# Package: bayescoveragedeploy (via r-universe)

June 10, 2026

**Title** Precompiled Stan Models for Deployment of Bayesian Coverage Model

**Version** 0.21

**Description** Provides precompiled Stan models for fitting local country-level Bayesian hierarchical transition models for health coverage indicators. This package uses the rstan backend with precompiled models for users without C++ compilers. All core functionality is provided by the bayescoveragemodel package.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Biarch** true

**Depends** R (>= 3.5.0)

**Imports** bayescoveragemodel, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), rstantools (>= 2.5.0)

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**SystemRequirements** GNU make

**URL** https://github.com/AlkemaLab/bayescoveragedeploy

**BugReports** https://github.com/AlkemaLab/bayescoveragedeploy/issues

**Config/testthat/edition** 3

**Remotes** AlkemaLab/localhierarchy, AlkemaLab/bayescoveragemodel

**Config/pak/sysreqs** cmake make libicu-dev libx11-dev

**Repository** https://alkemalab.r-universe.dev

**Date/Publication** 2026-06-10 01:31:02 UTC

**RemoteUrl** https://github.com/AlkemaLab/bayescoveragedeploy

**RemoteRef** HEAD

**RemoteSha** 0a2cb0d1a6ff6031a4f75ce8944435a25fd0ae7c

# Contents



---

`bayescoveragedeploy-package`
*The 'bayescoveragedeploy' package.*

---

## Description

Provides precompiled Stan models for fitting local country-level Bayesian hierarchical transition models for health coverage indicators (ANC4, institutional delivery, vaccination). This package uses the rstan backend with precompiled models for users without C++ compilers. All core functionality is provided by the bayescoveragemodel package.

## Details

This package contains precompiled Stan models for the following variants:

- fpem - Basic transition model
- fpem_routine - Model with routine data
- fpem_aggregates - Model with subnational aggregates
- fpem_routine_aggregates - Model with routine data and aggregates

The main user-facing function is `fit_local_model`.

## Author(s)

**Maintainer**: Leontine Alkema <lalkema@umass.edu>

## References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.7. https://mc-stan.org

## See Also

Useful links:

- https://github.com/AlkemaLab/bayescoveragedeploy
- Report bugs at https://github.com/AlkemaLab/bayescoveragedeploy/issues

`fit_local_model` *Fit local country model with precompiled Stan models*

## Description

This function provides a simplified interface for fitting country-specific models using precompiled Stan models. All data processing and model logic is delegated to bayescoveragemodel::fit_model().

## Usage

```
fit_local_model(
  survey_df,
  iso_select,
  indicator = "anc4",
  routine_df = NULL,
  chains = 4,
  seed = 1234,
  refresh = 10,
  iter_sampling = 300,
  iter_warmup = 150,
  adapt_delta = 0.95,
  max_treedepth = 14,
  ...
)
```

## Arguments

| | |
|---|---|
| `survey_df` | Survey data (processed via bayescoveragemodel::process_data) |
| `iso_select` | ISO code for the country |
| `indicator` | Indicator name ("anc4", "ideliv", "vdpt", etc.) |
| `routine_df` | Optional tibble with routine data |
| `chains` | Number of MCMC chains (default 4) |
| `seed` | Random seed (default 1234) |
| `refresh` | Progress update frequency (default 10) |
| `iter_sampling` | Number of sampling iterations (default 200) |
| `iter_warmup` | Number of warmup iterations (default 150) |
| `adapt_delta` | Target acceptance rate (default 0.9) |
| `max_treedepth` | Maximum tree depth (default 14) |
| `...` | Additional arguments passed to bayescoveragemodel::fit_model() |

## Value

Model fit object (same structure as bayescoveragemodel::fit_model)

## Examples

```
## Not run:
# ===== Quick Example =====
library(haven)
dat0 <- read_dta("data_raw/ICEH_national.dta")
regions_dat <- readr::read_csv("data_raw/regions_updated.csv")

# Process data
data <- bayescoveragemodel::process_data(
  dat = dat0,
  regions_dat = regions_dat,
  indicator = "anc4"
)

# Fit model (fast test)
fit <- fit_local_model(
  survey_df = data,
  iso_select = "KEN",
  indicator = "anc4",
  chains = 1,
  iter_sampling = 5,
  iter_warmup = 5
)

# Plot results
bayescoveragemodel::plot_estimates_local_all(fit)

# ===== Production Example =====
fit_prod <- fit_local_model(
  survey_df = data,
  iso_select = "KEN",
  indicator = "anc4",
  chains = 4,
  iter_sampling = 200,
  iter_warmup = 150,
  seed = 123
)

# ===== More Examples =====
# See BayesCoverage app!

## End(Not run)
```

# Index