Package 'tatooheene'

July 22, 2025

Title Technology Appraisal Toolbox for Health Economic Evaluations in the Netherlands

Version 0.19.0

Description Functions to support economic modelling in R based on the methods of the Dutch guideline for economic evaluations in healthcare <https://www.zorginstituutnederland.nl/publicaties/publicatie/2024/01/16/richtlijn-voor-het-uitvoeren-van-economische-evaluaties-in-de-gezondheidszorg>, CBS data <https://www.cbs.nl/>, and OECD data <https://www.oecd.org/en.html>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5)

LazyData true

Imports dplyr (>= 1.1.4), assertthat (>= 0.2.1), tidyr (>= 1.3.0), stats

Suggests cbsodataR, lubridate, here, testthat

NeedsCompilation no

Author Frederick Thielen [aut] (ORCID: <https://orcid.org/0000-0002-0312-5891>), Stijn Peeters [aut, cre] (ORCID: <https://orcid.org/0009-0004-3684-3584>)

Maintainer Stijn Peeters <s.b.peeters@eshpm.eur.nl>

Repository CRAN

Date/Publication 2024-12-18 15:50:06 UTC

Contents

`depreciation_interest` *A function to calculate the costs of medical equipment*

Description

[Experimental] A function to calculate the costs of medical equipment based on Section 3.3 of the Dutch EE guideline; k = annual depreciation and interest expense jaarlijkse afschrijvings- en rentekosten

Usage

```
depreciation_interest(
  v_replace_val,
  r_salvage_val,
  n_amortisation_period = 10,
  i_interest_rt = 0.025,
  output = c("data frame", "annuity factor", "annual cost")
)
```

Arguments

`v_replace_val`	V: vervangingswaarde; replacement value
`r_salvage_val`	R: restwaarde; salvage value
`n_amortisation_period`	N: afschrijvingstermijn,; amortization period
`i_interest_rt`	i: renteperceof ntage; interest rate
`output`	Default of output is a data frame with both the annuity factor and yearly depreciation and interest costs, but the values can be selected independently

Value

A data frame with the annuity factor, yearly depreciation and interest costs, or the values independently.

Examples

```
# Example usage of the depreciation_interest function
# Calculate both annuity factor and yearly depreciation and interest costs as a data frame
depreciation_interest(v_replace_val = 50000, r_salvage_val = 5000)

# Get only the annuity factor
depreciation_interest(v_replace_val = 50000, r_salvage_val = 5000, output = "annuity factor")

# Get only the annual depreciation and interest cost
depreciation_interest(v_replace_val = 50000, r_salvage_val = 5000, output = "annual cost")
```

---

df_cpi_combined *Consumer Price Index (CPI) data from CBS*

Description

A subset data frame of Consumentenprijzen; prijsindex 2015=100 from CBS. Identifier: 83131NED

Usage

```
df_cpi_combined
```

Format

`df_cpi_combined`**:**
A data frame with 11 rows and 8 columns:

Year from Year from in case of two consecutive years

Year to Year ending in case of two consecutive years

Percentage Percentage change in case of two consecutive years

Factor Factor for multiplication in case of two consecutive years

Year from' Year starting from, in the case of a range, the year up to the maximum year

Year to' Year ending in case of a range, the year up to the maximum year

Percentage' Percentage for multiplication in case of a range, the year up to the maximum year

Factor' Factor for multiplication in case of a range, the year up to the maximum year

Source

https://www.cbs.nl/nl-nl/cijfers

df_fp *Job vacancy data from CBS*

Description

A subset data frame of job vacancies; SBI 2008; by economic activity and company size from CBS. Identifier: 80472NED

Usage

```
df_fp
```

Format

`df_fp`**:**

A data frame with 27 rows and 7 columns:

Year Year

Filled vacancies Vervulde vacatures

Open vacancies Openstaande vacatures

Friction period in days Calculation of the friction period in days

Friction period in weeks Calculation of the friction period in weeks

Friction period days average over 5 years Calculation of the 5 year average friction period in days

Friction period weeks average over 5 years Calculation of the 5 year average friction period in weeks

Source

https://www.cbs.nl/nl-nl/cijfers

df_ppp *Purchasing Power Parity (PPP) data from OECD*

Description

A subset data frame of the OECD data set containing the Purchasing Power Parity (PPP) data.

