Package 'QALY'

October 17, 2017

Type Package

Title Calculate QALY Gains with Discounting and Inflated Costs

Version 0.1.0

Author Nathan Green

Maintainer Nathan Green <nathan.green@imperial.ac.uk>

Description S3 package for QALY calculation and plotting.

License MIT

Encoding UTF-8

LazyData true

Imports readr,
datapkg

Suggests testthat

RoxygenNote 6.0.1

R topics documented:

`adjusted_life_years` *Adjusted Life Years Object Constructor*

Description

For use as input to the QALY and DALY functions.

Usage

```
adjusted_life_years(start_year = 0, end_year = NA, age = NA,
  time_horizon = NA, utility, discount_rate = 0.035)
```

Arguments

`start_year`	Calendar year to begin calculation
`end_year`	Calendar year to end calculation
`age`	Age at start of period
`time_horizon`	Number of time periods from start to end date
`utility`	Proportion health detriment
`discount_rate`	Fixed proportion reduction over time

Value

Object of class adjusted_life_years

See Also

QALY, DALY

Examples

```
AdjLifeYears <- adjusted_life_years(
                    start_year = 2016,
                    end_year = 2020,
                    age = 33,
                    time_horizon = NA,
                    utility = 0.9,
                    discount_rate = 0.035)

total_QALYs(AdjLifeYears)
## 2.913622

total_QALY(1)
## "Error: Not an adjusted_life_years class input object."
```

`calc_QALY` *Calculate Quality-Adjusted Life Years*

Description

Discounted total QALYs upto a defined time horizon. This is a simpler function. An alternative Method is also available (see `total_QALYs`).

Usage

```
calc_QALY(utility = NA, age = NA, time_horizon = NA, halfend = FALSE,
  start_delay = 0)
```

Arguments

`utility`	Vector of values between 0 and 1 (1 - utility loss)
`age`	Year of age
`time_horizon`	Non-negative value how many time step into future as sum limit
`halfend`	Should the last year be a half year
`start_delay`	What time delay to origin, to shift discounting

Details

Uses the following formula, for year `i`:

$$\sum prop_year(i) * utility(i) * QoL(age(i)) * discount_factor(i)$$

for `i` = 1, ..., `time_horizon`.

`prop_year` is useful for fractions of years at the start and end of the period. However, since we may not know this then may not be necessary.

References

Sassi, Franco, Health Policy and Planning, 5, 402-408, Calculating QALYs, comparing QALY and DALY calculations, volume 21, 2006

See Also

`calc_QALY_CFR`, `calc_QALY_population` `total_QALYs`

Examples

```
calc_QALY(utility = 0.9, age = 13, time_horizon = 49)
```

calc_QALY_CFR *Calculate QALYs using Case-Fatality Rates*

Description

Sum to time of death or a prespecified time horizon. CFRs are dependent on age. Utility is fixed over time.

Usage

```
calc_QALY_CFR(AGES = NA, cfr_age_lookup = NULL, time_horizon = NA,
  utility = 0.9)
```

Arguments

AGES	Vector of ages at start
cfr_age_lookup	Data frame of case-fatality ratios for ages
time_horizon	Vector of end dates for each individual
utility	Proportion health detriment

See Also

calc_QALY

Examples

```
# 12 month case fatality rate
# Crofts et al (2008)
cfr_age_breaks <- c(15, 45, 65, 200)
cfr_age_levels <- levels(cut(0, cfr_age_breaks, right = FALSE))

cfr_age_lookup <- data.frame(age = cfr_age_levels,
                             cfr = c(0.0018, 0.0476, 0.1755),
                             a = c(1, 125, 413), #beta distn
                             b = c(564, 2500, 1940))

rownames(cfr_age_lookup) <- cfr_age_lookup$age

# status-quo
QALY.statusquo <- calc_QALY_CFR(AGES = IMPUTED_sample$cfr_age_groups[IMPUTED_sample$uk_tb==1],
                                cfr_age_lookup)

# screened imputed sample
QALY.screened <- calc_QALY_CFR(AGES = IMPUTED_sample$cfr_age_groups[IMPUTED_sample$uk_tb1==1],
                               cfr_age_lookup)

# who changed LTBI status
uk_TB.completedTx <- (IMPUTED_sample$uk_tb1==0 & IMPUTED_sample$uk_tb==1)

# fixed over time
QALY_uk_TB.completedTx <- years(death.isdtt[uk_TB.completedTx] - uk_tb.isdtt[uk_TB.completedTx])
```

```
QALY.screened <- c(QALY.screened, QALY_uk_TB.completedTx)

# (non-fixed) discounted all-cause (non-active TB)
notification_to_allcause_death <- (IMPUTED_sample$date_death1_issdt - IMPUTED_sample$fup_issdt)/365

calc_QALY_CFR(time_horizon = notification_to_allcause_death[uk_TB.completedTx],
              utility = 1.0)
```

`calc_QALY_population` *Calculate QALYs For Population*

Description

This is a wrapper function for `calc_QALY` over multiple time horizons (e.g. individuals).

Usage

```
calc_QALY_population(utility, age, time_horizons, start_delay = 0, ...)
```

Arguments

`utility`	Vector of utilities for each year in to the future, between 0 and 1
`age`	Vector of ages at start
`time_horizons`	Vector of non-negative durations
`start_delay`	What time delay to origin, to shift discounting
`...`	Additional arguments

Details

Assume that the utilities are the same for all individuals.

Value

QALY vector

See Also

`calc_QALY_CFR`, `calc_QALY`

CFR_time_horizon.adjusted_life_years *Case-Fatality Rate Determined Time Horizon*

Description

Using this function then don't require a separate CFR function to calculate the total QALYs. DRY priniciple.

