

Package ‘mqriskR’

April 27, 2026

Type Package

Title Actuarial Risk Modeling and Life Contingencies

Version 0.1.0

Description Provides functions for actuarial risk modeling, including survival models, life annuities, multiple-decrement models, and mortality improvement projections. The package is designed to align with standard actuarial notation and supports teaching, exam preparation, and reproducible actuarial analysis. The methods are based on standard actuarial references including Camilli, Duncan and London (2014, ISBN:9781625423474) ``Models for Quantifying Risk" and Dickson, Hardy and Waters (2020, ISBN:9781108478083) ``Actuarial Mathematics for Life Contingent Risks".

License MIT + file LICENSE

Encoding UTF-8

Suggests ggplot2, testthat (>= 3.0.0), expm, knitr, rmarkdown

Config/testthat/edition 3

Imports stats, utils

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Nii Okine [aut, cre]

Maintainer Nii Okine <okinean@appstate.edu>

Repository CRAN

Date/Publication 2026-04-27 14:40:02 UTC

Contents

A2barx	7
A2barxn	8
A2barxn1	9
A2nAbarx	9
A2nAx	10
A2nAx_m	10

A2nEx	11
A2x	12
A2xn	12
A2xn1	13
A2xn1_m	13
A2xn_m	14
A2x_m	14
AAL_PUC_db	15
AAL_TUC_db	16
Abarx	16
Abarxj_md	17
Abarxn	18
Abarxn1	18
Abarxn1_udd	19
Abarxn_udd	19
Abarxy	20
Abarxy1	20
Abarxy2	21
Abarxybar	22
abarxybar_ch12	22
abarxy_ch12	23
Abarx_udd	24
abarx_y	24
Abaryx1	25
Abaryx2	26
abary_x	26
AB_cae	27
AB_fas	28
adotxy	28
adotxybar	29
adotxybarn	30
adotxyn	30
ag38_prefunding_ratio	31
ag38_reserve_ul	32
alphaF	33
annuity_annual	33
annuity_approximations	36
annuity_approximations_udd	36
annuity_approximations_woolhouse2	38
annuity_approximations_woolhouse3	39
annuity_certain	40
annuity_mthly	41
annuity_mthly_accum_due	41
annuity_mthly_accum_immediate	42
annuity_mthly_deferred_due	42
annuity_mthly_deferred_immediate	43
annuity_mthly_temp_due	44
annuity_mthly_temp_immediate	45

annuity_mthly_whole_due	46
annuity_mthly_whole_immediate	47
annuity_relationships	48
annuity_varying_payments	49
APV_gross_premiums	51
APV_NR_db	52
AS_path	53
AS_path_md	54
AVz_dc	54
AV_path_ul_typeA	55
AV_path_ul_typeB	56
Ax	57
Axj_md	57
Axn	58
Axn1	59
Axn1_m	59
Axn1_m_udd	60
axn_improved	60
Axn_m	61
Axn_m_udd	62
Axy	62
Axybar	63
Axybarn	64
Axybarn1	64
axybarn_ch12	65
axybar_ch12	66
Axyn	67
Axyn1	67
axyn_ch12	68
axy_ch12	69
ax_improved	70
Ax_m	70
Ax_m_udd	71
ax_y	71
ay_x	72
betaF	73
ch12_multilife_cont_params	73
chapter15_spot_interest_apv	74
chapter15_variable_interest_apv	76
coi_ul_typeB	77
contribution_rate_target	78
cov_term_deferred	79
cov_term_endow	79
cumhaz0	80
DAbarn1	81
DAxn1	81
DbarAbarn1	82
decompGg_disc	82

discount	84
discounted_payback_period	85
dist0	86
double_force_delta	86
double_force_i	87
dx	87
dxj	88
dxtau	88
ELtx	89
ex_complete	89
ex_complete_tab	90
ex_curtate	91
ex_curtate_tab	91
ex_temp_complete_tab	92
ex_temp_curtate_tab	92
fnk_from_z	93
forward_matrix_from_z	93
fx	94
fx_tab	95
gain_loss_md	95
GI_cont	97
GI_disc	97
GMF_rollforward_ul	98
GM_cont	99
GM_disc	99
GTg_disc	100
GT_cont	101
GT_disc	102
hazard0	103
htVx	103
IABarx	104
IAx	104
IAxn1	105
IbarAbarx	106
IbarAbarxn1	106
iMA_eiul	107
Income_dc	107
interest_convert	108
iP_eiul	109
IRR_profit	109
i_credit_eiul	110
life_table	111
lx	111
lx_select	112
lx_to_S0	112
markov_nstep_prob	113
md_table	113
meanVx	114

mortality_improvement	115
multilife_ch12	115
multilife_ch12_ext	115
mux_tab	116
nAbarx	116
nAbarx_udd	117
nAx	117
naxn_improved	118
nAx_m	118
nAx_m_udd	119
NC_EAN_db	120
NC_PUC_db	120
NC_TUC_db	121
ndx	122
nEx	122
nExy	123
nExybar	123
nkqx	124
nmqx	125
nmqx_select	125
NPV_partial	126
NPV_profit	126
npx	127
npxtau_md	128
npx_select	128
nqx	129
nqxj_md	129
nqxtau_md	130
nqx_select	131
PAB_cae	131
PAB_fas	132
Pbar_trapz_ms	133
Pi_signature	134
PnAdotx	135
Pnax	135
premium_ch9	136
profit_margin	140
Pr_vector_disc	140
pv_cashflows	142
pv_spot_cashflows	142
pxtau	143
pxtau_ul	144
px_proj	145
px_to_lx	145
qxprime_mudd	146
qxprime_sudd	146
qxtau	147
qx_dep_cf	147

qx_dep_sudd	148
qx_proj	148
qx_tab	149
qx_to_lx	149
replacement_ratio_db	150
replacement_ratio_dc	150
reserve_ch10	151
reserve_ch10_ext	151
reserve_ch11	151
reserve_ch11_ext	152
reserve_deferred_insurance_ch10	152
rt_ul	153
S0	153
S0_to_lx	154
salary_scale	154
select_life_table	155
solve_yield	155
thiele_backward_path	156
thiele_backward_step	157
thiele_dVdt	157
thiele_dVdt_01	158
thiele_path_01	159
tp00_tp01_euler	160
tpx	161
tpxprimej_cf	162
tpxtau_ul	163
tpxy	163
tpxybar	164
tpx_improved	165
tpx_tab	166
tpx_tau_cf	166
tqx	167
tqxj_cf	167
tqxprimej_cf	168
tqxprime_mudd	168
tqxy	169
tqxy1	170
tqxy2	170
tqxybar	171
tqx_tab	172
tqyx1	172
tqyx2	173
tsVx	174
tsVxn	175
tsVxn1	176
tVbarAbarx	177
tVbarx	177
tVEx	178

tVFx	180
tVGx	181
tVnAdotx	182
tVnax	183
tVnEx	184
tVx	184
tVxn	185
tVxn1	186
tVxn1_ret	186
tVxn_ret	187
tVx_m	188
tVx_ret	188
udd_continuous_multiplier	189
udd_mthly_multiplier	189
varLtx	190
var_Abarx	190
var_Abarxn	191
var_Abarxn1	192
var_Ax	192
var_Axn	193
var_Axn1	193
var_Axn1_m	194
var_Axn_m	194
var_Ax_m	195
var_nAbarx	195
var_nAx	196
var_nAx_m	197
var_nEx	197
Vprefloor_crvn_ul	198
vt_var	198
V_zeroized	199
z_from_coupon_annual	200
z_from_coupon_semi	201
z_from_fn1	201

Index**203**

A2barx

*Second moment of continuous whole life insurance PV***Description**

Computes ${}^2\bar{A}_x$ by evaluating \bar{A}_x at doubled force.

Usage

A2barx(x, i, model, ...)

Arguments

x	Age.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of second moments.

A2barxn

Second moment of continuous endowment insurance PV

Description

Computes ${}^2\bar{A}_{x:\overline{n}|}$.

Usage

A2barxn(x, n, i, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of second moments.

A2barxn1	<i>Second moment of continuous term insurance PV</i>
----------	--

Description

Computes ${}^2\bar{A}_{x:\overline{n}|}^1$ by evaluating $\bar{A}_{x:\overline{n}|}^1$ at doubled force.

Usage

A2barxn1(x, n, i, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of second moments.

A2nAbarx	<i>Second moment of continuous deferred insurance PV</i>
----------	--

Description

Computes ${}^2{}_n\bar{A}_x$ by evaluating ${}_n\bar{A}_x$ at doubled force.

Usage

A2nAbarx(x, n, i, model, ...)

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of second moments.

A2nAx

Second moment of deferred insurance PV

Description

Computes ${}^2_n|A_x$ by evaluating ${}_n|A_x$ at doubled force.

Usage

A2nAx(x, n, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.
tol	Numerical tolerance for truncating infinite sums.
k_max	Maximum number of terms in the sum.

Value

Numeric vector of second moments.

A2nAx_m

Second moment of m-thly deferred insurance PV

Description

Second moment of m-thly deferred insurance PV

Usage

A2nAx_m(x, n, i, m, model, ..., tol = 1e-12, j_max = 100000L)

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.
tol	Numerical tolerance for truncating the infinite sum.
j_max	Maximum number of m-thly intervals in the sum.

Value

Numeric vector of second moments.

A2nEx	<i>Second moment of pure endowment PV</i>
-------	---

Description

Computes ${}^2_nE_x = (v')^n {}_n p_x$.

Usage

A2nEx(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of second moments.

A2x *Second moment of whole life insurance PV*

Description

Computes 2A_x by evaluating A_x at doubled force.

Usage

A2x(x, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)

Arguments

x	Age.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.
tol	Numerical tolerance for truncating infinite sums.
k_max	Maximum number of terms in the sum.

Value

Numeric vector of second moments.

A2xn *Second moment of endowment insurance PV*

Description

Computes ${}^2A_{x:\overline{n}|}$.

Usage

A2xn(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of second moments.

A2xn1	<i>Second moment of term insurance PV</i>
-------	---

Description

Computes ${}^2A_{x:\overline{n}|}^1$ by evaluating $A_{x:\overline{n}|}^1$ at doubled force.

Usage

A2xn1(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of second moments.

A2xn1_m	<i>Second moment of m-thly term insurance PV</i>
---------	--

Description

Second moment of m-thly term insurance PV

Usage

A2xn1_m(x, n, i, m, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of second moments.

A2xn_m

Second moment of m-thly endowment insurance PV

Description

Second moment of m-thly endowment insurance PV

Usage

A2xn_m(x, n, i, m, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of second moments.

A2x_m

Second moment of m-thly whole life insurance PV

Description

Computes ${}^2A_x^{(m)}$ by evaluating $A_x^{(m)}$ at doubled force.

Usage

A2x_m(x, i, m, model, ..., tol = 1e-12, j_max = 100000L)

Arguments

x	Age.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.
tol	Numerical tolerance for truncating the infinite sum.
j_max	Maximum number of m-thly intervals in the sum.

Value

Numeric vector of second moments.

AAL_PUC_db

Projected Unit Credit accrued liability for a DB plan

Description

Computes the PUC accrued liability as the APV of the portion of the projected benefit attributed to past service.

Usage

```
AAL_PUC_db(
  projected_benefit,
  past_service,
  total_service,
  v_to_ret,
  p_surv,
  adue_ret
)
```

Arguments

projected_benefit	Projected benefit at retirement.
past_service	Past service completed.
total_service	Total service at retirement.
v_to_ret	Discount factor to retirement.
p_surv	Active-service survival probability to retirement.
adue_ret	Retirement annuity factor.

Value

PUC accrued liability.

Examples

```
AAL_PUC_db(projected_benefit = 30000, past_service = 10, total_service = 30,
v_to_ret = 0.5, p_surv = 0.9, adue_ret = 12)
```

AAL_TUC_db

Traditional Unit Credit accrued liability for a DB plan

Description

Computes the TUC accrued liability as the APV of the accrued benefit.

Usage

```
AAL_TUC_db(accrued_benefit, v_to_ret, p_surv, adue_ret)
```

Arguments

accrued_benefit	Accrued benefit at the valuation date.
v_to_ret	Discount factor to retirement.
p_surv	Active-service survival probability to retirement.
adue_ret	Retirement annuity factor.

Value

TUC accrued liability.

Examples

```
AAL_TUC_db(accrued_benefit = 12000, v_to_ret = 0.5, p_surv = 0.9, adue_ret = 12)
```

Abarx

Continuous whole life insurance APV

Description

Computes $\bar{A}_x = \int_0^\infty v^t {}_t p_x \mu_{x+t} dt$.

Usage

```
Abarx(x, i, model, ...)
```


Arguments

x	Age.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of APVs.

Abarxj_md	<i>Continuous multiple-decrement insurance APV $\bar{A}_x^{(j)}$</i>
-----------	---

Description

Computes the actuarial present value of a benefit payable at the moment of decrement by Cause j , matching Equation (14.4) in Chapter 14.

Usage

```
Abarxj_md(t, ptau, muj, delta, benefit = 1)
```

Arguments

t	Numeric vector of time points.
ptau	Numeric vector of values ${}_t p_x^{(\tau)}$.
muj	Numeric vector of values $\mu_{x+t}^{(j)}$.
delta	Force of interest.
benefit	Benefit amount payable on decrement by Cause j .

Details

The integral is evaluated numerically by the trapezoidal rule:

$$\bar{A}_x^{(j)} = \int_0^T v^t {}_t p_x^{(\tau)} \mu_{x+t}^{(j)} dt$$

Value

A numeric scalar.

Examples

```
t <- seq(0, 20, by = 0.01)
ptau <- exp(-0.012 * t)
mu_ac <- rep(0.002, length(t))
Abarxj_md(t, ptau, mu_ac, delta = 0.05, benefit = 2000)
```

Abarxn1_udd *UDD approximation of continuous term insurance*

Description

Computes $\bar{A}_{x:\bar{n}|}^1 = (i/\delta)A_{x:\bar{n}|}^1$.

Usage

Abarxn1_udd(Axn1, i)

Arguments

Axn1 Discrete term insurance APV.
i Effective annual interest rate.

Value

Continuous term insurance APV under UDD.

Abarxn_udd *UDD approximation of continuous endowment insurance*

Description

Computes $\bar{A}_{x:\bar{n}|} = (i/\delta)A_{x:\bar{n}|}^1 + {}_nE_x$.

Usage

Abarxn_udd(Axn1, nEx, i)

Arguments

Axn1 Discrete term insurance APV.
nEx Pure endowment APV, ${}_nE_x$.
i Effective annual interest rate.

Value

Continuous endowment insurance APV under UDD.

Abarxy

Continuous joint-life whole life insurance

Description

Computes $\bar{A}_{xy} = 1 - \delta \bar{a}_{xy}$.

Usage

Abarxy(x, y, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

Abarxy(40, 50, i = 0.05, model = "uniform", omega = 100)

Abarxy1

Continuous contingent insurance: benefit on death of (x) if before (y)

Description

Computes $\bar{A}_{xy}^1 = \int_0^\infty v^t {}_t p_{xy} \mu_{x+t} dt$.

Usage

Abarxy1(x, y, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Abarxy1(40, 50, i = 0.05, model = "uniform", omega = 100)
```

Abarxy2	<i>Continuous contingent insurance: benefit on death of (x) if after (y)</i>
---------	--

Description

Computes $\bar{A}_{xy}^2 = \bar{A}_x - \bar{A}_{xy}^1$.

Usage

```
Abarxy2(x, y, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Abarxy2(40, 50, i = 0.05, model = "uniform", omega = 100)
```

 Abarxybar

Continuous last-survivor whole life insurance

Description

Computes $\bar{A}_{\overline{xy}} = \bar{A}_x + \bar{A}_y - \bar{A}_{xy}$.

Usage

Abarxybar(x, y, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Abarxybar(40, 50, i = 0.05, model = "uniform", omega = 100)
```

 abarxybar_ch12

Continuous last-survivor whole life annuity

Description

Computes $\bar{a}_{\overline{xy}} = \bar{a}_x + \bar{a}_y - \bar{a}_{xy}$.

Usage

abarxybar(x, y, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Details

Shared documentation topic used to avoid filename collisions with case-distinct function names on case-insensitive file systems.

Value

Numeric vector.

Examples

```
abarxybar(40, 50, i = 0.05, model = "uniform", omega = 100)
```

abarxy_ch12

Continuous joint-life whole life annuity

Description

Computes $\bar{a}_{xy} = \int_0^{\infty} v^t {}_t p_{xy} dt$.

Usage

```
abarxy(x, y, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Details

Shared documentation topic used to avoid filename collisions with case-distinct function names on case-insensitive file systems.

Value

Numeric vector.

Examples

```
abax_y(40, 50, i = 0.05, model = "uniform", omega = 100)
```

Abarx_udd	<i>UDD approximation of continuous whole life insurance</i>
-----------	---

Description

Computes $\bar{A}_x = (i/\delta)A_x$.

Usage

```
Abarx_udd(Ax, i)
```

Arguments

Ax	Discrete whole life insurance APV.
i	Effective annual interest rate.

Value

Continuous whole life insurance APV under UDD.

abax_y	<i>Continuous reversionary annuity to (y) after death of (x)</i>
--------	--

Description

Computes $\bar{a}_{x|y} = \bar{a}_y - \bar{a}_{xy}$.

Usage

```
abax_y(x, y, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
abaryx_y(40, 50, i = 0.05, model = "uniform", omega = 100)
```

Abaryx1

Continuous contingent insurance: benefit on death of (y) if before (x)

Description

Computes $\bar{A}_{xy}^1 = \int_0^\infty v^t {}_t p_{xy} \mu_{y+t} dt$.

Usage

```
Abaryx1(x, y, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Abaryx1(40, 50, i = 0.05, model = "uniform", omega = 100)
```

 Abaryx2

Continuous contingent insurance: benefit on death of (y) if after (x)

Description

Computes $\bar{A}_{xy}^2 = \bar{A}_y - \bar{A}_{xy}^1$.

Usage

Abaryx2(x, y, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

Abaryx2(40, 50, i = 0.05, model = "uniform", omega = 100)

 abary_x

Continuous reversionary annuity to (x) after death of (y)

Description

Computes $\bar{a}_{y|x} = \bar{a}_x - \bar{a}_{xy}$.

Usage

abary_x(x, y, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
abary_x(40, 50, i = 0.05, model = "uniform", omega = 100)
```

 AB_cae

Accrued benefit for a career average earnings plan

Description

Computes the accrued benefit using actual salary history only.

Usage

```
AB_cae(salary_history, p)
```

Arguments

salary_history	Numeric vector of annual salaries to date.
p	Accrual percentage.

Value

Accrued benefit.

Examples

```
AB_cae(salary_history = c(100000, 104000, 108160), p = 1)
```


Value

Numeric vector.

Examples

```
adotxy(40, 50, i = 0.05, model = "uniform", omega = 100)
```

adotxybar	<i>Last-survivor whole life annuity-due</i>
-----------	---

Description

Computes $\ddot{a}_{\overline{xy}}$.

Usage

```
adotxybar(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Value

Numeric vector.

Examples

```
adotxybar(40, 50, i = 0.05, model = "uniform", omega = 100)
```

adotxybarn *Last-survivor temporary annuity-due*

Description

Computes $\ddot{a}_{\overline{xy:\bar{n}}}$.

Usage

adotxybarn(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
adotxybarn(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

adotxyn *Joint-life temporary annuity-due*

Description

Computes $\ddot{a}_{\overline{xy:\bar{n}}}$.

Usage

adotxyn(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
adotxyn(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

ag38_prefunding_ratio *AG 38 prefunding ratio*

Description

Computes the prefunding ratio in Equation (16.18), capped at 1.

Usage

```
ag38_prefunding_ratio(excess_payment, nsp_required)
```

Arguments

excess_payment	Excess payment or shadow-fund amount.
nsp_required	Net single premium required to fully fund the guarantee.

Value

Numeric scalar.

Examples

```
ag38_prefunding_ratio(60000, 100000)
```

ag38_reserve_ul *AG 38 reserve calculation*

Description

Computes the main quantities in the Chapter 16 AG 38 reserve calculation, including the prefunding ratio, reduced deficiency reserve, Step (8) reserve, and final increased basic reserve.

Usage

```
ag38_reserve_ul(  
  basic_reserve,  
  deficiency_reserve = 0,  
  excess_payment,  
  nsp_required,  
  valuation_nsp,  
  surrender_charge = 0  
)
```

Arguments

`basic_reserve` Basic reserve.
`deficiency_reserve` Deficiency reserve.
`excess_payment` Excess payment or shadow-fund amount.
`nsp_required` Net single premium required to fully fund the guarantee.
`valuation_nsp` Valuation net single premium.
`surrender_charge` Applicable surrender charge.

Value

A named list.

Examples

```
ag38_reserve_ul(  
  basic_reserve = 10000,  
  deficiency_reserve = 0,  
  excess_payment = 60000,  
  nsp_required = 100000,  
  valuation_nsp = 150000,  
  surrender_charge = 5000  
)
```

alphaF	<i>Full preliminary term first-year modified premium</i>
--------	--

Description

Computes $\alpha^F = vq_x = A_{x:\overline{1}|}^1$.

Computes $\alpha^F = vq_x = A_{x:\overline{1}|}^1$.

Usage

