sim_fmacs_p=6_b

Setup for simulation

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.3.3
```

```r
library(gt)
```

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

```r
library(knitr)
```

```
Warning: package 'knitr' 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)
library(SimDesign)
```

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

```r
# TODO:
#   - Summarize the pattern of bias

# Define conditions: Testing different sample sizes
design <- createDesign(
  n = c(30, 100, 250, 1000, 5000)
)

# 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)
  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(.1, 0, 0, 0, 0, 0),
    c(.2, -.3, 0, 0, 0, 0),
    c(.3, -.3, 0, 0, 0, 0)
  ),
```

```r
  nu = c(2, 3, 1.5, 3.5, 2, 3),
  alpha = c(0, -0.25, 0.25, 0.5),
  psi = c(1, 0.85, 1.15, 0.7),
  theta = c(1, 1.2, .8, .9, 1, 1) - .1,
  dtheta = matrix(
    runif(24, min = -0.2, max = 0.2),
    nrow = 4
  ),
  # ucov = replicate(4, get_ucov(6), simplify = FALSE)
  ucov = replicate(4, diag(.1, 6), simplify = FALSE),
  ninv_ind = c(1, 2)
)
# lavaan syntax
fixed$mod <- paste(
  "f =~",
  paste0("y", seq_len(fixed$p), collapse = " + ")
)
# Compute implied means and covariances
fixed <- within(fixed, {
  lambdag <- lapply(dlambda, FUN = \(x) x + lambda)
  Thetag <- lapply(seq_along(ucov),
                   FUN = function(g) {
                     diag(theta + dtheta[g, ]) + ucov[[g]]
                   })
  covy <- mapply(\(lam, psi, th) tcrossprod(lam) * psi + th,
                 lam = lambdag, psi = psi, th = Thetag,
                 SIMPLIFY = FALSE)
  meany <- mapply(\(lam, al, nu) nu + lam * al,
                  lam = lambdag, al = alpha, nu = list(nu),
                  SIMPLIFY = FALSE)
})

# Population effect size
fixed$fmacs_pop <- local({
  pooled_sd <- lapply(fixed$covy, FUN = \(x) diag(x)) |>
    do.call(what = rbind) |>
    colMeans() |>
    sqrt()

  fmacs(
    intercepts = matrix(rep(fixed$nu, 4),
```

```r
                          nrow = 4,
                          byrow = TRUE
    ),
    loadings = sweep(
      do.call(rbind, fixed$dlambda),
      MARGIN = 2,
      STATS = fixed$lambda,
      FUN = "+"
    ),
    latent_mean = 0,
    latent_sd = 1,
    pooled_item_sd = pooled_sd
  )[1:2]
})

# 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)
# }
```

Running the simulation

```r
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]]
                    )
                    colnames(yg) <- paste0("y", seq_len(fixed_objects$p))
                    cbind(yg, group  = g)
                  })
  do.call(rbind, ylist)
}
sim1 <- generate(design[3, ], fixed_objects = fixed)


# Analysis
```

```r
analyze <- function(condition, dat, fixed_objects) {
  # Define lavaan syntax
  pinv_fit <- cfa(
    fixed_objects$mod,
    data = dat,
    group = "group", std.lv = TRUE,
    group.equal = c("loadings", "intercepts"),
    group.partial = c(
      paste0("f=~y", fixed_objects$ninv_ind),
      paste0("y", fixed_objects$ninv_ind, "~1")
    )
  )
  as.vector(pinsearch::pin_effsize(pinv_fit))
}

