Package 'dynamicpv'

January 15, 2026

Title Evaluates Present Values and Health Economic Models with Dynamic
Pricing and Uptake

Version 0.4.1

Description The goal of 'dynamicpv' is to provide a simple way to calculate (net) present values and outputs from health economic models (especially cost-effectiveness and budget impact) in discrete time that reflect dynamic pricing and dynamic uptake. Dynamic pricing is also known as life cycle pricing; dynamic uptake is also known as multiple or stacked cohorts, or dynamic disease prevalence. Shafrin (2024) <doi:10.1515/fhep-2024-0014> provides an explanation of dynamic value elements, in the context of Generalized Cost Effectiveness Analysis, and Puls (2024) <doi:10.1016/j.jval.2024.03.006> reviews challenges of incorporating such dynamic value elements. This package aims to reduce those challenges.

License GPL (>= 3)

URL https://MSDLLCpapers.github.io/dynamicpv/,

https://github.com/MSDLLCpapers/dynamicpv

BugReports https://github.com/MSDLLCpapers/dynamicpv/issues

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports dplyr, readr, rlang, tidyr

Suggests flexsurv, ggplot2, heemod, knitr, lubridate, purrr,
rmarkdown, scales, testthat (>= 3.0.0), usethis

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Dominic Muston [aut, cre] (ORCID:
<https://orcid.org/0000-0003-4876-7940>),
John Blischak [ctb] (ORCID: <https://orcid.org/0000-0003-2634-9879>),
Merck & Co., Inc., Rahway, NJ, USA and its affiliates [cph, fnd]

Maintainer Dominic Muston <dominic.muston@msd.com>

Repository CRAN

Date/Publication 2026-01-15 18:00:19 UTC

Contents

+.dynpv	*Method to add two dynpv objects together*

Description

Add together two objects each of class "dynpv"

Usage

```
## S3 method for class 'dynpv'
e1 + e2
```

Arguments

e1	First "dynpv" object
e2	Second "dynpv" object

Value

S3 object of class "dynpv"

-.dynpv   *Method to subtract one dynpv object from another*

Description

Subtract one object of S7 class "dynpv" from another

Usage

```
## S3 method for class 'dynpv'
e1 - e2
```

Arguments

e1   First "dynpv" object

e2   Second "dynpv" object

Details

Present value of `e1-e2` is the present values from `e1` less that from `e2`. Total uptake of `e1-e2` is the uptake from `e1` less that from `e2`. Take care of this when using $mean of the summed object.

Value

S3 object of class "dynpv"

---

addprod   *Method to add two dynpv objects together*

Description

Add together two objects each of S3 class "dynpv": e1 + mult * e2

Usage

```
addprod(e1, e2, mult)
```

Arguments

e1   First "dynpv" object

e2   Second "dynpv" object

mult   Numeric
                Present value is present value from `e1` plus `mult` times the present value from `e2`. Total uptake is the uptake from `e1` plus `mult` times the uptake from `e2`. Take care of this when using $mean of the summed object.

Value

S3 object of class "dynpv"

See Also

dynpv(), futurepv()

---

dynpv   *Present values with dynamic pricing and dynamic uptake*

Description

Calculate present value for a payoff with dynamic (lifecycle) pricing and dynamic uptake (stacked cohorts).

Usage

```
dynpv(
  uptakes = 1,
  payoffs,
  horizon = length(payoffs),
  tzero = 0,
  prices = rep(1, length(payoffs) + tzero),
  discrate = 0
)
```

Arguments

uptakes	Vector of patient uptake over time
payoffs	Vector of payoffs of interest (numeric vector)
horizon	Time horizon for the calculation (length must be less than or equal to the length of payoffs)
tzero	Time at the date of calculation, to be used in lookup in prices vector
prices	Vector of price indices through the time horizon of interest
discrate	Discount rate per timestep, corresponding to price index

Details

Suppose payoffs in relation to patients receiving treatment (such as costs or health outcomes) occur over timesteps $t = 1, ..., T$. Let us partition time as follows.

- Suppose $j = 1, ..., T$ indexes the time at which the patient begins treatment.
- Suppose $k = 1, ..., T$ indexes time since initiating treatment.