Usage

```
df_ppp
```

Format

`df_ppp`:

A data frame with 64 rows and 2 columns:

Year Year of PPP

PPP Purchase Power Parity

Source

https://sdmx.oecd.org/public

---

df_rp_medical — *Medical unit cost data from the Costing manual: Methods and Reference Prices for Economic Evaluations in Healthcare*

Description

A subset data frame of medical unit cost data; chapter 4 of the Dutch Costing Manual

Usage

```
df_rp_medical
```

Format

`df_rp_medical`:

A data frame with 116 rows and 4 columns:

Category Category of the medical unit cost

Unit Medical unit cost

2022 Year 2022 medical unit cost

2023 Year 2023 medical unit cost

Source

<Hakkaart - van Roijen, L., Peeters, S., Kanters, T., van Baal, P., Brouwer, W., Drost, R., Evers, S. M. A. A., van Exel, J., Reckers-Droog, V., Tannaoui, N.-E., Thielen, F., & Wijnen, B. (2024). Kostenhandleiding voor economische evaluaties in de gezondheidszorg: Methodologie en Referentieprijzen. (Herziene versie 2024 ed.).>

df_rp_patient *Patient & family unit cost data from the Costing manual: Methods and Reference Prices for Economic Evaluations in Healthcare*

Description

A subset data frame of patient & family unit cost data; chapter 5 of the Dutch Costing Manual

Usage

```
df_rp_patient
```

Format

df_rp_patient**:**

A data frame with 6 rows and 4 columns:

Category Category of the patient & family unit cost

Unit Patient & family unit cost

2022 Year 2022 patient & family unit cost

2023 Year 2023 patient & family unit cost

Source

<Hakkaart - van Roijen, L., Peeters, S., Kanters, T., van Baal, P., Brouwer, W., Drost, R., Evers, S. M. A. A., van Exel, J., Reckers-Droog, V., Tannaoui, N.-E., Thielen, F., & Wijnen, B. (2024). Kostenhandleiding voor economische evaluaties in de gezondheidszorg: Methodologie en Referentieprijzen. (Herziene versie 2024 ed.).>

df_rp_prod *Productivity & other unit cost data from the Costing manual: Methods and Reference Prices for Economic Evaluations in Healthcare*

Description

A subset data frame of productivity & other unit cost data; chapter 6 of the Dutch Costing Manual

Usage

```
df_rp_prod
```

Format

`df_rp_prod`**:**

A data frame with 38 rows and 4 columns:

Category Category of the productivity & other unit cost

Unit Productivity & other unit cost

2022 Year 2022 productivity & other unit cost

2023 Year 2023 productivity & other unit cost

Source

<Hakkaart - van Roijen, L., Peeters, S., Kanters, T., van Baal, P., Brouwer, W., Drost, R., Evers, S. M. A. A., van Exel, J., Reckers-Droog, V., Tannaoui, N.-E., Thielen, F., & Wijnen, B. (2024). Kostenhandleiding voor economische evaluaties in de gezondheidszorg: Methodologie en Referentieprijzen. (Herziene versie 2024 ed.).>

`discount_value` *A function to calculate the discounted value of future costs or effects*

Description

[Experimental] A function to calculate the discounted value of a future costs or effects based on the in paragraph 2.6.1.2 of the Dutch EE guideline mentioned discount rate and time period

Usage

```
discount_value(
  current_value,
  discount_rate = 0.03,
  time,
  time_unit = c("years", "months", "weeks", "days"),
  discount_year_one = FALSE
)
```

Arguments

`current_value`	The current value.
`discount_rate`	The discount rate to use for the calculation. Default is 0.03. The guideline stipulates 0.03 for costs and 0.015 for effects.
`time`	The time at which the future value occurs.
`time_unit`	The unit of time to use for the calculation. Default is "years", but "months", "weeks", and "days" are also valid options.
`discount_year_one`	Logical value indicating whether to discount the first year as well. Default is FALSE.

Value

A numeric value of the discounted future value.

