Package: rcbayes (via r-universe)

October 29, 2026

Title Estimate Rogers-Castro Migration Age Schedules with Bayesian Models

Version 0.3.0

Description A collection of functions to estimate Rogers-Castro migration age schedules using 'Stan'. This model which describes the fundamental relationship between migration and age in the form of a flexible multi-exponential migration model was most notably proposed in Rogers and Castro (1978) <doi:10.1068/a100475>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

Depends R (>= 3.4.0)

Imports methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), rstantools (>= 2.1.1), Rdpack, dplyr, rlang, stats, tibble, tidybayes, magrittr, shiny, shinythemes

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, ggplot2

SystemRequirements GNU make

VignetteBuilder knitr

RdMacros Rdpack

NeedsCompilation yes

Author Jessie Yeung [aut, cre] (ORCID: <https://orcid.org/0000-0002-0319-6892>), Monica Alexander [aut] (ORCID: <https://orcid.org/0000-0002-8135-3435>), Tim Riffe [aut] (ORCID: <https://orcid.org/0000-0002-2673-4622>)

Maintainer Jessie Yeung <jessieyeung1@gmail.com>

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2025-08-28 02:40:02 UTC

RemoteUrl https://github.com/cran/rcbayes

RemoteRef HEAD

RemoteSha 99f013262fbfb3fc14df05e1936274f7ef581eb2

Contents

rcbayes-package 2
init_rc 3
interact_rc 5
mig_calculate_rc 6
mig_estimate_rc 7

Index **10**

`rcbayes-package` *The 'rcbayes' package.*

Description

A collection of functions to estimate Rogers-Castro migration age schedules using 'Stan'. This model which describes the fundamental relationship between migration and age in the form of a flexible multi-exponential migration model was most notably proposed in Rogers and Castro (1978) doi:10.1068/a100475.

A DESCRIPTION OF THE PACKAGE

Author(s)

Maintainer: Jessie Yeung <jessieyeung1@gmail.com> (ORCID)

Authors:

- Monica Alexander (ORCID)
- Tim Riffe (ORCID)

References

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

`init_rc` *Set initial values for Rogers-Castro migration model*

Description

Choose initial values for parameters in the Rogers-Castro model in a strategic way based on your data. Provide these initial values to improve convergence of model. Intended to be used with rcbayes::mig_estimate_rc as an additional input into 'Stan'.

Usage

```
init_rc(
  ages,
  migrants,
  pop,
  mx,
  pre_working_age,
  working_age,
  retirement,
  post_retirement,
  nchains = 4,
  net_mig
)
```

Arguments

`ages`	numeric. A vector of integers for ages.
`migrants`	numeric. A vector of integers for observed age-specific migrants.
`pop`	numeric. A vector of integers for age-specific population or sample sizes, of which "migrants" experienced a migration event.
`mx`	numeric. A vector of age-specific migration rates.
`pre_working_age`	logical (TRUE/FALSE). Whether or not you are including pre working age component.
`working_age`	logical (TRUE/FALSE). Whether or not you are including working age component.
`retirement`	logical (TRUE/FALSE). Whether or not you are including retirement age component.
`post_retirement`	logical (TRUE/FALSE). Whether or not you are including post retirement age component.
`nchains`	numeric. A positive integer specifying the number of Markov chains. Should be 4 unless changed otherwise.
`net_mig`	numeric. Deprecated argument, use migrants instead.

Value

A list of length nchains. Each element of the list is a list of numeric values. Within the inner lists, there is one element for every model parameter.