```
alphaF(x, i, tbl = NULL, model = NULL, ...)
```

```
alphaF(x, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Issue age.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
alphaF(40, i = 0.05, model = "uniform", omega = 100)
alphaF(40, i = 0.05, model = "uniform", omega = 100)
```

annuity_annual	<i>Annual annuity functions (Chapter 8)</i>
----------------	---

Description

Annual whole life, temporary, deferred, and actuarial accumulated value annuity functions in immediate, due, and continuous forms.

Computes the annual whole life annuity-immediate $a_x = \sum_{t=1}^{\infty} v^t {}_t p_x$.

Computes the annual whole life annuity-due $\ddot{a}_x = \sum_{t=0}^{\infty} v^t {}_t p_x = 1 + a_x$.

Computes the continuous whole life annuity $\bar{a}_x = \int_0^{\infty} v^t {}_t p_x dt$.

Computes the annual temporary annuity-immediate $a_{x:\overline{n}|} = \sum_{t=1}^n v^t {}_t p_x$.

Computes the annual temporary annuity-due $\ddot{a}_{x:\overline{n}|} = \sum_{t=0}^{n-1} v^t {}_t p_x$.

Computes the continuous temporary annuity $\bar{a}_{x:\overline{n}|} = \int_0^n v^t {}_t p_x dt$.

Computes the annual deferred whole life annuity-immediate ${}_n|a_x = {}_n E_x a_{x+n}$.

Computes the annual deferred whole life annuity-due ${}_n|\ddot{a}_x = {}_n E_x \ddot{a}_{x+n}$.

Computes the continuous deferred whole life annuity ${}_n|\bar{a}_x = {}_n E_x \bar{a}_{x+n}$.

Computes $s_{x:\overline{n}|} = a_{x:\overline{n}|} / {}_n E_x$.

Computes $\ddot{s}_{x:\overline{n}|} = \ddot{a}_{x:\overline{n}|} / {}_n E_x$.

Computes $\bar{s}_{x:\overline{n}|} = \bar{a}_{x:\overline{n}|} / {}_n E_x$.

Usage

`ax(x, i, model, ..., k_max = 5000, tol = 1e-12)`

`adotx(x, i, model, ..., k_max = 5000, tol = 1e-12)`

`abarx(x, i, model, ..., tol = 1e-10)`

`axn(x, n, i, model, ...)`

`adotxn(x, n, i, model, ...)`

`abarxn(x, n, i, model, ...)`

`nax(x, n, i, model, ..., k_max = 5000, tol = 1e-12)`

`nadotx(x, n, i, model, ..., k_max = 5000, tol = 1e-12)`

`nabarx(x, n, i, model, ..., tol = 1e-10)`

`sxn(x, n, i, model, ...)`

`sdotxn(x, n, i, model, ...)`

`sbarxn(x, n, i, model, ...)`

Arguments

x	Age.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters passed to the survival model.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.
n	Term in years.

Details

Naming convention follows Chapter 8 notation:

- $ax() = a_x$
- $adotx() = \ddot{a}_x$
- $abarcx() = \bar{a}_x$
- $axn() = a_{x:\overline{n}|}$
- $adotxn() = \ddot{a}_{x:\overline{n}|}$
- $abarcxn() = \bar{a}_{x:\overline{n}|}$
- $nax() = {}_n|a_x$
- $nadotx() = {}_n|\ddot{a}_x$
- $nabarcx() = {}_n|\bar{a}_x$
- $sxn() = s_{x:\overline{n}|}$
- $sdotxn() = \ddot{s}_{x:\overline{n}|}$
- $sbarxn() = \bar{s}_{x:\overline{n}|}$

These functions work directly from the Chapter 5 survival model functions and the Chapter 7 pure endowment function $nEx()$.

Value

Numeric vector.

 annuity_approximations

Annuity approximations (Chapter 8)

Description

Chapter 8 approximation formulas for m-thly and continuous life annuities.

Details

This file implements:

- UDD approximations for m-thly annuities,
- UDD approximations for continuous annuities,
- Woolhouse 2-term approximations,
- Woolhouse 3-term approximations.

 annuity_approximations_udd

UDD annuity approximations

Description

UDD-based approximations for Chapter 8 annuity functions.

Computes

$$\ddot{a}_x^{(m)} \approx \alpha(m)\ddot{a}_x - \beta(m).$$

Computes

$$\ddot{a}_{x:\overline{n}|}^{(m)} \approx \alpha(m)\ddot{a}_{x:\overline{n}|} - \beta(m)(1 - {}_nE_x).$$

Computes

$${}_n\ddot{a}_x^{(m)} \approx \alpha(m) {}_n\ddot{a}_x - \beta(m) {}_nE_x.$$

Computes

$$a_x^{(m)} \approx \alpha(m)a_x + \gamma(m).$$

Computes

$$a_{x:\overline{n}|}^{(m)} \approx \alpha(m)a_{x:\overline{n}|} + \gamma(m)(1 - {}_nE_x).$$

Computes

$${}_n a_x^{(m)} \approx \alpha(m) {}_n a_x + \gamma(m) {}_n E_x.$$

Computes

$$\ddot{s}_{x:\overline{n}|}^{(m)} \approx \alpha(m)\ddot{s}_{x:\overline{n}|} - \beta(m) \left(\frac{1}{{}_nE_x} - 1 \right).$$

Computes

$$s_{x:\bar{n}|}^{(m)} \approx \alpha(m)s_{x:\bar{n}|} + \gamma(m) \left(\frac{1}{{}_nE_x} - 1 \right).$$

Computes

$$\bar{a}_x \approx \frac{id}{\delta^2} \ddot{a}_x - \frac{i - \delta}{\delta^2}.$$

Uses the identity

$$\bar{a}_{x:\bar{n}|} \approx \frac{1 - \bar{A}_{x:\bar{n}|}}{\delta}$$

together with the package's existing Chapter 7 UDD insurance approximation for $\bar{A}_{x:\bar{n}|}$.

Computes

$${}_n|\bar{a}_x \approx {}_nE_x \bar{a}_{x+n}.$$

Usage

`adotx_m_udd(x, m, i, model, ...)`

`adotxn_m_udd(x, n, m, i, model, ...)`

`nadotx_m_udd(x, n, m, i, model, ...)`

`ax_m_udd(x, m, i, model, ...)`

`axn_m_udd(x, n, m, i, model, ...)`

`nax_m_udd(x, n, m, i, model, ...)`

`sdotxn_m_udd(x, n, m, i, model, ...)`

`sxn_m_udd(x, n, m, i, model, ...)`

`abarx_udd(x, i, model, ...)`

`abarn_udd(x, n, i, model, ...)`

`nabarn_udd(x, n, i, model, ...)`

Arguments

<code>x</code>	Age.
<code>m</code>	Number of payments per year.
<code>i</code>	Effective annual interest rate.
<code>model</code>	Survival model.
<code>...</code>	Additional model parameters.
<code>n</code>	Term.

Details

These functions implement the standard Uniform Distribution of Deaths approximations linking annual, m-thly, and continuous annuity values.

The exported functions documented on this page are:

- `ax_m_udd()`
- `axn_m_udd()`
- `nax_m_udd()`
- `adotx_m_udd()`
- `adotxn_m_udd()`
- `nadotx_m_udd()`
- `sxn_m_udd()`
- `sdotxn_m_udd()`
- `abax_udd()`
- `abaxn_udd()`
- `nabax_udd()`

Note that this function relies on the already-existing `Abarxn_udd()` implementation in the package, so extra survival-model arguments are not used.

Value

Numeric vector.

annuity_approximations_woolhouse2

Woolhouse 2-term annuity approximations

Description

Woolhouse 2-term approximations for Chapter 8 annuity functions.

Usage

```
ax_m_woolhouse2(x, m, i, model, ...)
```

```
adotx_m_woolhouse2(x, m, i, model, ...)
```

```
nax_m_woolhouse2(x, n, m, i, model, ...)
```

```
nadotx_m_woolhouse2(x, n, m, i, model, ...)
```

```
axn_m_woolhouse2(x, n, m, i, model, ...)
```

```

adotxn_m_woolhouse2(x, n, m, i, model, ...)
sxn_m_woolhouse2(x, n, m, i, model, ...)
sdotxn_m_woolhouse2(x, n, m, i, model, ...)
abbarx_woolhouse2(x, i, model, ...)

```

Arguments

x	Age.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.
n	Term.

Value

Numeric vector.

annuity_approximations_woolhouse3
Woolhouse 3-term annuity approximations

Description

Woolhouse 3-term approximations for Chapter 8 annuity functions.

Usage

```

ax_m_woolhouse3(x, m, i, model, ...)
adotx_m_woolhouse3(x, m, i, model, ...)
nax_m_woolhouse3(x, n, m, i, model, ...)
nadotx_m_woolhouse3(x, n, m, i, model, ...)
axn_m_woolhouse3(x, n, m, i, model, ...)
adotxn_m_woolhouse3(x, n, m, i, model, ...)
abbarx_woolhouse3(x, i, model, ...)

```

Arguments

x	Age.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.
n	Term.

Value

Numeric vector.

annuity_certain	<i>Present value of a level annuity-certain</i>
-----------------	---

Description

Computes the present value of an n-period annuity certain with level payments of 1 per period.

Usage

```
annuity_certain(n, i, due = FALSE, m = 1, cont = FALSE)
```

Arguments

n	Number of payments or periods.
i	Effective interest rate per period.
due	If TRUE, annuity-due; otherwise annuity-immediate.
m	Payment frequency per period (m = 1 means annual).
cont	If TRUE, continuous payment model.

Value

Present value.

Examples

```
annuity_certain(n = 10, i = 0.05)
annuity_certain(n = 10, i = 0.05, due = TRUE)
annuity_certain(n = 10, i = 0.05, cont = TRUE)
```

annuity_mthly *m-thly contingent annuity functions (Chapter 8)*

Description

Chapter 8 functions for life annuities payable m-thly.

Details

These functions implement the exact m-thly formulas from Section 8.5 using the survival model functions already available in the package.

In all cases, the annuity is one unit per year payable in m equal installments, so each payment is of size $1/m$.

annuity_mthly_accum_due *Temporary m-thly annuity-due actuarial accumulated value*

Description

Computes

$$\ddot{s}_{x:\overline{n}|}^{(m)} = \ddot{a}_{x:\overline{n}|}^{(m)} / n E_x$$

Usage

`sdotxn_m(x, n, m, i, model, ...)`

Arguments

<code>x</code>	Age.
<code>n</code>	Term in years.
<code>m</code>	Number of payments per year.
<code>i</code>	Effective annual interest rate.
<code>model</code>	Survival model name.
<code>...</code>	Additional parameters passed to the survival model.

Value

Numeric vector of actuarial accumulated values.

Examples

```
sdotxn_m(40, n = 10, m = 12, i = 0.05, model = "uniform", omega = 100)
```

 annuity_mthly_accum_immediate

Temporary m-thly annuity-immediate actuarial accumulated value

Description

Computes

$$s_{x:\overline{n}|}^{(m)} = a_{x:\overline{n}|}^{(m)} / {}_nE_x$$

Usage

```
sxn_m(x, n, m, i, model, ...)
```

Arguments

x	Age.
n	Term in years.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.

Value

Numeric vector of actuarial accumulated values.

Examples

```
sxn_m(40, n = 10, m = 12, i = 0.05, model = "uniform", omega = 100)
```

 annuity_mthly_deferred_due

Deferred whole life m-thly annuity-due

Description

Computes the deferred whole life m-thly annuity-due using

$${}_nE_x \ddot{a}_{x+n}^{(m)}$$

Usage

```
nadotx_m(x, n, m, i, model, ..., k_max = 2e+05, tol = 1e-12)
```

Arguments

x	Age.
n	Deferral period in years.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.

Value

Numeric vector of annuity values.

Examples

```
nadotx_m(40, n = 10, m = 12, i = 0.05, model = "uniform", omega = 100)
```

annuity_mthly_deferred_immediate

Deferred whole life m-thly annuity-immediate

Description

Computes the deferred whole life m-thly annuity-immediate using

$${}_nE_x a_{x+n}^{(m)}$$

Usage

```
nax_m(x, n, m, i, model, ..., k_max = 2e+05, tol = 1e-12)
```

Arguments

x	Age.
n	Deferral period in years.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.

Value

Numeric vector of annuity values.

Examples

```
max_m(40, n = 10, m = 12, i = 0.05, model = "uniform", omega = 100)
```

annuity_mthly_temp_due

Temporary m-thly annuity-due

Description

Computes the exact temporary m-thly annuity-due.

Usage

```
adotxn_m(x, n, m, i, model, ...)
```

Arguments

x	Age.
n	Term in years.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.

Details

$$\ddot{a}_{x:\overline{n}|}^{(m)} = \frac{1}{m} \sum_{t=0}^{mn-1} v^{t/m} {}_{t/m}p_x$$

Value

Numeric vector of annuity values.

Examples

```
adotxn_m(40, n = 10, m = 12, i = 0.05, model = "uniform", omega = 100)
```

annuity_mthly_temp_immediate
Temporary m-thly annuity-immediate

Description

Computes the exact temporary m-thly annuity-immediate.

Usage

```
axn_m(x, n, m, i, model, ...)
```

Arguments

x	Age.
n	Term in years.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.

Details

$$a_{x:\overline{n}|}^{(m)} = \frac{1}{m} \sum_{t=1}^{mn} v^{t/m} {}_{t/m}p_x$$

Value

Numeric vector of annuity values.

Examples

```
axn_m(40, n = 10, m = 12, i = 0.05, model = "uniform", omega = 100)
```

 annuity_mthly_whole_due

Whole life m-thly annuity-due

Description

Computes the exact whole life m-thly annuity-due.

Usage

```
adotx_m(x, m, i, model, ..., k_max = 2e+05, tol = 1e-12)
```

Arguments

x	Age.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.

Details

$$\ddot{a}_x^{(m)} = \frac{1}{m} \sum_{t=0}^{\infty} v^{t/m} {}_{t/m}p_x$$

Value

Numeric vector of annuity values.

Examples

```
adotx_m(40, m = 12, i = 0.05, model = "uniform", omega = 100)
```

annuity_mthly_whole_immediate
Whole life m-thly annuity-immediate

Description

Computes the exact whole life m-thly annuity-immediate, with annual payment rate 1 split into m equal payments of size 1/m.

Usage

```
ax_m(x, m, i, model, ..., k_max = 2e+05, tol = 1e-12)
```

Arguments

x	Age.
m	Number of payments per year.
i	Effective annual interest rate.
model	Survival model name.
...	Additional parameters passed to the survival model.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.

Details

$$a_x^{(m)} = \frac{1}{m} \sum_{t=1}^{\infty} v^{t/m} {}_{t/m}p_x$$

Value

Numeric vector of annuity values.

Examples

```
ax_m(40, m = 12, i = 0.05, model = "uniform", omega = 100)
```

annuity_relationships *Annuity-insurance relationships (Chapter 8)*

Description

This file provides the core Chapter 8 identities linking annual and continuous annuity functions to the corresponding insurance functions.

Computes $a_x = (v - A_x)/d$.

Computes $\ddot{a}_x = (1 - A_x)/d$.

Computes $\bar{a}_x = (1 - \bar{A}_x)/\delta$.

Computes $a_{x:\overline{n}|} = \ddot{a}_{x:\overline{n}|} - 1 + {}_nE_x$ together with $\ddot{a}_{x:\overline{n}|} = (1 - A_{x:\overline{n}|})/d$.

Computes $\ddot{a}_{x:\overline{n}|} = (1 - A_{x:\overline{n}|})/d$.

Computes $\bar{a}_{x:\overline{n}|} = (1 - \bar{A}_{x:\overline{n}|})/\delta$.

Computes ${}_n|a_x = {}_nE_x a_{x+n}$.

Computes ${}_n|\ddot{a}_x = {}_nE_x \ddot{a}_{x+n}$.

Computes ${}_n|\bar{a}_x = {}_nE_x \bar{a}_{x+n}$.

Usage

`annuity_identity_ax(x, i, model, ...)`

`annuity_identity_adotx(x, i, model, ...)`

`annuity_identity_abarx(x, i, model, ...)`

`annuity_identity_axn(x, n, i, model, ...)`

`annuity_identity_adotxn(x, n, i, model, ...)`

`annuity_identity_abarxn(x, n, i, model, ...)`

`annuity_identity_nax(x, n, i, model, ..., k_max = 5000, tol = 1e-12)`

`annuity_identity_nadotx(x, n, i, model, ..., k_max = 5000, tol = 1e-12)`

`annuity_identity_nabarx(x, n, i, model, ..., tol = 1e-10)`

Arguments

<code>x</code>	Age.
<code>i</code>	Effective annual interest rate.
<code>model</code>	Survival model.
<code>...</code>	Additional model parameters passed to the survival model.

n	Term in years.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.

Details

Included identities:

- whole life immediate: $a_x = (v - A_x)/d$
- whole life due: $\ddot{a}_x = (1 - A_x)/d$
- whole life continuous: $\bar{a}_x = (1 - \bar{A}_x)/\delta$
- temporary immediate: $a_{x:\overline{n}|} = (1 - A_{x:\overline{n}|})/d - 1 + {}_nE_x$
- temporary due: $\ddot{a}_{x:\overline{n}|} = (1 - A_{x:\overline{n}|})/d$
- temporary continuous: $\bar{a}_{x:\overline{n}|} = (1 - \bar{A}_{x:\overline{n}|})/\delta$
- deferred immediate: ${}_n|a_x = {}_nE_x a_{x+n}$
- deferred due: ${}_n|\ddot{a}_x = {}_nE_x \ddot{a}_{x+n}$
- deferred continuous: ${}_n|\bar{a}_x = {}_nE_x \bar{a}_{x+n}$

These are wrapper functions that evaluate the Chapter 8 relationships using the Chapter 7 insurance functions already implemented in the package.

Hence $a_{x:\overline{n}|} = (1 - A_{x:\overline{n}|})/d - 1 + {}_nE_x$.

Value

Numeric vector.

annuity_varying_payments

Varying-payment annuity functions (Chapter 8)

Description

Chapter 8 non-level annuity functions for increasing and decreasing life annuities.

Computes

$$(Ia)_x = \sum_{t=1}^{\infty} t v^t {}_t p_x.$$

Computes

$$(Ia)_{x:\overline{n}|} = \sum_{t=1}^n t v^t {}_t p_x.$$

Computes

$$(Da)_{x:\overline{n}|} = \sum_{t=1}^n (n + 1 - t) v^t {}_t p_x.$$

Computes

$$(I\ddot{a})_x = \sum_{t=0}^{\infty} (t+1) v^t {}_t p_x.$$

Computes

$$(I\ddot{a})_{x:\overline{n}|} = \sum_{t=0}^{n-1} (t+1) v^t {}_t p_x.$$

Computes

$$(D\ddot{a})_{x:\overline{n}|} = \sum_{t=0}^{n-1} (n-t) v^t {}_t p_x.$$

Computes

$$(\bar{I}\bar{a})_x = \int_0^{\infty} t v^t {}_t p_x dt.$$

Computes

$$(\bar{I}\bar{a})_{x:\overline{n}|} = \int_0^n t v^t {}_t p_x dt.$$

Computes

$$(\bar{D}\bar{a})_{x:\overline{n}|} = \int_0^n (n-t) v^t {}_t p_x dt.$$

Usage

`Iax(x, i, model, ..., k_max = 5000, tol = 1e-12)`

`Iaxn(x, n, i, model, ...)`

`Daxn(x, n, i, model, ...)`

`Iadotx(x, i, model, ..., k_max = 5000, tol = 1e-12)`

`Iadotxn(x, n, i, model, ...)`

`Dadotxn(x, n, i, model, ...)`

`Iabarx(x, i, model, ..., tol = 1e-10)`

`Iabarxn(x, n, i, model, ...)`

`Dabarxn(x, n, i, model, ...)`

Arguments

<code>x</code>	Age.
<code>i</code>	Effective annual interest rate.
<code>model</code>	Survival model.

...	Additional model parameters.
k_max	Maximum summation horizon for non-terminating models.
tol	Truncation tolerance for non-terminating models.
n	Term in years.

Details

The functions implemented here match the notation in Section 8.6:

- $Iax() = (Ia)_x$
- $Iaxn() = (Ia)_{x:\overline{n}|}$
- $Daxn() = (Da)_{x:\overline{n}|}$
- $Iadotx() = (I\ddot{a})_x$
- $Iadotxn() = (I\ddot{a})_{x:\overline{n}|}$
- $Dadotxn() = (D\ddot{a})_{x:\overline{n}|}$
- $Iabarx() = (\bar{Ia})_x$
- $Iabarn() = (\bar{Ia})_{x:\overline{n}|}$
- $Dabarn() = (\bar{Da})_{x:\overline{n}|}$

Value

Numeric vector.

Numeric vector.

Numeric vector.

APV_gross_premiums *APV of gross premiums under a risk discount rate*

Description

Computes the actuarial present value of gross premiums:

$$APV_{GP} = \sum_{t=0}^{n-1} \frac{G_{t+1} \cdot {}_t p_x^{(\tau)}}{(1+r)^t}.$$

Usage

APV_gross_premiums(G, r, p_tau)

Arguments

G	Gross premium vector for policy years 1 through n .
r	Risk discount rate.
p_tau	One-year in-force probabilities. This may have length $n - 1$ or n . If length n , the final entry is ignored.

Value

Numeric scalar.

Examples

```
APV_gross_premiums(G = rep(95, 3), r = 0.10, p_tau = c(0.99858, 0.99847, 0.99834))
```

APV_NR_db

APV of normal retirement benefit for a DB plan

Description

Computes Equation (18.10).

Usage

```
APV_NR_db(PABz, v_to_ret, p_surv, adue_ret)
```

Arguments

PABz	Projected annual benefit at retirement.
v_to_ret	Discount factor from current age to retirement.
p_surv	Active-service survival probability to retirement.
adue_ret	Retirement annuity factor.

Value

Actuarial present value of the normal retirement benefit.

Examples

```
APV_NR_db(PABz = 108008.66, v_to_ret = 1 / 1.06^30, p_surv = 0.8, adue_ret = 12)
```

AS_path	<i>Projected asset share path $_kAS$</i>
---------	---

Description

Computes projected asset shares recursively using Equations (14.5b) and (14.6b) of Chapter 14, with optional support for a survival benefit payable at the end of year k .

Usage

AS_path(AS0, G, r, e, b1, b2, q1, q2, p_tau, i, b3 = NULL)

Arguments

AS0	Initial asset share $_0AS$.
G	Level annual premium.
r	Numeric vector of percent-of-premium expense factors.
e	Numeric vector of fixed contract expenses.
b1	Numeric vector of Cause 1 benefit amounts.
b2	Numeric vector of Cause 2 benefit amounts.
q1	Numeric vector of Cause 1 decrement probabilities.
q2	Numeric vector of Cause 2 decrement probabilities.
p_tau	Numeric vector of in-force probabilities.
i	Effective annual interest rate.
b3	Optional numeric vector of survival benefit amounts payable at the end of year k conditional on survival through year k . Defaults to a zero vector.

Details

For policy year k ,

$$[{}_{k-1}AS + G(1 - r_k) - e_k](1 + i) = b_k^{(1)} q_{x+k-1}^{(1)} + b_k^{(2)} q_{x+k-1}^{(2)} + p_{x+k-1}^{(\tau)} \left(b_k^{(3)} + {}_kAS \right)$$

so that

$${}_kAS = \frac{[{}_{k-1}AS + G(1 - r_k) - e_k](1 + i) - b_k^{(1)} q_{x+k-1}^{(1)} - b_k^{(2)} q_{x+k-1}^{(2)}}{p_{x+k-1}^{(\tau)}} - b_k^{(3)}$$

Value

A data frame with columns k and AS.

<code>AS_path_md</code>	<i>General projected asset share path (multiple decrements)</i>
-------------------------	---

Description

General projected asset share path (multiple decrements)

Usage

`AS_path_md(AS0, G, r, e, b_mat, q_mat, p_tau, i, b_surv = NULL)`

Arguments

<code>AS0</code>	Initial asset share.
<code>G</code>	Premium.
<code>r</code>	Expense percentages.
<code>e</code>	Fixed expenses.
<code>b_mat</code>	Matrix of benefits (rows = years, cols = causes).
<code>q_mat</code>	Matrix of decrement probabilities (same shape as <code>b_mat</code>).
<code>p_tau</code>	In-force probabilities.
<code>i</code>	Interest rate.
<code>b_surv</code>	Optional survival benefits.

Value

A data frame with columns `k` and `AS`. Column `k` gives the policy year from 0 to n , and column `AS` gives the corresponding projected asset share at each year.

<code>AVz_dc</code>	<i>Accumulated value of defined contribution plan contributions</i>
---------------------	---

Description

Computes the accumulated value at retirement age z for the defined contribution model in Equation (18.1).

Usage

`AVz_dc(x, z, Sx, c, i, g = NULL, s = NULL)`

Arguments

x	Entry age.
z	Retirement age.
Sx	Salary at age x.
c	Contribution rate as a proportion of salary.
i	Annual effective interest rate.
g	Optional constant annual salary growth rate.
s	Optional salary scale vector of length z - x.

Value

The accumulated value of contributions at age z.

Examples

AVz_dc(x = 30, z = 65, Sx = 50000, c = 0.10, i = 0.05, g = 0.04)

AV_path_ul_typeA	<i>Account-value path for Type A universal life</i>
------------------	---

Description

Computes the year-by-year account value roll-forward for Type A universal life using the explicit form in Equation (16.8), or Equation (16.9) when $i^q = i^c$.

Usage

AV_path_ul_typeA(G, r, e, qx, ic, B, iq = ic, AV0 = 0)

Arguments

G	Premium vector G_t .
r	Percent-of-premium expense vector r_t .
e	Fixed expense vector e_t .
qx	Mortality vector q_{x+t-1} .
ic	Credited interest rate vector i^c .
B	Fixed death benefit face amount.
iq	Interest rate vector i^q used in cost of insurance. Defaults to ic.
AV0	Initial account value. Defaults to 0.

Value

A data frame with columns t, premium, AV.

Examples

