

Package ‘SimBEL’

May 5, 2021

Type Package

Title Un package de calcul du best estimate epargne sous Solvabilite 2

Version 3.2.2

Description Un modele de simulation Monte-Carlo s'appuyant sur une projection d'un canton (actif et passif) permettant l'evaluation des provisions best estimate d'un contrat d'epargne et re-traité français en euros. Plusieurs chocs de la formule standard peuvent être effectués.

Author Prim'Act

URL <http://github.com/gquibert/SimBEL>

BugReports <http://github.com/gquibert/SimBEL/issues>

Maintainer Quentin Guibert <quentin.guibert@primact.fr>

Depends R (>= 4.0.1), rootSolve(>= 1.7), doParallel(>= 1.0.10),
RSQLite(>= 2.0), reshape2(>= 1.4.2), plyr(>= 1.8.4)

Imports Rcpp, utils, methods

Suggests

LinkingTo Rcpp

License GPL-2

LazyData TRUE

RoxygenNote 6.1.1

Collate 'Action_class.R' 'Action_buy.R' 'Action_calc_pmv1.R'
'Action_calc_vm.R' 'Action_internal.R' 'Action_load.R'
'Action_revalo.R' 'Action_sell.R' 'Action_sell_pvl.R'
'Action_update_dur_det.R' 'Action_update_vm.R'
'AlmEngine_class.R' 'AlmEngine_create_ptf_bought_action.R'
'Immo_class.R' 'AlmEngine_create_ptf_bought_immo.R'
'Oblig_class.R' 'AlmEngine_create_ptf_bought_oblig.R'
'AlmEngine_do_calc_nb_sold_action.R'
'AlmEngine_do_calc_nb_sold_immo.R'
'AlmEngine_do_calc_nb_sold_oblig.R' 'AlmEngine_internal.R'
'Treso_class.R' 'RC_class.R' 'PRE_class.R' 'FraisFin_class.R'
'PortFin_class.R' 'AlmEngine_reallocate.R' 'AlmEngine_update.R'
'AutresPassifs-class.R' 'AutresPassifs-internal.R'
'AutresPassifs-load.R' 'AutresPassifs-proj_annee.R'
'AutresReserves-class.R' 'AutresReserves-init_debut_pgg_psap.R'
'AutresReserves-internal.R' 'AutresReserves-load.R'

'AutresReserves-update_reserves.R' 'DataBase_class.R'
 'ESG_class.R' 'ParamBe_class.R' 'ParamRevaloEngine_class.R'
 'ParamAlmEngine_class.R' 'HypCanton_class.R' 'Ppb_class.R'
 'ModelPointESG_class.R' 'TabProbaEpEuroInd-class.R'
 'TabEpEuroInd-class.R' 'EpEuroInd-class.R' 'TauxPB-class.R'
 'FraisPassif-class.R' 'ParamRachDyn-class.R'
 'ParamComport-class.R' 'ParamTableRach-class.R'
 'ParamTableMort-class.R' 'HypTech-class.R' 'PortPassif-class.R'
 'Canton_class.R' 'Be_class.R' 'Be-run_be.R' 'Be-run_be_simu.R'
 'Be-write_results.R' 'Be_data.R' 'Be_internal.R'
 'Canton_calc_fin_proj.R' 'Canton_calc_result_technique_ap_pb.R'
 'Canton_data.R' 'Canton_internal.R' 'Canton_proj_an.R'
 'ParamChocSousc-class.R' 'ParamChocMket_class.R'
 'ChocSolvabilite2_class.R'
 'ChocSolvabilite2_do_choc_action_type1.R'
 'ChocSolvabilite2_do_choc_action_type2.R'
 'ChocSolvabilite2_do_choc_currency.R'
 'ChocSolvabilite2_do_choc_frais.R'
 'ChocSolvabilite2_do_choc_immo.R'
 'ChocSolvabilite2_do_choc_longevite.R'
 'ChocSolvabilite2_do_choc_mortalite.R'
 'ChocSolvabilite2_do_choc_rachat_down.R'
 'ChocSolvabilite2_do_choc_rachat_mass.R'
 'ChocSolvabilite2_do_choc_rachat_up.R'
 'ChocSolvabilite2_do_choc_spread.R'
 'ChocSolvabilite2_do_choc_spread_unitaire.R'
 'ChocSolvabilite2_do_choc_taux.R' 'ChocSolvabilite2_internal.R'
 'ChocSolvabilite2_load.R' 'DataBase_init_tables.R'
 'DataBase_insert_tables.R' 'DataBase_internal.R'
 'DataBase_merge_actifs.R' 'DataBase_merge_be.R'
 'DataBase_merge_flux_fin.R' 'DataBase_merge_pb.R'
 'DataBase_merge_table_be.R' 'DataBase_merge_table_produit.R'
 'DataBase_update_table_output_produit.R'
 'DataBase_update_table_pb.R' 'ESG-get_choc_inflation_frais.R'
 'ESG_internal.R' 'ESG_chargement.R' 'ESG_extract_mp.R'
 'EpEuroInd-calc_proba_dyn.R' 'EpEuroInd-internal.R'
 'EpEuroInd_load.R' 'FraisFin_calc.R' 'FraisFin_internal.R'
 'FraisFin_load.R' 'FraisPassif-calc_frais.R'
 'FraisPassif-internal.R' 'FraisPassif-load.R'
 'HypCanton_internal.R' 'HypCanton_load.R'
 'HypTech-convert_table.R' 'HypTech-get_choc_rach.R'
 'HypTech-get_choc_table.R' 'HypTech-get_comport.R'
 'HypTech-get_qx_mort.R' 'HypTech-get_qx_rach.R'
 'HypTech-get_rach_dyn.R' 'HypTech-internal.R'
 'Initialisation_class.R' 'Initialisation_load.R'
 'HypTech-load.R' 'Immo_buy.R' 'Immo_calc_pmv1.R'
 'Immo_calc_vm.R' 'Immo_internal.R' 'Immo_load.R'
 'Immo_revalo.R' 'Immo_sell.R' 'Immo_update_dur_det.R'
 'Immo_update_vm.R' 'Initialisation_create_folder.R'
 'Initialisation_initSimBEL.R' 'Initialisation_init_scenario.R'
 'Initialisation_internal.R' 'Initialisation_load_be.R'
 'Initialisation_set_architecture.R' 'ModelPointESG_internal.R'