Examples

```
# Example usage of the discount_value function
# Calculate the discounted value of 100 after 5 years, the first year is not discounted
discount_value(current_value = 100, discount_rate = 0.03, time = 5, time_unit = "years")

# Calculate the discounted value of 100 after 60 months, the first year is not discounted
discount_value(current_value = 100, discount_rate = 0.03, time = 60, time_unit = "months")

# Calculate the discounted value of 100 after 365 days, the first year is discounted
discount_value(current_value = 100, time = 365, time_unit = "days", discount_year_one = TRUE)
```

---

discount_vector *A function to create a discount vector for a given time period*

Description

[Experimental] A function to calculate a discount vector for a given time period based on the in paragraph 2.6.1.2 of the Dutch EE guideline mentioned discount rate and time period

Usage

```
discount_vector(
  discount_rate = 0.03,
  start_time = 0,
  end_time,
  time_unit = c("years", "months", "weeks", "days"),
  discount_year_one = FALSE
)
```

Arguments

discount_rate	The discount rate to use for the calculation. Default is 0.03. The guideline stipulates 0.03 for costs and 0.015 for effects.
start_time	The start time for the discount vector. Default is 0.
end_time	The end time for the discount vector.
time_unit	The unit of time to use for the calculation. Default is "years", but "months", "weeks", and "days" are also valid options.
discount_year_one	Logical value indicating whether to discount the first year as well. Default is FALSE.

Value

A numeric vector of discounted values for each time period.

Examples

```
# Example usage of the discount_vector function
# Calculate the discount vector for 5 years, a discount rate of 0.015, first year is not discounted
discount_vector(discount_rate = 0.015, end_time = 5, time_unit = "years")

# Calculate the discount vector for 60 months, a start time of 2, the first year is not discounted
discount_vector(discount_rate = 0.03, start_time = 2, end_time = 60, time_unit = "months")

# Calculate the discount vector for 365 days, a discount rate of 0.06, the first year is discounted
discount_vector(discount_rate = 0.06, end_time = 365, time_unit = "days", discount_year_one = TRUE)
```

---

friction_period *A function to download the friction period over one or multiple years*

Description

[Experimental]

Usage

```
friction_period(year = "all", period = "all", type = "5_year_avg")
```

Arguments

year	The year of which the friction period should be downloaded, multiple years are possible. The default is the whole data frame
period	The friction period that should be included (days/weeks), default is including the whole dataframe
type	If period is chosen, a decision can be made between the 5 year average and 1 year friction period in days/weeks. The default is the 5 year average

Value

A data frame with friction periods for all years or a selection of years and variables.

Examples

```
# Example usage of the depreciation_interest function
friction_period(year = 2019, period = "weeks", type = "5_year_avg")
```

nl_med_prices *A function to download the Medical Reference prices of the Dutch Costing Manual for one or multiple years*

Description

[Experimental] A function downloads the Medical Reference prices of the Dutch Costing Manual for one or multiple years. The prices are available in Euro (EUR) or International Dollar (INT$).

Usage

```
nl_med_prices(
  year = "all",
  category = "all",
  unit = "all",
  currency = c("EUR", "INT$")
)
```

Arguments

year	The year of which the reference price should be downloaded, multiple years are possible, default is the whole dataset
category	The category of prices that should be included (one or more categories), default is including all categories
unit	The reference price that should be included (one or multiple reference prices), default is including the whole dataframe
currency	The currency of the output of the prices. A decision can be made between EUR and INT$, the default is EUR.

Value

A dataframe or value with the Medical Reference price(s) of the Dutch Costing Manual for the specified years

Examples

```
# Example usage of the nl_med_prices function
# Calculate for year 2023 with the category Nursing
nl_med_prices(year = "2023", category = "Nursing")

# Calculate for year 2022 and 2023 the category Nursing
nl_med_prices(year = "all", category = "Nursing")

# Calculate for year 2022 with the category Nursing in INT$
nl_med_prices(year = "2022", category = "Nursing" , currency = "INT$")
```

`nl_pat_fam_prices` *A function to download the Patient & Family Reference prices of the Dutch Costing Manual for one or multiple years*

Description

[Experimental] This function downloads the Patient & Family Reference prices of the Dutch Costing Manual for one or multiple years. The prices are available in Euro (EUR) or International Dollar (INT$).

Usage

```
nl_pat_fam_prices(
  year = "all",
  category = "all",
  unit = "all",
  currency = c("EUR", "INT$")
)
```

Arguments

`year`	The year of which the reference price should be downloaded, multiple years are possible, default is the whole dataset (year = "all")
`category`	The category of prices that should be included (one or more categories), default is including all categories
`unit`	The reference price that should be included (one or multiple reference prices), default is including the whole data frame
`currency`	The currency of the output of the prices. A decision can be made between EUR and INT$, the default is EUR.