Examples

```
# Ex. 1: Using ages, migrants, and population
ages <- 0:80
migrants <- c(202,215,167,188,206,189,164,
              158,197,185,176,173,167,198,
              203,237,249,274,319,345,487,
              491,521,505,529,527,521,529,
              507,484,467,439,399,399,380,
              368,310,324,289,292,270,269,
              285,254,245,265,257,258,263,
              253,346,293,332,346,349,355,
              386,346,344,352,331,320,307,
              320,310,258,254,243,256,263,
              183,169,172,160,166,113,132,
              111,130,110,113)
pop <- c(105505,105505,105505,105505,105505,
         106126,106126,106126,106126,106126,
         100104,100104,100104,100104,100104,
         114880,114880,114880,114880,114880,
         136845,136845,136845,136845,136845,
         136582,136582,136582,136582,136582,
         141935,141935,141935,141935,141935,
         134097,134097,134097,134097,134097,
         130769,130769,130769,130769,130769,
         133718,133718,133718,133718,133718,
         154178,154178,154178,154178,154178,
         145386,145386,145386,145386,145386,
         126270,126270,126270,126270,126270,
         108314,108314,108314,108314,108314,
         79827,79827,79827,79827,79827,59556,
         59556,59556,59556,59556,59556)

#compute initial values
iv <- init_rc(ages=ages, migrants=migrants, pop=pop,
              pre_working_age=TRUE,
              working_age=TRUE,
              retirement=TRUE,
              post_retirement=TRUE)

# Ex 2: Using ages and mx
ages <- 0:80
mx <- c(0.001914601, 0.002037818, 0.001582863, 0.001781906,
         0.001952514, 0.001780902, 0.001545333, 0.001488796,
         0.001856284, 0.001743211, 0.001758172, 0.001728203,
         0.001668265, 0.001977943, 0.002027891, 0.002063022,
         0.002167479, 0.002385097, 0.002776811, 0.003003134,
         0.003558771, 0.003588001, 0.003807227, 0.003690307,
```

```
          0.003865687, 0.003858488, 0.003814558, 0.003873131,
          0.003712056, 0.003543659, 0.003290238, 0.003092965,
          0.002811146, 0.002811146, 0.002677282, 0.002744282,
          0.002311759, 0.002416161, 0.002155156, 0.002177528,
          0.002064710, 0.002057062, 0.002179416, 0.001942356,
          0.001873533, 0.001981783, 0.001921955, 0.001929434,
          0.001966826, 0.001892041, 0.002244159, 0.001900401,
          0.002153355, 0.002244159, 0.002263617, 0.002441776,
          0.002655001, 0.002379872, 0.002366115, 0.002421141,
          0.002621367, 0.002534252, 0.002431298, 0.002534252,
          0.002455057, 0.002381964, 0.002345034, 0.002243477,
          0.002363499, 0.002428126, 0.002292457, 0.002117078,
          0.002154659, 0.002004334, 0.002079497, 0.001897374,
          0.002216401, 0.001863792, 0.002182820, 0.001847001,
          0.001897374)

# compute initial values
iv <- init_rc(ages=ages, mx=mx,
              pre_working_age=TRUE,
              working_age=TRUE,
              retirement=TRUE,
              post_retirement=TRUE)
```

`interact_rc` *Run Interactive Rogers-Castro App*

Description

Run an interactive Rogers-Castro app. Use interactive sliders to see how parameters affect the Rogers-Castro age schedules.

Usage

```
interact_rc()
```

Value

No return value, called for interactive widget

Examples

```
## Not run:
interact_rc()

## End(Not run)
```

`mig_calculate_rc` *Calculate Rogers-Castro migration age schedule*

Description

Given a set of ages and parameters, calculate the migration age schedule based on the Rogers and Castro formula. Choose between a 7, 9, 11 or 13 parameter model.

Usage

```
mig_calculate_rc(ages, pars)
```

Arguments

`ages`	numeric. A vector of ages for migration rates to be calculated.
`pars`	numeric. A named list of parameters. See below for details.

Details

In the full 13 parameter model, the migration rate at age x, $m(x)$ is defined as

$$m(x) = a1*exp(-1*alpha1*x)+a2*exp(-1*alpha2*(x-mu2)-exp(-1*lambda2*(x-mu2)))+a3*exp(-1*alpha3*$$

The first, second, third and fourth pieces of the equation represent pre-working age, working age, retirement and post-retirement age patterns, respectively. Models with less parameters gradually remove terms at the older ages. Parameters in each family are:

- pre-working age: a1, alpha1
- working age: a2, alpha2, mu2, lambda2
- retirement: a3, alpha3, mu3, lambda3
- post retirement: a4, lambda4

For a specific family to be included, values for all parameters in that family must be specified.

Value

A vector the same length as `ages`. Values represent migration rate for each age in `ages`.

References

Rogers A, Castro LJ (1981). *Model migration schedules*. RR-81-030.

Examples

```
pars <- c(a1= 0.09, alpha1= 0.1, a2= 0.2,
alpha2= 0.1, mu2= 21, lambda2= 0.39, a3= 0.001,
alpha3= 1, mu3= 67, lambda3= 0.6, c= 0.01)
ages <- 0:75
mx <- mig_calculate_rc(ages = ages, pars = pars)

plot(ages, mx, type = 'l')
```

`mig_estimate_rc` *Estimate Rogers-Castro migration age schedule*

Description

Given a set of ages and observed age-specific migrants, estimate the parameters of a Roger-Castro model migration schedule. Choose between a 7, 9, 11 or 13 parameter model.

Usage

```
mig_estimate_rc(
  ages,
  migrants,
  pop,
  mx,
  sigma,
  pre_working_age,
  working_age,
  retirement,
  post_retirement,
  net_mig,
  ...
)
```

Arguments

`ages`	numeric. A vector of integers for ages.
`migrants`	numeric. A vector of integers for observed age-specific migrants.
`pop`	numeric. A vector of integers for age-specific population or sample sizes, of which "migrants" experienced a migration event.
`mx`	numeric. A vector of age-specific migration rates.
`sigma`	numeric. Standard deviation of migration rates for Normal model. Argument is option, standard deviation is estimated if Normal model is run without being specified.
`pre_working_age`	logical (TRUE/FALSE). Whether or not to include pre working age component.

working_age	logical (TRUE/FALSE). Whether or not to include working age component.
retirement	logical (TRUE/FALSE). Whether or not to include retirement age component.
post_retirement	logical (TRUE/FALSE). Whether or not to include post retirement age component.
net_mig	numeric. Deprecated argument, use migrants instead.
...	additional inputs to stan, see ?rstan::stan for details.

