

Package: p0bservations (via r-universe)

October 1, 2026

Type Package
Title Assorted Functions and Observations by P0bs
Version 0.4.0
Description Provides assorted functions by p0bs.
URL <https://github.com/p0bs/p0bservations>
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Depends R (>= 4.1)
Imports dplyr, rlang, rvest, stats, tidyr
Suggests covr, testthat (>= 3.0.0)
Config/testthat/edition 3
Config/roxygen2/version 8.0.0
Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev
Repository <https://p0bs.r-universe.dev>
Date/Publication 2026-05-22 12:00:10 UTC
RemoteUrl <https://github.com/p0bs/p0bservations>
RemoteRef HEAD
RemoteSha 3b6166fffeb6bc620cffd07f544afb2570d4fb0b

Contents

liability_tax	2
probability_ruin	2
probability_ruin_rate	4
rate_annuity	5
tax_parameters	6
Index	7

liability_tax

Calculate UK income and/or National Insurance taxes

Description

This function applies the prevailing tax bands and rates to give a simple calculation for UK Income Tax and National Insurance (to the nearest couple of pounds). Please note that these are approximations, so do not rely on them for financial returns or planning. The output is a named list containing: 'income_net' for the net income after Income Tax and National Insurance; 'income_tax' for the Income Tax liability; 'ni' for the National Insurance liability; and 'total_tax' for the combined liability. 'income_taxable' may be a numeric vector; all output list elements will be vectors of the same length.

Usage

```
liability_tax(income_taxable, tax_year_end)
```

Arguments

income_taxable The taxable income level (i.e. after deductions for things like pension contributions). May be a numeric vector.

tax_year_end The calendar year in which the tax year ends, as a YYYY integer. For example, tax year 2023/24 would be 2024.

Examples

```
liability_tax(
  income_taxable = 38000,
  tax_year_end = 2024
)$total_tax

# Vectorised over multiple incomes
liability_tax(
  income_taxable = c(22000, 40000, 65000),
  tax_year_end = 2024
)$total_tax
```

probability_ruin

Calculate the probability of retirement ruin

Description

This function uses the Milevsky-Robinson method to analyse the probability of retirement ruin, by parsimoniously meshing investment risk and return, mortality estimates and spending rates without resorting to opaque Monte Carlo simulations. For further details, see: Milevsky, M. and C. Robinson; "A Sustainable Spending Rate without Simulation"; Financial Analysts Journal, Volume 61, Number 6. (2005). Please note that these are approximations, so do not rely on them for financial returns or planning. All arguments are vectorised and are recycled in the standard R fashion.

Usage

```
probability_ruin(  
  return_expected,  
  return_sd,  
  life_remaining_expected,  
  rate_spend  
)
```

Arguments

return_expected	The expected real return of the entire pension portfolio.
return_sd	The projected standard deviation of the returns of the entire pension portfolio.
life_remaining_expected	The median projected remaining lifespan of the individual in question.
rate_spend	The annual spending rate applied by the individual to their pension portfolio.

Examples

```
probability_ruin(  
  return_expected = 0.07,  
  return_sd = 0.2,  
  life_remaining_expected = 28.1,  
  rate_spend = 0.05  
)  
  
# Vectorised: sweep over spending rates  
probability_ruin(  
  return_expected = 0.07,  
  return_sd = 0.2,  
  life_remaining_expected = 28.1,  
  rate_spend = c(0.03, 0.04, 0.05)  
)
```

probability_ruin_rate *Calculate the spend rate for a given probability of retirement ruin*

Description

This function uses the Milevsky-Robinson method to find the spend rate corresponding to a given probability of retirement ruin, by parsimoniously meshing investment risk and return, mortality estimates and spending rates without resorting to opaque Monte Carlo simulations. For further details, see: Milevsky, M. and C. Robinson; "A Sustainable Spending Rate without Simulation"; Financial Analysts Journal, Volume 61, Number 6. (2005). Please note that these are approximations, so do not rely on them for financial returns or planning. All arguments are vectorised and are recycled in the standard R fashion.