'Oblig_buy.R' 'Oblig_calc_coupon.R' 'Oblig_calc_dur.R'
'Oblig_calc_nominal.R' 'Oblig_calc_pmv1.R'
'Oblig_calc_sur_dec.R' 'Oblig_calc_vm.R' 'Oblig_calc_vnc.R'
'Oblig_calc_z_spread.R' 'Oblig_flux_annee.R' 'Oblig_internal.R'
'Oblig_load.R' 'Oblig_sell.R' 'Oblig_update_cc.R'
'Oblig_update_dur.R' 'Oblig_update_mat_res.R'
'Oblig_update_sd.R' 'Oblig_update_vm.R' 'Oblig_update_vnc.R'
'Oblig_update_zsp.R' 'Oblig_yield_to_maturity.R' 'PRE_calc.R'
'PRE_do_update_val_courante.R' 'PRE_do_update_val_debut.R'
'PRE_internal.R' 'PRE_load.R' 'ParamAlmEngine_internal.R'
'ParamAlmEngine_load.R' 'ParamBe_internal.R'
'ParamChocMket_internal.R' 'ParamChocSousc-internal.R'
'ParamComport-internal.R' 'ParamRachDyn-internal.R'
'ParamRevaloEngine_internal.R' 'ParamRevaloEngine_load.R'
'ParamTableMort-calc_coef_rente.R'
'ParamTableMort-calc_proba_decès.R'
'ParamTableMort-calc_proba_survie.R' 'ParamTableMort-calc_qx.R'
'ParamTableMort-internal.R' 'ParamTableRach-calc_rach.R'
'ParamTableRach-internal.R' 'TabProbaRetEuroRest-class.R'
'TabRetEuroRest_class.R' 'RetraiteEuroRest_class.R'
'PassifBase-calc_pm.R' 'PassifBase-calc_prest.R'
'PassifBase-calc_primes.R' 'PassifBase-calc_proba_flux.R'
'PassifBase-calc_revalo_pm.R' 'PassifBase-calc_tx_cible.R'
'PassifBase-calc_tx_min.R' 'PassifBase-vieilli_mp.R'
'PortFin_calc_pmv1.R' 'PortFin_calc_rdt.R'
'PortFin_calc_resultat_fin.R' 'PortFin_calc_tra.R'
'PortFin_chargement.R' 'PortFin_chargement_reference.R'
'PortFin_do_update_pmv1.R'
'PortFin_do_update_vm_vnc_precedent.R' 'PortFin_internal.R'
'PortFin_print_alloc.R' 'PortFin_update.R'
'PortFin_update_reference.R' 'PortFin_vieillissement_action.R'
'PortFin_vieillissement_immo.R'
'PortFin_vieillissement_oblig.R'
'PortFin_vieillissement_treso.R'
'PortPassif-calc_rdt_marche_ref.R' 'PortPassif-internal.R'
'PortPassif-load.R' 'PortPassif-proj_annee_av_pb.R'
'PortPassif-vieillissement_ap_pb.R'
'PortPassif-vieillissement_av_pb.R' 'Ppb_calc_ppb_attrib.R'
'Ppb_dotation_reprise.R' 'Ppb_internal.R' 'Ppb_load.R'
'Ppb_ppb_8ans.R' 'Ppb_vieillissement_ppb.R' 'RC_calc.R'
'RC_do_update_val_courante.R' 'RC_do_update_val_debut.R'
'RC_internal.R' 'RC_load.R' 'RcppExports.R'
'RetraiteEuroRest-get_coef_rente.R'
'RetraiteEuroRest_internal.R' 'RetraiteEuroRest_load.R'
'RevaloEngine_base_prod_fin.R' 'RevaloEngine_calc_marge_fin.R'
'RevaloEngine_calc_result_technique.R'
'RevaloEngine_calc_revalo.R' 'RevaloEngine_class.R'
'RevaloEngine_finance_cible_marge.R'
'RevaloEngine_finance_cible_pmv1.R'
'RevaloEngine_finance_cible_ppb.R'
'RevaloEngine_finance_contrainte_legale.R'
'RevaloEngine_finance_tmg.R' 'RevaloEngine_internal.R'