Value

A dataframe or value with the Patient & Family Reference price(s) of the Dutch Costing Manual for the specified years

Examples

```
# Example usage of the nl_pat_fam_prices function
# Calculate for 2023 with the category Transportation and the unit Car, cost per kilometer in EURO
nl_pat_fam_prices(year = "2022", category = "Transportation", unit = "Car, cost per kilometer")

# Calculate for year 2022 and 2023 the unit Car, cost per kilometer in EURO
nl_pat_fam_prices(year = "all", unit = "Car, cost per kilometer")

# Calculate for the year 2022 with the category Transportation in INT$
nl_pat_fam_prices(year = "2022", category = "Transportation", currency = "INT$")
```

nl_ppp — *A function to obtain the Dutch PPP factor values in International Dollar (INT$)*

Description

[Experimental] This function downloads the Purchasing Power Parity (PPP) factor values for the Netherlands from the OECD website per year in International Dollar (Int$).

Usage

```
nl_ppp(year = "all")
```

Arguments

year	The year of which the PPP factor should be downloaded, multiple years are possible, default is the whole dataset.

Value

A dataframe or value with the PPP factor values for the specified years.

Examples

```
# Example usage of the nl_ppp function
nl_ppp(year = 2019)
```

nl_price_index — *A function to calculate the Consumer Price Index (CPI) for a given year range.*

Description

[Experimental] This function provides the Consumer Price Index (CPI) for a given year range both in a factor or dataframe based on CBS data and further described in 2.6.1.1 of the Dutche EE guideline

Usage

```
nl_price_index(
  start_year = 2013,
  end_year = 2023,
  output = c("table", "factor")
)
```

Arguments

start_year	start year for CPI output table or factor
end_year	End year for CPI output table or factor
output	Which output we would like to see. "factor": is the factor from start to end year, "table" is the table of all CPIs from start to end year

Value

Dataframe or factor with CPI data from start year to end year

Examples

```
# Example usage of the nl_price_index function
# Get the CPI factor from 2013 to 2023
nl_price_index(start_year = 2013, end_year = 2023, output = "factor")

# Get the CPI table from 2013 to 2023
nl_price_index(start_year = 2013, end_year = 2023, output = "table")
```

---

nl_prod_oth_prices *A function to download the Productivity and other societal Reference prices of the Dutch Costing Manual for one or multiple years*

Description

[Experimental] This function downloads the Productivity and other societal Reference prices of the Dutch Costing Manual for one or multiple years. The prices are available in Euro (EUR) or International Dollar (INT$).

Usage

```
nl_prod_oth_prices(
  year = "all",
  category = "all",
  unit = "all",
  currency = c("EUR", "INT$")
)
```

Arguments

year	The year of which the reference price should be downloaded, multiple years are possible, default is the whole dataset
category	The category of prices that should be included (one or more categories), default is including all categories
unit	The reference price that should be included (one or multiple reference prices), default is including the whole dataframe

currency	The currency in which the reference price should be included (EUR or INT$), default is EUR

Value

A dataframe or value with the Productivity and/other societal Reference price(s) of the Dutch Costing Manual

Examples

```
# Example usage of the nl_prod_oth_prices function:
# Calculate for 2023 with the category Productivity loss - Paid work in EUR
nl_prod_oth_prices(year = "2023", category = "Productivity loss - Paid work")

# Calculate for 2022 and 2023 with the category Productivity loss - Paid work in EUR
nl_prod_oth_prices(year = "all", category = "Productivity loss - Paid work")

# Calculate for 2022 with the category Productivity loss - Paid work in INT$
nl_prod_oth_prices(year = "2022", category = "Productivity loss - Paid work", currency = "INT$")
```

---

pretty_price *A function to write pretty prices in* bookdown *reports*

Description

[Experimental] This function writes pretty prices in bookdown reports. The function uses the `formatC()` function to format the number and adds the currency to the end of the number.

Usage

```
pretty_price(x, digi = 2, currency = "EUR", ...)
```

Arguments

x	A number to be printed
digi	Number of digits
currency	The name of the currency
...	Extra arguments for `formatC()`

Value

A pretty price with the currency

Examples

```
# Example usage of the pretty_price function
pretty_price(1000, currency = "EUR")
```

Index