Clustered bootstrapping for selective reporting models in meta-analysis with dependent effects

ESMAR 2023

James E. Pustejovsky & Megha Joshi

2023-02-27

Selective reporting of study results

- Selective reporting occurs if *affirmative* findings are *more likely to be reported* and available for inclusion in meta-analysis.
  - *Affirmative* meaning statistically significant and in the theoretically expected direction.
  - Bias in the publication process (journal/editor/reviewer incentives)
  - Strategic decisions by authors
- Selective reporting distorts the evidence base available for systematic review/meta-analysis.
  - Inflates average effect size estimates from meta-analyses.
  - Biases estimates of heterogeneity (Augusteijn et al., 2019).

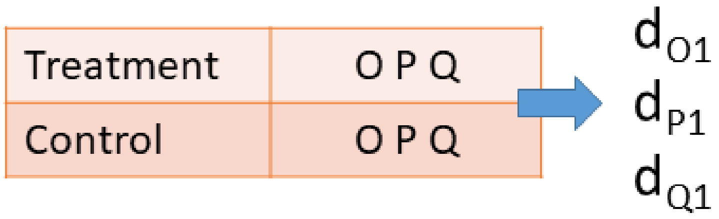
Tools for investigating selective reporting

- Graphical diagnostics
    - Funnel plots
    - Contour-enhanced funnel plots
    - Power-enhanced funnel plots (sunset plots)

- Tests/adjustments for funnel plot asymmetry
    - Trim-and-fill
    - Egger's regression
    - PET/PEESE
    - Kinked meta-regression
- Selection models
    - Weight-function models
    - Copas models
    - Sensitivity analysis
- p-value diagnostics
    - p-curve
    - p-uniform / $\boldsymbol{p}$**-uniform***

Dependent effect size estimates

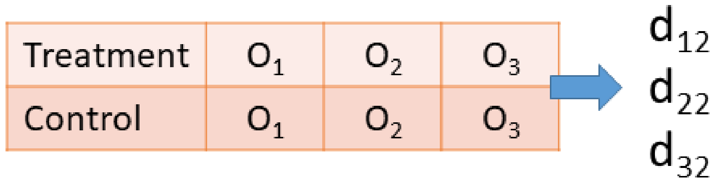
Multiple outcomes measured on a common set of participants



Treatment	O P Q
Control	O P Q

d_{O1}
d_{P1}
d_{Q1}

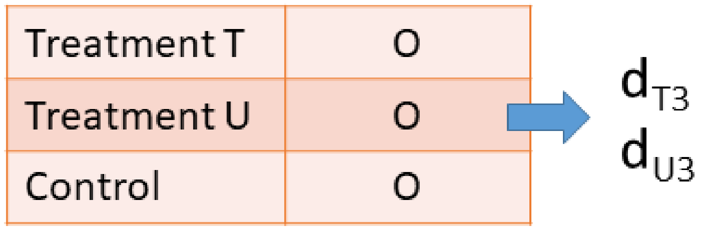
Outcomes measured at multiple follow-up times



Treatment	O_1	O_2	O_3
Control	O_1	O_2	O_3

d_{12}
d_{22}
d_{32}

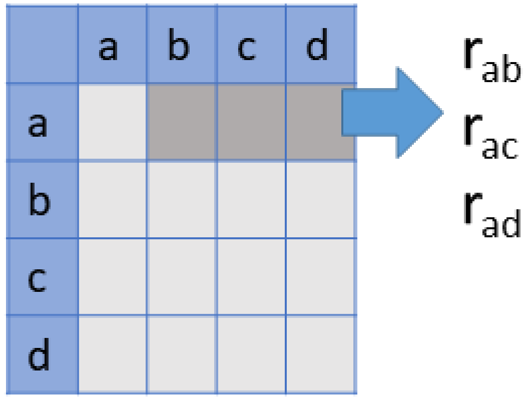
Multiple treatment conditions compared to a common control



Treatment T	O
Treatment U	O
Control	O

d_{T3}
d_{U3}

Multiple correlations from a common sample



	a	b	c	d
a		▨	▨	▨
b				
c				
d				

r_{ab}
r_{ac}
r_{ad}

Motivation

- Dependent effect sizes are very common in social science meta-analyses.
    - Good methods available for handling dependence in meta-analysis / meta-regression.
- However, few methods for investigating selective reporting bias can handle dependent effect sizes (Rodgers & Pustejovsky, 2021).
- Using existing bias-correction methods without accounting for dependency risks misleading conclusions
    - too-narrow confidence intervals
    - hypothesis tests inflated Type 1 error rates

