent) will be unbiased for the average treatment effect and also for the standard error around that effect. An "interaction weighted" estimator performs similarly. On the other hand, incorporating block-level indicators into a regression model will not correctly weight the block-level estimates, leading to bias and imprecision.

When we report results from a blocked experiment, we prefer the interaction estimator of Lin (2013). In that estimator, which includes recentered indicators for blocks interacted with the covariates, the coefficient on the treatment indicator estimates the treatment effect averaged over the blocks. Reporting this average estimate circumvents the need to specify a particular block when producing a predicted outcome for the treatment condition (which is our preferred approach in graphs).

Example Code

To create blocks on a small number of discrete covariates, we tend to use the randomizr package in R. From the example above, if the data are in df, and we want to randomize within hospitals and attached the assignments to the data:

```
df$blocked_assg <- block_ra(blocks = df$hospital)
```

Then, we can estimate the average treatment effect with the Lin estimator via

```
estimatr::lm_lin(Y ~ blocked_assg, covariates = ~ hospital, data = df)
```

To create blocks on several numeric covariates, the blockTools package takes the blocking variables and an ID variable. It first creates the blocks, then assigns conditions within them:

```
blocks_out <- block(df, id.vars = "id", block.vars = c("hosp_A", "age"))
assg_out <- assignment(blocks_out)
```

Then, we can estimate the average treatment with our preferred standard errors via

```
df$blocked_assg <- extract_condition(assg_out, df, id.var = "id")   # as of blockTools 0.6-4

estimatr::lm_robust(Y ~ blocked_assg, data = df)
```

We prefer a non-interactive model here, since (1) we will have equal assignment probabilities across blocks and (2) linear dependence will preclude the estimation of coefficients for every condition-times-block interaction.

References

DeclareDesign Blog. 2018a. "Sometimes Blocking Can Reduce Your Precision." https://declaredesign.org/blog/2018-09-24-bad_blocks_bad.html.

DeclareDesign Blog. 2018b. "Improve Power Using Your Answer Strategy, Not Just Your Data Strategy." https://declaredesign.org/blog/2018-10-02-power-strategies.html.

Gerber, Alan S., and Donald P. Green. 2012. Field Experiments: Design, Analysis, and Interpretation. New York, NY: WW Norton.

Imai, Kosuke, Gary King, and Elizabeth A. Stuart. 2008. "Misunderstandings Between Experimentalists and Observationalists About Causal Inference." Journal of the Royal Statistical Society, Series A 171 (2): 481–502.

Lin, Winston. 2013. "Agnostic Notes on Regression Adjustments to Experimental Data: Reexamining Freedman's Critique." The Annals of Applied Statistics 7 (1): 295–318.

Moore, Ryan T. 2012. "Multivariate Continuous Blocking to Improve Political Science Experiments." Political Analysis 20 (4): 460–79. https://doi.org/10.1093/pan/mps025.