'RevaloEngine_pb_contr.R' 'SimBEL.R' 'TabEpEuroInd-internal.R'
 'TabProbaBase-update_tab_proba.R'
 'TabProbaEpEuroInd-internal.R' 'TabProbaRetEuroRest-internal.R'
 'TabRetEuroRest_internal.R' 'TauxPB-internal.R' 'TauxPB_load.R'
 'Treso_calc_vm.R' 'Treso_internal.R' 'Treso_load.R'
 'Treso_revalo.R' 'Treso_revenu.R' 'Treso_update.R' 'help.R'
 'taux_period-function.R'

Archs x64

R topics documented:

Action	8
AlmEngine	9
AutresPassifs	9
AutresReserves	10
autres_passif_load	10
autres_reserves_load	11
base_prod_fin	12
Be	12
Be_DataSet	13
buy_action	13
buy_immo	14
buy_oblig	15
calc_ax	15
calc_axy	16
calc_coupon	17
calc_dotation_ppb	17
calc_fin_proj	18
calc_flux_annee	19
calc_frais	19
calc_frais_fin	20
calc_marge_fin	21
calc_nominal	22
calc_pb_attrib	22
calc_pm	23
calc_pmv1	25
calc_pmv1_action	25
calc_pmv1_immo	26
calc_pmv1_oblig	26
calc_PRE	27
calc_prest	27
calc_primes	30
calc_proba_deces	30
calc_proba_dyn	31
calc_proba_flux	32
calc_proba_survie	33
calc_qx	33
calc_rach	34
calc_rach_dyn	34
calc_RC	35
calc_rdt	35

calc_rdt_marche_ref	36
calc_reprise_ppb	36
calc_resultat_fin	37
calc_result_technique	38
calc_result_technique_ap_pb	38
calc_revalo	39
calc_revalo_pm	40
calc_sur_dec	41
calc_tra	42
calc_tx_cible	42
calc_tx_cible_ref_marche	43
calc_tx_min	44
calc_vm_action	44
calc_vm_immo	45
calc_vm_oblig	46
calc_vm_treso	46
calc_vnc	47
calc_z_spread	47
Canton	48
Canton_DataSet	49
chargement_choc	49
chargement_ESG	50
chargement_PortFin	51
chargement_PortFin_reference	51
chgt_period	52
ChocSolvabilite2	53
convert_table	54
create_ptf_bought_action	54
create_ptf_bought_immo	55
create_ptf_bought_oblig	56
DataBase	56
do_calc_nb_sold_action	57
do_calc_nb_sold_immo	57
do_calc_nb_sold_oblig	58
do_choc_action_type1	59
do_choc_action_type2	59
do_choc_currency	60
do_choc_frais	61
do_choc_immo	61
do_choc_longevite	62
do_choc_mortalite	63
do_choc_rachat_down	64
do_choc_rachat_mass	64
do_choc_rachat_up	65
do_choc_spread	66
do_choc_spread_unitaire	67
do_choc_taux	67
do_update_pmv1	68
do_update_PRE_val_courante	69
do_update_PRE_val_debut	69
do_update_RC_val_courante	70
do_update_RC_val_debut	70