In general, $t = j + k - 1$, and we are interested in the set of (j, k) such that $1 \leq t \leq T$.

For example, $t = 3$ comprises:

- patients who are in the third timestep of treatment that began in timestep 1: (j,k)=(1,3);
- patients who are in the second timestep of treatment that began in timestep 2, (j,k)=(2,2); and
- patients who are in the first timestep of treatment that began in timestep 3, (j,k)=(3,1)

The Present Value of a cashflow p_k for the u_j patients who began treatment at time j and who are in their kth timestep of treatment is as follows

$$PV(j,k,l) = u_j \cdot p_k \cdot R_{j+k+l-1} \cdot (1+i)^{2-j-k}$$

where i is the risk-free discount rate per timestep, p_k is the cashflow amount in today's money, and $p_k \cdot R_{j+k+l-1}$ is the nominal amount of the cashflow at the time it is incurred, allowing for an offset of $l = tzero$.

The total present value, $TPV(l)$, is therefore the sum over all j and k within the time horizon T, namely:

$$TPV(l) = \sum_{j=1}^{T} \sum_{k=1}^{T-j+1} PV(j,k,l) = \sum_{j=1}^{T} \sum_{k=1}^{T-j+1} u_j \cdot p_k \cdot R_{l+j+k-1} \cdot (1+i)^{2-j-k}$$

This function calculates $PV(j,k,l)$ for all values of j, k and l, and returns this in a tibble.

Value

A tibble of class "dynpv" with the following columns:

- j: Time at which patients began treatment
- k: Time since patients began treatment
- l: Time offset for the price index (from `tzero`)
- t: Equals $j + k - 1$
- uj: Uptake of patients beginning treatment at time j (from `uptakes`)
- pk: Cashflow amount in today's money in respect of patients at time k since starting treatment (from `payoffs`)
- R: Index of prices over time $l + t$ (from `prices`)
- v: Discounting factors, $(1+i)^{1-t}$, where `i` is the discount rate per timestep
- pv: Present value, $PV(j,k,l)$

Examples

```
# Simple example
pv1 <- dynpv(
  uptakes = (1:10), # 1 patient uptakes in timestep 1, 2 patients in timestep 2, etc
  tzero = c(0,5), # Calculations are performed with prices at times 0 and 5
  payoffs = 90 + 10*(1:10), # Payoff vector of length 10 = (100, 110, ..., 190)
  prices = 1.02^((1:15)-1), # Prices increase at 2\% per timestep in future
  discrate = 0.05 # The nominal discount rate is 5\% per timestep;
                  # the real discount rate per timestep is 3\% (=5\% - 3\%)
)
pv1
```

```
summary(pv1)

# More complex example, using cashflow output from a heemod model

# Obtain dataset
democe <- get_dynfields(
    heemodel = oncpsm,
    payoffs = c("cost_daq_new", "cost_total", "qaly"),
    discount = "disc"
    )

# Obtain short payoff vector of interest
payoffs <- democe |>
    dplyr::filter(int=="new", model_time<11) |>
    dplyr::mutate(cost_oth = cost_total - cost_daq_new)

# Example calculation
pv2 <- dynpv(
    uptakes = rep(1, nrow(payoffs)),
    payoffs = payoffs$cost_oth,
    prices = 1.05^(0:(nrow(payoffs)-1)),
    discrate = 0.08
)
summary(pv2)
```

---

futurepv   *Calculate present value for a payoff in a single cohort with dynamic pricing across multiple timepoints*

Description

Present value of a series of payoffs for a single given cohort, entering at given future time, allowing for dynamic pricing. This function is a wrapper for [dynpv()](#) restricted to evaluation of a single cohort.