Usage

```
CFR_time_horizon.adjusted_life_years(adjusted_life_years, cfr_modelframe)
```

Arguments

adjusted_life_years	An object of class adjusted_life_years
cfr_modelframe	Data frame with CFR and ages

Value

An object of class adjusted_life_years

Examples

```
AdjLifeYears <- adjusted_life_years(
                    start_year = 2016,
                    end_year = 2020,
                    age = NA,
                    time_horizon = NA,
                    utility = 0.9,
                    discount_rate = 0.035)

cfr_modelframe <- model.frame(cfr ~ age, data = cfr_age_lookup)

CFR_time_horizon.adjusted_life_years(AdjLifeYears, cfr_modelframe)
```

discount *Discounted Values Over Time*

Description

Dscounted value, e.g. cost or health (QALY), for each time point, e.g. year. E.g. a QALY in the future is worth less to us now because of 'interest' or conversely we'd need more QALYs now to have a QALY further in the future.

Usage

```
discount(discount_rate = 0.035, t_limit = 100)
```

Arguments

`discount_rate`	Discount factor, default at 3.5%
`t_limit`	Time period (positive integer) to discount over starting from 1

Details

Formula used is

$$1/(1 + discount_rate)^{years}$$

Value

Discounted value for each time point up to `t_limit`

References

Severens, Johan L and Milne, Richard J, Value in Health, 4, Discounting Health Outcomes in Economic Evaluation : The Ongoing Debate, volume 7, 2004

Examples

```
D <- discount(t_limit = 10)
utility <- 0.97

# Discounted QALYs upto 10 years
sum(utility * D)
```

`fillin_missing_utilities` *Fill-in Missing Trailing Utilities*

Description

Repeat last value up to final period.

Usage

```
fillin_missing_utilities(utility, time_horizon)
```

Arguments

`utility`	Vector of health quality of life values between 0 and 1
`time_horizon`	Non-negative integer

Value

Vector of utilities

See Also

`calc_QALY_population`

inflation_adjust_cost *Calculate Inflation Adjusted Costs*

Description

Up to the present time inflated upwards. Option to use the ONS GDP_Deflators_Qtrly_National_Accounts or a fixed 3.5%. Can't download directly into function because the .csv on the website is too messy as-is. This would be good to do though so that can always use latest version. TODO: webscraping? regular expressions?

Usage

```
inflation_adjust_cost(from_year, to_year, from_cost, reference = NA,
  fixed = TRUE)
```

Arguments

from_year	Date of cost to convert from
to_year	Date to convert cost to
from_cost	Cost at from_year
reference	Source of data (string)
fixed	Fixed 3.5% rate of inflation?

Examples

```
from_year <- 2012
to_year <- 2015
from_cost <- 96.140

inflation_adjust_cost(from_year, to_year, from_cost, fixed = FALSE)
#100

inflation_adjust_cost(from_year = 2010, to_year = 2016, from_cost = 1)
#1.229255
1*(1+0.035)^6
```

make_discount *Make an Encapsulated Discount Function*

Description

This format doesn't need to keep track of time.

Usage

```
make_discount(discount_rate = 0.035)
```

Arguments

discount_rate	Discount factor, default at 3.5%

Value

Function with global scoped index

---

plot_QALY *QALY Plot as a Non-Increasing Step Function Over Time*

Description

QALY Plot as a Non-Increasing Step Function Over Time

Usage

```
plot_QALY(QALYs, overlay = FALSE, XLIM = c(0, 80), COL = "light grey",
  age_annotate = FALSE)
```

Arguments

QALYs	QALY class object
overlay	Overlay lines on current plot?
age_annotate	

See Also

total_QALYs.adjusted_life_years

Examples

```
QALYs <- total_QALYs(AdjLifeYears)
plot_QALY(QALYs)

## list of objects
plot_QALY(QALY_diseasefree[[1]], overlay = F)
map(QALY_diseasefree, plot_QALY, overlay = T, COL = rgb(0, 0, 0, 0.1))
map(QALYloss_diseasefree, plot_QALY, overlay = T, COL = "red")

# cumulative total QALY loss
sapply(QALY_diseasefree,
       FUN = function(x) attr(x, "yearly_QALYs")) %>%
       rowSums(na.rm = TRUE) %>%
       cumsum %>%
       plot(type = 'l')
```

`total_QALYs` *Calculate Life-Time QALYs*

Description

Calculate Life-Time QALYs

Usage

```
total_QALYs(adjusted_life_years)
```

Arguments

`adjusted_life_years`
: Object of class `adjusted_life_years`

Value

QALYs object

`total_QALYs.adjusted_life_years` *Calculate Life-Time QALYs*

Description

Calculate Life-Time QALYs

Usage

```
## S3 method for class 'adjusted_life_years'
total_QALYs(adj_years)
```

Arguments

`adjusted_life_years`
: Object of class `adjusted_life_years`

Value

QALYs object

Examples

```
AdjLifeYears <- adjusted_life_years(
                    start_year = 2016,
                    end_year = 2020,
                    age = NA,
                    time_horizon = NA,
                    utility = 0.9,
                    discount_rate = 0.035)

total_QALYs(AdjLifeYears)
## 2.913622

total_QALYs(1)
## "Error: Not an adjusted_life_years class input object."
```

---

total_QALYs.default *Calculate Life-Time QALYs*

Description

Calculate Life-Time QALYs

Usage

```
## Default S3 method:
total_QALYs(adjusted_life_years)
```

Arguments

adjusted_life_years	Object of class adjusted_life_years

Value

QALYs object