do_update_vm_vnc_precedent	71
duration_sensi	72
echeancier	72
EpEuroInd	73
ESG	73
extract_ESG	74
finance_cible_marge	75
finance_cible_pmv1	76
finance_cible_ppb	77
finance_contrainte_legale	78
finance_tmg	79
FraisFin	80
FraisPassif	80
frais_fin_load	81
frais_passif_load	81
get_choc_inflation_frais	82
get_choc_rach	83
get_choc_table	83
get_coef_rente	84
get_comport	84
get_qx_mort	85
get_qx_rach	86
get_rach_dyn	86
HypCanton	87
HypTech	88
hyp_canton_load	88
Immo	89
Initialisation	89
initialisation_load	90
init_create_folder	91
init_debut_pgg_psap	92
init_scenario	92
init_SimBEL	93
init_tables	93
insert_tables	94
load_action	94
load_be	95
load_epeuroind	95
load_ht	96
load_immo	96
load_oblig	97
load_pp	98
load_reteurorest	98
load_treso	99
merge_actifs	99
merge_be	100
merge_flux_fin	100
merge_pb	101
merge_table_be	101
merge_table_produit	102
ModelPointESG	102
Oblig	103

ParamAlmEngine	103
ParamBe	104
ParamChocMket	104
ParamChocSousc	104
ParamComport	105
ParamRachDyn	105
ParamRevaloEngine	106
ParamTableMort	106
ParamTableRach	107
param_alm_engine_load	108
param_revalo_load	108
pb_contr	109
PortFin	110
PortPassif	111
Ppb	112
ppb_8ans	112
ppb_load	113
PRE	114
pre_load	114
print_alloc	115
proj_an	115
proj_annee_autres_passifs	116
proj_annee_av_pb	117
RC	118
rc_load	118
reallocate	119
RetraiteEuroRest	120
RevaloEngine	120
revalo_action	121
revalo_immo	121
revalo_treso	122
revenu_treso	122
run_be	123
run_be_simu	124
sell_action	125
sell_immo	125
sell_oblig	126
sell_pvl_action	127
set_architecture	127
SimBEL	128
TabEpEuroInd	128
TabProbaEpEuroInd	129
TabProbaRetEuroRest	129
TabRetEuroRest	130
TauxPB	130
tauxpb_load	130
taux_period	131
Treso	132
update_cc_oblig	132
update_dur_det_action	133
update_dur_det_immo	133
update_dur_oblig	134

update_mat_res	134
update_PortFin	135
update_PortFin_reference	136
update_reserves	136
update_sd_oblig	137
update_table_output_pb	138
update_table_output_produit	138
update_tab_proba	139
update_treso	139
update_vm_action	140
update_vm_immo	141
update_vm_oblig	141
update_vnc_oblig	142
update_zsp_oblig	142
vieillissement_action_PortFin	143
vieillissement_immo_PortFin	143
vieillissement_oblig_PortFin	144
vieillissement_ppb	145
vieillissement_treso_PortFin	145
vieillissement_ap_pb	146
vieilli_mp	147
viellissement_av_pb	148
write_be_results	149
yield_to_maturity	149

Index**150**

Action

*La classe Action.***Description**

Classe pour les actifs de type Action.

Slots

ptf_action est un data.frame, chaque ligne represente un actif action du portefeuille d'action.

Author(s)

Prim'Act

See Also

Les operations d'achat-vente action [buy_action](#), [sell_action](#) et [sell_pvl_action](#).

AlmEngine

La classe ALMEngine

Description

Classe ayant pour principal vocation de contenir des methodes de reallocation.

Slots

journal_achat_vente outil permettant de memoriser l'ensemble des operations d'achat-vente.

Author(s)

Prim'Act

See Also

La fonction de reallocation du Portefeuille [reallocate](#)

AutresPassifs

La classe AutresPassifs.

Description

Une classe pour la gestion des passifs hors modele.

Slots

mp un objet data.frame au format fige contenant les flux des passifs hors modele.

Author(s)

Prim'Act

See Also

La lecture des flux d'une annee [proj_annee_autres_passifs](#).

AutresReserves

La classe AutreReserves

Description

Une classe de parametres permettant de gerer le stock de provision globale de gestion (PGG) et de provision pour sinistres a payer (PSAP).