Usage

```
futurepv(tzero = 0, payoffs, prices, discrate)
```

Arguments

tzero	Time at the date of calculation, to be used in lookup in prices vector
payoffs	Vector of payoffs of interest (numeric vector)
prices	Vector of price indices through the time horizon of interest
discrate	Discount rate per timestep, corresponding to price index

Value

A tibble of class "dynpv" with the following columns:

- j: Time at which patients began treatment
- k: Time since patients began treatment
- l: Time offset for the price index (from `tzero`)
- t: Equals $j + k - 1$
- uj: Uptake of patients beginning treatment at time j (from `uptakes`)
- pk: Cashflow amount in today's money in respect of patients at time k since starting treatment (from `payoffs`)
- R: Index of prices over time $l + t$ (from `prices`)
- v: Discounting factors, $(1 + i)^{1-t}$, where i is the discount rate per timestep
- pv: Present value, $PV(j, k, l)$

See Also

[dynpv()]

Examples

```
library(dplyr)

# Obtain dataset
democe <- get_dynfields(
   heemodel = oncpsm,
   payoffs = c("cost_daq_new", "cost_total", "qaly"),
   discount = "disc"
   )

# Obtain discount rate
discrate <- get_param_value(oncpsm, "disc")

# Obtain payoff vector of interest
payoffs <- democe |>
   filter(int=="new") |>
   mutate(cost_oth_rup = cost_total_rup - cost_daq_new_rup)
Nt <- nrow(payoffs)

# Run calculation for times 0-9
fpv <- futurepv(
  tzero = (0:9)*52,
  payoffs = payoffs$cost_oth_rup,
  prices = 1.001^(1:(2*Nt)-1), # Approx 5.3% every 52 steps
  discrate = 0.001 + discrate
)
fpv
summary(fpv)
```

---

get_dynfields   *Helper function to get a tibble of the relevant fields from heemod output*

Description

Helper function to get a tibble of the relevant fields from heemod output

Usage

```
get_dynfields(heemodel, payoffs, discount, fname = NA)
```

Arguments

heemodel	A health economic model object from the *heemod* package (see heemod::heemod-package).
payoffs	Field names of payoffs of interest (string vector)
discount	Name of parameter providing discount rate per cycle (string)
fname	Export data to a .CSV file of this name, if given (character)

Value

Tibble of payoffs taken from the heemod model, by intervention and model timestep (model_time).

The field vt is calculated as (1+i)^(1-model_time), where i is the discount rate per model timestep set in the *heemod* model through the parameter disc_cycle. This can be useful in 'rolling-up' payoff values to the timestep in which they were incurred.

An additional set of payoffs (identified with a "_rup" suffix) provides calculations of the payoffs as at the start of the timestep in which they were incurred, i.e. original payoff / vt.

See Also

heemod::heemod-package

Examples

```
democe <- get_dynfields(
   heemodel = oncpsm,
   payoffs = c("cost_daq_new", "cost_total", "qaly"),
   discount = "disc"
   )
head(democe)
```

---

get_param_value   *Obtain parameter value(s) from a heemod output*

Description

Obtain parameter value(s) from a heemod output

Usage

```
get_param_value(heemodel, param)
```

Arguments

heemodel	A health economic model object from the *heemod* package (see heemod::heemod-package).
param	Name of parameter to extract from the heemod model

Value

Value of the parameter from the heemod model

See Also

heemod::heemod-package

Examples

```
get_param_value(
   heemodel = oncpsm,
   param = "disc"
   )
```

---

mean.dynpv   *Mean present value per uptaking patient*

Description

Mean of the Present Value per uptaking patient, by time at which the calculation is performed (tzero input to dynpv()).

Usage

```
## S3 method for class 'dynpv'
mean(x, ...)
```

Arguments

x   Tibble of class "dynpv" created by dynpv() or futurepv()

...   Currently unused

Details

This is equal to total() divided by uptake().

Value

A number or tibble

See Also

dynpv(), futurepv()

---

ncoh   *Number of cohorts of uptaking patients*

Description

Number of cohorts of uptaking patients, calculated as the length of the uptakes input to dynpv()

Usage

```
ncoh(df)
```

Arguments

df   Tibble of class "dynpv" created by dynpv() or futurepv()

Value

A number

See Also

dynpv(), futurepv()

---

ntimes *Number of times at which present value calculations are performed*

Description

Number of times at which present value calculations are performed, calculated as the length of the tzero input to dynpv()

Usage

```
ntimes(df)
```

Arguments

df   Tibble of class "dynpv" created by dynpv() or futurepv()

Value

A number

See Also

dynpv(), futurepv()

---

oncpsm *Heemod cost-effectiveness model example*

Description

An example three state cost-effectiveness model in oncology built using heemod::heemod-package()
according to the assumptions and specification in the accompanying paper.