```
qx <- c(.00076, .00081, .00085, .00090, .00095)
r <- c(.75, .10, .10, .10, .10)
e <- c(100, 20, 20, 20, 20)
G <- rep(5000, 5)
```

```
AV_path_ul_typeA(G = G, r = r, e = e, qx = qx, ic = 0.03, B = 100000)
```

AV_path_ul_typeB	<i>Account-value path for Type B universal life</i>
------------------	---

Description

Computes the year-by-year account value roll-forward for Type B universal life using Equations (16.4) and (16.5a).

Usage

```
AV_path_ul_typeB(G, r, e, qx, ic, B, iq = ic, AV0 = 0)
```

Arguments

G	Premium vector G_t .
r	Percent-of-premium expense vector r_t .
e	Fixed expense vector e_t .
qx	Mortality vector q_{x+t-1} .
ic	Credited interest rate vector i^c .
B	Face amount.
iq	Interest rate vector i^q used in cost of insurance. Defaults to ic.
AV0	Initial account value. Defaults to 0.

Value

A data frame with columns t, premium, net_contribution, COI, and AV.

Examples

```
qx <- c(.00076, .00081, .00085, .00090, .00095)
r <- c(.75, .10, .10, .10, .10)
e <- c(100, 20, 20, 20, 20)
G <- rep(5000, 5)
```

```
AV_path_ul_typeB(G = G, r = r, e = e, qx = qx, ic = 0.03, B = 100000)
```

Ax	<i>Whole life insurance APV</i>
----	---------------------------------

Description

Computes $A_x = \sum_{k=0}^{\infty} v^{k+1} {}_k|q_x$.

Usage

`Ax(x, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)`

Arguments

x	Age.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.
tol	Numerical tolerance for truncating infinite sums.
k_max	Maximum number of terms in the sum.

Value

Numeric vector of APVs.

Axj_md	<i>Discrete multiple-decrement insurance APV $A_{x^{\wedge}(j)}$</i>
--------	---

Description

Computes the actuarial present value of a benefit payable at the end of the year of decrement if decrement occurs by Cause j , matching Equation (14.3b) in Chapter 14.

Usage

`Axj_md(qj, ptau, i, benefit = 1)`

Arguments

qj	Numeric vector of conditional probabilities $q_{x+k}^{(j)}$ for Cause j .
ptau	Numeric vector of survival probabilities ${}_k p_x^{(\tau)}$ of remaining in force to duration k .
i	Effective annual interest rate.
benefit	Benefit amount payable on decrement by Cause j .

Details

The function evaluates

$$A_x^{(j)} = \sum_{k=0}^{n-1} v^{k+1} {}_kP_x^{(\tau)} q_{x+k}^{(j)}$$

with an optional benefit amount multiplier.

Value

A numeric scalar.

Examples

```
q1 <- c(.02, .02, .02, .02, .02)
q2 <- c(.03, .04, .05, .06, .00)
q3 <- c(.00, .00, .00, .00, .98)
qtau <- q1 + q2 + q3

ptau <- numeric(length(qtau))
ptau[1] <- 1
for (k in 2:length(qtau)) {
  ptau[k] <- prod(1 - qtau[1:(k - 1)])
}

Axj_md(qj = q1, ptau = ptau, i = 0.06, benefit = 1000)
```

Axn

Endowment insurance APV

Description

Computes $A_{x:\overline{n}|} = A_{x:\overline{n}|}^1 + {}_nE_x$.

Usage

```
Axn(x, n, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of APVs.

Axn1	<i>Term insurance APV</i>
------	---------------------------

Description

Computes $A_{x:\overline{n}|}^1 = \sum_{k=0}^{n-1} v^{k+1} {}_k|q_x$.

Usage

Axn1(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of APVs.

Axn1_m	<i>m-thly term insurance APV</i>
--------	----------------------------------

Description

Computes $A_{x:\overline{n}|}^{1(m)} = \sum_{j=0}^{mn-1} v^{(j+1)/m} \Pr(j/m < T_x \leq (j+1)/m)$.

Usage

Axn1_m(x, n, i, m, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of APVs.

Axn1_m_udd

UDD approximation of m-thly term insurance

Description

Computes $A_{x:\overline{n}|}^{1(m)} = (i/i^{(m)})A_{x:\overline{n}|}^1$.

Usage

Axn1_m_udd(Axn1, i, m)

Arguments

Axn1 Discrete term insurance APV.
 i Effective annual interest rate.
 m Positive integer payment frequency.

Value

m-thly term insurance APV under UDD.

axn_improved

Temporary annuity-immediate under mortality improvement

Description

Computes

$$a_{x:\overline{n}|} = \sum_{t=1}^n v^t {}_t p_x^{(\text{improved})}$$

using projected mortality rates.

Usage

axn_improved(x0, n, i, qx_base_vec, AAx_vec, base_year, issue_year)

Arguments

x0	Issue age.
n	Term in years.
i	Effective annual interest rate.
qx_base_vec	Base-year one-year death probabilities for ages x0, x0+1, . . . , x0+n-1.
AAx_vec	Mortality improvement factors for ages x0, x0+1, . . . , x0+n-1.
base_year	Base year.
issue_year	Issue year.

Value

Numeric scalar.

Axn_m	<i>m</i> -thly endowment insurance APV
-------	--

Description

Computes $A_{x:\overline{n}|}^{(m)} = A_{x:\overline{n}|}^{1(m)} + v^n {}_n p_x$.

Usage

Axn_m(x, n, i, m, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of APVs.

 Axn_m_udd

UDD approximation of m-thly endowment insurance

Description

Computes $A_{x:\overline{n}|}^{(m)} = (i/i^{(m)})A_{x:\overline{n}|}^1 + {}_nE_x$.

Usage

Axn_m_udd(Axn1, nEx, i, m)

Arguments

Axn1	Discrete term insurance APV.
nEx	Pure endowment APV.
i	Effective annual interest rate.
m	Positive integer payment frequency.

Value

m-thly endowment insurance APV under UDD.

 Axy

Joint-life whole life insurance

Description

Computes A_{xy} .

Usage

Axy(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Value

Numeric vector.

Examples

```
Axy(40, 50, i = 0.05, model = "uniform", omega = 100)
```

Axybar

Last-survivor whole life insurance

Description

Computes $A_{\overline{xy}} = A_x + A_y - A_{xy}$.

Usage

```
Axybar(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Value

Numeric vector.

Examples

```
Axybar(40, 50, i = 0.05, model = "uniform", omega = 100)
```

Axybarn

Last-survivor endowment insurance

Description

Computes $A_{\overline{xy:\overline{n}|}}$.

Usage

Axybarn(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

Axybarn(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)

Axybarn1

Last-survivor term insurance

Description

Computes $A_{\overline{xy:\overline{n}|}}^1$.

Usage

Axybarn1(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Axybarn1(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

 axybarn_ch12

Last-survivor temporary annuity-immediate

Description

Computes $a_{\overline{xy:\overline{n}}}$.

Usage

```
axybarn(x, y, n, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Details

Shared documentation topic used to avoid filename collisions with case-distinct function names on case-insensitive file systems.

Value

Numeric vector.

Examples

```
axybarn(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

 axybar_ch12

Last-survivor whole life annuity-immediate

Description

Computes $a_{\overline{xy}}$.

Usage

```
axybar(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Details

Shared documentation topic used to avoid filename collisions with case-distinct function names on case-insensitive file systems.

Value

Numeric vector.

Examples

```
axybar(40, 50, i = 0.05, model = "uniform", omega = 100)
```

Axyn *Joint-life endowment insurance*

Description

Computes $A_{xy:\overline{n}|}$.

Usage

Axyn(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

Axyn(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)

Axyn1 *Joint-life term insurance*

Description

Computes $A^1_{xy:\overline{n}|}$.

Usage

Axyn1(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Axyn1(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

axyn_ch12

Joint-life temporary annuity-immediate

Description

Computes $a_{xy:\overline{n}|}$.

Usage

```
axyn(x, y, n, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Details

Shared documentation topic used to avoid filename collisions with case-distinct function names on case-insensitive file systems.

Value

Numeric vector.

Examples

```
axy_n(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

axy_ch12

Joint-life whole life annuity-immediate

Description

Computes a_{xy} .

Usage

```
axy(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Details

Shared documentation topic used to avoid filename collisions with case-distinct function names on case-insensitive file systems.

Value

Numeric vector.

Examples

```
axy(40, 50, i = 0.05, model = "uniform", omega = 100)
```

ax_improved	<i>Whole life annuity-immediate under mortality improvement</i>
-------------	---

Description

Computes a truncated whole life annuity-immediate under projected mortality.

Usage

```
ax_improved(x0, i, qx_base_vec, AAx_vec, base_year, issue_year)
```

Arguments

x0	Issue age.
i	Effective annual interest rate.
qx_base_vec	Base-year one-year death probabilities for successive ages.
AAx_vec	Mortality improvement factors for successive ages.
base_year	Base year.
issue_year	Issue year.

Value

Numeric scalar.

Ax_m	<i>m-thly whole life insurance APV</i>
------	--

Description

Computes $A_x^{(m)} = \sum_{j=0}^{\infty} v^{(j+1)/m} \Pr(j/m < T_x \leq (j+1)/m)$.

Usage

```
Ax_m(x, i, m, model, ..., tol = 1e-12, j_max = 100000L)
```

Arguments

x	Age.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.
tol	Numerical tolerance for truncating the infinite sum.
j_max	Maximum number of m-thly intervals in the sum.

Value

Numeric vector of APVs.

Ax_m_udd

UDD approximation of m-thly whole life insurance

Description

Computes $A_x^{(m)} = (i/i^{(m)})A_x$.

Usage

Ax_m_udd(Ax, i, m)

Arguments

Ax	Discrete whole life insurance APV.
i	Effective annual interest rate.
m	Positive integer payment frequency.

Value

m-thly whole life insurance APV under UDD.

ax_y

Reversionary annuity to (y) after death of (x)

Description

Computes $a_{x|y} = a_y - a_{xy}$.

Usage

ax_y(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Value

Numeric vector.

Examples

```
ax_y(40, 50, i = 0.05, model = "uniform", omega = 100)
```

 ay_x

Reversionary annuity to (x) after death of (y)

Description

Computes $a_{y|x} = a_x - a_{xy}$.

Usage

```
ay_x(x, y, i, tbl = NULL, model = NULL, ..., k_max = 5000, tol = 1e-12)
```

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.
k_max	Maximum number of terms.
tol	Convergence tolerance.

Value

Numeric vector.

Examples

```
ay_x(40, 50, i = 0.05, model = "uniform", omega = 100)
```

betaF	<i>Full preliminary term renewal modified premium</i>
-------	---

Description

Computes the FPT renewal premium $\beta^F = P_{x+1}$ for whole life insurance.

Computes the FPT renewal premium $\beta^F = P_{x+1}$ for whole life insurance.

Usage

```
betaF(x, i, tbl = NULL, model = NULL, ...)
```

```
betaF(x, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Issue age.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
betaF(40, i = 0.05, model = "uniform", omega = 100)
betaF(40, i = 0.05, model = "uniform", omega = 100)
```

ch12_multilife_cont_params

Shared parameters for Chapter 12 continuous multi-life functions

Description

Shared parameters for Chapter 12 continuous multi-life functions

Arguments

x	Age of first life.
y	Age of second life.
i	Effective annual interest rate.
n	Term in years.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

 chapter15_spot_interest_apv

Spot-rate actuarial present value functions

Description

Chapter 15 functions for actuarial present values when discounting uses spot rates by maturity. If z_t denotes the annual effective spot rate for maturity t , then the discount factor is $(1 + z_t)^{-t}$.

Computes

$${}_nE_x = (1 + z_n)^{-n} \cdot {}_n p_x.$$

Computes

$$A_{x:\overline{n}|}^1 = \sum_{t=1}^n (1 + z_t)^{-t} \cdot {}_{t-1} p_x \cdot q_{x+t-1}.$$

Computes

$$A_{x:\overline{n}|} = A_{x:\overline{n}|}^1 + {}_nE_x$$

using spot-rate discount factors.

For an immediate annuity,

$$a_{x:\overline{n}|} = \sum_{t=1}^n (1 + z_t)^{-t} \cdot {}_t p_x.$$

Usage

```
nEx_spot(qx, z, benefit = 1)
```

```
Axn1_spot(qx, z, benefit = 1)
```

```
Axn_spot(qx, z, benefit = 1)
```

```
axn_spot(qx, z, type = c("immediate", "due"), benefit = 1)
```

Arguments

qx	Numeric vector of one-year mortality rates.
z	Numeric vector of annual effective spot rates for maturities 1, . . . , n.
benefit	Amount of each annuity payment.
type	Either "immediate" or "due".

Details

For an annuity-due,

$$\ddot{a}_{x:\overline{n}|} = \sum_{t=0}^{n-1} (1 + z_t)^{-t} \cdot {}_t p_x,$$

where the time-0 discount factor is 1.

Value

A numeric scalar.

A numeric scalar.

A numeric scalar.

A numeric scalar.

Examples

```
qx <- c(.02, .03, .04, .05, .06)
z <- c(.03, .04, .05, .06, .07)
nEx_spot(qx, z, benefit = 1000000)
```

```
qx <- c(.02, .03, .04, .05, .06)
z <- c(.03, .04, .05, .06, .07)
Axn1_spot(qx, z)
```

```
qx <- c(.02, .03, .04, .05, .06)
z <- c(.03, .04, .05, .06, .07)
Axn_spot(qx, z)
```

```
qx <- c(.02, .03, .04, .05, .06)
z <- c(.03, .04, .05, .06, .07)
axn_spot(qx, z, type = "due")
```

 chapter15_variable_interest_apv

Variable-interest actuarial present value functions

Description

Chapter 15 functions for actuarial present values under variable annual effective interest rates interpreted as a yearly scenario i_1, i_2, \dots, i_n .

Computes the APV of an n -year pure endowment under a variable annual interest scenario:

$${}_nE_x = v_n \cdot {}_n p_x.$$

Computes the APV of an n -year term insurance with benefit paid at the end of the year of death under a variable annual interest scenario:

$$A_{x:\overline{n}|}^1 = \sum_{t=1}^n v_t \cdot {}_{t-1} p_x \cdot q_{x+t-1}.$$

Computes the APV of an n -year endowment insurance under a variable annual interest scenario:

$$A_{x:\overline{n}|} = A_{x:\overline{n}|}^1 + {}_nE_x.$$

Computes the APV of an n -year temporary life annuity under a variable annual interest scenario.

Usage

```
nEx_var(qx, i, benefit = 1)
```

```
Axn1_var(qx, i, benefit = 1)
```

```
Axn_var(qx, i, benefit = 1)
```

```
axn_var(qx, i, type = c("immediate", "due"), benefit = 1)
```

Arguments

qx	Numeric vector of one-year mortality rates $q_x, q_{x+1}, \dots, q_{x+n-1}$.
i	Numeric vector of annual effective interest rates i_1, i_2, \dots, i_n .
benefit	Amount of each annuity payment.
type	Either "immediate" or "due".

Details

If a benefit amount is supplied, the function returns that benefit times the APV factor.

If a benefit amount is supplied, the function returns that benefit times the APV factor.

If a benefit amount is supplied, the function returns that benefit times the APV factor.

For an immediate annuity,

$$a_{x:\overline{n}|} = \sum_{t=1}^n v_t \cdot {}_t p_x.$$

For an annuity-due,

$$\ddot{a}_{x:\overline{n}|} = \sum_{t=0}^{n-1} v_t \cdot {}_t p_x,$$

with $v_0 = 1$.

Value

A numeric scalar.

A numeric scalar.

A numeric scalar.

A numeric scalar.

Examples

```
qx <- c(.03, .04, .05, .06, .07)
nEx_var(qx = qx, i = c(.06, .07, .08, .09, .10), benefit = 1000)

qx <- c(.03, .04, .05, .06, .07)
Axn1_var(qx = qx, i = c(.06, .07, .08, .09, .10))

qx <- c(.03, .04, .05, .06, .07)
Axn_var(qx = qx, i = c(.06, .07, .08, .09, .10))

qx <- rep(.02, 5)
axn_var(qx = qx, i = c(.06, .05, .04, .03, .03), type = "immediate")
axn_var(qx = qx, i = c(.03, .04, .05, .06, .07), type = "due")
```

coi_ul_typeB

Cost of insurance for Type B universal life

Description

Computes the one-period cost of insurance for a Type B universal life policy using Equation (16.5a):

$$COI_t = \frac{Bq_{x+t-1}}{1 + i^q}.$$

Usage

```
coi_ul_typeB(B, qx, iq)
```

Arguments

B	Face amount.
qx	Mortality rate for the period.
iq	Interest rate used in the cost-of-insurance calculation.

Value

Numeric vector.

Examples

```
coi_ul_typeB(B = 100000, qx = 0.00076, iq = 0.03)
```

contribution_rate_target

Target contribution rate for a defined contribution plan

Description

Solves Equation (18.5) for the contribution rate required to achieve a target replacement ratio.

Usage

```
contribution_rate_target(x, z, Sx, RR_target, i, adue_z, g = NULL, s = NULL)
```

Arguments

x	Entry age.
z	Retirement age.
Sx	Salary at age x.
RR_target	Target replacement ratio.
i	Annual effective interest rate.
adue_z	Whole life annuity-due factor at age z.
g	Optional constant annual salary growth rate.
s	Optional salary scale vector of length z - x.

Value

Required contribution rate.

Examples

```
contribution_rate_target(
  x = 30, z = 65, Sx = 60000, RR_target = 0.50,
  i = 0.06, adue_z = 11, g = 0.04
)
```

cov_term_deferred *Covariance of term and deferred insurance PVs*

Description

Computes $\text{Cov}(Z_{x:\overline{n}|}^1, {}_n|Z_x) = -A_{x:\overline{n}|}^1 \cdot {}_n|A_x$.

Usage

```
cov_term_deferred(x, n, i, tbl = NULL, model = NULL, ...)
```

Arguments

x Age.
n Term.
i Effective annual interest rate.
tbl Optional life table object.
model Optional parametric survival model name.
... Additional arguments passed to survival-model functions.

Value

Numeric vector of covariances.

cov_term_endow *Covariance of term insurance and pure endowment PVs*

Description

Computes $\text{Cov}(Z_{x:\overline{n}|}^1, Z_{x:\overline{n}|}^{1(\text{pure endow})}) = -A_{x:\overline{n}|}^1 \cdot {}_nE_x$.

Usage

```
cov_term_endow(x, n, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of covariances.

cumhaz0	<i>Cumulative hazard for age-at-failure T0</i>
---------	--

Description

Computes $\Lambda_0(t) = \int_0^t \lambda_0(y) dy$.

Usage

cumhaz0(t, model, ...)

Arguments

t	Numeric vector of times ($t \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale

Value

Numeric vector of cumulative hazard values.

DAbarxn1	<i>Piecewise-continuous decreasing n-year term insurance</i>
----------	--

Description

Computes

$$(\overline{DA})_{x:\overline{n}|}^1 = \int_0^n [n + 1 - t] v^t {}_t p_x \mu_{x+t} dt.$$

Usage

DAbarxn1(x, n, i, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

DAxn1	<i>Decreasing n-year term insurance</i>
-------	---

Description

Computes

$$(\overline{DA})_{x:\overline{n}|}^1 = \sum_{k=0}^{n-1} (n - k) v^{k+1} \Pr(K_x = k).$$

Usage

DAxn1(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

DbarAbarxn1

Fully continuous decreasing n-year term insurance

Description

Computes

$$(\bar{D}\bar{A})_{x:\bar{n}|}^1 = \int_0^n (n-t) v^t {}_t p_x \mu_{x+t} dt.$$

Usage

DbarAbarxn1(x, n, i, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

decompGg_disc

Ordered gross gain decomposition

Description

Decomposes total gross gain into interest, mortality, and expense components in a user-specified order.

Decomposes total gross gain into interest, mortality, and expense components in a user-specified order. The first two components are computed sequentially, and the last component is taken as the balancing item so that the components sum exactly to total gain.

Usage

```

decompGg_disc(
  VtG,
  Vt1G,
  G,
  i_assumed,
  q_assumed,
  r_assumed = 0,
  e_assumed = 0,
  s_assumed = 0,
  i_actual,
  q_actual,
  r_actual = 0,
  e_actual = 0,
  s_actual = 0,
  b = 1,
  order = c("interest", "mortality", "expense")
)

```

```

decompGg_disc(
  VtG,
  Vt1G,
  G,
  i_assumed,
  q_assumed,
  r_assumed = 0,
  e_assumed = 0,
  s_assumed = 0,
  i_actual,
  q_actual,
  r_actual = 0,
  e_actual = 0,
  s_actual = 0,
  b = 1,
  order = c("interest", "mortality", "expense")
)

```

Arguments

VtG	Gross reserve at duration t.
Vt1G	Gross reserve at duration t+1.
G	Gross premium.
i_assumed	Assumed annual effective interest rate.
q_assumed	Assumed mortality rate.
r_assumed	Assumed percent-of-premium expense rate.
e_assumed	Assumed per-policy expense.

s_assumed	Assumed settlement expense.
i_actual	Actual annual effective interest rate.
q_actual	Actual mortality rate.
r_actual	Actual percent-of-premium expense rate.
e_actual	Actual per-policy expense.
s_actual	Actual settlement expense.
b	Benefit amount. Default 1.
order	Character vector giving the order of decomposition.

Value

Named numeric vector.

