Blocking

Ryan T. Moore

American University

The Lab @ DC

2024-06-11

Table of contents I

Why?

Why do we block?

- Covariate **balance**

Why do we block?

▶ Covariate **balance**
▶ Estimate **closer to truth**

Why do we block?

- Covariate **balance**
- Estimate **closer to truth**
- Increased **efficiency**

Why do we block?

- Covariate **balance**
- Estimate **closer to truth**
- Increased **efficiency**
- Triply-robust estimates: block, randomize, adjust

Why do we block?

- Covariate **balance**
- Estimate **closer to truth**
- Increased **efficiency**
- Triply-robust estimates: block, randomize, adjust
- Block-level effects
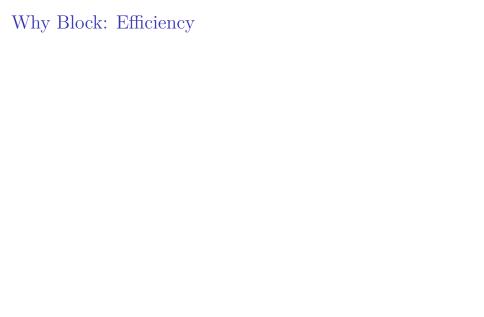 ⇝ different actors interested in different effects

Why do we block?

- Covariate **balance**
- Estimate **closer to truth**
- Increased **efficiency**
- Triply-robust estimates: block, randomize, adjust
- Block-level effects
  - ↝ different actors interested in different effects
- Guidelines for limited/uncertain resources

Why Block: Balance

Simulation study: 100 units, $X_1 \sim N(0,1)$, $X_2 \sim \text{Unif}(0,1)$, $X_3 \sim \chi_2^2$; 1000 such experiments. Assg treatmnt in 3 ways.

Why Block: Balance

Simulation study: 100 units, $X_1 \sim N(0,1)$, $X_2 \sim \text{Unif}(0,1)$, $X_3 \sim \chi_2^2$; 1000 such experiments. Assg treatmnt in 3 ways.

Why Block: Efficiency

Blocking in Applications: Balance and Efficiency

Moore (2012): Perry Preschool Experiment

Left: QQ plot of balance (100 blocked vs. unblocked)
Right: Est TE under sharp null (100 blocked vs. unblocked)

(SES, sex, IQ)

Considering more variables …

(+ siblings, AFDC, mom empl, educ, father, …)

How?

Blocking with `blockTools`

Start with some sample data:

```
library(blockTools)
data(x100)

x100 |> head()
```

```
    id id2  b1  b2 g  ig
1 1001 101 156 795 b 729
2 1002 102 813 469 a 627
3 1003 103 950 978 a 959
4 1004 104 991 781 a 661
5 1005 105 613 759 a 819
6 1006 106 654 838 b 643
```

(Moore 2012; Moore and Schnakenberg 2023)

Blocking with `blockTools`

```r
b <- block(x100, id.vars = "id",
           block.vars = c("b1", "b2"))
```

```r
bl <- b$blocks$`1`
```

```r
bl |> head()
```

```
  Unit 1 Unit 2   Distance
1   1043   1040 0.01240000
2   1100   1020 0.02259275
3   1065   1027 0.02912651
4   1085   1081 0.03498815
5   1088   1061 0.04789253
6   1064   1014 0.07985116
```

Blocking with `blockTools`

Why all this?

```
bl <- b$blocks$`1`
```

We are extracting just the blocked pairs themselves.

▶ Why b$blocks? Since b has 3 components:

```
names(b)
```

```
[1] "blocks"    "level.two" "call"
```

▶ Why blocks$1'? Since this is (default-named) first (and only) "group":

```
names(b$blocks)
```

```
[1] "1"
```

Blocking with `blockTools`

What else could we do?

```r
b_3groups_3conditions <- block(
  x100,
  groups = "g",                  # (Factor variable in data)
  n.tr = 3,
  id.vars = "id",
  block.vars = c("b1", "b2"),
  distance = "mve"
)
```

```r
b_3groups_3conditions$blocks
```

```
$a
   Unit 1 Unit 2 Unit 3 Max Distance
1    1076   1039   1056    0.2611685
2    1084   1058   1017    0.4028237
3    1073   1029   1098    0.4256919
4    1065   1061   1002    0.4380198
```

