sim_dmacs_cat_p=6

```r
library(SimDesign)
```

```
Warning: package 'SimDesign' was built under R version 4.3.3
```

```r
library(lavaan)
```

```
Warning: package 'lavaan' was built under R version 4.3.3
```

```
This is lavaan 0.6-18
lavaan is FREE software! Please report any bugs.
```

```r
library(pinsearch)

# TODO:
#   - Increase the number of replications to 500
#   - Summarize the pattern of bias

# Define conditions
design <- createDesign(
    n = c(30, 100, 250, 1000)
)

# Fixed objects
set.seed(1855)
# Helper
get_ucov <- function(p, scale = sqrt(0.1), n = 5) {
    W <- matrix(rnorm(p * n), nrow = n)
    WtW <- crossprod(W)
```

```r
    D <- diag(1 / sqrt(diag(WtW))) * scale
    D %*% WtW %*% D
}
fixed <- list(
    p = 6,
    lambda = c(.3, .7, .4, .5, .6, .4),
    dlambda = list(
        c(0, 0, 0, 0, 0, 0),
        c(.3, 0, 0, 0, 0, 0)
    ),
    # nu = c(2, 3, 1.5, 3.5, 2, 3),
    tau = c(0.5, -0.5, 1, 0, -2, 0),
    alpha = c(0, -0.25),
    psi = c(1, 1.15),
    ninv_ind = c(1)
)
# lavaan syntax
fixed$mod <- paste(
    "f =~",
    paste0("y", seq_len(fixed$p), collapse = " + "), "\n",
    paste0("y", seq_len(fixed$p), "~~ 1 * y", seq_len(fixed$p),
           collapse = "\n")
)
# Compute implied means and covariances
fixed <- within(fixed, {
    lambdag <- lapply(dlambda, FUN = \(x) x + lambda)
    covy <- mapply(\(lam, psi) tcrossprod(lam) * psi + diag(length(lam)),
                   lam = lambdag, psi = psi,
                   SIMPLIFY = FALSE)
    meany <- mapply(\(lam, al) lam * al,
                    lam = lambdag, al = alpha,
                    SIMPLIFY = FALSE)
})

# Population effect size
fixed$dmacs_pop <- local({
    pooled_sd <- mapply(pinsearch:::var_from_thres,
        thres = fixed$tau[1],
        mean = lapply(fixed$meany, FUN = \(x) x[1]),
        sd = lapply(fixed$covy, FUN = \(x) sqrt(x[1, 1]))
    ) |>
```

```r
        mean() |>
        sqrt()

    dmacs_ordered(
        thresholds = matrix(rep(fixed$tau, 2),
            nrow = 2,
            byrow = TRUE
        ) |> `colnames<-`(1:6),
        loadings = sweep(
            do.call(rbind, fixed$dlambda),
            MARGIN = 2,
            STATS = fixed$lambda,
            FUN = "+"
        ),
        latent_mean = 0,
        latent_sd = 1,
        pooled_item_sd = pooled_sd
    )[1]
})

# Function for data generation
# sim_y <- function(n, lambda, nu, alpha, psi, Theta) {
#     covy <- tcrossprod(lambda) * psi + Theta
#     meany <- nu + lambda * alpha
#     MASS::mvrnorm(n, mu = meany, Sigma = covy)
# }
generate <- function(condition, fixed_objects) {
    ylist <- lapply(seq_along(fixed_objects$covy),
        FUN = function(g) {
            yg <- MASS::mvrnorm(
                condition$n,
                mu = fixed_objects$meany[[g]],
                Sigma = fixed_objects$covy[[g]]
            )
            yg <- vapply(seq_len(ncol(yg)), FUN = \(j) {
                findInterval(yg[, j], fixed_objects$tau[j])
            }, FUN.VALUE = integer(nrow(yg)))
            colnames(yg) <- paste0("y", seq_len(fixed_objects$p))
            cbind(yg, group  = g)
        })
    do.call(rbind, ylist)
```

```r
}
sim1 <- generate(design[3, ], fixed_objects = fixed)


# Analysis
analyze <- function(condition, dat, fixed_objects) {
    # Define lavaan syntax
    pinv_fit <- cfa(
        fixed_objects$mod,
        data = dat,
        group = "group", std.lv = TRUE,
        ordered = TRUE,
        group.equal = c("loadings", "thresholds"),
        group.partial = c(
            paste0("f=~y", fixed_objects$ninv_ind),
            paste0("y", fixed_objects$ninv_ind, "|t1")
        ),
        parameterization = "theta"
    )
    as.vector(pinsearch::pin_effsize(pinv_fit))
}

analyze_bc <- function(condition, dat, fixed_objects) {
    # Define lavaan syntax
    pinv_fit <- cfa(
        fixed_objects$mod,
        data = dat,
        group = "group", std.lv = TRUE,
        ordered = TRUE,
        group.equal = c("loadings", "thresholds"),
        group.partial = c(
            paste0("f=~y", fixed_objects$ninv_ind),
            paste0("y", fixed_objects$ninv_ind, "|t1")
        ),
        parameterization = "theta"
    )
    f_orig <- as.vector(pinsearch::pin_effsize(pinv_fit))
    f_boot <- lavaan::bootstrapLavaan(pinv_fit,
        R = 250,
        FUN = pinsearch::pin_effsize,
        parallel = "snow",
```

```r
        ncpus = 19
    )
    pmax(0, 2 * f_orig - colMeans(f_boot, na.rm = TRUE))
}

analyze_bc2 <- function(condition, dat, fixed_objects) {
    # Define lavaan syntax
    pinv_fit <- cfa(
        fixed_objects$mod,
        data = dat,
        group = "group", std.lv = TRUE,
        ordered = TRUE,
        group.equal = c("loadings", "thresholds"),
        group.partial = c(
            paste0("f=~y", fixed_objects$ninv_ind),
            paste0("y", fixed_objects$ninv_ind, "|t1")
        ),
        parameterization = "theta"
    )
    f_orig <- pinsearch::pin_effsize(pinv_fit)
    ns <- lavInspect(pinv_fit, what = "nobs")
    ng <- length(ns)
    f2_bias <- (ng - 1) / ng * sum(1 / ns)
    sqrt(pmax(0, f_orig^2 - f2_bias))
}

# Evaluate/Summarize
evaluate <- function(condition, results, fixed_objects) {
    results <- as.matrix(results)
    c(
        bias = colMeans(results) - fixed_objects$dmacs_pop,
        robust_bias = apply(results, 2, mean, trim = .1) -
            fixed_objects$dmacs_pop,
        emp_sd = apply(results, 2, sd),
        emp_mad = apply(results, 2, mad)
    )
}

SimClean()
# out <- runSimulation(design,
#     replications = 500,
```

```r
#     parallel = TRUE,
#     generate = generate,
#     analyse = list(naive = analyze,
#                    bc_boot = analyze_bc,
#                    bc_form = analyze_bc2),
#     summarise = evaluate,
#     filename = "results-dmacs-cat-trial",
#     save_results = TRUE,
#     packages = c("MASS", "lavaan", "pinsearch"),
#     fixed_objects = fixed
# )
```

```r
data <- readRDS("C:/Users/alex/OneDrive/Documents/results-dmacs-cat-trial.rds")

bias_table <- data.frame(
  SampleSize = data$n,
  Bias_Naive1 = data$bias.naive,
  Bias_BC_Boot1 = data$bias.bc_boot,
  Bias_BC_Form1 = data$bias.bc_form
)

robust_bias_table <- data.frame(
  SampleSize = data$n,
  Robust_Bias_Naive = data$robust_bias.naive,
  Robust_Bias_BC_Boot = data$robust_bias.bc_boot,
  Robust_Bias_BC_Form = data$robust_bias.bc_form
)

emp_sd_table <- data.frame(
  SampleSize = data$n,
  Empirical_SD_Naive = data$emp_sd.naive,
  Empirical_SD_BC_Boot = data$emp_sd.bc_boot,
  Empirical_SD_BC_Form = data$emp_sd.bc_form
)

emp_mad_table <- data.frame(
  SampleSize = data$n,
  Empirical_MAD_Naive = data$emp_mad.naive,
  Empirical_MAD_BC_Boot = data$emp_mad.bc_boot,
  Empirical_MAD_BC_Form = data$emp_mad.bc_form
)
```

```r
# Create and display tables using knitr
knitr::kable(bias_table, caption = "Bias for Naive, BC Bootstrap, and Bias-Corrected acros
```