Named numeric vector.

Examples

```
decompGg_disc(
  VtG = 3950.73, Vt1G = 4607.07, G = 685,
  i_assumed = 0.06, q_assumed = 0.00592,
  r_assumed = 0.05, e_assumed = 0, s_assumed = 300,
  i_actual = 0.065, q_actual = 0.005,
  r_actual = 0.06, e_actual = 0, s_actual = 100,
  b = 50000,
  order = c("interest", "mortality", "expense")
)
decompGg_disc(
  VtG = 3950.73, Vt1G = 4607.07, G = 685,
  i_assumed = 0.06, q_assumed = 0.00592,
  r_assumed = 0.05, e_assumed = 0, s_assumed = 300,
  i_actual = 0.065, q_actual = 0.005,
  r_actual = 0.06, e_actual = 0, s_actual = 100,
  b = 50000,
  order = c("interest", "mortality", "expense")
)
```

discount

Discount factor for compound interest

Description

Discount factor for compound interest

Usage

discount(i, t)

Arguments

i Effective interest rate.
t Time (can be vector).

Value

Discount factor at time *t*.

Examples

```
discount(0.05, 0:5)
```

discounted_payback_period
Discounted payback period

Description

Returns the first duration *t* for which the partial net present value $NPV(t)$ is nonnegative.

Usage

```
discounted_payback_period(Pi, r)
```

Arguments

Pi Profit signature vector.
r Risk discount rate.

Value

Integer scalar, or *NA_integer_* if the payback period is not reached.

Examples

```
Pi <- c(-15.00, 8.42, 8.39, 8.58)  
discounted_payback_period(Pi, r = 0.10)
```

dist0	<i>Distribution functions for age-at-failure T_0</i>
-------	---

Description

Convenience functions for the CDF and density of T_0 .

Usage

$F_0(t, \text{model}, \dots)$

$f_0(t, \text{model}, \dots)$

Arguments

t	Numeric vector of times ($t \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale

Details

- $F_0(t)$ computes $F_0(t) = Pr(T_0 \leq t) = 1 - S_0(t)$
- $f_0(t)$ computes $f_0(t) = \frac{d}{dt} F_0(t)$

Value

Numeric vector. For F_0 : CDF values in $[0, 1]$. For f_0 : density values (≥ 0).

double_force_delta	<i>Doubled force of interest</i>
--------------------	----------------------------------

Description

Computes $\delta' = 2\delta$.

Usage

double_force_delta(delta)

Arguments

delta Numeric vector of forces of interest.

Value

Numeric vector of doubled forces of interest.

double_force_i	<i>Effective annual interest at doubled force</i>
----------------	---

Description

If $\delta' = 2\delta$, then $i' = (1 + i)^2 - 1$.

Usage

double_force_i(i)

Arguments

i Numeric vector of effective annual interest rates.

Value

Numeric vector of effective annual rates corresponding to doubled force.

dx	<i>Compute deaths between ages x and x+1</i>
----	--

Description

Compute deaths between ages x and x+1

Usage

dx(tbl, x)

Arguments

tbl A life_table object.
x Ages.

Value

Numeric vector of d_x values.

dxj	<i>Cause-specific decrements $d_x^{(j)}$</i>
-----	---

Description

Cause-specific decrements $d_x^{(j)}$

Usage

dxj(lxtau, qxj)

Arguments

lxtau	Number alive at age x in the multiple-decrement table.
qxj	Numeric vector of cause-specific decrement probabilities.

Value

Numeric vector.

Examples

dxj(1000, c(0.011, 0.100))

dxtau	<i>Total decrements $d_x^{(\tau)}$</i>
-------	---

Description

Total decrements $d_x^{(\tau)}$

Usage

dxtau(lxtau, qxj)

Arguments

lxtau	Number alive at age x in the multiple-decrement table.
qxj	Numeric vector of cause-specific decrement probabilities.

Value

Numeric scalar.

Examples

dxtau(1000, c(0.011, 0.100))

Arguments

x	Numeric vector of ages ($x \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale
tol	Tolerance used to choose a finite integration bound (numeric).

Value

Numeric vector of complete expectations.

ex_complete_tab	<i>Complete expectation of life from a life table</i>
-----------------	---

Description

Computes ${}^{\circ}e_x = \int_0^{\infty} {}_t p_x dt$ using a within-year assumption: "udd", "cf", or "balducci".

Usage

```
ex_complete_tab(tbl, x, assumption = c("udd", "cf", "balducci"))
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
assumption	One of "udd", "cf", "balducci".

Value

Numeric vector of complete expectations ${}^{\circ}e_x$.

ex_curtate	<i>Curtate expectation of life at age x</i>
------------	---

Description

Computes $e_x = E[K_x] = \sum_{k=1}^{\infty} k p_x$ with truncation.

Usage

```
ex_curtate(x, model, ..., k_max = 5000, tol = 1e-12)
```

Arguments

x	Numeric vector of ages ($x \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale
k_max	Maximum integer duration to sum to.
tol	Stop early if the summand is < tol for several steps.

Value

Numeric vector of curtate expectations.

ex_curtate_tab	<i>Curtate expectation of life from a life table</i>
----------------	--

Description

Computes the curtate expectation of life $e_x = \sum_{k=1}^{\infty} k p_x$ in the discrete tabular setting.

Usage

```
ex_curtate_tab(tbl, x)
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.

Value

Numeric vector of curtate expectations e_x .

ex_temp_complete_tab *Temporary complete expectation of life from a life table*

Description

Computes ${}^{\circ}e_{x:\overline{n}|} = \int_0^n {}_t p_x dt$ using a within-year assumption: "udd", "cf", or "balducci".

Usage

```
ex_temp_complete_tab(tbl, x, n, assumption = c("udd", "cf", "balducci"))
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
n	Numeric vector of nonnegative numbers.
assumption	One of "udd", "cf", "balducci".

Value

Numeric vector of temporary complete expectations.

ex_temp_curtate_tab *Temporary curtate expectation of life from a life table*

Description

Computes $e_{x:\overline{n}|} = \sum_{k=1}^n {}_k p_x$ for integer n in the discrete tabular setting.

Usage

```
ex_temp_curtate_tab(tbl, x, n)
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
n	Numeric vector of nonnegative integers.

Value

Numeric vector of temporary curtate expectations.

fnk_from_z *Forward rate $f_{n,k}$ from spot rates*

Description

Computes the n -year forward k -year annual effective rate implied by annual effective spot rates:

$$(1 + z_{n+k})^{n+k} = (1 + z_n)^n (1 + f_{n,k})^k.$$

Usage

```
fnk_from_z(z, n, k)
```

Arguments

z	Numeric vector of annual effective spot rates.
n	Forward start in years.
k	Forward maturity in years.

Details

The input vector z should contain annual effective spot rates for maturities 1, 2, ..., length(z).

Value

A numeric scalar.

Examples

```
z <- c(0.03, 0.04, 0.05, 0.06, 0.07)
fnk_from_z(z, n = 1, k = 4)
fnk_from_z(z, n = 2, k = 2)
```

forward_matrix_from_z *Matrix of all determinable forward rates from spot rates*

Description

Constructs an upper-left triangular matrix of annual effective forward rates $f_{n,k}$ implied by annual effective spot rates z_1, \dots, z_m .

Usage

```
forward_matrix_from_z(z)
```

Arguments

`z` Numeric vector of annual effective spot rates.

Details

Rows correspond to $n = 1, \dots, m - 1$ and columns correspond to $k = 1, \dots, m - 1$. Entries that are not determinable are returned as NA.

Value

A numeric matrix.

Examples

```
z <- c(0.03, 0.04, 0.05, 0.06, 0.07)
forward_matrix_from_z(z)
```

<code>fx</code>	<i>Conditional density for T_x</i>
-----------------	---

Description

Computes $f_x(t) = f_0(x + t)/S_0(x)$.

Usage

```
fx(t, x, model, ...)
```

Arguments

`t` Numeric vector of durations ($t \geq 0$).

`x` Numeric vector of ages ($x \geq 0$).

`model` One of "uniform", "exponential", "gompertz", "makeham", "weibull".

`...` Model parameters:

- uniform: omega
- exponential: lambda
- gompertz: B, c
- makeham: A, B, c
- weibull: shape, scale

Details

Vectorization rule: - If `t` and `x` are the same length, values are computed elementwise. - If one of `t` or `x` has length 1, it is recycled to match the other.

Value

Numeric vector of densities (≥ 0).

fx_tab	<i>Fractional conditional density from a life table</i>
--------	---

Description

Computes the conditional density $f_x(t | T_0 > x) = {}_t p_x \mu_{x+t}$ for $0 < t < 1$.

Usage

```
fx_tab(tbl, x, t, assumption = c("udd", "cf", "balducci"))
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
t	Numeric vector of fractional durations with $0 \leq t \leq 1$.
assumption	One of "udd", "cf", "balducci".

Value

Numeric vector of conditional density values.

gain_loss_md	<i>Gain or loss in a multiple-decrement model</i>
--------------	---

Description

Computes the gain or loss expression from Section 14.6.

Usage

```
gain_loss_md(
  Vt,
  G,
  r,
  e,
  i,
  b1,
  b2,
  s1 = 0,
  s2 = 0,
```

```

    q1,
    q2,
    Vt1,
    year_end_cause2 = FALSE,
    q1prime = NULL,
    q2prime = NULL
)

```

Arguments

Vt	Gross premium reserve at time t .
G	Gross premium for the year.
r	Percent-of-premium expense factor.
e	Fixed expense at the beginning of the year.
i	Earned interest rate.
b1	Cause 1 benefit.
b2	Cause 2 benefit.
s1	Claim settlement expense for Cause 1.
s2	Claim settlement expense for Cause 2.
q1	Cause 1 decrement probability.
q2	Cause 2 decrement probability.
Vt1	Gross premium reserve at time $t + 1$.
year_end_cause2	Logical; if TRUE, use the year-end Cause 2 form.
q1prime	Single-decrement Cause 1 probability for the year-end Cause 2 case.
q2prime	Single-decrement Cause 2 probability for the year-end Cause 2 case.

Details

With within-year decrement probabilities, the function evaluates

$$[{}_tV^G + G(1 - r) - e](1 + i) - \left[(b^{(1)} + s^{(1)})q^{(1)} + (b^{(2)} + s^{(2)})q^{(2)} + p^{(\tau)} {}_{t+1}V^G \right]$$

If year_end_cause2 = TRUE, the Cause 2 decrement is treated as occurring only at year end, matching Equation (14.30).

Value

A numeric scalar.

Examples

```

gain_loss_md(
  Vt = 115.00, G = 16, r = 0, e = 3, i = 0.06,
  b1 = 1000, b2 = 110, s1 = 0, s2 = 0,
  q1 = 0.01, q2 = 0.10, Vt1 = 128.83
)

```


Arguments

Vt	Reserve at duration t.
Vt1	Reserve at duration t+1.
P	Net premium for the year.
i_actual	Actual annual effective interest rate.
q_assumed	Assumed mortality rate for the year.
B	Benefit amount. Defaults to 1.

Value

Numeric vector.

Examples

GI_disc(Vt = 0.1, Vt1 = 0.11, P = 0.02, i_actual = 0.05, q_assumed = 0.01)

GMF_rollforward_ul *Guaranteed maturity fund roll-forward*

Description

Computes the one-period guaranteed maturity fund roll-forward used in Example 16.9.

Usage

GMF_rollforward_ul(GMF_prev, GMP, r, policy_charge, i)

Arguments

GMF_prev	Prior guaranteed maturity fund.
GMP	Guaranteed maturity premium.
r	Expense factor applied to GMP.
policy_charge	Guaranteed policy charge.
i	Guaranteed interest rate.

Value

Numeric scalar.

Examples

GMF_rollforward_ul(140.40, 14.49, 0.04, 11.80, 0.03)

GM_cont *Mortality gain helper for continuous-style recursion*

Description

Mortality gain helper for continuous-style recursion

Usage

GM_cont(Vt, Vt1, P, delta_assumed, p_actual, benefit = 0, h = 1)

Arguments

Vt	Reserve at time t.
Vt1	Reserve at time t+h.
P	Premium rate.
delta_assumed	Assumed force of interest.
p_actual	Actual survival probability over the step.
benefit	Benefit paid at start of step. Default 0.
h	Step length. Default 1.

Value

Numeric vector.

Examples

GM_cont(Vt = 10, Vt1 = 11, P = 1, delta_assumed = 0.05, p_actual = 0.99)

GM_disc *Mortality gain for a discrete insurance contract*

Description

Mortality gain for a discrete insurance contract

Usage

GM_disc(Vt, Vt1, P, i_assumed, q_actual, B = 1)

Arguments

Vt	Reserve at duration t.
Vt1	Reserve at duration t+1.
P	Net premium for the year.
i_assumed	Assumed annual effective interest rate.
q_actual	Actual mortality rate for the year.
B	Benefit amount. Defaults to 1.

Value

Numeric vector.

Examples

```
GM_disc(Vt = 0.1, Vt1 = 0.11, P = 0.02, i_assumed = 0.04, q_actual = 0.01)
```

GTg_disc	<i>Total gross gain for a discrete insurance contract</i>
----------	---

Description

Computes the Chapter 11 total gain under gross premiums and gross reserves.

Computes the Chapter 11 total gain under gross premiums and gross reserves.

Usage

```
GTg_disc(
  VtG,
  Vt1G,
  G,
  i_actual,
  q_actual,
  r_actual = 0,
  e_actual = 0,
  s_actual = 0,
  b = 1
)
```

```
GTg_disc(
  VtG,
  Vt1G,
  G,
  i_actual,
  q_actual,
  r_actual = 0,
```

```

    e_actual = 0,
    s_actual = 0,
    b = 1
)

```

Arguments

VtG	Gross reserve at duration t.
Vt1G	Gross reserve at duration t+1.
G	Gross premium.
i_actual	Actual annual effective interest rate.
q_actual	Actual mortality rate.
r_actual	Actual percent-of-premium expense rate.
e_actual	Actual per-policy expense.
s_actual	Actual settlement expense.
b	Benefit amount. Default 1.

Value

Numeric vector.
 Numeric vector.

Examples

```

GTg_disc(
  VtG = 0.10, Vt1G = 0.12, G = 0.02,
  i_actual = 0.05, q_actual = 0.01,
  r_actual = 0.03, e_actual = 0, s_actual = 0.01, b = 1
)
GTg_disc(
  VtG = 0.10, Vt1G = 0.12, G = 0.02,
  i_actual = 0.05, q_actual = 0.01,
  r_actual = 0.03, e_actual = 0, s_actual = 0.01, b = 1
)

```

 GT_cont

Total gain for a continuous-style one-step recursion

Description

Total gain for a continuous-style one-step recursion

Usage

```
GT_cont(Vt, Vt1, P, delta_actual, p_actual, benefit = 0, h = 1)
```

Arguments

Vt	Reserve at time t.
Vt1	Reserve at time t+h.
P	Premium rate.
delta_actual	Actual force of interest.
p_actual	Actual survival probability over the step.
benefit	Benefit paid at start of step. Default 0.
h	Step length. Default 1.

Value

Numeric vector.

Examples

```
GT_cont(Vt = 10, Vt1 = 11, P = 1, delta_actual = 0.05, p_actual = 0.99)
```

GT_disc	<i>Total gain for a discrete insurance contract</i>
---------	---

Description

Computes the Chapter 10 total gain: amount on hand at year-end minus amount required.

Usage

```
GT_disc(Vt, Vt1, P, i_actual, q_actual, B = 1)
```

Arguments

Vt	Reserve at duration t.
Vt1	Reserve at duration t+1.
P	Net premium for the year.
i_actual	Actual annual effective interest rate.
q_actual	Actual mortality rate for the year.
B	Benefit amount. Defaults to 1.

Value

Numeric vector.

Examples

```
GT_disc(Vt = 0.1, Vt1 = 0.11, P = 0.02, i_actual = 0.05, q_actual = 0.01)
```

hazard0	<i>Hazard / force for age-at-failure T0</i>
---------	---

Description

Computes $\lambda_0(t) = f_0(t)/S_0(t)$.

Usage

hazard0(t, model, ...)

Arguments

t	Numeric vector of times ($t \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale

Value

Numeric vector of hazard/force values.

htVx	<i>h-pay whole life net level premium reserve</i>
------	---

Description

Computes the Chapter 10 prospective reserve for an h-pay whole life policy.

Usage

htVx(x, h, t, i, model, ...)

Arguments

x	Issue age.
h	Premium-paying period in years.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
htVx(40, h = 10, t = 5, i = 0.05, model = "uniform", omega = 100)
```

IAbarx

*Piecewise-continuous increasing whole life insurance***Description**

Computes

$$(I\bar{A})_x = \int_0^{\infty} [t + 1] v^t {}_t p_x \mu_{x+t} dt.$$

Usage

```
IAbarx(x, i, model, ...)
```

Arguments

x	Age.
i	Effective annual interest rate.
model	Parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

IAx

*Increasing whole life insurance***Description**

Computes

$$(IA)_x = \sum_{k=0}^{\infty} (k + 1) v^{k+1} \Pr(K_x = k).$$

Usage

```
IAx(x, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)
```


Arguments

x	Age.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.
tol	Numerical tolerance for truncation.
k_max	Maximum number of terms.

Value

Numeric vector.

IAxn1	<i>Increasing n-year term insurance</i>
-------	---

Description

Computes

$$(IA)_{x:\overline{n}|}^1 = \sum_{k=0}^{n-1} (k+1)v^{k+1} \Pr(K_x = k).$$

Usage

IAxn1(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

IbarAbarx *Fully continuous increasing whole life insurance*

Description

Computes

$$(\bar{I}\bar{A})_x = \int_0^{\infty} t v^t {}_t p_x \mu_{x+t} dt.$$

Usage

IbarAbarx(x, i, model, ...)

Arguments

x	Age.
i	Effective annual interest rate.
model	Parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

IbarAbarxn1 *Fully continuous increasing n-year term insurance*

Description

Computes

$$(\bar{I}\bar{A})_{x:\overline{n}|}^1 = \int_0^n t v^t {}_t p_x \mu_{x+t} dt.$$

Usage

IbarAbarxn1(x, n, i, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

iMA_eiul	<i>Monthly-average index growth rate</i>
----------	--

Description

Computes the monthly-average index growth rate from an initial index value and 12 monthly closing values, matching Equation (16.13).

Usage

```
iMA_eiul(index)
```

Arguments

index	Numeric vector of length 13 containing the initial index value followed by the 12 monthly closing values.
-------	---

Value

Numeric scalar.

Examples

```
idx <- c(1000, 1020, 1100, 1150, 1080, 1040, 960, 1030, 1000, 1070, 1150, 1200, 1150)
iMA_eiul(idx)
```

Income_dc	<i>Retirement income from a defined contribution accumulation</i>
-----------	---

Description

Converts a defined contribution accumulation to annual annuity-due income.

Usage

```
Income_dc(AVz, adue_z)
```

Arguments

AVz	Accumulated value at retirement.
adue_z	Whole life annuity-due factor at retirement age.

Value

Annual retirement income.

Examples

```
Income_dc(AVz = 824211.35, adue_z = 12)
```

interest_convert *Convert between compound-interest quantities*

Description

Provides consistent conversions between: - effective interest rate i - effective discount rate d - force of interest δ

Usage

```
interest_convert(i = NULL, d = NULL, delta = NULL, m = NULL)
```

Arguments

i	Effective interest rate.
d	Effective discount rate.
δ	Force of interest.
m	Optional compounding frequency for the nominal rate convertible m -thly.

Details

Exactly one of i , d , or δ must be provided.

Value

A list with elements i , d , δ and, if m is supplied, im (the nominal rate convertible m -thly).

Examples

```
interest_convert(i = 0.05)
interest_convert(d = 0.04761905)
interest_convert(delta = log(1.05))
```

iP_eiul	<i>Point-to-point index growth rates</i>
---------	--

Description

Computes annual point-to-point index growth rates from successive index values, matching Equation (16.12).

Usage

```
iP_eiul(index)
```

Arguments

index	Numeric vector of index closing values.
-------	---

Value

Numeric vector of growth rates.

Examples

```
iP_eiul(c(1000, 1050, 1200, 1100))
```

IRR_profit	<i>Internal rate of return of a profit signature</i>
------------	--

Description

Computes the internal rate of return (IRR), defined as the rate r for which the net present value is zero.

Usage

```
IRR_profit(Pi, interval = c(0, 1), tol = .Machine$double.eps^0.5)
```

Arguments

Pi	Profit signature vector.
interval	Numeric vector of length 2 giving the search interval for <code>uniroot()</code> .
tol	Tolerance passed to <code>uniroot()</code> .

Value

Numeric scalar.

Examples

```
Pi <- c(-15.00, 8.42, 8.39, 8.58)
IRR_profit(Pi)
```

<i>i_credit_eiul</i>	<i>Credited rates from raw index growth rates</i>
----------------------	---

Description

Applies participation, floor, cap, and optional margin to raw index growth rates for an indexed universal life contract.

Usage

```
i_credit_eiul(
  i_raw,
  part = 1,
  floor = 0,
  cap = Inf,
  margin = 0,
  margin_after_participation = TRUE
)
```

Arguments

<i>i_raw</i>	Numeric vector of raw index growth rates.
<i>part</i>	Participation rate.
<i>floor</i>	Minimum credited rate.
<i>cap</i>	Maximum credited rate.
<i>margin</i>	Index margin. Defaults to 0.
<i>margin_after_participation</i>	Logical; if TRUE, subtract the margin after applying the participation rate.

Value

Numeric vector of credited rates.

Examples

```
raw <- iP_eiul(c(1000, 1050, 1200, 1100))
i_credit_eiul(raw, part = 1.10, floor = 0.01, cap = 0.10)
```

life_table	<i>Construct a life table</i>
------------	-------------------------------

Description

Build a discrete life table from one of lx, qx, px, or S0.

Usage

```
life_table(x, lx = NULL, qx = NULL, px = NULL, S0 = NULL, radix = 1e+05)
```

Arguments

x	Numeric vector of ages.
lx	Numeric vector of l_x values.
qx	Numeric vector of q_x values.
px	Numeric vector of p_x values.
S0	Numeric vector of S_0(x) values.
radix	Radix used when converting S0 to lx, or when building from qx/px.

Value

A data.frame with class "life_table".

lx	<i>Extract life-table survivor values</i>
----	---

Description

Extract life-table survivor values

Usage

```
lx(tbl, x)
```

Arguments

tbl	A life_table object.
x	Ages.

Value

Numeric vector of l_x values.

lx_select	<i>Extract select-table survivor value</i>
-----------	--

Description

Returns $l_{[x]+t}$ from a select life table.

Usage

```
lx_select(tbl, x_sel, t)
```

Arguments

tbl	A select_life_table object.
x_sel	Numeric vector of ages at selection.
t	Numeric vector of durations since selection.