Usage

```
oncpsm
```

Format

oncpsm:
A heemod object

Source

Created based on assumptions.

---

summary.dynpv   *Summarize a dynpv object*

Description

Summarize a dynpv object

Usage

```
## S3 method for class 'dynpv'
summary(object, ...)
```

Arguments

object   Tibble of class "dynpv" created by dynpv() or futurepv()

...   Currently unused

Value

A list of class "dynpv_summary" with the following elements:

- ncoh: Number of cohorts of uptaking patients, from ncoh()
- ntimes: Number of times at which present value calculations are performed, from ntimes()
- uptake: Total number of uptaking patients, from uptake()
- sum_by_coh: Present value for each uptake cohort and calculation time, from sum_by_coh()
- total: Total present value, from total()
- mean: Mean present value per uptaking patient, from mean(), equal to total/uptake.

See Also

dynpv(), futurepv()

---

sum_by_coh   *Present value for each uptake cohort and calculation time*

Description

Calculates the sum of the Present Value by uptake cohort (j) and time at which the calculation is performed (tzero input to dynpv())

Usage

```
sum_by_coh(df)
```

Arguments

df   Tibble of class "dynpv" created by `dynpv()` or `futurepv()`

Details

The Present Value of a cashflow p_k for the u_j patients who began treatment at time j and who are in their kth timestep of treatment is as follows

$$PV(j, k, l) = u_j \cdot p_k \cdot R_{j+k+l-1} \cdot (1+i)^{2-j-k}$$

where i is the risk-free discount rate per timestep, p_k is the cashflow amount in today's money, and $p_k \cdot R_{j+k+l-1}$ is the nominal amount of the cashflow at the time it is incurred, allowing for an offset of $l = tzero$.

This method returns $\sum_{k=1}^{T-j+1} PV(j, k, l)$ for each value of j and l, where T is the time horizon of the calculation.

Value

A number or tibble

See Also

`dynpv()`, `futurepv()`

---

total   *Total present value*

Description

Sum of the Present Value, by time at which the calculation is performed (`tzero` input to `dynpv()`)

Usage

```
total(df)
```

Arguments

df   Tibble of class "dynpv" created by `dynpv()` or `futurepv()`

Details

The Present Value of a cashflow p_k for the u_j patients who began treatment at time j and who are in their kth timestep of treatment is as follows

$$PV(j, k, l) = u_j \cdot p_k \cdot R_{j+k+l-1} \cdot (1+i)^{2-j-k}$$

where i is the risk-free discount rate per timestep, p_k is the cashflow amount in today's money, and $p_k \cdot R_{j+k+l-1}$ is the nominal amount of the cashflow at the time it is incurred, allowing for an offset of $l = tzero$.

The total present value by time at which the calculation is performed, $TPV(l)$, is therefore the sum of $PV(j, k, l)$ over all j and k within the time horizon T, namely:

$$TPV(l) = \sum_{j=1}^{T} \sum_{k=1}^{T-j+1} PV(j, k, l) = \sum_{j=1}^{T} \sum_{k=1}^{T-j+1} u_j \cdot p_k \cdot R_{l+j+k-1} \cdot (1+i)^{2-j-k}$$

Value

A number or tibble

See Also

dynpv(), futurepv()

---

trim_vec   *Trim the tailing zeroes from a long vector*

Description

Trim the tailing zeroes from a long vector

Usage

```
trim_vec(vec)
```

Arguments

vec   Vector. Final elements may be zero.

Value

A vector whose length is shorter than the original, if there were trailing zero elements

Examples

```
trim_vec(c(1:10, rep(0,3)))
```

---

uptake	*Total number of uptaking patients*

Description

Total number of uptaking patients, calculated as the sum of the uptake input to [dynpv()](), or $\sum_{j=1}^{T} u_j$

Usage

```
uptake(df)
```

Arguments

df	Tibble of class "dynpv" created by [dynpv()]() or [futurepv()]()

Value

A number or tibble

See Also

[dynpv()](), [futurepv()]()

Index