analyze_bc <- function(condition, dat, fixed_objects) {
  # Define lavaan syntax
  pinv_fit <- cfa(
    fixed_objects$mod,
    data = dat,
    group = "group", std.lv = TRUE,
    group.equal = c("loadings", "intercepts"),
    group.partial = c(
      paste0("f=~y", fixed_objects$ninv_ind),
      paste0("y", fixed_objects$ninv_ind, "~1")
    )
  )
  f_orig <- as.vector(pinsearch::pin_effsize(pinv_fit))
  f_boot <- lavaan::bootstrapLavaan(pinv_fit,
                                    R = 100,
                                    FUN = pinsearch::pin_effsize
  )
  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,
```

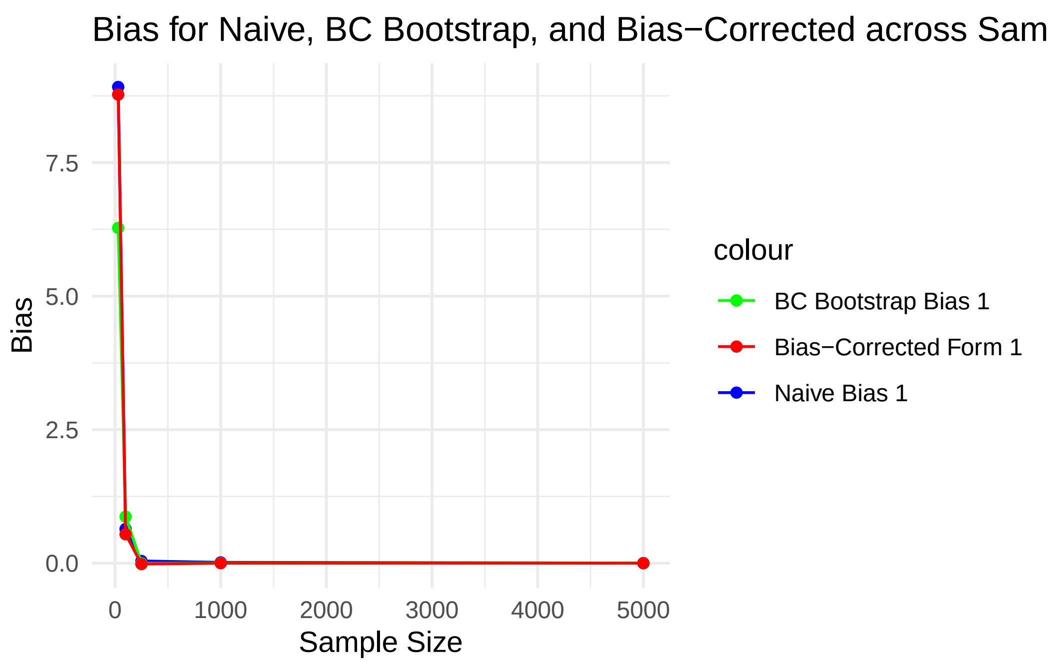
```
    group.equal = c("loadings", "intercepts"),
    group.partial = c(
      paste0("f=~y", fixed_objects$ninv_ind),
      paste0("y", fixed_objects$ninv_ind, "~1")
    )
  )
  f_orig <- pinsearch::pin_effsize(pinv_fit)
  ns <- lavInspect(pinv_fit, what = "nobs")
  ng <- length(ns)
  f2_bias <- (ng - 1) / ng * sum(1 / ns)
  sqrt(pmax(0, f_orig^2 - f2_bias))
}

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

```
# out <- runSimulation(design,
#                      replications = 250,
#                      parallel = TRUE,
#                      generate = generate,
#                      analyse = list(naive = analyze,
#                                     bc_boot = analyze_bc,
#                                     bc_form = analyze_bc2),
#                      summarise = evaluate,
#                      filename = "results-trial-bc",
#                      packages = c("MASS", "lavaan", "pinsearch"),
#                      fixed_objects = fixed
# )
data <- readRDS("C:/Users/alex/OneDrive/Documents/results-trial-bc.rds")
```