Value

Numeric vector of survivor values.

lx_to_S0	<i>Convert life-table values to survival probabilities</i>
----------	--

Description

Converts Chapter 6 life-table values l_x into Chapter 5 survival probabilities $S_0(x) = l_x/l_0$.

Usage

```
lx_to_S0(lx)
```

Arguments

lx	Numeric vector of life-table survivor values.
----	---

Value

Numeric vector of $S_0(x)$ values.

markov_nstep_prob	<i>n-step transition probability for a discrete-time Markov chain</i>
-------------------	---

Description

Computes the (i, j) entry of P^n , useful for Chapter 14 examples involving discrete-time multi-state models such as CCRC and risk-class models.

Usage

```
markov_nstep_prob(P, n, i, j)
```

Arguments

P	Transition probability matrix.
n	Nonnegative integer number of steps.
i	Starting state index.
j	Ending state index.

Value

A numeric scalar.

Examples

```
P <- matrix(
  c(0.94, 0.03, 0.02, 0.01,
    0.50, 0.30, 0.18, 0.02,
    0.00, 0.00, 0.93, 0.07,
    0.00, 0.00, 0.00, 1.00),
  nrow = 4, byrow = TRUE
)

markov_nstep_prob(P, n = 3, i = 1, j = 1)
markov_nstep_prob(P, n = 3, i = 1, j = 3)
```

md_table	<i>Build a multiple-decrement table</i>
----------	---

Description

Build a multiple-decrement table

Usage

```
md_table(x, qxj, radix = 1e+05)
```

Arguments

x Integer vector of ages or durations.

qxj Matrix/data.frame of cause-specific decrement probabilities. Rows correspond to ages in **x**, columns correspond to causes.

radix Starting $l_x^{(\tau)}$.

Value

Data frame containing $q^{(j)}$, $q^{(\tau)}$, $p^{(\tau)}$, $l^{(\tau)}$, $d^{(j)}$, and $d^{(\tau)}$.

Examples

```
x <- 45:50
qmat <- cbind(
  q1 = c(.011, .012, .013, .014, .015, .016),
  q2 = c(.100, .100, .100, .100, .100, .100)
)
md_table(x, qmat, radix = 1000)
```

meanVx

Mean reserve for whole life insurance

Description

Computes the mean reserve ${}_{t+1/2}V$.

Computes the mean reserve ${}_{t+1/2}V$.

Usage

```
meanVx(x, t, i, tbl = NULL, model = NULL, ...)
```

```
meanVx(x, t, i, tbl = NULL, model = NULL, ...)
```

Arguments

x Issue age.

t Integer contract duration.

i Effective annual interest rate.

tbl Optional life table object.

model Optional parametric survival model.

... Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
meanVx(40, t = 10, i = 0.05, model = "uniform", omega = 100)
```

```
meanVx(40, t = 10, i = 0.05, model = "uniform", omega = 100)
```

mortality_improvement *Mortality improvement projection functions (Chapter 8)*

Description

Functions for Chapter 8 Section 8.8 mortality improvement projection.

Details

These functions project one-year death and survival probabilities forward from a base year and evaluate annuity values under projected mortality.

The standard projection used is

$$q_x^{[Y]} = q_x^{[B]}(1 - AA_x)^{Y-B}$$

where:

- B is the base year,
- Y is the projection year,
- AA_x is the mortality improvement factor at age x .

multilife_ch12 *Multi-life functions for Chapter 12*

Description

Core Chapter 12 functions for two-life joint-life and last-survivor models, including survival probabilities, annuities, insurances, and reversionary annuities.

multilife_ch12_ext *Extended multi-life functions for Chapter 12*

Description

Additional Chapter 12 functions for contingent probabilities, continuous contingent insurances, and continuous multi-life annuities.

mux_tab	<i>Fractional force of mortality from a life table</i>
---------	--

Description

Computes μ_{x+t} under UDD, constant force, or Balducci.

Usage

```
mux_tab(tbl, x, t, assumption = c("udd", "cf", "balducci"))
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
t	Numeric vector of fractional durations with $0 \leq t \leq 1$.
assumption	One of "udd", "cf", "balducci".

Value

Numeric vector of μ_{x+t} values.

nAbarx	<i>Continuous deferred insurance APV</i>
--------	--

Description

Computes ${}_n|\bar{A}_x = v^n {}_n p_x \bar{A}_{x+n}$.

Usage

```
nAbarx(x, n, i, model, ...)
```

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of APVs.

nAbarx_udd	<i>UDD approximation of continuous deferred insurance</i>
------------	---

Description

Computes ${}_n\bar{A}_x = (i/\delta) {}_nA_x$.

Usage

nAbarx_udd(nAx, i)

Arguments

nAx	Discrete deferred insurance APV.
i	Effective annual interest rate.

Value

Continuous deferred insurance APV under UDD.

nAx	<i>Deferred insurance APV</i>
-----	-------------------------------

Description

Computes ${}_nA_x = {}_nE_x \cdot A_{x+n}$.

Usage

nAx(x, n, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.
tol	Numerical tolerance for truncating infinite sums.
k_max	Maximum number of terms in the sum.

Value

Numeric vector of APVs.

naxn_improved	<i>Deferred temporary annuity-immediate under mortality improvement</i>
---------------	---

Description

Computes

$${}_u|a_{x:\bar{n}} = \sum_{t=u+1}^{u+n} v^t {}_tP_x^{(\text{improved})}$$

Usage

naxn_improved(x0, u, n, i, qx_base_vec, AAx_vec, base_year, issue_year)

Arguments

x0	Issue age.
u	Deferral period in years.
n	Temporary payment period in years.
i	Effective annual interest rate.
qx_base_vec	Base-year one-year death probabilities for ages x0, x0+1, . . . , x0+u+n-1.
AAx_vec	Mortality improvement factors for ages x0, x0+1, . . . , x0+u+n-1.
base_year	Base year.
issue_year	Issue year.

Value

Numeric scalar.

nAx_m	<i>m-thly deferred insurance APV</i>
-------	--------------------------------------

Description

Computes ${}_n|A_x^{(m)} = v^n {}_n p_x A_{x+n}^{(m)}$.

Usage

nAx_m(x, n, i, m, model, . . . , tol = 1e-12, j_max = 100000L)

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.
tol	Numerical tolerance for truncating the infinite sum.
j_max	Maximum number of m-thly intervals in the sum.

Value

Numeric vector of APVs.

nAx_m_udd

UDD approximation of m-thly deferred insurance

Description

Computes ${}_n|A_x^{(m)} = (i/i^{(m)}) {}_n|A_x$.

Usage

nAx_m_udd(nAx, i, m)

Arguments

nAx	Discrete deferred insurance APV.
i	Effective annual interest rate.
m	Positive integer payment frequency.

Value

m-thly deferred insurance APV under UDD.

NC_EAN_db *Entry Age Normal normal cost for a DB plan*

Description

Computes Equation (18.13).

Usage

NC_EAN_db(APV_total, adue_active)

Arguments

APV_total Total actuarial present value of benefits.
 adue_active Active-service annuity-due factor.

Value

Entry Age Normal normal cost.

Examples

NC_EAN_db(APV_total = 25000, adue_active = 15)

NC_PUC_db *Projected Unit Credit normal cost for a DB plan*

Description

Computes the PUC normal cost as the APV of the portion of the projected benefit attributed to the current year of service.

Usage

NC_PUC_db(projected_benefit, total_service, v_to_ret, p_surv, adue_ret)

Arguments

projected_benefit Projected benefit at retirement.
 total_service Total service at retirement.
 v_to_ret Discount factor to retirement.
 p_surv Active-service survival probability to retirement.
 adue_ret Retirement annuity factor.

Value

PUC normal cost.

Examples

```
NC_PUC_db(
  projected_benefit = 30000,
  total_service = 30,
  v_to_ret = 0.5,
  p_surv = 0.9,
  adue_ret = 12
)
```

 NC_TUC_db

Traditional Unit Credit normal cost for a DB plan

Description

Computes the TUC normal cost as the APV of the current year's accrual.

Usage

```
NC_TUC_db(accrual_benefit, v_to_ret, p_surv, adue_ret)
```

Arguments

accrual_benefit	Benefit accrued in the current year.
v_to_ret	Discount factor to retirement.
p_surv	Active-service survival probability to retirement.
adue_ret	Retirement annuity factor.

Value

TUC normal cost.

Examples

```
NC_TUC_db(accrual_benefit = 1560, v_to_ret = 0.5, p_surv = 0.9, adue_ret = 12)
```

ndx	<i>Compute deaths over an n-year interval from a life table</i>
-----	---

Description

Compute deaths over an n-year interval from a life table

Usage

ndx(tbl, x, n)

Arguments

tbl	A life_table object.
x	Ages.
n	Nonnegative integer durations.

Value

Numeric vector of ${}_n d_x$ values.

nEx	<i>Pure endowment APV</i>
-----	---------------------------

Description

Computes ${}_n E_x = v^n {}_n p_x$.

Usage

nEx(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of APVs.

nExy	<i>Joint-life pure endowment</i>
------	----------------------------------

Description

Computes ${}_nE_{xy} = v^n {}_n p_{xy}$.

Usage

nExy(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

nExy(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)

nExybar	<i>Last-survivor pure endowment</i>
---------	-------------------------------------

Description

Computes ${}_nE_{\overline{xy}} = v^n {}_n p_{\overline{xy}}$.

Usage

nExybar(x, y, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term.
i	Effective annual interest rate.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
nExybar(40, 50, n = 10, i = 0.05, model = "uniform", omega = 100)
```

 nkqx

Curtate death probability from a life table

Description

Computes ${}_k|q_x = {}_k p_x - {}_{k+1} p_x$

Usage

```
nkqx(tbl, x, k)
```

Arguments

tbl	A life_table object.
x	Numeric vector of ages.
k	Nonnegative integer.

Value

Numeric vector of ${}_k|q_x$ values.

nmxq	<i>Deferred death probability from a life table</i>
------	---

Description

Computes the probability that a life aged x survives n years and then dies within the following m years: ${}_n|{}_m q_x = {}_n p_x \cdot {}_m q_{x+n}$

Usage

```
nmxq(tbl, x, n, m)
```

Arguments

tbl	A life_table object.
x	Numeric vector of ages.
n	Nonnegative integer deferred period.
m	Nonnegative integer subsequent period.

Details

This function is for integer n and m in the discrete tabular setting.

Value

Numeric vector of ${}_n|{}_m q_x$ values.

nmxq_select	<i>Deferred select-life death probability</i>
-------------	---

Description

Computes ${}_n|{}_m q_{[x]+t} = {}_n p_{[x]+t} \cdot {}_m q_{[x]+t+n}$.

Usage

```
nmxq_select(tbl, x_sel, t, n, m)
```

Arguments

tbl	A select_life_table object.
x_sel	Numeric vector of ages at selection.
t	Numeric vector of current durations since selection.
n	Numeric vector of nonnegative integer deferred periods.
m	Numeric vector of nonnegative integer death windows.

Value

Numeric vector of deferred death probabilities.

NPV_partial	<i>Partial net present values</i>
-------------	-----------------------------------

Description

Computes the sequence $NPV(0), NPV(1), \dots, NPV(n)$ of partial net present values from a profit signature.

Usage

NPV_partial(Pi, r)

Arguments

Pi	Profit signature vector.
r	Risk discount rate.

Value

Numeric vector.

Examples

```
Pi <- c(-15.00, 8.42, 8.39, 8.58)
NPV_partial(Pi, r = 0.10)
```

NPV_profit	<i>Net present value of a profit signature</i>
------------	--

Description

Computes the Chapter 17 net present value:

$$NPV = \sum_{t=0}^n \frac{\Pi_t}{(1+r)^t}$$

Usage

NPV_profit(Pi, r)

Arguments

Pi	Profit signature vector.
r	Risk discount rate.

Value

Numeric scalar.

Examples

```
Pi <- c(-15.00, 8.42, 8.39, 8.58)
NPV_profit(Pi, r = 0.10)
```

npx

Compute n-year survival probability from a life table

Description

Compute n-year survival probability from a life table

Usage

```
npx(tbl, x, n)
```

Arguments

tbl	A life_table object.
x	Ages.
n	Nonnegative integer durations.

Value

Numeric vector of ${}_n p_x$ values.

npxtau_md	${}_n p_x^{(\tau)}$ from a multiple-decrement table
-----------	---

Description

${}_n p_x^{(\tau)}$ from a multiple-decrement table

Usage

```
npxtau_md(tbl, x, n)
```

Arguments

tbl	Output from md_table().
x	Starting age.
n	Number of years.

Value

Numeric scalar.

Examples

```
x <- 45:50
qmat <- cbind(q1 = c(.011, .012, .013, .014, .015, .016), q2 = rep(.1, 6))
tbl <- md_table(x, qmat, radix = 1000)
npxtau_md(tbl, x = 46, n = 3)
```

npx_select	<i>Select-life survival probability</i>
------------	---

Description

Computes ${}_n p_{[x]+t} = l_{[x]+t+n} / l_{[x]+t}$

Usage

```
npx_select(tbl, x_sel, t, n)
```

Arguments

tbl	A select_life_table object.
x_sel	Numeric vector of ages at selection.
t	Numeric vector of current durations since selection.
n	Numeric vector of nonnegative integer future durations.

Details

in the discrete select-table setting.

Value

Numeric vector of survival probabilities.

nqx	<i>Compute n-year death probability from a life table</i>
-----	---

Description

Compute n-year death probability from a life table

Usage

```
nqx(tbl, x, n)
```

Arguments

tbl	A life_table object.
x	Ages.
n	Nonnegative integer durations.

Value

Numeric vector of ${}_nq_x$ values.

${}_nq_{x md}$	<i>${}_nq_x^{(j)}$ from a multiple-decrement table</i>
----------------	---

Description

${}_nq_x^{(j)}$ from a multiple-decrement table

Usage

```
nqxj_md(tbl, x, n, j)
```

Arguments

tbl	Output from md_table().
x	Starting age.
n	Number of years.
j	Cause index.

Value

Numeric scalar.

Examples

```
x <- 45:50
qmat <- cbind(q1 = c(.011, .012, .013, .014, .015, .016), q2 = rep(.1, 6))
tbl <- md_table(x, qmat, radix = 1000)
nqxj_md(tbl, x = 46, n = 2, j = 1)
```

nqxtau_md	${}_nq_x^{(\tau)}$ from a multiple-decrement table
-----------	--

Description

${}_nq_x^{(\tau)}$ from a multiple-decrement table

Usage

```
nqxtau_md(tbl, x, n)
```

Arguments

tbl	Output from md_table().
x	Starting age.
n	Number of years.

Value

Numeric scalar.

Examples

```
x <- 45:50
qmat <- cbind(q1 = c(.011, .012, .013, .014, .015, .016), q2 = rep(.1, 6))
tbl <- md_table(x, qmat, radix = 1000)
nqxtau_md(tbl, x = 46, n = 2)
```

nqx_select	<i>Select-life death probability</i>
------------	--------------------------------------

Description

Computes ${}_nq_{[x]+t} = 1 - {}_n p_{[x]+t}$.

Usage

```
nqx_select(tbl, x_sel, t, n)
```

Arguments

tbl	A select_life_table object.
x_sel	Numeric vector of ages at selection.
t	Numeric vector of current durations since selection.
n	Numeric vector of nonnegative integer future durations.

Value

Numeric vector of death probabilities.

PAB_cae	<i>Projected annual benefit under a career average earnings DB plan</i>
---------	---

Description

Computes the projected annual benefit for a CAE plan using Equation (18.9).

Usage

```
PAB_cae(x, z, CASx, p, past_salary_total = 0, g = NULL, s = NULL)
```

Arguments

x	Current or entry age.
z	Retirement age.
CASx	Current annual salary at age x.
p	Accrual percentage, e.g. 1 for 1 percent.
past_salary_total	Optional total of actual past salaries.
g	Optional constant annual salary growth rate.
s	Optional salary scale vector of length z - x.

Value

Projected annual benefit.

Examples

PAB_cae($x = 30$, $z = 65$, $CASx = 100000$, $p = 1$, $g = 0.04$)

PAB_fas

Projected annual benefit under a final average salary DB plan

Description

Computes the projected annual benefit for a final average salary plan using Equation (18.6).

Usage

PAB_fas(x , z , $CASx$, p , $fas_years = 3$, $past_service = 0$, $g = NULL$, $s = NULL$)

Arguments

x	Current or entry age.
z	Retirement age.
$CASx$	Current annual salary at age x .
p	Accrual percentage, e.g. 2 for 2 percent.
fas_years	Number of years in the final average salary period.
$past_service$	Past years of service already completed at age x .
g	Optional constant annual salary growth rate.
s	Optional salary scale vector of length $z - x$.

Value

Projected annual benefit.

Examples

PAB_fas($x = 35$, $z = 65$, $CASx = 60000$, $p = 2$, $fas_years = 3$, $g = 0.04$)

Pbar_trapz_ms

*Continuous premium approximation \bar{P} by trapezoidal rule***Description**

Approximates the annual continuous premium in the disability model allowing for recovery, as in Example 14.18.

Usage

```
Pbar_trapz_ms(t, tp00, tp01, delta, mu02, mu12, B02 = 1, B12 = 1, R = 0)
```

Arguments

t	Numeric vector of time points.
tp00	Numeric vector of values ${}_t p_x^{00}$.
tp01	Numeric vector of values ${}_t p_x^{01}$.
delta	Force of interest.
mu02	Function of time returning μ_{x+t}^{02} .
mu12	Function of time returning μ_{x+t}^{12} .
B02	Benefit payable on death while healthy.
B12	Benefit payable on death while disabled.
R	Continuous income rate while disabled.

Details

The numerator is

$$\int v^t [{}_t p_x^{00} \mu_{x+t}^{02} B^{02} + {}_t p_x^{01} \mu_{x+t}^{12} B^{12} + {}_t p_x^{01} R] dt$$

and the denominator is

$$\int v^t {}_t p_x^{00} dt$$

Value

A numeric scalar.

Examples

```
mu01 <- function(t) 0.10 * t + 0.20
mu02 <- function(t) 0.20
mu10 <- function(t) 0.50
mu12 <- function(t) 0.125 * t + 0.20

ex1410 <- tp00_tp01_euler(
  h = 0.10, n = 2.0,
```

```

mu01 = mu01, mu02 = mu02, mu10 = mu10, mu12 = mu12
)

Pbar_trapz_ms(
  t = ex1410$t,
  tp00 = ex1410$tp00,
  tp01 = ex1410$tp01,
  delta = 0.04,
  mu02 = mu02,
  mu12 = mu12,
  B02 = 1000,
  B12 = 1000,
  R = 1000
)

```

Pi_signature

Profit signature from a profit vector

Description

Converts a Chapter 17 profit vector $\mathbf{Pr} = (Pr_0, \dots, Pr_n)$ into the corresponding profit signature $\mathbf{\Pi} = (\Pi_0, \dots, \Pi_n)$ using Equation (17.3).

Usage

```
Pi_signature(Pr, p_tau)
```

Arguments

Pr	Profit vector of length $n + 1$.
p_tau	One-year in-force probabilities. This may have length $n - 1$ or n . If length n , the final entry is ignored for the profit-signature calculation.

Value

Numeric vector of length $n + 1$.

Examples

```

Pr <- c(-15.00, 8.42, 8.40, 8.61)
Pi_signature(Pr, p_tau = c(0.99858, 0.99847, 0.99834))

```

PnAdotx *Deferred annuity-due premium*

Description

Computes $P({}_n|\ddot{a}_x) = {}_n|\ddot{a}_x / \ddot{a}_{x:\overline{n}|}$.

Usage

PnAdotx(x, n, i, model, ...)

Arguments

x	Issue age.
n	Deferral period.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

PnAdotx(40, n = 20, i = 0.05, model = "uniform", omega = 100)

Pnax *Deferred annuity-immediate premium*

Description

Computes $P({}_n|a_x) = {}_n|a_x / \ddot{a}_{x:\overline{n}|}$.

Usage

Pnax(x, n, i, model, ...)

Arguments

x	Issue age.
n	Deferral period.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
Pnax(40, n = 20, i = 0.05, model = "uniform", omega = 100)
```

```
premium_ch9
```

Chapter 9 premium, loss, and expense functions

Description

Functions in this file implement Chapter 9 funding-plan formulas, including:

- net annual premiums under the equivalence principle,
- limited-payment premiums,
- continuous-payment premium rates,
- fully continuous premium rates,
- true fractional premiums,
- present-value-of-loss means and variances,
- a basic gross premium formula for whole life insurance.

Usage

```
Px(x, i, tbl = NULL, model = NULL, ...)
```

```
Pxn1(x, n, i, tbl = NULL, model = NULL, ...)
```

```
PnEx(x, n, i, tbl = NULL, model = NULL, ...)
```

```
Pxn(x, n, i, tbl = NULL, model = NULL, ...)
```

```
tPx(x, t, i, tbl = NULL, model = NULL, ...)
```

```
tPxn1(x, n, t, i, tbl = NULL, model = NULL, ...)
```

```
tPnEx(x, n, t, i, tbl = NULL, model = NULL, ...)
```

```
tPxn(x, n, t, i, tbl = NULL, model = NULL, ...)
```

```
PnAx(x, n, i, tbl = NULL, model = NULL, ...)
```

```
tPnAx(x, n, t, i, tbl = NULL, model = NULL, ...)
```

```
Pbarx(x, i, model, ..., tol = 1e-10)
```



```
Pbarxn1(x, n, i, model, ...)  
Pbarxn(x, n, i, model, ...)  
PbarAbarx(x, i, model, ..., tol = 1e-10)  
PbarAbarxn1(x, n, i, model, ...)  
PbarAbarxn(x, n, i, model, ...)  
Px_m(x, m, i, tbl = NULL, model = NULL, ...)  
Pxn1_m(x, n, m, i, tbl = NULL, model = NULL, ...)  
Pxn_m(x, n, m, i, tbl = NULL, model = NULL, ...)  
PnAx_m(x, n, m, i, tbl = NULL, model = NULL, ...)  
EL0x(x, P, i, tbl = NULL, model = NULL, ...)  
varL0x(x, P, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)  
EL0xn1(x, n, P, i, tbl = NULL, model = NULL, ...)  
varL0xn1(x, n, P, i, tbl = NULL, model = NULL, ...)  
EL0xn(x, n, P, i, tbl = NULL, model = NULL, ...)  
varL0xn(x, n, P, i, tbl = NULL, model = NULL, ...)  
EL0barAbarx(x, P, i, model, ..., tol = 1e-10)  
varL0barAbarx(x, P, i, model, ...)  
  
Gx(  
  x,  
  i,  
  benefit = 1,  
  first_premium_pct = 0,  
  renewal_premium_pct = 0,  
  first_policy_exp = 0,  
  renewal_policy_exp = 0,  
  settlement_exp = 0,  
  tbl = NULL,  
  model = NULL,  
  ...  
)
```

Arguments

<code>x</code>	Age.
<code>i</code>	Effective annual interest rate.
<code>tbl</code>	Optional life table object.
<code>model</code>	Optional parametric survival model name.
<code>...</code>	Additional arguments passed to survival-model functions.
<code>n</code>	Term.
<code>t</code>	Premium-paying period.
<code>tol</code>	Numerical tolerance for functions that truncate infinite sums.
<code>m</code>	Number of payments per year.
<code>P</code>	Premium amount or premium rate.
<code>k_max</code>	Maximum summation horizon for functions that truncate infinite sums.
<code>benefit</code>	Benefit amount.
<code>first_premium_pct</code>	First-year premium expense proportion.
<code>renewal_premium_pct</code>	Renewal premium expense proportion.
<code>first_policy_exp</code>	First-year fixed expense.
<code>renewal_policy_exp</code>	Renewal fixed expense each year after the first.
<code>settlement_exp</code>	Settlement expense incurred at benefit payment.