Value

A list of length 3. The first element, pars_df, is a data frame that provides parameter estimates with 95% credible intervals. The second element, fit_df, is a data frame that shows the data and estimated migration rates at each age. The third element, check_converge, is a data frame that provides the R-hat values and effective sample sizes.

Examples

```
# Ex 1: Run poisson model using ages, migrants, and population
ages <- 0:80
migrants <- c(202,215,167,188,206,189,164,
              158,197,185,176,173,167,198,
              203,237,249,274,319,345,487,
              491,521,505,529,527,521,529,
              507,484,467,439,399,399,380,
              368,310,324,289,292,270,269,
              285,254,245,265,257,258,263,
              253,346,293,332,346,349,355,
              386,346,344,352,331,320,307,
              320,310,258,254,243,256,263,
              183,169,172,160,166,113,132,
              111,130,110,113)
pop <- c(105505,105505,105505,105505,105505,
         106126,106126,106126,106126,106126,
         100104,100104,100104,100104,100104,
         114880,114880,114880,114880,114880,
         136845,136845,136845,136845,136845,
         136582,136582,136582,136582,136582,
         141935,141935,141935,141935,141935,
         134097,134097,134097,134097,134097,
         130769,130769,130769,130769,130769,
         133718,133718,133718,133718,133718,
         154178,154178,154178,154178,154178,
         145386,145386,145386,145386,145386,
         126270,126270,126270,126270,126270,
         108314,108314,108314,108314,108314,
         79827,79827,79827,79827,79827,59556,
         59556,59556,59556,59556,59556)


# fit the model
res <- mig_estimate_rc(ages = ages, migrants = migrants, pop = pop,
```

```
                        pre_working_age = TRUE,
                        working_age = TRUE,
                        retirement = TRUE,
                        post_retirement = FALSE,
                        #optional inputs into stan
                        control = list(adapt_delta = 0.95, max_treedepth = 10),
                        iter = 10, chains = 1 #to speed up example
                        )
# plot the results and data
plot(ages, migrants/pop, ylab = "migration rate", xlab = "age")
lines(ages, res[["fit_df"]]$median, col = "red")
legend("topright", legend=c("data", "fit"), col=c("black", "red"), lty=1, pch = 1)

# Ex 2: Run normal model using ages and mx
ages <- 0:80
mx <- c(0.001914601, 0.002037818, 0.001582863, 0.001781906,
        0.001952514, 0.001780902, 0.001545333, 0.001488796,
        0.001856284, 0.001743211, 0.001758172, 0.001728203,
        0.001668265, 0.001977943, 0.002027891, 0.002063022,
        0.002167479, 0.002385097, 0.002776811, 0.003003134,
        0.003558771, 0.003588001, 0.003807227, 0.003690307,
        0.003865687, 0.003858488, 0.003814558, 0.003873131,
        0.003712056, 0.003543659, 0.003290238, 0.003092965,
        0.002811146, 0.002811146, 0.002677282, 0.002744282,
        0.002311759, 0.002416161, 0.002155156, 0.002177528,
        0.002064710, 0.002057062, 0.002179416, 0.001942356,
        0.001873533, 0.001981783, 0.001921955, 0.001929434,
        0.001966826, 0.001892041, 0.002244159, 0.001900401,
        0.002153355, 0.002244159, 0.002263617, 0.002441776,
        0.002655001, 0.002379872, 0.002366115, 0.002421141,
        0.002621367, 0.002534252, 0.002431298, 0.002534252,
        0.002455057, 0.002381964, 0.002345034, 0.002243477,
        0.002363499, 0.002428126, 0.002292457, 0.002117078,
        0.002154659, 0.002004334, 0.002079497, 0.001897374,
        0.002216401, 0.001863792, 0.002182820, 0.001847001,
        0.001897374)

# fit the model
res <- mig_estimate_rc(ages = ages, mx = mx,
                        pre_working_age = TRUE,
                        working_age = TRUE,
                        retirement = TRUE,
                        post_retirement = FALSE,
                        #optional inputs into stan
                        control = list(adapt_delta = 0.95, max_treedepth = 10),
                        iter = 10, chains = 1 #to speed up example
                        )
```

Index

init_rc, 3
interact_rc, 5

mig_calculate_rc, 6
mig_estimate_rc, 7

rcbayes *(*rcbayes-package*)*, 2
rcbayes-package, 2