Package: healthsim (via r-universe)

November 14, 2026

Title Simulate Synthetic Health Insurance Portfolios and Panels

Version 0.1.0

Description Generates synthetic South African health insurance data with a known data-generating process, for teaching and testing pricing models. One function simulates a cross-sectional portfolio of principal members with negative binomial claim counts and a lognormal-Pareto severity mixture. Another simulates a multi-year open-cohort panel with ageing, smoking cessation, wearable activity and sleep data joined by a Clayton copula, claims inflation, annual benefit limits, lapses and deaths. Every parameter can be changed, and the true expected frequency and severity are available for checking fitted models.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/Sbusiso-Mdingi/healthsim, https://www.kaggle.com/sbusisomdingiayesbu

BugReports https://github.com/Sbusiso-Mdingi/healthsim/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Sbusiso Mdingi [aut, cre, cph] (ORCID: <https://orcid.org/0009-0003-4417-968X>)

Maintainer Sbusiso Mdingi <sbusiso@mdingi.co.za>

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

Date/Publication 2026-11-13 18:00:02 UTC

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

RemoteRef HEAD

RemoteSha 4ce9eecaf99efe599912a27e122b0a93f6f22001

Contents

hs_provinces 2
panel_params 3
portfolio_params 3
set_levels 4
simulate_panel 5
simulate_portfolio 6
true_frequency 7
z_mean 8

Index **9**

`hs_provinces` *Provinces and benefit tiers*

Description

The nine South African provinces and the three benefit tiers used by the simulators, in the order that parameter vectors follow.

Usage

```
hs_provinces()

hs_plans()
```

Details

Essential covers in-hospital treatment; Standard adds out-of-hospital specialists, scans and tests; Comprehensive adds GP visits, prescribed medicines, dental and optical.

Value

A character vector.

Examples

```
hs_provinces()
hs_plans()
```

panel_params *Parameters of the panel*

Description

Returns the default parameters of `simulate_panel()`. Claims follow `portfolio_params()`, passed as `portfolio`, plus the panel terms below.

Usage

```
panel_params(..., portfolio = portfolio_params())
```

Arguments

`...`	Panel parameters to change, for example `lapse_base = 0.15`.
`portfolio`	Claims parameters, from `portfolio_params()`.

Value

A named list.

Examples

```
p <- panel_params(claims_inflation = 0.10,
                  portfolio = portfolio_params(freq_smoker = 0.4))
```

portfolio_params *Parameters of the cross-sectional portfolio*

Description

Returns the default data-generating parameters of `simulate_portfolio()`, with any named argument replacing its default. Unknown names are an error, so a typo cannot silently leave a default in place.

Usage

```
portfolio_params(...)
```

Arguments

`...`	Parameters to change, for example `freq_smoker = 0.4`.

Details

Frequency terms are on the log scale of claims per policy year, relative to a base profile: a 40-year-old woman in Gauteng on the Essential plan, non-smoker, BMI 25, no children. Severity terms are on the log scale of the expected cost per claim, in rand. Each claim is the severity scale times Z, where Z is lognormal with probability `1 - large_prob` and a Pareto large claim otherwise.

The defaults are illustrative assumptions, not estimates from real data.

Value

A named list.

Examples

```
p <- portfolio_params(freq_smoker = 0.4, frailty_var = 0)
p$freq_smoker
```

---

set_levels *Set factor levels with the base profile first*

Description

Converts `Sex`, `Smoker`, `Province` and `Plan` to factors whose first level is the base profile, so GLM coefficients line up with the parameters.

Usage

```
set_levels(df)
```

Arguments

`df`	A data frame from `simulate_portfolio()` or `simulate_panel()`.

Value

`df` with factor columns.

`simulate_panel` *Simulate a multi-year health insurance panel*

Description

Follows an open cohort of principal members for several calendar years, one row per member per year. The book starts with `n` members in force, and each year new members join, members age, change BMI, quit smoking, have children, claim, and lapse or die.

Usage

```
simulate_panel(
  n = 1e+05,
  years = 5,
  seed = 123,
  params = panel_params(),
  truth = FALSE
)
```

Arguments

`n`	Members in force at the start.
`years`	Number of calendar years.
`seed`	Random seed, or `NULL` to use the current RNG state.
`params`	Parameters from `panel_params()`.
`truth`	If `TRUE`, add the latent values used to simulate each row: `True_Frequency` (expected claims per year given the member's activity, without frailty), `True_Severity` (expected cost per claim before the limit), `Frailty`, and `Steps_Latent` and `Sleep_Latent` for all members.