Details

Naming follows Chapter 9 notation as closely as possible:

- $P_x()$ = whole life annual premium
- $P_{x:n}()$ = term insurance annual premium
- $P_nEx()$ = pure endowment annual premium
- $P_{x:n}()$ = endowment insurance annual premium
- $tP_x()$ = limited-payment whole life annual premium
- $tP_{x:n}()$ = limited-payment term insurance annual premium
- $tP_nEx()$ = limited-payment pure endowment annual premium
- $tP_{x:n}()$ = limited-payment endowment insurance annual premium
- $P_nAx()$ = deferred insurance annual premium
- $tP_nAx()$ = limited-payment deferred insurance annual premium
- $\bar{P}_x()$ = continuous-payment premium for discrete whole life insurance
- $\bar{P}_{x:n}()$ = continuous-payment premium for discrete term insurance
- $\bar{P}_{x:n}()$ = continuous-payment premium for discrete endowment insurance

profit_margin	<i>Profit margin</i>
---------------	----------------------

Description

Computes the Chapter 17 profit margin:

$$\text{Profit Margin} = \frac{NPV}{APV_{GP}}.$$

Usage

profit_margin(NPV, APV_GP)

Arguments

NPV	Net present value of profits.
APV_GP	Actuarial present value of gross premiums.

Value

Numeric scalar.

Examples

profit_margin(6.03, 259.52)

Pr_vector_disc	<i>Profit vector for a discrete profit-analysis model</i>
----------------	---

Description

Computes the Chapter 17 profit vector

$$\mathbf{Pr} = (Pr_0, Pr_1, \dots, Pr_n)$$

where Pr_0 is the negative pre-contract expense and the yearly expected profit values are calculated from the general discrete expression in Equation (17.1).

Usage

```
Pr_vector_disc(
  V,
  G,
  i,
  r = 0,
  e = 0,
  q1,
  q2 = 0,
  b1,
  b2 = 0,
  s1 = 0,
  s2 = 0,
  p_tau = NULL,
  pre_contract_expense = 0
)
```

Arguments

V	Vector of gross premium reserves ${}_tV^G$ of length $n + 1$, including the issue-time reserve and the terminal reserve.
G	Gross premium vector for policy years 1 through n .
i	Interest-rate vector for policy years 1 through n .
r	Percent-of-premium expense vector.
e	Fixed expense vector.
q1	First decrement probabilities, typically death.
q2	Second decrement probabilities, typically surrender or lapse. Defaults to 0.
b1	Benefit vector for decrement 1.
b2	Benefit vector for decrement 2. Defaults to 0.
s1	Settlement-expense vector for decrement 1. Defaults to 0.
s2	Settlement-expense vector for decrement 2. Defaults to 0.
p_tau	Optional vector of in-force probabilities $p_{x+t}^{(\tau)}$. If omitted, it is computed as $1 - q^{(1)} - q^{(2)}$.
pre_contract_expense	Positive pre-contract expense amount. The returned first element is $Pr_0 = -\text{pre_contract_expense}$.

Details

This implementation allows for two decrements, typically death and withdrawal/surrender.

Value

Numeric vector of length $n + 1$.

Examples

```
V <- c(0, 5.66, 6.17, 0)
qx <- c(0.00142, 0.00153, 0.00166)
Pr_vector_disc(
  V = V, G = 95, i = 0.06, r = 0.05, e = 10,
  q1 = qx, b1 = 50000, pre_contract_expense = 15
)
```

pv_cashflows

Present value of cash flows at time 0

Description

Present value of cash flows at time 0

Usage

```
pv_cashflows(cf, t, i)
```

Arguments

cf	Cash flow amounts (positive = inflow, negative = outflow).
t	Times of cash flows (same length as cf).
i	Effective interest rate.

Value

Present value at time 0.

Examples

```
pv_cashflows(c(-100, 60, 60), c(0, 1, 2), i = 0.10)
```

pv_spot_cashflows

Present value of cash flows using spot rates

Description

Discounts deterministic cash flows using spot rates matched to their maturities.

Usage

```
pv_spot_cashflows(
  amounts,
  times,
  spot,
  compounding = c("annual", "semiannual")
)
```

Arguments

amounts	Numeric vector of cash flow amounts.
times	Numeric vector of payment times in years.
spot	Numeric vector of spot rates matched elementwise to times. Use 0 for any time-0 entry.
compounding	Either "annual" or "semiannual".

Details

For annual compounding, each positive-time cash flow at time t is discounted by $(1 + z_t)^{-t}$.

For semiannual nominal compounding, each positive-time cash flow at time t is discounted by $(1 + z_t/2)^{-2t}$.

Time-0 cash flows are left undiscounted.

Value

A numeric scalar.

Examples

```
pv_spot_cashflows(
  amounts = c(200000, 50000, 50000, 100000),
  times = c(0, 0.5, 1, 2),
  spot = c(0, 0.02440, 0.02601, 0.02936),
  compounding = "semiannual"
)
```

pxtau

Survival probability $p_x^{(\tau)}$

Description

Survival probability $p_x^{(\tau)}$

Usage

```
pxtau(qxj)
```

Arguments

qxj Numeric vector of cause-specific decrement probabilities.

Value

Numeric scalar.

Examples

```
pxtau(c(0.011, 0.100))
```

pxtau_ul	<i>One-year persistency rates for universal life</i>
----------	--

Description

Computes the one-year persistency rates $p_x^{(\tau)}$ under either Equation (16.14) or Equation (16.15).

Usage

```
pxtau_ul(qd, qw, year_end_withdrawal = TRUE)
```

Arguments

qd Mortality probabilities.

qw Withdrawal probabilities.

year_end_withdrawal

Logical; if TRUE, use $(1 - q^{(d)})(1 - q^{(w)})$. Otherwise use $1 - q^{(d)} - q^{(w)}$.

Value

Numeric vector.

Examples

```
qd <- c(.001, .002, .003)
qw <- c(.02, .02, .03)
pxtau_ul(qd, qw)
```

px_proj *Project one-year survival probability under mortality improvement*

Description

Computes projected one-year survival probability

$$p_x^{[Y]} = 1 - q_x^{[Y]}$$

Usage

px_proj(qx_base, AAx, base_year, proj_year)

Arguments

qx_base	Base-year one-year death probability $q_x^{[B]}$.
AAx	Mortality improvement factor AA_x .
base_year	Base year B .
proj_year	Projection year Y . May be scalar or vector.

Value

Numeric vector of projected one-year survival probabilities.

px_to_lx *Construct life-table values from p_x values*

Description

Builds life-table survivor values recursively from $l_{x+1} = l_x p_x$, starting from a chosen radix.

Usage

px_to_lx(px, radix = 1e+05)

Arguments

px	Numeric vector of one-year survival probabilities p_x .
radix	Positive radix l_0 .

Value

Numeric vector of l_x values of length $\text{length}(px)+1$.

qxprime_mudd	<i>Independent probabilities $q_x^{t(j)}$ from dependent probabilities $q_x^{(j)}$ under MUDD</i>
--------------	---

Description

Independent probabilities $q_x^{t(j)}$ from dependent probabilities $q_x^{(j)}$ under MUDD

Usage

```
qxprime_mudd(qxj)
```

Arguments

qxj Numeric vector of dependent probabilities $q_x^{(j)}$.

Value

Numeric vector of independent probabilities $q_x^{t(j)}$.

Examples

```
qxprime_mudd(c(.20, .10))
```

qxprime_sudd	<i>Independent probabilities $q_x^{t(j)}$ from dependent probabilities $q_x^{(j)}$ under SUDD</i>
--------------	---

Description

Two-decrement case only.

Usage

```
qxprime_sudd(q1, q2)
```

Arguments

q1 Dependent probability for decrement 1.
q2 Dependent probability for decrement 2.

Value

Numeric vector c(q1prime, q2prime).

Examples

```
qxprime_sudd(0.20, 0.10)
```

qxtau	<i>Total probability of decrement $q_x^{(\tau)}$</i>
-------	---

Description

Total probability of decrement $q_x^{(\tau)}$

Usage

qxtau(qxj)

Arguments

qxj Numeric vector of cause-specific decrement probabilities.

Value

Numeric scalar.

Examples

qxtau(c(0.011, 0.100))

qx_dep_cf	<i>Dependent probabilities $q_x^{(j)}$ from independent probabilities $q_x^{t(j)}$ under constant force</i>
-----------	---

Description

Dependent probabilities $q_x^{(j)}$ from independent probabilities $q_x^{t(j)}$ under constant force

Usage

qx_dep_cf(qxprime)

Arguments

qxprime Numeric vector of independent probabilities.

Value

Numeric vector of dependent probabilities.

Examples

qx_dep_cf(c(0.20, 0.10))

qx_dep_sudd	<i>Dependent probabilities $q_x^{(j)}$ from independent probabilities $q_x^{(j)}$ under SUDD</i>
-------------	--

Description

Two-decrement case only.

Usage

qx_dep_sudd(q1prime, q2prime)

Arguments

q1prime	Independent probability for decrement 1.
q2prime	Independent probability for decrement 2.

Value

Numeric vector c(q1, q2).

Examples

qx_dep_sudd(0.20, 0.10)

qx_proj	<i>Project one-year death probability under mortality improvement</i>
---------	---

Description

Computes projected one-year death probability

$$q_x^{[Y]} = q_x^{[B]}(1 - AA_x)^{Y-B}$$

Usage

qx_proj(qx_base, AAx, base_year, proj_year)

Arguments

qx_base	Base-year one-year death probability $q_x^{[B]}$.
AAx	Mortality improvement factor AA_x .
base_year	Base year B .
proj_year	Projection year Y . May be scalar or vector.

Value

Numeric vector of projected one-year death probabilities.

qx_tab	<i>Compute one-year death probability from a life table</i>
--------	---

Description

Compute one-year death probability from a life table

Usage

```
qx_tab(tbl, x)
```

Arguments

tbl	A life_table object.
x	Ages.

Value

Numeric vector of q_x values.

qx_to_lx	<i>Construct life-table values from q_x values</i>
----------	---

Description

Builds life-table survivor values recursively from $l_{x+1} = l_x(1 - q_x)$, starting from a chosen radix.

Usage

```
qx_to_lx(qx, radix = 1e+05)
```

Arguments

qx	Numeric vector of one-year death probabilities q_x .
radix	Positive radix l_0 .

Value

Numeric vector of l_x values of length $\text{length}(qx)+1$.

replacement_ratio_db *Replacement ratio for a defined benefit plan*

Description

Computes a DB replacement ratio as benefit divided by a chosen salary measure.

Usage

replacement_ratio_db(benefit, salary)

Arguments

benefit	Annual retirement benefit.
salary	Salary measure used in the denominator.

Value

Replacement ratio.

Examples

replacement_ratio_db(benefit = 108008.66, salary = 187119.09)

replacement_ratio_dc *Replacement ratio for a defined contribution plan*

Description

Computes the replacement ratio defined in Equation (18.4).

Usage

replacement_ratio_dc(x, z, Sx, c, i, adue_z, g = NULL, s = NULL)

Arguments

x	Entry age.
z	Retirement age.
Sx	Salary at age x.
c	Contribution rate.
i	Annual effective interest rate.
adue_z	Whole life annuity-due factor at age z.
g	Optional constant annual salary growth rate.
s	Optional salary scale vector of length z - x.

Value

Replacement ratio.

Examples

```
replacement_ratio_dc(
  x = 30, z = 65, Sx = 50000, c = 0.10, i = 0.05,
  adue_z = 12, g = 0.04
)
```

 reserve_ch10

Reserve functions for Chapter 10

Description

Net level premium reserve functions aligned with Chapter 10 notation.

Details

These functions implement prospective reserve formulas for whole life, term insurance, pure endowment, endowment insurance, selected continuous premium cases, selected m-thly premium cases, basic gain/loss calculations, and duration-t loss moments.

 reserve_ch10_ext

Extended reserve functions for Chapter 10

Description

Additional Chapter 10 reserve functions: retrospective reserves, deferred insurance reserves, annuity reserves, continuous-time gain/loss helpers, and Euler/Thiele approximation helpers.

 reserve_ch11

Reserve functions for Chapter 11

Description

Chapter 11 functions for modified reserves, fractional-duration reserves, gross premium reserves, expense reserves, and gross gain analysis.

reserve_ch11_ext *Extended reserve functions for Chapter 11*

Description

Additional Chapter 11 reserve functions: FPT modified reserves, fractional-duration reserves, gross premium reserves, expense reserves, and ordered gross gain decomposition.

reserve_deferred_insurance_ch10
Deferred insurance reserve functions

Description

Chapter 10 reserve functions for deferred insurance contracts.

Usage

`tVnAx(x, n, t, i, model, ...)`

`htVnAx(x, n, h, t, i, model, ...)`

Arguments

<code>x</code>	Issue age.
<code>n</code>	Deferral period.
<code>t</code>	Duration.
<code>i</code>	Effective annual interest rate.
<code>model</code>	Survival model.
<code>...</code>	Additional model parameters.
<code>h</code>	Premium-paying period.

Details

`tVnAx()` computes the reserve for an n -year deferred insurance funded over the deferred period.

`htVnAx()` computes the reserve when premiums are limited to the first h years, with $h \leq n$.

Value

Numeric vector.

Numeric vector.

Examples

`tVnAx(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)`

`htVnAx(40, n = 20, h = 10, t = 5, i = 0.05, model = "uniform", omega = 100)`

rt_ul	<i>Ratio $r_t = AV_t/GMF_t$ capped at 1</i>
-------	--

Description

Computes the ratio in Equation (16.16).

Usage

rt_ul(AV, GMF)

Arguments

AV	Account value.
GMF	Guaranteed maturity fund.

Value

Numeric scalar.

Examples

rt_ul(AV = 4918.20, GMF = 14678.57)

S0	<i>Survival function for age-at-failure T0</i>
----	--

Description

Computes the survival distribution function $S_0(t) = Pr(T_0 > t)$.

Usage

S0(t, model, ...)

Arguments

t	Numeric vector of times ($t \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale

Details

Supported models (Chapter 5): uniform (de Moivre), exponential, Gompertz, Makeham, Weibull.

Value

Numeric vector of survival probabilities in $[0, 1]$.

<code>S0_to_lx</code>	<i>Convert survival probabilities to life-table values</i>
-----------------------	--

Description

Converts Chapter 5 survival function values $S_0(x)$ into Chapter 6 life-table values $l_x = l_0 S_0(x)$ using a chosen radix.

Usage

```
S0_to_lx(S0, radix = 1e+05)
```

Arguments

<code>S0</code>	Numeric vector of survival probabilities.
<code>radix</code>	Positive radix l_0 .

Value

Numeric vector of l_x values.

<code>salary_scale</code>	<i>Salary scale under constant annual growth</i>
---------------------------	--

Description

Constructs salary scale factors s_k under a constant annual growth rate.

Usage

```
salary_scale(k, g, base_age = min(k), s_base = 1)
```

Arguments

<code>k</code>	Numeric vector of ages.
<code>g</code>	Annual salary growth rate.
<code>base_age</code>	Age at which the scale is normalized.
<code>s_base</code>	Salary scale value at <code>base_age</code> .

Value

Numeric vector of salary scale factors.

Examples

```
salary_scale(k = 30:34, g = 0.04, base_age = 30)
```

select_life_table	<i>Construct a select life table</i>
-------------------	--------------------------------------

Description

Builds a select-life-table object from vectors of selection age, duration since selection, attained age, and survivor values.

Usage

```
select_life_table(x_sel, duration, attained_age, lx)
```

Arguments

x_sel	Numeric vector of ages at selection.
duration	Numeric vector of durations since selection.
attained_age	Numeric vector of attained ages.
lx	Numeric vector of select-table survivor values.

Value

A data.frame with class "select_life_table".

solve_yield	<i>Solve the yield rate by the equation of value</i>
-------------	--

Description

Finds the interest rate i such that the present value of the cash flows is 0.

Usage

```
solve_yield(cf, t, interval = c(-0.99, 1), tol = 1e-10)
```

Arguments

cf	Cash flows.
t	Times.
interval	Two-length numeric vector bracketing the root.
tol	Tolerance passed to uniroot.

Value

Yield rate i .

Examples

```
solve_yield(c(-100, 60, 60), c(0, 1, 2), interval = c(-0.5, 1))
```

thiele_backward_path *Backward Euler reserve path from maturity*

Description

Starting from a terminal reserve value at time T , computes reserves backward on a grid using the backward Euler-style Thiele step.

Usage

```
thiele_backward_path(times, V_terminal, P, delta, mu, benefit = 1)
```

Arguments

times	Vector of times in increasing order.
V_terminal	Reserve at the final time.
P	Premium rate, scalar or vector of length $\text{length}(\text{times})-1$.
delta	Force of interest, scalar or vector of length $\text{length}(\text{times})-1$.
mu	Force of mortality, scalar or vector of length $\text{length}(\text{times})-1$.
benefit	Benefit amount, scalar or vector of length $\text{length}(\text{times})-1$.

Value

Numeric vector of reserve values on the grid.

Examples

```
times <- seq(19, 20, by = 0.25)
thiele_backward_path(times, V_terminal = 1000, P = 26.96, delta = 0.058, mu = 0.002, benefit = 1000)
```

thiele_backward_step *One backward Euler-style Thiele step*

Description

Approximates the reserve at time t from a known reserve at time $t + h$.

Usage

```
thiele_backward_step(V_next, P, delta, mu, benefit = 1, h = 1)
```

Arguments

V_next	Reserve at time $t+h$.
P	Premium rate.
delta	Force of interest.
mu	Force of mortality at time t .
benefit	Benefit amount. Defaults to 1.
h	Step size.

Value

Numeric vector.

Examples

```
thiele_backward_step(V_next = 1000, P = 26.96, delta = 0.058, mu = 0.002, benefit = 1000, h = 1)
```

thiele_dVdt *Reserve derivative from Thiele's equation*

Description

Computes $dV/dt = P + \delta V - \mu(B - V)$.

Usage

```
thiele_dVdt(V, P, delta, mu, benefit = 1)
```

Arguments

V	Reserve at time t .
P	Premium rate.
delta	Force of interest.
mu	Force of mortality.
benefit	Benefit amount. Defaults to 1.

Value

Numeric vector.

Examples

```
thiele_dVdt(V = 900, P = 25, delta = 0.05, mu = 0.002, benefit = 1000)
```

```
thiele_dVdt_01
```

Reserve derivatives for the disability model with recovery

Description

Computes the right-hand sides of the coupled Thiele differential equations in Equations (14.25) and (14.26) for the healthy-life reserve ${}_t\bar{V}^{(0)}$ and the disabled-life reserve ${}_t\bar{V}^{(1)}$.

Usage

```
thiele_dVdt_01(t, V0, V1, delta, Pbar, B, R, mu01, mu02, mu10, mu12)
```

Arguments

t	Time.
V0	Value of ${}_t\bar{V}^{(0)}$.
V1	Value of ${}_t\bar{V}^{(1)}$.
delta	Force of interest.
Pbar	Continuous premium rate.
B	Death benefit.
R	Continuous disability income rate.
mu01	Function of time returning μ_{x+t}^{01} .
mu02	Function of time returning μ_{x+t}^{02} .
mu10	Function of time returning μ_{x+t}^{10} .
mu12	Function of time returning μ_{x+t}^{12} .

Details

The equations are

$$\frac{d}{dt}{}_t\bar{V}^{(0)} = \bar{P} + \delta_t\bar{V}^{(0)} - \mu_{x+t}^{02}(B - {}_t\bar{V}^{(0)}) - \mu_{x+t}^{01}({}_t\bar{V}^{(1)} - {}_t\bar{V}^{(0)})$$

and

$$\frac{d}{dt}{}_t\bar{V}^{(1)} = \delta_t\bar{V}^{(1)} - R - \mu_{x+t}^{12}(B - {}_t\bar{V}^{(1)}) - \mu_{x+t}^{10}({}_t\bar{V}^{(0)} - {}_t\bar{V}^{(1)})$$

Value

A named numeric vector with components dV_0 and dV_1 .

Examples

```
mu01 <- function(t) 0.10 * t + 0.20
mu02 <- function(t) 0.20
mu10 <- function(t) 0.50
mu12 <- function(t) 0.125 * t + 0.20

thiele_dVdt_01(
  t = 2.0, V0 = 0, V1 = 0,
  delta = 0.04, Pbar = 446.95,
  B = 1000, R = 1000,
  mu01 = mu01, mu02 = mu02, mu10 = mu10, mu12 = mu12
)
```

 thiele_path_01

Backward reserve path for the disability model with recovery

Description

Computes the backward Euler reserve path for the healthy-life reserve ${}_t\bar{V}^{(0)}$ and disabled-life reserve ${}_t\bar{V}^{(1)}$ using Equations (14.27) and (14.28).

Usage

```
thiele_path_01(
  h,
  n,
  delta,
  Pbar,
  B,
  R,
  mu01,
  mu02,
  mu10,
  mu12,
  V0_n = 0,
  V1_n = 0
)
```

Arguments

h Step size.
 n Final time.

delta	Force of interest.
Pbar	Continuous premium rate.
B	Death benefit.
R	Continuous disability income rate.
mu01	Function of time returning μ_{x+t}^{01} .
mu02	Function of time returning μ_{x+t}^{02} .
mu10	Function of time returning μ_{x+t}^{10} .
mu12	Function of time returning μ_{x+t}^{12} .
V0_n	Terminal value of ${}_n\bar{V}^{(0)}$.
V1_n	Terminal value of ${}_n\bar{V}^{(1)}$.

Value

A data frame with columns t, tV0, and tV1.

Examples

```
mu01 <- function(t) 0.10 * t + 0.20
mu02 <- function(t) 0.20
mu10 <- function(t) 0.50
mu12 <- function(t) 0.125 * t + 0.20

thiele_path_01(
  h = 0.10, n = 2.0, delta = 0.04, Pbar = 446.95,
  B = 1000, R = 1000,
  mu01 = mu01, mu02 = mu02, mu10 = mu10, mu12 = mu12
)
```

 tp00_tp01_euler

Euler approximation for ${}_{tp}x^{00}$ and ${}_{tp}x^{01}$

Description

Computes the Euler approximations in the disability model allowing for recovery, as in Equations (14.20) and (14.21).