Slots

pgg_debut la valeur de la PGG en debut de periode.
 psap_debut la valeur de la PSAP en debut de periode.
 pgg_valeur la valeur courant de la PGG.
 psap_valeur la valeur courant de la PSAP.
 tx_pgg_ep le taux de PGG applique sur l'epargne.
 tx_pgg_autres le taux de PGG applique sur les autres passifs.
 tx_psap_ep le taux de PGG applique sur l'epargne.
 tx_psap_autres le taux de PGG applique sur les autres passifs.

Author(s)

Prim'Act

See Also

Le calcul et la mise a jour des autres reserves [update_reserves](#) et [init_debut_pgg_psap](#).

autres_passif_load

Methode permettant de charger la valeur initiale des autres passifs.

Description

autres_passif_load est une methode permettant de charger les donnees associees a un objet de classe [AutresPassifs](#).

Usage

```
autres_passif_load(file_autres_passif_address)
```

Arguments

file_autres_passif_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner un objet [AutresPassifs](#).

Value

L'objet de la classe [AutresPassifs](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

autres_reserves_load	<i>Methode permettant de charger la valeur initiale de la PSAP et de la PGG.</i>
----------------------	--

Description

autres_reserves_load est une methode permettant de charger les donnees associees a un objet de classe [AutresReserves](#).

Usage

```
autres_reserves_load(file_autres_reserves_address)
```

Arguments

file_autres_reserves_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner un objet [AutresReserves](#).

Value

L'objet de la classe [AutresReserves](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

base_prod_fin	<i>Calcule la base de produits financiers attribuables.</i>
---------------	---

Description

base_prod_fin est une methode permettant de calculer la base de produits financiers attribuables pour la revalorisation des contrats.

Usage

```
base_prod_fin(tra, pm_moy, ppb)
```

Arguments

tra	est une valeur numeric donnant le taux de rendement de l'actif.
pm_moy	est un vecteur numeric comprenant le montant de PM moyenne par produit.
ppb	est un objet de la classe Ppb qui renvoie l'etat courant de la PPB.

Value

La valeur de la base de produit financier par produit et au total pour le portefeuille.

Author(s)

Prim'Act

See Also

[Ppb](#).

Be	<i>La classe Be.</i>
----	----------------------

Description

Une classe pour le calcul du best estimate d'un assureur.

Slots

param_be un objet [ParamBe](#) qui regroupe les parametres de base du calcul d'un best estimate.
 canton un objet de type [Canton](#) correspond au canton parametre en date initiale.
 esg un objet de type [ESG](#).
 base un objet de type [DataBase](#).
 tab_flux une liste qui contient les flux moyens de best estimate et de ses composantes.
 tab_be est une liste qui contient la valeur du best estimate et de ses composantes.

Author(s)

Prim'Act

See Also

Le calcul d'un best estimate : [run_be](#). Le calcul d'une simulation de best estimate : [run_be_simu](#). L'initialisation d'un best estimate dans les situations centrales et choquées : [init_scenario](#). La sortie des résultats au format ".csv" : [write_be_results](#). La classe [Canton](#). La classe [ESG](#). La classe [ParamBe](#).

Be_DataSet	<i>Dataset Be.</i>
------------	--------------------

Description

Dataset de la classe [Be](#) contenant des données correspondant aux classes : [Canton](#), [ParamBe](#), [ESG](#).

Usage

```
data("be")
```

Format

Objet de la classe [Be](#).

Note

Il s'agit de données simulées, elles ont été créées afin de tester les différentes fonctions du package.

Author(s)

Prim'Act

Examples

```
data(be)
param_be <- be@param_be
portefeuille_passif <- be@canton@ptf_passif
```

buy_action	<i>Mise à jour de chaque composante d'un portefeuille action suite à un achat d'un autre portefeuille action.</i>
------------	---

Description

buy_action est une méthode permettant de mettre à jour le portefeuille action suite à l'achat d'un autre portefeuille action. de chaque composante d'un portefeuille action.

Usage

```
buy_action(x, ptf_bought)
```

Arguments

- x objet de la classe [Action](#) (decrivant le portefeuille action en detention).
ptf_bought objet de la classe [Action](#) (decrivant le portefeuille action achete).

Value

L'objet x complete des elements de ptf_bought.

Author(s)

Prim'Act

buy_immo	<i>Mise a jour de chaque composante d'un portefeuille action suite a un achat d'un autre portefeuille immobilier.</i>
----------	---

Description

buy_immo est une methode permettant de mettre a jour le portefeuille action suite a l'achat d'un autre portefeuille immobilier. de chaque composante d'un portefeuille action.