Table 1: Bias for Naive, BC Bootstrap, and Bias-Corrected across Sample Sizes

SampleSize	Bias_Naive1	Bias_BC_Boot1	Bias_BC_Form1
30	0.4075979	0.3038443	0.3635886
100	0.1438503	0.0555087	0.1201242
250	0.0559425	0.0059237	0.0424477
1000	0.0075601	-0.0057407	0.0043066

```r
knitr::kable(robust_bias_table, caption = "Robust Bias for Naive, BC Bootstrap, and Bias-C
```

Table 2: Robust Bias for Naive, BC Bootstrap, and Bias-Corrected across Sample Sizes

SampleSize	Robust_Bias_Naive	Robust_Bias_BC_Boot	Robust_Bias_BC_Form
30	0.3826305	0.2412953	0.3441872
100	0.1163620	0.0059315	0.0957243
250	0.0455814	-0.0097682	0.0348200
1000	0.0041979	-0.0076320	0.0014530

```r
knitr::kable(emp_sd_table, caption = "Empirical Standard Deviation for Naive, BC Bootstrap
```

Table 3: Empirical Standard Deviation for Naive, BC Bootstrap, and Bias-Corrected across Sample Sizes

SampleSize	Empirical_SD_Naive	Empirical_SD_BC_Boot	Empirical_SD_BC_Form
30	0.3504513	0.4840956	0.3772408
100	0.2182071	0.2834244	0.2315206
250	0.1418793	0.1684096	0.1511943
1000	0.0837288	0.0907899	0.0856249

```r
knitr::kable(emp_mad_table, caption = "Empirical MAD for Naive, BC Bootstrap, and Bias-Cor
```

Table 4: Empirical MAD for Naive, BC Bootstrap, and Bias-Corrected across Sample Sizes

SampleSize	Empirical_MAD_Naive	Empirical_MAD_BC_Boot	Empirical_MAD_BC_Form
30	0.3577492	0.5347278	0.3877167
100	0.1805724	0.2358567	0.1945414
250	0.1593400	0.1971070	0.1645365
1000	0.0834076	0.0918275	0.0848776