Usage

```
tp00_tp01_euler(h, n, mu01, mu02, mu10, mu12, p00_0 = 1, p01_0 = 0)
```


Arguments

h	Step size.
n	Final time.
mu01	Function of time returning μ_{x+t}^{01} .
mu02	Function of time returning μ_{x+t}^{02} .
mu10	Function of time returning μ_{x+t}^{10} .
mu12	Function of time returning μ_{x+t}^{12} .
p00_0	Initial value of ${}_0p_x^{00}$.
p01_0	Initial value of ${}_0p_x^{01}$.

Details

The model uses three states:

- State 0: healthy
- State 1: disabled
- State 2: deceased

Value

A data frame with columns t, tp00, tp01, and tp02.

Examples

```
mu01 <- function(t) 0.10 * t + 0.20
mu02 <- function(t) 0.20
mu10 <- function(t) 0.50
mu12 <- function(t) 0.125 * t + 0.20

tp00_tp01_euler(
  h = 0.10, n = 2.0,
  mu01 = mu01, mu02 = mu02, mu10 = mu10, mu12 = mu12
)
```

 tpx

Conditional survival probability for Tx

Description

Computes ${}_t p_x = S_0(x+t)/S_0(x)$.

Usage

```
tpx(t, x, model, ...)
```

Arguments

t	Numeric vector of durations ($t \geq 0$).
x	Numeric vector of ages ($x \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale

Details

Vectorization rule: - If t and x are the same length, values are computed elementwise. - If one of t or x has length 1, it is recycled to match the other.

Value

Numeric vector in $[0, 1]$.

tpxprimej_cf

Single-decrement survival probability ${}_t p_x^{(j)}$ under constant force

Description

Single-decrement survival probability ${}_t p_x^{(j)}$ under constant force

Usage

```
tpxprimej_cf(mu, t)
```

Arguments

mu	Force of decrement for cause j.
t	Time.

Value

Numeric scalar/vector.

Examples

```
tpxprimej_cf(0.10, 5)
```

tpxtau_ul	<i>Cumulative persistency to the end of each policy year</i>
-----------	--

Description

Computes ${}_t p_x^{(\tau)}$ from the one-year persistency rates.

Usage

```
tpxtau_ul(qd, qw, year_end_withdrawal = TRUE)
```

Arguments

qd	Mortality probabilities.
qw	Withdrawal probabilities.
year_end_withdrawal	Logical; if TRUE, use Equation (16.15).

Value

Numeric vector.

Examples

```
qd <- c(.001, .002, .003)
qw <- c(.02, .02, .03)
tpxtau_ul(qd, qw)
```

tpxy	<i>Joint-life survival probability</i>
------	--

Description

Computes ${}_t p_{xy} = {}_t p_x {}_t p_y$ under independence.

Usage

```
tpxy(x, y, t, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
t	Time.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tpxy(40, 50, t = 10, model = "uniform", omega = 100)
```

tpxybar	<i>Last-survivor survival probability</i>
---------	---

Description

Computes ${}_t p_{\overline{xy}} = {}_t p_x + {}_t p_y - {}_t p_{xy}$.

Usage

```
tpxybar(x, y, t, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
t	Time.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tpxybar(40, 50, t = 10, model = "uniform", omega = 100)
```

tpx_improved	<i>Multi-year survival probability under mortality improvement</i>
--------------	--

Description

Computes survival over n years starting at age x_0 in year `issue_year`, using base-year mortality `qx_base_vec` and improvement factors `AAx_vec`.

Usage

```
tpx_improved(x0, n, qx_base_vec, AAx_vec, base_year, issue_year)
```

Arguments

<code>x0</code>	Issue age.
<code>n</code>	Number of years.
<code>qx_base_vec</code>	Base-year one-year death probabilities for successive ages.
<code>AAx_vec</code>	Mortality improvement factors for successive ages.
<code>base_year</code>	Base year.
<code>issue_year</code>	Issue year.

Details

The vectors should correspond to ages $x_0, x_0+1, \dots, x_0+n-1$.

The survival probability is

$$\prod_{j=0}^{n-1} \left(1 - q_{x+j}^{[issue_year+j]} \right)$$

Value

Numeric scalar.

tpx_tab	<i>Fractional survival probability from a life table</i>
---------	--

Description

Computes ${}_t p_x$ for $0 \leq t \leq 1$ from a discrete life table under one of the standard Chapter 6 assumptions: UDD, constant force, or Balducci.

Usage

```
tpx_tab(tbl, x, t, assumption = c("udd", "cf", "balducci"))
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
t	Numeric vector of fractional durations with $0 \leq t \leq 1$.
assumption	One of "udd", "cf", "balducci".

Value

Numeric vector of ${}_t p_x$ values.

tpx_tau_cf	<i>Total survival probability ${}_t p_x^{(\tau)}$ under constant forces</i>
------------	--

Description

Total survival probability ${}_t p_x^{(\tau)}$ under constant forces

Usage

```
tpx_tau_cf(mu, t)
```

Arguments

mu	Numeric vector of cause-specific forces.
t	Time.

Value

Numeric scalar/vector.

Examples

```
tpx_tau_cf(c(0.10, 0.20), 5)
```

tqx	<i>Conditional failure probability for Tx</i>
-----	---

Description

Computes ${}_tq_x = 1 - {}_tP_x$.

Usage

tqx(t, x, model, ...)

Arguments

t	Numeric vector of durations ($t \geq 0$).
x	Numeric vector of ages ($x \geq 0$).
model	One of "uniform", "exponential", "gompertz", "makeham", "weibull".
...	Model parameters: <ul style="list-style-type: none"> • uniform: omega • exponential: lambda • gompertz: B, c • makeham: A, B, c • weibull: shape, scale

Value

Numeric vector in $[0, 1]$.

tqxj_cf	<i>Cause-specific probability ${}_tq_x^{(j)}$ under constant forces</i>
---------	--

Description

Cause-specific probability ${}_tq_x^{(j)}$ under constant forces

Usage

tqxj_cf(mu, j, t)

Arguments

mu	Numeric vector of cause-specific forces.
j	Cause index.
t	Time.

Value

Numeric scalar/vector.

Examples

tqxj_cf(c(0.10, 0.20), j = 1, t = 5)

tqxprimej_cf	<i>Single-decrement failure probability ${}_tq_x^{(j)}$ under constant force</i>
--------------	---

Description

Single-decrement failure probability ${}_tq_x^{(j)}$ under constant force

Usage

tqxprimej_cf(mu, t)

Arguments

mu	Force of decrement for cause j.
t	Time.

Value

Numeric scalar/vector.

Examples

tqxprimej_cf(0.10, 5)

tqxprime_mudd	<i>Independent probabilities ${}_tq_x^{(j)}$ under MUDD</i>
---------------	--

Description

Independent probabilities ${}_tq_x^{(j)}$ under MUDD

Usage

tqxprime_mudd(qxj, t)

Arguments

qxj	Numeric vector of dependent probabilities $q_x^{(j)}$.
t	Time in [0,1].

Value

Numeric vector.

Examples

```
tqxprime_mudd(c(.20, .10), t = 0.5)
```

tqxy	<i>Joint-life failure probability</i>
------	---------------------------------------

Description

Computes ${}_tq_{xy} = 1 - {}_tp_{xy}$.

Usage

```
tqxy(x, y, t, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
t	Time.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tqxy(40, 50, t = 10, model = "uniform", omega = 100)
```

tqxy1	<i>Probability that (x) fails before (y) within n years</i>
-------	---

Description

Computes ${}_nq_{xy}^1 = \int_0^n {}_t p_{xy} \mu_{x+t} dt$ under independence.

Usage

tqxy1(x, y, n, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term in years.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

tqxy1(40, 50, n = 10, model = "uniform", omega = 100)

tqxy2	<i>Probability that (x) fails after (y) within n years</i>
-------	--

Description

Computes ${}_nq_{xy}^2 = {}_nq_x - {}_nq_{xy}^1$.

Usage

tqxy2(x, y, n, tbl = NULL, model = NULL, ...)

Arguments

x	Age of first life.
y	Age of second life.
n	Term in years.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tqxy2(40, 50, n = 10, model = "uniform", omega = 100)
```

tqxybar	<i>Last-survivor failure probability</i>
---------	--

Description

Computes ${}_tq_{\overline{xy}} = 1 - {}_tp_{\overline{xy}}$.

Usage

```
tqxybar(x, y, t, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
t	Time.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tqxybar(40, 50, t = 10, model = "uniform", omega = 100)
```

tqx_tab	<i>Fractional failure probability from a life table</i>
---------	---

Description

Computes ${}_tq_x = 1 - {}_tp_x$ for $0 \leq t \leq 1$.

Usage

```
tqx_tab(tbl, x, t, assumption = c("udd", "cf", "balducci"))
```

Arguments

tbl	A life_table object.
x	Numeric vector of integer ages.
t	Numeric vector of fractional durations with $0 \leq t \leq 1$.
assumption	One of "udd", "cf", "balducci".

Value

Numeric vector of ${}_tq_x$ values.

tqyx1	<i>Probability that (y) fails before (x) within n years</i>
-------	---

Description

Computes ${}_nq_{xy}^1 = \int_0^n {}_tp_{xy} \mu_{y+t} dt$ under independence.

Usage

```
tqyx1(x, y, n, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
n	Term in years.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tqyx1(40, 50, n = 10, model = "uniform", omega = 100)
```

tqyx2	<i>Probability that (y) fails after (x) within n years</i>
-------	--

Description

Computes ${}_nq_{xy}^2 = {}_nq_y - {}_nq_{xy}^1$.

Usage

```
tqyx2(x, y, n, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age of first life.
y	Age of second life.
n	Term in years.
tbl	Life table.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tqyx2(40, 50, n = 10, model = "uniform", omega = 100)
```

tsVx

*Fractional-duration whole life reserve***Description**

Computes the interpolated reserve ${}_{t+s}V = ({}_tV + P_x)(1 - s) + {}_{t+1}V \cdot s$ for $0 \leq s \leq 1$.

Computes the interpolated reserve ${}_{t+s}V = ({}_tV + P_x)(1 - s) + {}_{t+1}V \cdot s$ for $0 \leq s \leq 1$.

Usage

```
tsVx(x, t, s, i, tbl = NULL, model = NULL, ...)
```

```
tsVx(x, t, s, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Issue age.
t	Integer contract duration.
s	Fractional part of duration in [0, 1].
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
tsVx(40, t = 10, s = 0.5, i = 0.05, model = "uniform", omega = 100)
tsVx(40, t = 10, s = 0.5, i = 0.05, model = "uniform", omega = 100)
```

tsVxn

*Fractional-duration endowment reserve***Description**

Computes the interpolated reserve ${}_{t+s}V = ({}_tV + P)(1 - s) + {}_{t+1}V \cdot s$ for an n-year endowment insurance.

Computes the interpolated reserve ${}_{t+s}V = ({}_tV + P)(1 - s) + {}_{t+1}V \cdot s$ for an n-year endowment insurance.

Usage

```
tsVxn(x, n, t, s, i, tbl = NULL, model = NULL, ...)
```

```
tsVxn(x, n, t, s, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Issue age.
n	Term in years.
t	Integer duration with $t < n$.
s	Fractional part in $[0, 1]$.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
tsVxn(40, n = 20, t = 10, s = 0.5, i = 0.05, model = "uniform", omega = 100)
tsVxn(40, n = 20, t = 10, s = 0.5, i = 0.05, model = "uniform", omega = 100)
```

tsVxn1	<i>Fractional-duration term reserve</i>
--------	---

Description

Computes the interpolated reserve ${}_{t+s}V = ({}_tV + P)(1 - s) + {}_{t+1}V \cdot s$ for an n-year term insurance.

Computes the interpolated reserve ${}_{t+s}V = ({}_tV + P)(1 - s) + {}_{t+1}V \cdot s$ for an n-year term insurance.

Usage

```
tsVxn1(x, n, t, s, i, tbl = NULL, model = NULL, ...)
```

```
tsVxn1(x, n, t, s, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Issue age.
n	Term in years.
t	Integer duration with $t < n$.
s	Fractional part in $[0, 1]$.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
tsVxn1(40, n = 20, t = 10, s = 0.5, i = 0.05, model = "uniform", omega = 100)
tsVxn1(40, n = 20, t = 10, s = 0.5, i = 0.05, model = "uniform", omega = 100)
```

tVbarAbarx	<i>Fully continuous whole life reserve</i>
------------	--

Description

Computes the Chapter 10 reserve for a whole life insurance with continuous premiums and immediate payment of claims.

Usage

```
tVbarAbarx(x, t, i, model, ...)
```

Arguments

x	Issue age.
t	Duration, allowed to be any nonnegative numeric value.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Details

In the fully continuous setting, reserve time t may be any nonnegative real value.

Value

Numeric vector.

Examples

```
tVbarAbarx(40, t = 10, i = 0.05, model = "uniform", omega = 100)
tVbarAbarx(40, t = c(19, 19.25, 19.5, 19.75, 20), i = 0.06,
           model = "uniform", omega = 100)
```

tVbarx	<i>Whole life reserve with continuous premiums</i>
--------	--

Description

Computes the Chapter 10 reserve for a discrete whole life insurance funded by continuous premiums.

Usage

```
tVbarx(x, t, i, model, ...)
```

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVbarx(40, t = 10, i = 0.05, model = "uniform", omega = 100)
```

tVEx

Whole life expense reserve

Description

Computes the Chapter 11 expense reserve for a fully discrete whole life insurance with renewal expenses.

Computes the Chapter 11 expense reserve for a fully discrete whole life insurance with renewal expenses.

Usage

```
tVEx(
  x,
  t,
  i,
  G,
  benefit = 1,
  renewal_premium_pct = 0,
  renewal_policy_exp = 0,
  settlement_exp = 0,
  tbl = NULL,
  model = NULL,
  ...
)
```

```
tVEx(
  x,
  t,
  i,
```

```

    G,
    benefit = 1,
    renewal_premium_pct = 0,
    renewal_policy_exp = 0,
    settlement_exp = 0,
    tbl = NULL,
    model = NULL,
    ...
)

```

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
G	Gross annual premium.
benefit	Insurance amount. Default 1.
renewal_premium_pct	Renewal percent-of-premium expense.
renewal_policy_exp	Renewal per-policy expense.
settlement_exp	Settlement expense at death.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```

tVEx(
  x = 40, t = 10, i = 0.05, G = 0.03,
  benefit = 1, renewal_premium_pct = 0.10,
  renewal_policy_exp = 0.002, settlement_exp = 0.02,
  model = "uniform", omega = 100
)
tVEx(
  x = 40, t = 10, i = 0.05, G = 0.03,
  benefit = 1, renewal_premium_pct = 0.10,
  renewal_policy_exp = 0.002, settlement_exp = 0.02,
  model = "uniform", omega = 100
)

```

tVF_x *Full preliminary term reserve for whole life insurance***Description**

Computes the FPT reserve for a whole life insurance. For whole life insurance, ${}_1V^F = 0$ and for $t \geq 1$, ${}_tV^F = {}_{t-1}V_{x+1}^{NLP}$.

Computes the FPT reserve for a whole life insurance. For whole life insurance, ${}_1V^F = 0$ and for $t \geq 1$, ${}_tV^F = {}_{t-1}V_{x+1}^{NLP}$.

Usage

```
tVFx(x, t, i, tbl = NULL, model = NULL, ...)
```

```
tVFx(x, t, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model.
...	Additional model parameters.

Value

Numeric vector.

Numeric vector.

Examples

```
tVFx(40, t = 5, i = 0.05, model = "uniform", omega = 100)
tVFx(40, t = 5, i = 0.05, model = "uniform", omega = 100)
```

tVGx	<i>Whole life gross premium reserve</i>
------	---

Description

Computes the Chapter 11 prospective gross premium reserve for a fully discrete whole life insurance with annual premiums and renewal expenses.

Computes the Chapter 11 prospective gross premium reserve for a fully discrete whole life insurance with annual premiums and renewal expenses.

Usage

```
tVGx(
  x,
  t,
  i,
  G,
  benefit = 1,
  renewal_premium_pct = 0,
  renewal_policy_exp = 0,
  settlement_exp = 0,
  tbl = NULL,
  model = NULL,
  ...
)
```

```
tVGx(
  x,
  t,
  i,
  G,
  benefit = 1,
  renewal_premium_pct = 0,
  renewal_policy_exp = 0,
  settlement_exp = 0,
  tbl = NULL,
  model = NULL,
  ...
)
```

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
G	Gross annual premium.

benefit Insurance amount. Default 1.
 renewal_premium_pct Renewal percent-of-premium expense.
 renewal_policy_exp Renewal per-policy expense.
 settlement_exp Settlement expense at death.
 tbl Optional life table object.
 model Optional parametric survival model.
 ... Additional model parameters.

Details

This function is intended for durations after issue, where future expenses are modeled through renewal premium expenses, renewal per-policy expenses, and settlement expense.

Value

Numeric vector.

Numeric vector.

Examples

```

tVGx(
  x = 40, t = 10, i = 0.05, G = 0.03,
  benefit = 1, renewal_premium_pct = 0.10,
  renewal_policy_exp = 0.002, settlement_exp = 0.02,
  model = "uniform", omega = 100
)
tVGx(
  x = 40, t = 10, i = 0.05, G = 0.03,
  benefit = 1, renewal_premium_pct = 0.10,
  renewal_policy_exp = 0.002, settlement_exp = 0.02,
  model = "uniform", omega = 100
)

```

$tVnAdotx$

Deferred annuity-due reserve

Description

Computes the Chapter 10 reserve ${}_tV({}_n|\ddot{a}_x)$ for $t < n$.

Usage

$tVnAdotx(x, n, t, i, model, \dots)$

Arguments

x	Issue age.
n	Deferral period.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVnAdotx(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)
```

tV_{nax}	<i>Deferred annuity-immediate reserve</i>
------------	---

Description

Computes the reserve for an n-year deferred annuity-immediate for $t < n$.

Usage

```
tVnax(x, n, t, i, model, ...)
```

Arguments

x	Issue age.
n	Deferral period.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVnax(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)
```

 $tVnEx$ *Pure endowment net level premium reserve*

Description

Computes the Chapter 10 prospective reserve for an n-year pure endowment.

Usage

$tVnEx(x, n, t, i, model, \dots)$

Arguments

x	Issue age.
n	Term in years.
t	Duration.
i	Effective annual interest rate.
$model$	Survival model.
\dots	Additional model parameters.

Value

Numeric vector.

Examples

$tVnEx(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)$

 tVx *Whole life net level premium reserve*

Description

Computes the Chapter 10 prospective reserve for a whole life insurance with annual premiums: reserve at duration t equals future APV of benefits minus future APV of net premiums.

Usage

$tVx(x, t, i, model, \dots)$

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVx(40, t = 10, i = 0.05, model = "uniform", omega = 100)
```

tV_{xn}	<i>Endowment insurance net level premium reserve</i>
-----------	--

Description

Computes the Chapter 10 prospective reserve for an n-year endowment insurance.

Usage

```
tVxn(x, n, t, i, model, ...)
```

Arguments

x	Issue age.
n	Term in years.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVxn(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)
```

tV_{xn1}	<i>Term insurance net level premium reserve</i>
------------	---

Description

Computes the Chapter 10 prospective reserve for an n-year term insurance.

Usage

$tV_{xn1}(x, n, t, i, model, \dots)$

Arguments

x	Issue age.
n	Term in years.
t	Duration.
i	Effective annual interest rate.
$model$	Survival model.
\dots	Additional model parameters.

Value

Numeric vector.

Examples

$tV_{xn1}(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)$

tV_{xn1_ret}	<i>Term insurance reserve by retrospective method</i>
-----------------	---

Description

Computes the retrospective term reserve for $t \leq n$.

Usage

$tV_{xn1_ret}(x, n, t, i, model, \dots)$

Arguments

x	Issue age.
n	Term in years.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVxn1_ret(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)
```

tV_{xn_ret}	<i>Endowment insurance reserve by retrospective method</i>
----------------	--

Description

Computes the Chapter 10 retrospective reserve ${}_tV_{x:\overline{n}|} = P_{x:\overline{n}|}\ddot{s}_{x:\overline{t}|} - {}_tk_x$ for $t \leq n$.