Usage

```
buy_immo(x, ptf_bought)
```

Arguments

- x objet de la classe [Immo](#) (decrivant le portefeuille immo en detention).
ptf_bought objet de la classe [Immo](#) (decrivant le portefeuille immo achete).

Value

L'objet x complete des elements de ptf_bought.

Author(s)

Prim'Act

buy_oblig	<i>Mise a jour de chaque composante d'un portefeuille obligataire suite a un achat d'un autre portefeuille obligataire.</i>
-----------	---

Description

buy_oblig est une methode permettant de mettre a jour le portefeuille obligataire suite a l'achat d'un autre portefeuille obligataire. de chaque composante d'un portefeuille obligataire.

Usage

```
buy_oblig(x, ptf_bought)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
ptf_bought	objet de la classe Oblig (decrivant le portefeuille obligataire achete).

Value

L'objet x complete des elements de ptf_bought.

Author(s)

Prim'Act

calc_ax	<i>Calcule la valeur d'un coefficient actuariel d'une rente sur une tete.</i>
---------	---

Description

calc_ax est une methode permettant de calculer les coefficients actuariels pour des rentes versant potentiellement plusieurs flux dans l'annee.

Usage

```
calc_ax(tx_tech, freq_rente, echu, table_mort, age, gen)
```

Arguments

tx_tech	un numeric correspondant au taux technique utilise pour l'actualisation.
freq_rente	un integer correspondant au nombre de versements dans l'annee.
echu	un logical valant TRUE si la rente est a terme echue et FALSE si la rente est a terme echoir.
table_mort	un objet de la classe ParamTableMort designant la table de mortalite de la tete principale.
age	la valeur integer de l'age de la tete principale.
gen	la valeur integer de la generation de la tete principale.

Value

Le coefficient actuariel calcule.

Author(s)

Prim'Act

See Also

Le calcul des probabilités de survies [calc_proba_survie](#).

calc_axy	<i>Calcule la valeur d'un coefficient actuariel d'une rente avec reversion.</i>
----------	---

Description

calc_axy est une methode permettant de calculer les coefficients actuariels pour des rentes avec reversion versant potentiellement plusieurs flux dans l'annee.

Usage

```
calc_axy(tx_tech, tx_rvs, freq_rente, echu, table_mort_1, age_1, gen_1,
         table_mort_2, age_2, gen_2)
```

Arguments

tx_tech	un numeric correspondant au taux technique utilise pour l'actualisation.
tx_rvs	un numeric correspondant au taux de reversion de la rente.
freq_rente	un integer correspondant au nombre de versements dans l'annee.
echu	un logical valant TRUE si la rente est a terme echue et FALSE si la rente est a terme echoir.
table_mort_1	un objet de la classe ParamTableMort designant la table de mortalite de la tete principale.
age_1	la valeur integer de l'age de la tete principale.
gen_1	la valeur integer de la generation de la tete principale.
table_mort_2	un objet de la classe ParamTableMort designant la table de mortalite de la tete secondaire.
age_2	la valeur integer de l'age de la tete secondaire.
gen_2	la valeur integer de la generation de la tete secondaire.

Value

Le coefficient actuariel calcule.

Author(s)

Prim'Act

See Also

Le calcul des probabilités de survies [calc_proba_survie](#).

calc_coupon	Calcule les coupons d'un portefeuille obligataire.
-------------	--

Description

calc_coupon est une methode permettant de calculer les valeurs de coupon de l'ensemble des obligations composant un portefeuille obligataire.

Usage

```
calc_coupon(x)
```

Arguments

x	un objet de la classe Oblig , dont on souhaite calculer le coupon annuel pour chacune de ses composantes.
---	---

Value

un vecteur dont chaque element correspond a la valeur du coupon de l'obligation consideree , i.e $tx_coupon * parite * nominal * nb_unit$. Le vecteur renvoie a autant d'elements que le portefeuille obligataire en input a de lignes.

Author(s)

Prim'Act

calc_dotation_ppb	Dote la valeur de la PPB
-------------------	--------------------------

Description

calc_dotation_ppb est une methode permettant de doter la PPB. La dotation est effectuee si les limites de dotation de la PPB sur l'annee ne sont pas atteintes. La valeur de cette limite est mise a jour suite a la dotation.

Usage