Usage

```
probability_ruin_rate(
  return_expected,
  return_sd,
  life_remaining_expected,
  value_probability_ruin
)
```

Arguments

```
return_expected      The expected real return of the entire pension portfolio.
return_sd            The projected standard deviation of the returns of the entire pension portfolio.
life_remaining_expected
                    The median projected remaining lifespan of the individual in question.
value_probability_ruin
                    The desired probability of retirement ruin, strictly between 0 and 1.
```

Examples

```
probability_ruin_rate(
  return_expected = 0.07,
  return_sd = 0.2,
  life_remaining_expected = 28.1,
  value_probability_ruin = 0.1
)

# Vectorised: sweep over target ruin probabilities
probability_ruin_rate(
  return_expected = 0.07,
  return_sd = 0.2,
  life_remaining_expected = 28.1,
  value_probability_ruin = c(0.05, 0.1, 0.2)
)
```

`rate_annuity`*Get the latest annuity rates for the UK*

Description

This function retrieves data on UK annuity rates from Sharing Pensions. Rates are available for different specifications and ages. See the website for further details. Please note that these are approximations, so do not rely on them for financial returns or planning.

Usage

```
rate_annuity(value_age, value_specifications)
```

Arguments

value_age These are the choices for the age of the annuitant, namely: '55', '60', '65', '70' or '75'. The ensuing annuity rates imply that the annuity begins to be paid immediately and thereafter on a monthly basis.

value_specifications

These are the specifications available, varying by: whether the annuity is on a single basis or a joint basis (and, if so, whether the last remaining spouse receives half or all of the annuity); whether the annuity is guaranteed for ten years or not; and whether the payments will escalate at 3 are available:

single_simple single, level rate, no guarantee

single_guaranteed single, level rate, guaranteed

single_escalating single, escalating rate, no guarantee

half_simple joint, half upon first death, level rate, no guarantee

full_simple joint, all upon first death, level rate, no guarantee

half_escalating joint, half upon first death, escalating rate, no guarantee

Source

<https://www.sharingpensions.co.uk/annuity_rates.htm>

Examples

```
## Not run:
rate_annuity(value_age = "65", value_specifications = "single_simple")

## End(Not run)
```

tax_parameters

Key income tax parameters for the UK.

Description

A dataset containing the main tax parameters used for calculating net income for the UK (outside of Scotland).

Usage

```
tax_parameters
```

Format

A data frame with 3 rows (one for each tax year) and 14 variables:

year_tax_end the calendar year in which the end of the tax year occurs

level_ni_lower the lower level breakpoint for National Insurance

level_ni_upper the upper level breakpoint for National Insurance

level_allowance_lower the lower level breakpoint for personal allowance

level_allowance_upper the upper level breakpoint for personal allowance

level_tax_higher the income breakpoint for higher rate income tax

level_tax_upper the income breakpoint for upper rate income tax

rate_ni_lower the lower level tax rate for National Insurance

rate_ni_upper the upper level tax rate for National Insurance

rate_allowance_drop the rate at which personal allowance drops after 'level_allowance_upper'

rate_tax_basic the basic rate of income tax

rate_tax_higher the higher rate of income tax

rate_tax_upper the upper rate of income tax

rate_tax_sales the rate of sales tax, also called VAT

Source

<https://www.crunch.co.uk/knowledge-tax/tax-rates-thresholds-and-allowances-for-current-tax-year/>

Index

- * **Retirement**

- probability_ruin, [2](#)
 - probability_ruin_rate, [4](#)
 - rate_annuity, [5](#)

- * **Tax**

- liability_tax, [2](#)

- * **datasets**

- tax_parameters, [6](#)

liability_tax, [2](#)

probability_ruin, [2](#)

probability_ruin_rate, [4](#)

rate_annuity, [5](#)

tax_parameters, [6](#)