Blocking with `blockTools`

Some rows from each "group":

```
rows_a <- b_3groups_3conditions$blocks$a |> slice(1:2) |> r
rows_b <- b_3groups_3conditions$blocks$b |> slice(1:2) |> r
rows_c <- b_3groups_3conditions$blocks$c |> slice(1:2) |> r
bind_rows(rows_a, rows_b, rows_c)
```

```
  Unit 1 Unit 2 Unit 3 Max Distance group
1   1076   1039   1056    0.2611685     a
2   1084   1058   1017    0.4028237     a
3   1043   1040   1009    0.1716535     b
4   1048   1031   1025    0.2025934     b
5   1095   1092   1049    0.3468516     c
6   1088   1027   1066    0.3577024     c
```

Blocking with `blockTools`

Other arguments to `block()`

- ▶ `vcov.data`
- ▶ `groups`: for exact-blocks
- ▶ `n.tr`
- ▶ `id.vars`
- ▶ `block.vars`
- ▶ `algorithm`: `optGreedy`, `optimal`, `naiveGreedy`, `randGreedy`, `sortGreedy`
- ▶ `distance`: `mahalanobis`, `mcd`, `mve`, `euclidean`, $k \times k$ matx
- ▶ `weight`
- ▶ `level.two`: block states by most similar cities
- ▶ `valid.var`, `valid.range`: Goldilocks
- ▶ `seed.dist`: (for `mcd` and `mve`)

Assign

```
a <- assignment(b, seed = 71573706)
a
```

```
Assignments:

   Treatment 1  Treatment 2  Distance
1  1040         1043         0.01240000
2  1100         1020         0.02259275
3  1065         1027         0.02912651
4  1081         1085         0.03498815
5  1088         1061         0.04789253
6  1014         1064         0.07985116
7  1032         1070         0.08279625
8  1097         1098         0.08882421
9  1038         1018         0.09316331
10 1031         1048         0.10391953
11 1084         1058         0.10835825
```

Assign

```
a3 <- assignment(b_3groups_3conditions, seed = 979677744)
a3
```

```
Assignments:

Group: a
    Treatment 1   Treatment 2   Treatment 3   Max Distance
1   1056          1076          1039          0.2611685
2   1017          1058          1084          0.4028237
3   1029          1073          1098          0.4256919
4   1061          1002          1065          0.4380198
5   1059          1081          1046          0.4609745
6   1060          1067          1004          0.6265911
7   1089          1024          1032          0.6945585
8   1052          1054          1030          0.8135706
9   1093          1026          1068          0.8689412
10  1036          1008          1091          1.3792664
```

Blocking with `randomizr::block_ra()`

Material

Coppock (2023)

Then what?

blockTools: diagnose, get block IDs, check balance

Diagnose:

```
diagnose(a, data = x100, id.vars = "id",
         suspect.var = "b1", suspect.range = c(0, 5))
```

blockTools: diagnose, get block IDs, check balance

Diagnose:

```
diagnose(a, data = x100, id.vars = "id",
         suspect.var = "b1", suspect.range = c(0, 5))
```

Get block IDs

```
createBlockIDs(a, data = x100, id.var = "id")
```

blockTools: diagnose, get block IDs, check balance

Diagnose:

```
diagnose(a, data = x100, id.vars = "id",
         suspect.var = "b1", suspect.range = c(0, 5))
```

Get block IDs

```
createBlockIDs(a, data = x100, id.var = "id")
```

Get balance:

```
assg2xBalance(a, x100, id.var = "id",
              bal.vars = c("b1", "b2"))
```

Thanks!

rtm@american.edu

www.ryantmoore.org

References I

Coppock, Alexander. 2023. *randomizr: Easy-to-Use Tools for Common Forms of Random Assignment and Sampling.* https://CRAN.R-project.org/package=randomizr.

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.

Moore, Ryan T., and Keith Schnakenberg. 2023. *blockTools: Blocking, Assignment, and Diagnosing Interference in Randomized Experiments.* http://www.ryantmoore.org/html/software.blockTools.html.