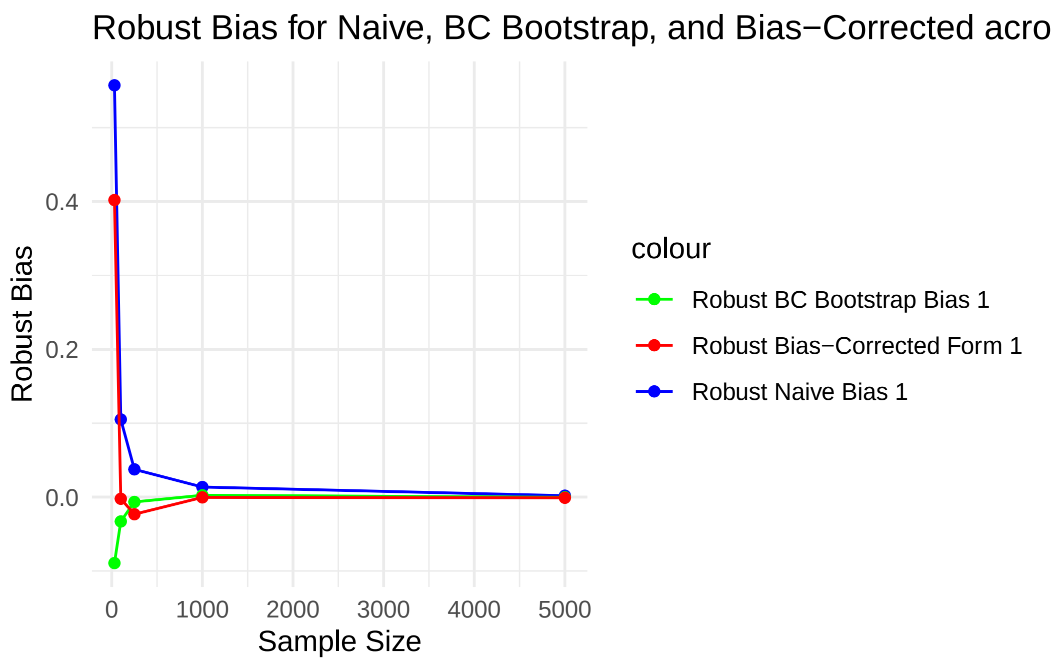
Graphs to visualize simulation results

```r
# Bias
bias_plot <- ggplot(data, aes(x = n)) +
  geom_line(aes(y = bias.naive1, color = "Naive Bias 1")) +
  geom_point(aes(y = bias.naive1, color = "Naive Bias 1")) +
  geom_line(aes(y = bias.bc_boot1, color = "BC Bootstrap Bias 1")) +
  geom_point(aes(y = bias.bc_boot1, color = "BC Bootstrap Bias 1")) +
  geom_line(aes(y = bias.bc_form1, color = "Bias-Corrected Form 1")) +
  geom_point(aes(y = bias.bc_form1, color = "Bias-Corrected Form 1")) +
  labs(title = "Bias for Naive, BC Bootstrap, and Bias-Corrected across Sample Sizes",
       x = "Sample Size", y = "Bias") +
  scale_color_manual(values = c("Naive Bias 1" = "blue",
                                "BC Bootstrap Bias 1" = "green",
                                "Bias-Corrected Form 1" = "red")) +
  theme_minimal()
bias_plot
```



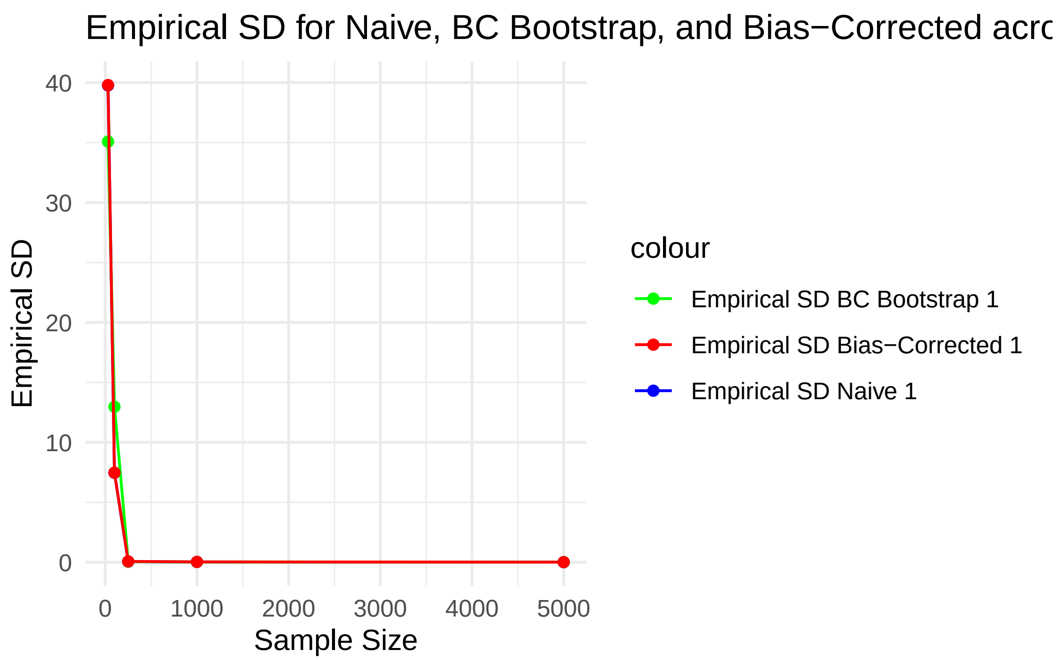

```r
# Robust Bias
robust_bias_plot <- ggplot(data, aes(x = n)) +
  geom_line(aes(y = robust_bias.naive1, color = "Robust Naive Bias 1")) +
  geom_point(aes(y = robust_bias.naive1, color = "Robust Naive Bias 1")) +
  geom_line(aes(y = robust_bias.bc_boot1, color = "Robust BC Bootstrap Bias 1")) +
```

```r
  geom_point(aes(y = robust_bias.bc_boot1, color = "Robust BC Bootstrap Bias 1")) +
  geom_line(aes(y = robust_bias.bc_form1, color = "Robust Bias-Corrected Form 1")) +
  geom_point(aes(y = robust_bias.bc_form1, color = "Robust Bias-Corrected Form 1")) +
  labs(title = "Robust Bias for Naive, BC Bootstrap, and Bias-Corrected across Sample Size
       x = "Sample Size", y = "Robust Bias") +
  scale_color_manual(values = c("Robust Naive Bias 1" = "blue",
                                "Robust BC Bootstrap Bias 1" = "green",
                                "Robust Bias-Corrected Form 1" = "red")) +
  theme_minimal()
robust_bias_plot
```

