

Package ‘funbootband’

September 18, 2026

Title Simultaneous Prediction and Confidence Bands for Functional Data

Version 0.3.0

Description Computes simultaneous prediction and confidence bands for densely sampled functional data on a common grid. The calibration builds on the functional bootstrap approach of Lenhoff et al. (1999) [doi:10.1016/S0966-6362\(98\)00043-5](https://doi.org/10.1016/S0966-6362(98)00043-5); hierarchical measurement designs are motivated by Koska et al. (2023) [doi:10.1016/j.jbiomech.2023.111506](https://doi.org/10.1016/j.jbiomech.2023.111506). Independent curves are resampled individually. Clustered data use an intact-subject bootstrap with equal subject weighting, and the clustered prediction target is one future curve from a new subject. Curves are represented by finite Fourier series, and an 'Rcpp' backend performs the bootstrap calibration.

License GPL-3

URL <https://github.com/koda86/funbootband-cran>

BugReports <https://github.com/koda86/funbootband-cran/issues>

Depends R (>= 3.5)

Imports Rcpp, stats

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

SystemRequirements C++17

Config/testthat/edition 3

ByteCompile true

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Daniel Koska [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-8245-5222>)

Maintainer Daniel Koska <dkoska@proton.me>

Repository CRAN

Date/Publication 2026-09-18 14:30:08 UTC

Contents

band	2
Index	6

band	<i>Simultaneous Bands for Functional Data</i>
------	---

Description

Create simultaneous bootstrap bands for dense functional data (rows are time points, columns are curves). For clustered designs, subjects are treated as the independent sampling units and are re-sampled intact.

Usage

```
band(
  data,
  type = c("prediction", "confidence"),
  alpha = 0.05,
  iid = TRUE,
  id = NULL,
  B = 1000L,
  k.coef = 50L
)
```

Arguments

data	Numeric matrix with T rows (time) and n columns (curves). A data.frame of numeric columns is also accepted and coerced to a matrix.
type	Character, either "prediction" or "confidence".
alpha	Numeric in (0, 1). Use 0.05 for 95% bands.
iid	Logical; if FALSE, use an intact-cluster bootstrap (requires id or infers clusters from column-name prefixes). The clustered prediction target is one future curve from a new subject. Subjects are weighted equally, and curves are weighted equally within subject.
id	Optional integer/factor vector of length ncol(data) giving a cluster id for each curve (used when iid = FALSE). If NULL and iid = FALSE, clusters are inferred from column names by prefix (up to the first underscore, hyphen, or dot).
B	Integer, number of bootstrap iterations (e.g., 1000 for final results; use smaller values in examples/tests).
k.coef	Integer; number of Fourier harmonics (default 50). Automatically clamped to $\lfloor (T - 1)/2 \rfloor$ based on the grid length. Larger values fit more high-frequency detail; smaller values smooth more.

Details

For `iid = FALSE`, the clustered prediction band is marginal over the subject population. It targets one curve from an independent new subject; it is not conditional on an already observed subject and does not target joint coverage of several future curves. The construction assumes independent subjects and exchangeable repeated curves within subject. Since calibration follows Fourier preprocessing, the formal target is the future curve in the same Fourier-reconstructed representation, rather than raw pointwise measurement noise that the chosen basis does not retain.

The i.i.d. calibration follows the curve-level functional-bootstrap target of Lenhoff et al. (1999). The clustered construction implemented here is an intact-subject bootstrap with subject-first empirical weighting. It is a revision of, rather than a literal implementation of, the hierarchical resampling description in Koska et al. (2023). Clustered confidence inference treats subject mean curves as the independent units and studentizes every bootstrap replicate with its own pointwise standard error.

Value

A list with elements `lower`, `mean`, `upper` (each of length `T`) and `meta`. For clustered prediction, `meta$target` records the estimand `"new_subject_new_curve"` and `meta$weighting` records the subject-first weighting convention.

References

- Koska, D., Oriwol, D., & Maiwald, C. (2023). Comparison of statistical models for characterizing continuous differences between two biomechanical measurement systems. *Journal of Biomechanics*, 149, 111506. doi:[10.1016/j.jbiomech.2023.111506](https://doi.org/10.1016/j.jbiomech.2023.111506)
- Lenhoff, M. W., Santner, T. J., Otis, J. C., Peterson, M. G. E., Williams, B. J., & Backus, S. I. (1999). Bootstrap prediction and confidence bands: a superior statistical method for analysis of gait data. *Gait & Posture*, 9(1), 10–17. doi:[10.1016/S0966-6362\(98\)00043-5](https://doi.org/10.1016/S0966-6362(98)00043-5)
- Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap Methods and Their Application*. Cambridge University Press. doi:[10.1017/cbo9780511802843](https://doi.org/10.1017/cbo9780511802843)