```
calc_dotation_ppb(x, montant)
```

Arguments

x	objet de la classe Ppb .
montant	une valeur numeric a doter.

Value

ppb l'objet x mis a jour.
dotation le montant de la dotation effectuee.

Author(s)

Prim'Act

calc_fin_proj

calcule le flux et les resultats ajustes en fin de projection.

Description

calc_fin_proj est une methode permettant de calculer au niveau du canton les resultats financier, technique, brut et net d'impot, ainsi que le flux de passifs soldant une projection.

Usage

```
calc_fin_proj(x, resultat_fin, result_tech, pm_fin_ap_pb, tx_pb,  
             tx_enc_moy)
```

Arguments

x	est un objet de la classe Canton .
resultat_fin	est la valeur numeric du resultat financier avant fin de projection.
result_tech	est la valeur numeric du resultat technique avant fin de projection.
pm_fin_ap_pb	est un vecteur numeric par produit correspond au PM de fin avant application de la fin de projection.
tx_pb	est un vecteur numeric par produit correspond au taux de PB contractuel.
tx_enc_moy	est un vecteur numeric par produit correspond au taux chargement sur encours moyens.

Value

flux_fin_passif un vecteur de flux de fin par produit.
result_tech le montant de resultat technique en fin de projection.
result_fin le montant de resultat finanacier en fin de projection.
result_brut le montant de resultat brut d'impot en fin de projection.
result_net le montant de resultat net d'impot en fin de projection.
impot le montant d'impot sur le resultat en fin de projection.

calc_flux_annee	Calcule les flux perçus dans l'année du portefeuille obligataire.
-----------------	---

Description

calc_flux_annee est une méthode permettant de calculer les tombées de coupons et les échéances l'ensemble des obligations composant un portefeuille obligataire.

Usage

```
calc_flux_annee(x)
```

Arguments

x	un objet de la classe Oblig .
---	---

Value

tombée_coupon un vecteur correspondant aux tombées de coupon dans l'année. Ce vecteur a autant d'éléments que le portefeuille obligataire d'inputs a de lignes.

tombée_echeance un vecteur correspondant aux tombées d'échéances dans l'année. Ce vecteur a autant d'éléments que le portefeuille obligataire d'inputs a de lignes.

Author(s)

Prim'Act

calc_frais	Calcule des frais de passif.
------------	------------------------------

Description

calc_frais est une méthode générique permettant de calculer les frais sur prestations, sur primes et sur encours.

Usage

```
calc_frais(x, type, nom_prod, nb, mt, coef_inf)
```

Arguments

x	objet de la classe FraisPassif .
type	un caractère designant le type de frais appliqué.
nom_prod	est le nom de produit de type caractère.
nb	correspond à un nombre de contrats, utilise comme assiette de frais fixe par contrat.
mt	correspond à un montant, utilise comme assiette de frais variable.
coef_inf	correspond au coefficient d'inflation appliqué.

Details

Le type du contrat prend pour valeur prime pour les frais sur primes, prest pour les frais sur prestations et enc pour les frais sur encours.

Value

Une liste contenant les montants de frais fixes et de frais variables.

Author(s)

Prim'Act

calc_frais_fin	<i>Calcul des frais financier.</i>
----------------	------------------------------------

Description

calc_frais_fin est une methode permettant de calculer les frais financiers.

Usage

```
calc_frais_fin(x, vm_moy, coef_inflation)
```

Arguments

x	est un objet de type FraisFin contenant les parametres des frais financiers associes a un canton.
vm_moy	est un objet de type numeric correspondant a la valeur moyenne de l'actif en valeur de marche.
coef_inflation	est un objet de type numeric correspondant au coefficient d'inflation des frais.

Value

La valeur des frais financiers : un reel de type numeric.

Author(s)

Prim'Act

calc_marge_fin	Calcule la marge financiere de l'assureur.
----------------	--

Description

calc_marge_fin est une methode permettant de calculer la marge financiere de l'assureur apres attribution d'un certain niveau de revalorisation.

Usage

```
calc_marge_fin(base_fin, rev_prest_nette, rev_stock_nette,  
               contrib_tmg_prest, contrib_tmg_stock, contrib_ppb_tx_cible)
```

Arguments

base_fin	est un vecteur de type numeric comprenant par produit la base de produits financiers.
rev_prest_nette	est un vecteur de type numeric comprenant par produit la revalorisation nette sur prestations.
rev_stock_nette	est un vecteur de type numeric comprenant par produit la revalorisation nette sur stock.
contrib_tmg_prest	est une valeur numeric comprenant par produit la contribution de la PPB au financement des TMG sur prestations.
contrib_tmg_stock	est une valeur numeric comprenant par produit la contribution de la PPB au financement des TMG sur stock.
contrib_ppb_tx_cible	une valeur de type numeric comprenant par produit la contribution de la PPB au financement du taux cible sur stock.

Value

Le montant de la marge de l'assureur.

Author(s)

Prim'Act

calc_nominal	<i>Calcule le nominal des obligations constituant le portefeuille obligataire.</i>
--------------	--

Description

calc_nominal est une methode permettant de calculer les valeurs de nominal de l'ensemble des obligations composant un portefeuille obligataire.

Usage