A pragmatic strategy:

Cluster-bootstrapping a selection model

- Fit a regular selection model (ignoring dependency issues).
  - Using the `metafor` package
- Re-sample clusters of dependent effect sizes to assess uncertainty.
  - Using the `boot` package

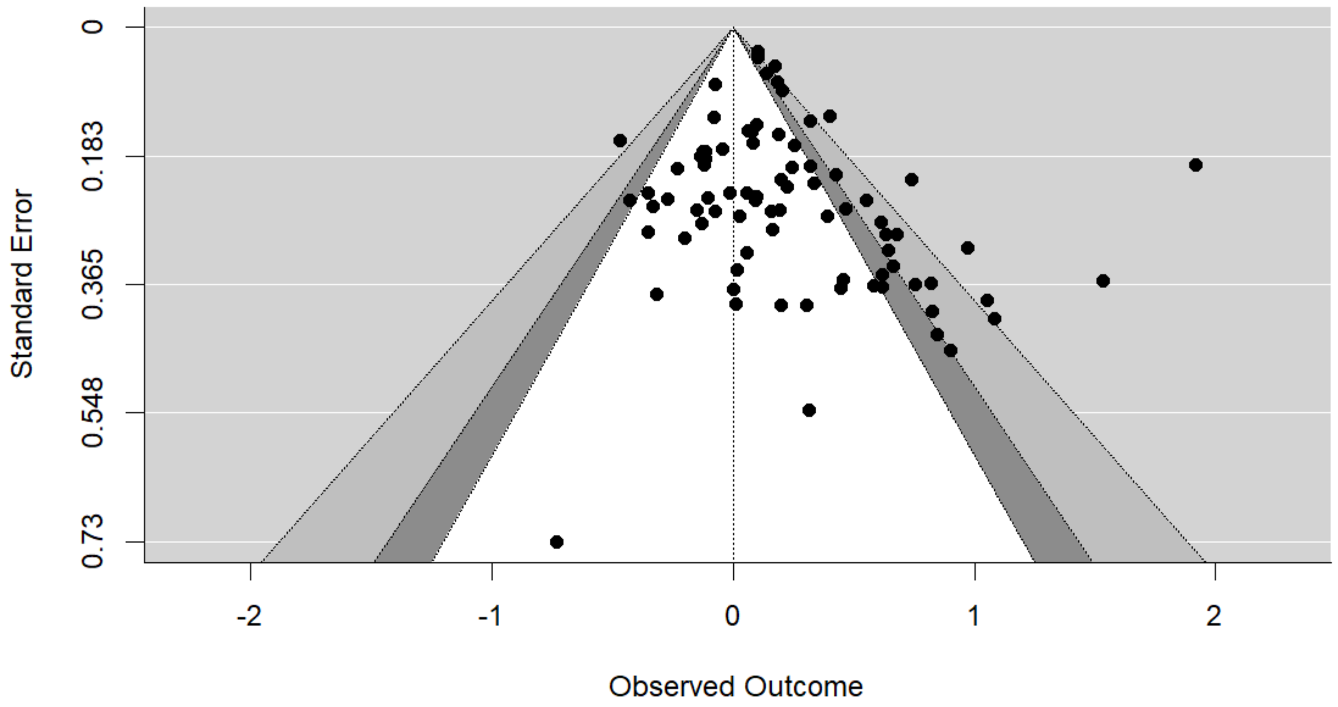
Lehmann, Elliot, & Calin-Jageman (2018). Meta-analysis of the effect of red on perceived attractiveness.

Show 10 entries Search:

	study	presentation	yi	vi
1	Banas, 2014	Paper	0.057	0.103
2	Berthold, 2013	Screen	0.554	0.06
3	Bigelow et al., 2013	Screen	0.315	0.295
4	Bigelow et al., 2013	Screen	-0.733	0.534
5	Blech, 2014	Screen	0.079	0.027
6	Blech, 2015	Screen	-0.354	0.055
7	Boelk & Madden, 2014	Paper	-0.27	0.059
8	Buechner et al., 2015	Paper	0.679	0.087
9	Costello et al., 2017	Paper	-0.125	0.031
10	Costello et al., 2017	Paper	0.075	0.022

Random effects model

- Average ES: $\hat{\mu} = \mathbf{0.207}$, robust 95% CI [0.089, 0.325]
- Heterogeneity $\hat{\tau} = \mathbf{0.318}$



Cobbling a clustered bootstrap

```r
fit_selmodel <- function(dat, index, ...) {

  # take subset of data
  boot_dat <- dat[index,]

  # fit selection model

  # compile results?

}
```

- To use `boot::boot()`, we need a function to fit the selection model.
  - `dat` argument: dataset with one row per cluster
  - `index` argument: vector of row indexes used to create bootstrap sample.
  - `...`: any further arguments

How to get one row per cluster?