Examples

```
## Independent-curve example

set.seed(1)
T <- 101L
n <- 30L
x <- seq(0, 1, length.out = T)
mu_true <- 0.7 * sin(2 * pi * x) - 0.2 * cos(4 * pi * x)

generate_curve <- function() {
  mu_true +
    rnorm(1, sd = 0.35) +
    rnorm(1, sd = 0.30) * sin(2 * pi * x) +
    rnorm(1, sd = 0.20) * cos(2 * pi * x) +
    rnorm(1, sd = 0.15) * sin(4 * pi * x)
}

Y <- replicate(n, generate_curve())
```

```

fit_pred <- band(Y, type = "prediction", alpha = 0.10,
  iid = TRUE, B = 500L, k.coef = 4L)
fit_conf <- band(Y, type = "confidence", alpha = 0.10,
  iid = TRUE, B = 500L, k.coef = 4L)

ylim <- range(c(Y, fit_pred$lower, fit_pred$upper), finite = TRUE)
plot(x, fit_pred$mean, type = "n", ylim = ylim,
  xlab = "Normalized time", ylab = "Value",
  main = "Simultaneous bands (i.i.d.)")
matlines(x, Y, col = grDevices::adjustcolor("gray40", 0.25), lty = 1)
polygon(c(x, rev(x)), c(fit_pred$lower, rev(fit_pred$upper)),
  col = grDevices::adjustcolor("steelblue", 0.25), border = NA)
polygon(c(x, rev(x)), c(fit_conf$lower, rev(fit_conf$upper)),
  col = grDevices::adjustcolor("darkorange", 0.30), border = NA)
lines(x, fit_pred$mean, lwd = 2)
lines(x, mu_true, col = "red", lwd = 2, lty = 2)
## Clustered example: repeated curves nested within subjects

set.seed(2)
T <- 101L
x <- seq(0, 1, length.out = T)

# Twelve independent subjects contribute unequal numbers of repeated curves.
K_subject <- 12L
m <- rep(c(2L, 3L, 4L), length.out = K_subject)
id <- rep(seq_len(K_subject), m)

mu_true <- 0.7 * sin(2 * pi * x) - 0.2 * cos(4 * pi * x)

# Smooth subject-specific deviations from the population mean.
subject_effect <- sapply(seq_len(K_subject), function(i) {
  rnorm(1, sd = 0.35) +
    rnorm(1, sd = 0.30) * sin(2 * pi * x) +
    rnorm(1, sd = 0.20) * cos(2 * pi * x)
})

# Smooth curve-to-curve deviations within a subject.
within_subject_effect <- function() {
  rnorm(1, sd = 0.18) * sin(4 * pi * x) +
    rnorm(1, sd = 0.12) * cos(4 * pi * x)
}

Y <- sapply(seq_along(id), function(j) {
  mu_true + subject_effect[, id[j]] + within_subject_effect()
})

trial <- ave(id, id, FUN = seq_along)
colnames(Y) <- paste0("subject", id, "_trial", trial)

# The prediction target is one Fourier-reconstructed curve from a new subject.
fit_pred <- band(Y, type = "prediction", alpha = 0.10,
  iid = FALSE, id = id, B = 500L, k.coef = 4L)

```

```
# The confidence target is the equally subject-weighted population mean curve.
fit_conf <- band(Y, type = "confidence", alpha = 0.10,
               iid = FALSE, id = id, B = 500L, k.coef = 4L)

fit_pred$meta[c("target", "weighting", "bootstrap_unit", "n_clusters")]

# Plot the results.
ylim <- range(c(Y, fit_pred$lower, fit_pred$upper), finite = TRUE)
plot(x, fit_pred$mean, type = "n", ylim = ylim,
     xlab = "Normalized time", ylab = "Value",
     main = "Simultaneous bands (clustered)")
matlines(x, Y, col = grDevices::adjustcolor("gray40", 0.20), lty = 1)
polygon(c(x, rev(x)), c(fit_pred$lower, rev(fit_pred$upper)),
       col = grDevices::adjustcolor("steelblue", 0.25), border = NA)
polygon(c(x, rev(x)), c(fit_conf$lower, rev(fit_conf$upper)),
       col = grDevices::adjustcolor("darkorange", 0.30), border = NA)
lines(x, fit_pred$mean, lwd = 2)
lines(x, mu_true, col = "red", lwd = 2, lty = 2)
```

Index

band, [2](#)