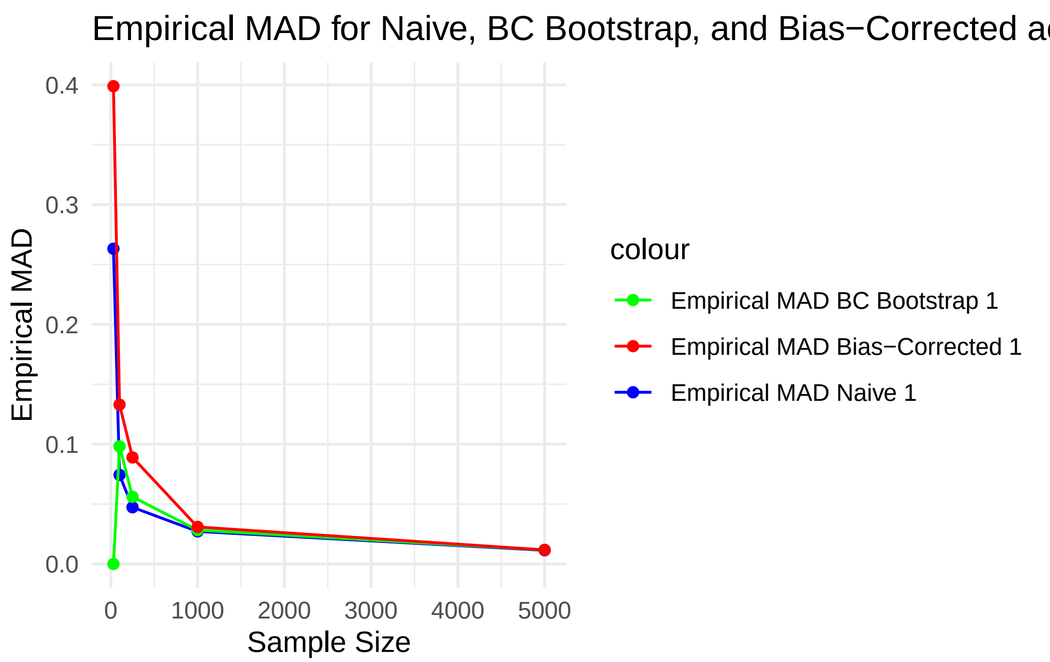



```r
# Empirical Standard Deviation
emp_sd_plot <- ggplot(data, aes(x = n)) +
  geom_line(aes(y = emp_sd.naive1, color = "Empirical SD Naive 1")) +
  geom_point(aes(y = emp_sd.naive1, color = "Empirical SD Naive 1")) +
  geom_line(aes(y = emp_sd.bc_boot1, color = "Empirical SD BC Bootstrap 1")) +
  geom_point(aes(y = emp_sd.bc_boot1, color = "Empirical SD BC Bootstrap 1")) +
  geom_line(aes(y = emp_sd.bc_form1, color = "Empirical SD Bias-Corrected 1")) +
  geom_point(aes(y = emp_sd.bc_form1, color = "Empirical SD Bias-Corrected 1")) +
  labs(title = "Empirical SD for Naive, BC Bootstrap, and Bias-Corrected across Sample Siz
       x = "Sample Size", y = "Empirical SD") +
```

```
  scale_color_manual(values = c("Empirical SD Naive 1" = "blue",
                               "Empirical SD BC Bootstrap 1" = "green",
                               "Empirical SD Bias-Corrected 1" = "red")) +
  theme_minimal()
emp_sd_plot
```




```
# Median Absolute Deviation
emp_mad_plot <- ggplot(data, aes(x = n)) +
  geom_line(aes(y = emp_mad.naive1, color = "Empirical MAD Naive 1")) +
  geom_point(aes(y = emp_mad.naive1, color = "Empirical MAD Naive 1")) +
  geom_line(aes(y = emp_mad.bc_boot1, color = "Empirical MAD BC Bootstrap 1")) +
  geom_point(aes(y = emp_mad.bc_boot1, color = "Empirical MAD BC Bootstrap 1")) +
  geom_line(aes(y = emp_mad.bc_form1, color = "Empirical MAD Bias-Corrected 1")) +
  geom_point(aes(y = emp_mad.bc_form1, color = "Empirical MAD Bias-Corrected 1")) +
  labs(title = "Empirical MAD for Naive, BC Bootstrap, and Bias-Corrected across Sample Si
       x = "Sample Size", y = "Empirical MAD") +
  scale_color_manual(values = c("Empirical MAD Naive 1" = "blue",
                               "Empirical MAD BC Bootstrap 1" = "green",
                               "Empirical MAD Bias-Corrected 1" = "red")) +
  theme_minimal()
emp_mad_plot
```