Use two datasets

```
# Make a dataset of cluster IDs

cluster_IDs <-
  lehmann_dat %>%
  group_by(study) %>%
  summarise()

# Merge with full data

full_dat <-
  merge(
    cluster_IDs,
    lehmann_dat,
    by = "study"
  )
```

Use `nest_by()`

```
library(dplyr)

# Nest the data for each study
lehmann_nested <-
  lehmann_dat %>%
  nest_by(study, .key = "data")

# Recover the full dataset
full_dat <-
  lehmann_nested %>%
  unnest(data)
```

selmodel() with error handling

```r
run_sel_model <- function(dat, type, steps) {

  # initial random effects model
  RE_mod <- metafor::rma.uni(
    yi = yi, vi = vi, data = dat,
    method = "ML"
  )

  # fit selection model
  res <- metafor::selmodel(
    RE_mod, type = type, steps = steps,
    skiphes = TRUE, # turn off SE calculation
    skiphet = TRUE # turn off heterogeneity test
  )

  # compile parameter estimates into a vector
  c(beta = res$beta[,1],
    tau = sqrt(res$tau2),
    delta = if (type == "stepfun") res$delta[-1] else res$delta)

}

run_sel_model <- purrr::possibly(run_sel_model,
                                 otherwise = rep(NA_real_, 3))
```

The completed fitting function

```r
fit_selmodel <- function(dat, index = 1:nrow(dat),
                         type = "stepfun", steps = 0.025) {

  # take subset of data
  boot_dat_cluster <- dat[index, ]

  # expand to one row per effect size
  boot_dat <- tidyr::unnest(boot_dat_cluster, data)

  # build run_selmodel
  run_sel_model <- function(dat, type, steps) {
    ...
  }
  p <- 2L + length(steps)
  run_sel_model <- purrr::possibly(run_sel_model,
                                   otherwise = rep(NA_real_, p))

  # fit selection model, return vector
  run_sel_model(boot_dat, type = type, steps = steps)

}
```

Generate cluster bootstraps

```r
# Nest the data for each study
lehmann_nested <- nest_by(lehmann_dat, study, .key = "data")

fit_selmodel(lehmann_nested)
```

```
## beta.intrcpt          tau        delta
##        0.133        0.285        0.548
```

```r
tictoc::tic()

# Generate bootstraps
set.seed(20230222)

boots <- boot(
  data = lehmann_nested,
  statistic = fit_selmodel, steps = .025,
  R = 1999,
  parallel = "snow", ncpus = 8 # your mileage may vary
)

tictoc::toc()
```

```
## 46.85 sec elapsed
```

Bootstrap confidence intervals

For overall average ES

```r
boot.ci(boots, type = "perc", index = 1)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1997 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boots, type = "perc", index = 1)
##
## Intervals :
## Level     Percentile
## 95%   (-0.0014,  0.4149 )
## Calculations and Intervals on Original Scale
```

Bootstrap confidence intervals

For heterogeneity (τ)

```r
boot.ci(boots, type = "perc", index = 2)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1997 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boots, type = "perc", index = 2)
##
## Intervals :
## Level     Percentile
## 95%   ( 0.0011,  0.4951 )
## Calculations and Intervals on Original Scale
```

Bootstrap confidence intervals

For selection weight

```r
boot.ci(boots, type = "perc", index = 3)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1997 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boots, type = "perc", index = 3)
##
## Intervals :
## Level     Percentile
## 95%   ( 0.0599,  2.6702 )
## Calculations and Intervals on Original Scale
```

Discussion

- In principle, cluster bootstrap could be applied to other selective reporting detection/adjustment methods.
- We are currently studying the performance of bootstrapping a three-parameter selection model.
  - Initial results suggest that CIs have reasonable coverage.
- Future directions
  - Exploring other resampling methods such as fractional weighted bootstrap, but this requires modifying `selmodel()` implementation.
  - Turning this workflow into a more user-friendly function.

THANK YOU!