Coverage of confidence intervals with missing data

A simple demonstration

Jonas Moss

7/12/2022

This document contains a simple demonstration of the package used on multivariate Laplace distributed data. For each column $j = 1, 2, 3$ and each row i, the probability that x_{ij} is observed depends only on j and equals $p_1 = 1, p_2 = 0.75, p_3 = 0.5$.

Simulation function

```
#' Simulate data from multivariate normal with missing observations.
#'
#' @param n Number of observations.
#' @param mu Mean vector.
#' @param sigma Covariance matrix.
#' @param p Vector of probabilities for being missing.
#' @param n_reps The number of repetitions desired.
#' @param f Function applied to each row.
#' @param laplace If `TRUE`, simulates from the multivariate Laplace.
#' @return Simulated values with `f` applied to them.
simulate <- function(n, mu, sigma, p, n_reps, f = mean, laplace = FALSE) {
  r <- ncol(sigma)
  future.apply::future_replicate(n_reps,
    {
      z <- if (laplace) {
        LaplacesDemon::rmvl(n, mu = mu, Sigma = sigma)
      } else {
        MASS::mvrnorm(n, mu = mu, Sigma = sigma)
      }
      for (i in seq(r)) {
```

```
        indices <- sample(n, n - p[i] * n)
        z[indices, i] <- NA
      }
      f(z)
    },
    future.seed = TRUE
  )
}
```

Setup

Let's define a population mean vector and covariance matrix to sample from.

```
mu <- c(1, 2, 3)
sigma <- matrix(c(
  1, 0.5, 0.6,
  0.5, 1, 0.7,
  0.6, 0.7, 1
), nrow = 3)
```

Then the population value of Conger's kappa is

```
library("fleissci")
par <- conger_pop(mu, sigma)
par
```

```
[1] 0.3
```

Let's simulate some data using the observation probabilities p_1, p_2, p_3 defined above.

```
set.seed(313)
p <- c(1, 0.75, 0.5)
x <- simulate(100, mu, sigma, p, 1, f = \(x) x, laplace = TRUE)[, , 1]
head(x)
```

```
            [,1]      [,2]     [,3]
[1,]  1.33079259        NA       NA
[2,]  1.51863973 1.9186068 2.961135
[3,] -0.79813836 0.5790096       NA
```

```
[4,]  0.03915211 0.8303862 2.105929
[5,]  1.79225416 1.1677473 1.115970
[6,]  1.76370426        NA       NA
```

And it's easy to calculate confidence intervals for this data

```r
fleissci::congerci(x)
```

```
Call: fleissci::congerci(x = x)

95% confidence interval (n = 100).
    0.025     0.975
0.1299563 0.4052523

Sample estimates.
    kappa        sd
0.2676043 0.6902372
```

We can use the function `simulate` to simulate the coverage of confidence intervals.

```r
set.seed(313)
n = 100
n_reps = 10000
f <- \(x) {
  ci = fleissci::congerci(x)
  (c(ci[1] <= par) & (par <= ci[2]))
}
results <- simulate(n, mu, sigma, p, n_reps, f = f, laplace = TRUE)
mean(results)
```

```
[1] 0.8748
```

This coverage is OK (it's supposed to be 95%), but not very impressive. But we know that the multivariate Laplace distribution is elliptical, so let's try the elliptical option instead.

```r
set.seed(313)
n = 100
n_reps = 10000
f <- \(x) {
  ci = fleissci::congerci(x, type = "elliptical")
```

```
  (c(ci[1] <= par) & (par <= ci[2]))
}
results <- simulate(n, mu, sigma, p, n_reps, f = f, laplace = TRUE)
mean(results)
```

```
[1] 0.9129
```

This coverage is better, but maybe it can be improved further using the Fisher transform?

```
set.seed(313)
n = 100
n_reps = 10000
f <- \(x) {
  ci = fleissci::congerci(x, type = "elliptical", transform = "fisher")
  (c(ci[1] <= par) & (par <= ci[2]))
}
results <- simulate(n, mu, sigma, p, n_reps, f = f, laplace = TRUE)
mean(results)
```

```
[1] 0.915
```