Set up for tables

```
bias_table <- data.frame(
  SampleSize = data$n,
  Bias_Naive1 = data$bias.naive1,
  Bias_BC_Boot1 = data$bias.bc_boot1,
  Bias_BC_Form1 = data$bias.bc_form1
)

robust_bias_table <- data.frame(
  SampleSize = data$n,
  Robust_Bias_Naive1 = data$robust_bias.naive1,
  Robust_Bias_BC_Boot1 = data$robust_bias.bc_boot1,
  Robust_Bias_BC_Form1 = data$robust_bias.bc_form1
)

emp_sd_table <- data.frame(
  SampleSize = data$n,
  Empirical_SD_Naive1 = data$emp_sd.naive1,
  Empirical_SD_BC_Boot1 = data$emp_sd.bc_boot1,
  Empirical_SD_BC_Form1 = data$emp_sd.bc_form1
)
```

```r
emp_mad_table <- data.frame(
  SampleSize = data$n,
  Empirical_MAD_Naive1 = data$emp_mad.naive1,
  Empirical_MAD_BC_Boot1 = data$emp_mad.bc_boot1,
  Empirical_MAD_BC_Form1 = data$emp_mad.bc_form1
)

# 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	8.9145626	6.2761694	8.7733024
100	0.6400178	0.8718441	0.5412443
250	0.0417918	-0.0037044	-0.0163374
1000	0.0148040	0.0034603	0.0000516
5000	0.0022918	0.0000941	-0.0006132

```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_Naive1	Robust_Bias_BC_Boot1	Robust_Bias_BC_Form1
30	0.5573468	-0.0893366	0.4020016
100	0.1050621	-0.0329775	-0.0023121
250	0.0375246	-0.0065947	-0.0232138
1000	0.0136022	0.0024601	-0.0004156
5000	0.0019113	-0.0002633	-0.0009805

```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_Naive1	Empirical_SD_BC_Boot1	Empirical_SD_BC_Form1
30	39.7623512	35.0814255	39.7931779

SampleSize	Empirical_SD_Naive1	Empirical_SD_BC_Boot1	Empirical_SD_BC_Form1
100	7.4620651	12.9672387	7.4695273
250	0.0526387	0.0561582	0.0738196
1000	0.0283332	0.0300447	0.0329846
5000	0.0122437	0.0124034	0.0125824

```
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_Naive1	Empirical_MAD_BC_Boot1	Empirical_MAD_BC_Form1
30	0.2631611	0.0000000	0.3988596
100	0.0743302	0.0981924	0.1329826
250	0.0473194	0.0560259	0.0889281
1000	0.0271608	0.0282376	0.0308964
5000	0.0114088	0.0117638	0.0117348