Details

Claims follow the portfolio model (`simulate_portfolio()`) with three additions:

- Frailty has a lasting part and a yearly part, so past claims predict future claims.
- Daily steps and sleep efficiency lower claim frequency for everyone, but are only recorded for members of the wellness programme. The two are joined by a Clayton copula, so low activity and poor sleep tend to come together. Active members join the programme more often, so the missing values are not missing at random.
- Cost per claim grows with `claims_inflation`, while annual benefit limits stay fixed in rand, so more members reach the limit each year.

Lapses happen at the end of a year and are more likely in the first policy year and for members who did not claim, so the remaining book gets worse over time. Deaths happen during the year and end exposure early.

Value

A data frame with one row per member per calendar year: `Policy_ID`, `Calendar_Year`, `Entry_Year`, `Duration` (policy year, 1 in the year of joining), `Age`, `Sex`, `BMI`, `Smoker`, `Province`, `Plan`, `Children`, `Wellness`, `Avg_Daily_Steps` and `Sleep_Efficiency` (`NA` outside the wellness programme), `Exposure`, `ClaimCount`, `Charges` (incurred, rand), `Annual_Limit`, `Charges_Paid` (capped at the limit), `Limit_Reached`, and `Exit` (`"none"`, `"lapse"` or `"death"`). Members with `Exit == "none"` in the last year are still in force.

Examples

```
pn <- simulate_panel(2000, years = 3, seed = 1)
table(pn$Calendar_Year, pn$Exit)

# Faster claims inflation
pp <- panel_params(claims_inflation = 0.12)
pn2 <- simulate_panel(2000, years = 3, seed = 1, params = pp)
```

---

simulate_portfolio *Simulate a cross-sectional health insurance portfolio*

Description

Simulates one year of a portfolio of principal members with their dependent children. Claim counts are Poisson given a gamma frailty, so negative binomial overall, and every claim has its own cost from a lognormal-Pareto mixture. A policy's `Charges` are the sum of its claims.

Usage

```
simulate_portfolio(
  n = 1e+05,
  seed = 123,
  params = portfolio_params(),
  claims = FALSE,
  id = FALSE
)
```

Arguments

`n`	Number of policies.
`seed`	Random seed, or `NULL` to use the current RNG state.
`params`	Parameters from `portfolio_params()`.
`claims`	If `TRUE`, return a list with the policies and a claim-level table (one row per claim) instead of the policies alone.
`id`	If `TRUE`, add a `Policy_ID` column.

Details

With the defaults, `simulate_portfolio(1e5, seed = 123)` gives the Kaggle dataset "Health Insurance Actuarial Playground".

Value

A data frame of policies with columns `Age`, `Sex`, `BMI`, `Smoker`, `Province`, `Plan`, `Children`, `Exposure`, `ClaimCount` and `Charges` (rand), or a list of `policies` and `claims` when `claims = TRUE`.

Examples

```
df <- simulate_portfolio(1000, seed = 1)
head(df)
sum(df$ClaimCount) / sum(df$Exposure)

# A portfolio where smoking costs more
df2 <- simulate_portfolio(1000, seed = 1, portfolio_params(freq_smoker = 0.5))
```

`true_frequency` *True expected frequency and severity*

Description

The expected claims per policy year and the expected cost per claim under the data-generating parameters, for comparing with fitted models. `true_frequency()` is the mean over the frailty, so it is the target of a Poisson or negative binomial GLM with log exposure as offset.

Usage

```
true_frequency(df, params = portfolio_params())

true_severity(df, params = portfolio_params())
```

Arguments

`df`	A data frame with the portfolio columns.
`params`	Parameters from `portfolio_params()`.

Value

A numeric vector, one value per row of `df`.

Examples

```
df <- simulate_portfolio(1000, seed = 1)
mean(true_frequency(df))
mean(true_severity(df))
```

z_mean *Moments of the claim-size multiplier*

Description

`z_mean()` is the mean of the multiplier Z, used to scale claims to the severity means. `z_cv2()` is the squared coefficient of variation of a single claim, which is the true dispersion of a Gamma severity GLM.

Usage

```
z_mean(params = portfolio_params())

z_cv2(params = portfolio_params())
```

Arguments

`params`	Parameters from `portfolio_params()`.

Value

A number.

Examples

```
z_cv2()
```

Index

hs_plans *(*hs_provinces*)*, 2
hs_provinces, 2

panel_params, 3
panel_params(), *5*
portfolio_params, 3
portfolio_params(), *3*, *6–8*

set_levels, 4
simulate_panel, 5
simulate_panel(), *3*, *4*
simulate_portfolio, 6
simulate_portfolio(), *3–5*

true_frequency, 7
true_severity *(*true_frequency*)*, 7

z_cv2 *(*z_mean*)*, 8
z_mean, 8