Usage

```
tVxn_ret(x, n, t, i, model, ...)
```

Arguments

x	Issue age.
n	Term in years.
t	Duration.
i	Effective annual interest rate.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
tVxn_ret(40, n = 20, t = 10, i = 0.05, model = "uniform", omega = 100)
```

tVx_m	<i>Whole life reserve with m-thly premiums</i>
---------	--

Description

Computes the Chapter 10 reserve for a whole life insurance funded by true m-thly premiums.

Usage

$tVx_m(x, t, m, i, model, \dots)$

Arguments

x	Issue age.
t	Duration.
m	Number of premium payments per year.
i	Effective annual interest rate.
<code>model</code>	Survival model.
\dots	Additional model parameters.

Value

Numeric vector.

Examples

$tVx_m(40, t = 10, m = 12, i = 0.05, model = "uniform", omega = 100)$

tVx_{ret}	<i>Whole life net level premium reserve by retrospective method</i>
-------------	---

Description

Computes the Chapter 10 retrospective reserve $tV_x = P_x \ddot{s}_{x:\bar{t}|} - tk_x$.

Usage

$tVx_{ret}(x, t, i, model, \dots)$

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
<code>model</code>	Survival model.
\dots	Additional model parameters.

Value

Numeric vector.

Examples

```
tVx_ret(40, t = 10, i = 0.05, model = "uniform", omega = 100)
```

udd_continuous_multiplier

UDD multiplier for continuous insurance approximations

Description

Under UDD, $\bar{A}_x = (i/\delta)A_x$ and similarly for term and deferred insurance.

Usage

```
udd_continuous_multiplier(i)
```

Arguments

i Numeric vector of effective annual interest rates.

Value

Numeric vector equal to i/δ .

udd_mthly_multiplier

UDD multiplier for m-thly insurance approximations

Description

Under UDD, $A_x^{(m)} = (i/i^{(m)})A_x$ and similarly for term and deferred insurance.

Usage

```
udd_mthly_multiplier(i, m)
```

Arguments

i Numeric vector of effective annual interest rates.

m Positive integer payment frequency.

Value

Numeric vector equal to $i/i^{(m)}$.

varLtx	<i>Variance of present value of loss at duration t for whole life insurance</i>
--------	---

Description

Computes the Chapter 10 conditional variance $\text{Var}({}_tL_x \mid K_x \geq t)$ for a fully discrete whole life insurance.

Usage

```
varLtx(x, t, i, P, model, ...)
```

Arguments

x	Issue age.
t	Duration.
i	Effective annual interest rate.
P	Annual premium.
model	Survival model.
...	Additional model parameters.

Value

Numeric vector.

Examples

```
prem <- Px(40, i = 0.05, model = "uniform", omega = 100)
varLtx(40, t = 10, i = 0.05, P = prem, model = "uniform", omega = 100)
```

var_Abarx	<i>Variance of continuous whole life insurance PV</i>
-----------	---

Description

Computes $\text{Var}(\bar{Z}_x) = {}^2\bar{A}_x - \bar{A}_x^2$.

Usage

```
var_Abarx(x, i, model, ...)
```

Arguments

x	Age.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of variances.

var_Abarxn	<i>Variance of continuous endowment insurance PV</i>
------------	--

Description

Variance of continuous endowment insurance PV

Usage

```
var_Abarxn(x, n, i, model, ...)
```

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of variances.

var_Abarxn1	<i>Variance of continuous term insurance PV</i>
-------------	---

Description

Computes $\text{Var}(\bar{Z}_{x:\bar{n}}^1) = {}^2\bar{A}_{x:\bar{n}}^1 - (\bar{A}_{x:\bar{n}}^1)^2$.

Usage

```
var_Abarxn1(x, n, i, model, ...)
```

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of variances.

var_Ax	<i>Variance of whole life insurance PV</i>
--------	--

Description

Computes $\text{Var}(Z_x) = {}^2A_x - A_x^2$.

Usage

```
var_Ax(x, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)
```

Arguments

x	Age.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.
tol	Numerical tolerance for truncating infinite sums.
k_max	Maximum number of terms in the sum.

Value

Numeric vector of variances.

var_Axn	<i>Variance of endowment insurance PV</i>
---------	---

Description

Variance of endowment insurance PV

Usage

var_Axn(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of variances.

var_Axn1	<i>Variance of term insurance PV</i>
----------	--------------------------------------

Description

Computes $\text{Var}(Z_{x:\overline{n}|}^1) = {}^2A_{x:\overline{n}|}^1 - (A_{x:\overline{n}|}^1)^2$.

Usage

var_Axn1(x, n, i, tbl = NULL, model = NULL, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of variances.

var_Axn1_m	<i>Variance of m-thly term insurance PV</i>
------------	---

Description

Variance of m-thly term insurance PV

Usage

var_Axn1_m(x, n, i, m, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of variances.

var_Axn_m	<i>Variance of m-thly endowment insurance PV</i>
-----------	--

Description

Variance of m-thly endowment insurance PV

Usage

var_Axn_m(x, n, i, m, model, ...)

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of variances.

var_Ax_m	<i>Variance of m-thly whole life insurance PV</i>
----------	---

Description

Computes $\text{Var}(Z_x^{(m)}) = {}^2A_x^{(m)} - (A_x^{(m)})^2$.

Usage

```
var_Ax_m(x, i, m, model, ..., tol = 1e-12, j_max = 100000L)
```

Arguments

x	Age.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.
tol	Numerical tolerance for truncating the infinite sum.
j_max	Maximum number of m-thly intervals in the sum.

Value

Numeric vector of variances.

var_nAbarx	<i>Variance of continuous deferred insurance PV</i>
------------	---

Description

Variance of continuous deferred insurance PV

Usage

```
var_nAbarx(x, n, i, model, ...)
```

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.

Value

Numeric vector of variances.

var_nAx	<i>Variance of deferred insurance PV</i>
---------	--

Description

Variance of deferred insurance PV

Usage

```
var_nAx(x, n, i, tbl = NULL, model = NULL, ..., tol = 1e-12, k_max = 5000)
```

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.
tol	Numerical tolerance for truncating infinite sums.
k_max	Maximum number of terms in the sum.

Value

Numeric vector of variances.

var_nAx_m	<i>Variance of m-thly deferred insurance PV</i>
-----------	---

Description

Variance of m-thly deferred insurance PV

Usage

```
var_nAx_m(x, n, i, m, model, ..., tol = 1e-12, j_max = 100000L)
```

Arguments

x	Age.
n	Deferral period.
i	Effective annual interest rate.
m	Positive integer payment frequency.
model	Parametric survival model name.
...	Additional model parameters passed to survival-model functions.
tol	Numerical tolerance for truncating the infinite sum.
j_max	Maximum number of m-thly intervals in the sum.

Value

Numeric vector of variances.

var_nEx	<i>Variance of pure endowment PV</i>
---------	--------------------------------------

Description

Variance of pure endowment PV

Usage

```
var_nEx(x, n, i, tbl = NULL, model = NULL, ...)
```

Arguments

x	Age.
n	Term.
i	Effective annual interest rate.
tbl	Optional life table object.
model	Optional parametric survival model name.
...	Additional arguments passed to survival-model functions.

Value

Numeric vector of variances.

<code>Vprefloor_crvm_ul</code>	<i>Pre-floor CRVM reserve for universal life</i>
--------------------------------	--

Description

Computes the pre-floor CRVM reserve from Equation (16.17).

Usage

`Vprefloor_crvm_ul(r, pvfb_minus_pvfp)`

Arguments

<code>r</code>	Ratio r_t .
<code>pvfb_minus_pvfp</code>	Difference $(PVFB)_t - (PVFP)_t$.

Value

Numeric scalar.

Examples

`Vprefloor_crvm_ul(r = 0.33506, pvfb_minus_pvfp = 70)`

<code>vt_var</code>	<i>Discount factors under a variable annual interest scenario</i>
---------------------	---

Description

Computes the sequence of discount factors

$$v_1, v_2, \dots, v_n$$

where

$$v_t = \prod_{k=1}^t (1 + i_k)^{-1}.$$

Usage

`vt_var(i)`

Arguments

`i` Numeric vector of annual effective interest rates i_1, i_2, \dots, i_n .

Details

This corresponds to the Chapter 15 notation ${}_jv^t$ for a fixed scenario j .

Value

Numeric vector of discount factors of the same length as `i`.

Examples

```
vt_var(c(0.06, 0.07, 0.08))
```

V_zeroized	<i>Zeroized reserves for a discrete death-only contract</i>
------------	---

Description

Computes the zeroized reserve sequence by backward recursion, setting negative reserves equal to zero.

Usage

```
V_zeroized(qx, i, G, benefit, r = 0, e = 0, V_terminal = 0, floor_zero = TRUE)
```

Arguments

<code>qx</code>	Mortality vector.
<code>i</code>	Interest-rate vector.
<code>G</code>	Gross premium vector.
<code>benefit</code>	Death-benefit vector.
<code>r</code>	Percent-of-premium expense vector.
<code>e</code>	Fixed-expense vector.
<code>V_terminal</code>	Terminal reserve. Defaults to 0.
<code>floor_zero</code>	Logical; if TRUE, negative reserves are reset to 0.

Details

For a death-only contract with no settlement expense and no second decrement, the recursion sets

$$Pr_{t+1} = ({}_tV^Z + G_{t+1}(1 - r_{t+1}) - e_{t+1})(1 + i_{t+1}) - [Bq_{x+t} + {}_{t+1}V^Z p_{x+t}]$$

equal to zero, solving backward for ${}_tV^Z$.

Value

Numeric vector of zeroized reserves of length $n + 1$.

Examples

```
V_zeroized(
  qx = c(.015, .017, .019, .021, .024),
  i = 0.06,
  G = 19279,
  benefit = 1000000,
  e = 240
)
```

z_from_coupon_annual *Bootstrap annual spot rates from annual coupon-bond yields*

Description

Bootstraps annual effective zero-coupon yields from par annual coupon-bearing bond yields of the same maturities.

Usage

```
z_from_coupon_annual(maturity, coupon_yield, par = 1000)
```

Arguments

maturity	Integer vector of maturities in years, in increasing order.
coupon_yield	Numeric vector of annual coupon yields.
par	Par value of each bond.

Value

Numeric vector of annual effective spot rates.

Examples

```
maturity <- 1:4
coupon_yield <- c(0.02, 0.04, 0.06, 0.08)
z_from_coupon_annual(maturity, coupon_yield)
```

z_from_coupon_semi *Bootstrap semiannual nominal spot rates from coupon-bond yields*

Description

Bootstraps the semiannual nominal annual zero-coupon yields from par coupon-bearing bond yields of the same maturities.

Usage

```
z_from_coupon_semi(maturity, coupon_yield, par = 1000)
```

Arguments

maturity	Numeric vector of maturities in years, typically 0.5, 1.0, 1.5, . . . , in increasing order.
coupon_yield	Numeric vector of nominal annual coupon yields convertible semiannually.
par	Par value of each bond.

Details

Both coupon yields and spot yields are interpreted as nominal annual rates convertible semiannually.

Value

Numeric vector of semiannual nominal annual spot rates.

Examples

```
maturity <- c(0.5, 1.0, 1.5, 2.0)
coupon_yield <- c(0.0244, 0.0260, 0.0276, 0.0293)
z_from_coupon_semi(maturity, coupon_yield)
```

z_from_fn1 *Spot rates from forward one-year rates*

Description

Converts annual effective forward one-year rates $f_{0,1}, f_{1,1}, \dots, f_{n-1,1}$ into annual effective spot rates z_1, z_2, \dots, z_n .

Usage

```
z_from_fn1(fn1)
```

Arguments

fn1 Numeric vector of annual effective forward one-year rates.

Details

Since

$$(1 + z_n)^n = \prod_{j=0}^{n-1} (1 + f_{j,1}),$$

the spot rates are recovered directly.

Value

Numeric vector of annual effective spot rates.

Examples

```
z_from_fn1(c(0.04, 0.05, 0.06, 0.07, 0.08))
```

Index

A2barx, 7
A2barxn, 8
A2barxn1, 9
A2nAbarx, 9
A2nAx, 10
A2nAx_m, 10
A2nEx, 11
A2x, 12
A2x_m, 14
A2xn, 12
A2xn1, 13
A2xn1_m, 13
A2xn_m, 14
AAL_PUC_db, 15
AAL_TUC_db, 16
AB_cae, 27
AB_fas, 28
Abarx, 16
abarx (annuity_annual), 33
Abarx_udd, 24
abarx_udd (annuity_approximations_udd),
36
abarx_woolhouse2
(annuity_approximations_woolhouse2),
38
abarx_woolhouse3
(annuity_approximations_woolhouse3),
39
abarx_y, 24
Abarxj_md, 17
Abarxn, 18
abarxn (annuity_annual), 33
Abarxn1, 18
Abarxn1_udd, 19
Abarxn_udd, 19
abarxn_udd
(annuity_approximations_udd),
36
Abarxy, 20
abarxy (abarxy_ch12), 23
Abarxy1, 20
Abarxy2, 21
abarxy_ch12, 23
Abarxybar, 22
abarxybar (abarxybar_ch12), 22
abarxybar_ch12, 22
abary_x, 26
Abaryx1, 25
Abaryx2, 26
adotx (annuity_annual), 33
adotx_m (annuity_mthly_whole_due), 46
adotx_m_udd
(annuity_approximations_udd),
36
adotx_m_woolhouse2
(annuity_approximations_woolhouse2),
38
adotx_m_woolhouse3
(annuity_approximations_woolhouse3),
39
adotxn (annuity_annual), 33
adotxn_m (annuity_mthly_temp_due), 44
adotxn_m_udd
(annuity_approximations_udd),
36
adotxn_m_woolhouse2
(annuity_approximations_woolhouse2),
38
adotxn_m_woolhouse3
(annuity_approximations_woolhouse3),
39
adotxy, 28
adotxybar, 29
adotxybarn, 30
adotxyn, 30
ag38_prefunding_ratio, 31
ag38_reserve_ul, 32
alphaF, 33

- annuity_annual, 33
- annuity_approximations, 36
- annuity_approximations_udd, 36
- annuity_approximations_woolhouse2, 38
- annuity_approximations_woolhouse3, 39
- annuity_certain, 40
- annuity_identity_abarx
(annuity_relationships), 48
- annuity_identity_abarxn
(annuity_relationships), 48
- annuity_identity_adotx
(annuity_relationships), 48
- annuity_identity_adotxn
(annuity_relationships), 48
- annuity_identity_ax
(annuity_relationships), 48
- annuity_identity_axn
(annuity_relationships), 48
- annuity_identity_nabarx
(annuity_relationships), 48
- annuity_identity_nadotx
(annuity_relationships), 48
- annuity_identity_nax
(annuity_relationships), 48
- annuity_mthly, 41
- annuity_mthly_accum_due, 41
- annuity_mthly_accum_immediate, 42
- annuity_mthly_deferred_due, 42
- annuity_mthly_deferred_immediate, 43
- annuity_mthly_temp_due, 44
- annuity_mthly_temp_immediate, 45
- annuity_mthly_whole_due, 46
- annuity_mthly_whole_immediate, 47
- annuity_relationships, 48
- annuity_varying_payments, 49
- APV_gross_premiums, 51
- APV_NR_db, 52
- AS_path, 53
- AS_path_md, 54
- AV_path_ul_typeA, 55
- AV_path_ul_typeB, 56
- AVz_dc, 54
- Ax, 57
- ax (annuity_annual), 33
- ax_improved, 70
- Ax_m, 70
- ax_m (annuity_mthly_whole_immediate), 47
- Ax_m_udd, 71
- ax_m_udd (annuity_approximations_udd),
36
- ax_m_woolhouse2
(annuity_approximations_woolhouse2),
38
- ax_m_woolhouse3
(annuity_approximations_woolhouse3),
39
- ax_y, 71
- Axj_md, 57
- Axn, 58
- axn (annuity_annual), 33
- Axn1, 59
- Axn1_m, 59
- Axn1_m_udd, 60
- Axn1_spot
(chapter15_spot_interest_apv),
74
- Axn1_var
(chapter15_variable_interest_apv),
76
- axn_improved, 60
- Axn_m, 61
- axn_m (annuity_mthly_temp_immediate), 45
- Axn_m_udd, 62
- axn_m_udd (annuity_approximations_udd),
36
- axn_m_woolhouse2
(annuity_approximations_woolhouse2),
38
- axn_m_woolhouse3
(annuity_approximations_woolhouse3),
39
- Axn_spot (chapter15_spot_interest_apv),
74
- axn_spot (chapter15_spot_interest_apv),
74
- Axn_var
(chapter15_variable_interest_apv),
76
- axn_var
(chapter15_variable_interest_apv),
76
- Axy, 62
- axy (axy_ch12), 69
- axy_ch12, 69
- Axybar, 63
- axybar (axybar_ch12), 66

- axybar_ch12, 66
- Axybarn, 64
- axybarn (axybarn_ch12), 65
- Axybarn1, 64
- axybarn_ch12, 65
- Axyn, 67
- axyn (axyn_ch12), 68
- Axyn1, 67
- axyn_ch12, 68
- ay_x, 72

- betaF, 73

- ch12_multilife_cont_params, 73
- chapter15_spot_interest_apv, 74
- chapter15_variable_interest_apv, 76
- coi_ul_typeB, 77
- contribution_rate_target, 78
- cov_term_deferred, 79
- cov_term_endow, 79
- cumhaz0, 80

- Dabarxn (annuity_varying_payments), 49
- DABarxn1, 81
- Dadotxn (annuity_varying_payments), 49
- Daxn (annuity_varying_payments), 49
- DAxn1, 81
- DbarAbarxn1, 82
- decompGg_disc, 82
- discount, 84
- discounted_payback_period, 85
- dist0, 86
- double_force_delta, 86
- double_force_i, 87
- dx, 87
- dxj, 88
- dxtau, 88

- EL0barAbarx (premium_ch9), 136
- EL0x (premium_ch9), 136
- EL0xn (premium_ch9), 136
- EL0xn1 (premium_ch9), 136
- ELtx, 89
- ex_complete, 89
- ex_complete_tab, 90
- ex_curtate, 91
- ex_curtate_tab, 91
- ex_temp_complete_tab, 92
- ex_temp_curtate_tab, 92

- F0 (dist0), 86
- f0 (dist0), 86
- fnk_from_z, 93
- forward_matrix_from_z, 93
- fx, 94
- fx_tab, 95

- gain_loss_md, 95
- GI_cont, 97
- GI_disc, 97
- GM_cont, 99
- GM_disc, 99
- GMF_rollforward_ul, 98
- GT_cont, 101
- GT_disc, 102
- GTg_disc, 100
- Gx (premium_ch9), 136

- hazard0, 103
- htVnAx
 - (reserve_deferred_insurance_ch10), 152
- htVx, 103

- i_credit_eiul, 110
- IABarx, 104
- Iabarx (annuity_varying_payments), 49
- Iabarxn (annuity_varying_payments), 49
- Iadotx (annuity_varying_payments), 49
- Iadotxn (annuity_varying_payments), 49
- IAx, 104
- Iax (annuity_varying_payments), 49
- Iaxn (annuity_varying_payments), 49
- IAxn1, 105
- IbarAbarx, 106
- IbarAbarxn1, 106
- iMA_eiul, 107
- Income_dc, 107
- interest_convert, 108
- iP_eiul, 109
- IRR_profit, 109

- life_table, 111
- lx, 111
- lx_select, 112
- lx_to_S0, 112

- markov_nstep_prob, 113
- md_table, 113

- meanVx, 114
- mortality_improvement, 115
- multilife_ch12, 115
- multilife_ch12_ext, 115
- mux_tab, 116

- nAbarx, 116
- nabarx (annuity_annual), 33
- nAbarx_udd, 117
- nabarx_udd
 - (annuity_approximations_udd), 36
- nadotx (annuity_annual), 33
- nadotx_m (annuity_mthly_deferred_due), 42
- nadotx_m_udd
 - (annuity_approximations_udd), 36
- nadotx_m_woolhouse2
 - (annuity_approximations_woolhouse2), 38
- nadotx_m_woolhouse3
 - (annuity_approximations_woolhouse3), 39
- nAx, 117
- nax (annuity_annual), 33
- nAx_m, 118
- nax_m
 - (annuity_mthly_deferred_immediate), 43
- nAx_m_udd, 119
- nax_m_udd (annuity_approximations_udd), 36
- nax_m_woolhouse2
 - (annuity_approximations_woolhouse2), 38
- nax_m_woolhouse3
 - (annuity_approximations_woolhouse3), 39
- naxn_improved, 118
- NC_EAN_db, 120
- NC_PUC_db, 120
- NC_TUC_db, 121
- ndx, 122
- nEx, 122
- nEx_spot (chapter15_spot_interest_apv), 74
- nEx_var
 - (chapter15_variable_interest_apv), 76
- nExy, 123
- nExybar, 123
- nkqx, 124
- nmxq, 125
- nmxq_select, 125
- NPV_partial, 126
- NPV_profit, 126
- npx, 127
- npx_select, 128
- npxtau_md, 128
- nqx, 129
- nqx_select, 131
- nqxj_md, 129
- nqxtau_md, 130

- PAB_cae, 131
- PAB_fas, 132
- Pbar_trapz_ms, 133
- PbarAbarx (premium_ch9), 136
- PbarAbarxn (premium_ch9), 136
- PbarAbarxn1 (premium_ch9), 136
- Pbarx (premium_ch9), 136
- Pbarxn (premium_ch9), 136
- Pbarxn1 (premium_ch9), 136
- Pi_signature, 134
- PnAdotx, 135
- PnAx (premium_ch9), 136
- Pnax, 135
- PnAx_m (premium_ch9), 136
- PnEx (premium_ch9), 136
- Pr_vector_disc, 140
- premium_ch9, 136
- profit_margin, 140
- pv_cashflows, 142
- pv_spot_cashflows, 142
- Px (premium_ch9), 136
- Px_m (premium_ch9), 136
- px_proj, 145
- px_to_lx, 145
- Pxn (premium_ch9), 136
- Pxn1 (premium_ch9), 136
- Pxn1_m (premium_ch9), 136
- Pxn_m (premium_ch9), 136
- pxtau, 143
- pxtau_ul, 144

- qx_dep_cf, 147
- qx_dep_sudd, 148

- qx_proj, 148
- qx_tab, 149
- qx_to_lx, 149
- qxprime_mudd, 146
- qxprime_sudd, 146
- qxtau, 147

- replacement_ratio_db, 150
- replacement_ratio_dc, 150
- reserve_ch10, 151
- reserve_ch10_ext, 151
- reserve_ch11, 151
- reserve_ch11_ext, 152
- reserve_deferred_insurance_ch10, 152
- rt_ul, 153

- S0, 153
- S0_to_lx, 154
- salary_scale, 154
- sbarxn (annuity_annual), 33
- sdotxn (annuity_annual), 33
- sdotxn_m (annuity_mthly_accum_due), 41
- sdotxn_m_udd
 - (annuity_approximations_udd), 36
- sdotxn_m_woolhouse2
 - (annuity_approximations_woolhouse2), 38
- select_life_table, 155
- solve_yield, 155
- sxn (annuity_annual), 33
- sxn_m (annuity_mthly_accum_immediate), 42
- sxn_m_udd (annuity_approximations_udd), 36
- sxn_m_woolhouse2
 - (annuity_approximations_woolhouse2), 38

- thiele_backward_path, 156
- thiele_backward_step, 157
- thiele_dVdt, 157
- thiele_dVdt_01, 158
- thiele_path_01, 159
- tp00_tp01_euler, 160
- tPnAx (premium_ch9), 136
- tPnEx (premium_ch9), 136
- tPx (premium_ch9), 136
- tpx, 161
- tpx_improved, 165
- tpx_tab, 166
- tpx_tau_cf, 166
- tPxn (premium_ch9), 136
- tPxn1 (premium_ch9), 136
- tpxprimej_cf, 162
- tpxtau_ul, 163
- tpxy, 163
- tpxybar, 164
- tqx, 167
- tqx_tab, 172
- tqxj_cf, 167
- tqxprime_mudd, 168
- tqxprimej_cf, 168
- tqxy, 169
- tqxy1, 170
- tqxy2, 170
- tqxybar, 171
- tqyx1, 172
- tqyx2, 173
- tsVx, 174
- tsVxn, 175
- tsVxn1, 176
- tVbarAbarx, 177
- tVbarx, 177
- tVEx, 178
- tVfx, 180
- tVGx, 181
- tVnAdotx, 182
- tVnAx
 - (reserve_deferred_insurance_ch10), 152
- tVnax, 183
- tVnEx, 184
- tVx, 184
- tVx_m, 188
- tVx_ret, 188
- tVxn, 185
- tVxn1, 186
- tVxn1_ret, 186
- tVxn_ret, 187

- udd_continuous_multiplier, 189
- udd_mthly_multiplier, 189

- V_zeroized, 199
- var_Abarx, 190
- var_Abarxn, 191
- var_Abarxn1, 192

var_Ax, 192
var_Ax_m, 195
var_Axn, 193
var_Axn1, 193
var_Axn1_m, 194
var_Axn_m, 194
var_nAbarx, 195
var_nAx, 196
var_nAx_m, 197
var_nEx, 197
varL0barAbarx (premium_ch9), 136
varL0x (premium_ch9), 136
varL0xn (premium_ch9), 136
varL0xn1 (premium_ch9), 136
varLtx, 190
Vprefloor_crvm_u1, 198
vt_var, 198

z_from_coupon_annual, 200
z_from_coupon_semi, 201
z_from_fn1, 201