```
calc_nominal(x)
```

Arguments

x un objet de la classe [Oblig](#).

Value

Un vecteur dont chaque element correspond a la valeur du nominal de l'obligation consideree : parite * nominal * nb_unit. Le vecteur renvoie a autant d'elements que le portefeuille obligataire en input a de lignes.

Author(s)

Prim'Act

calc_pb_attrib	<i>Calcule le montant de PB attribuee depuis le debut d'annee.</i>
----------------	--

Description

calc_pb_attrib est une methode permettant de calculer le montant de PB attribuee au cours d'une annee.

Usage

```
calc_pb_attrib(x)
```

Arguments

x un objet de la classe [Ppb](#).

Value

la valeur numeric correspondant au montant de la pb attribuee.

Author(s)

Prim'Act

calc_pm	<i>Calcul les PM pour des contrats epargne en euros et les contrats de retraite euro en phases de restitution.</i>
---------	--

Description

calc_pm est une methode permettant de calculer les provisions mathematiques (PM) de fin de periode avant application de la revalorisation au titre de la participation aux benefices.

Usage

```
calc_pm(x, method, an, tx_cible, y)
```

Arguments

x	un objet de la classe EpEuroInd ou de la classe RetraiteEuroRest contenant les model points epargne euros ou retraite euro en phases de restitution.
method	un character prenant pour valeur normal pour le calcul des flux avec application de la revalorisation au titre de la participation aux benefices, et la valeur gar pour le calcul avec uniquement les flux garanti (calcul de la FDB).
an	une valeur integer correspondant a l'annee du calcul des prestations.
tx_cible	une liste conteant les taux cibles annuel et semestriel par model points. Le format de cette liste correspond a la sortie de la methode calc_tx_cible
y	une liste a remplir uniquement si x est de type EpEuroInd contenant les parametres : tab_prime : une liste contenant les flux de primes pour chaque ligne de model points. Le format de cette liste correspond a la sortie flux de la methode calc_primes . A remplir uniquement si l'objet x est de type EpEuroInd . tab_prest : une liste contenant le taux de revalorisation minimum associes a chaque ligne de model points. Le format de cette liste correspond a la sortie flux de la methode calc_prest . A remplir uniquement si x est de type EpEuroInd . tx_min : une liste contenant le taux de revalorisation minimum associes a chaque ligne de model points. Le format de cette liste correspond a la sortie de la methode calc_tx_min . A remplir uniquement si x est de type EpEuroInd . tx_soc : est une valeur numeric correspondant au taux de prelevements sociaux EpEuroInd .

Details

En epargne, cette methode permet de calculer les montants de PM de fin d'annee, avec une revalorisation minimale pour les inputs. Les chargements sur encours sont egalement preleves. Cette methode permet de gerer les contrats a taux de revalorisation net negatif. Cette methode permet egalement de calculer le besoin de financement necessaire pour atteindre les exigences de revalorisation des assures. Pour la retraite, cette methode renvoie les elements de PM ainsi que le besoin de financement afferent.

Value

Une liste contenant :

method : la valeur de l'argument method.

flux : une liste comprenant les flux de l'annee.

stock : une liste comprenant les nombres de sorties.

Le format de la liste flux est :

rev_stock_brut : un vecteur contenant la revalorisation minimale brute de l'annee appliquee au PM (nul en cas de typage [RetraiteEuroRest](#)).

rev_stock_nette : un vecteur contenant la revalorisation minimale nette de l'annee appliquee au PM (nul en cas de typage [RetraiteEuroRest](#)).

enc_charg_stock : un vecteur contenant les chargement sur encours de l'annee, calcules en prenant en compte la revalorisation minimale (nul en cas de typage [RetraiteEuroRest](#)).

enc_charg_base_th : un vecteur contenant les chargements sur encours theoriques de l'annee, evalues sur la base de la PM non revalorisees (nul en cas de typage [RetraiteEuroRest](#)).

enc_charg_rmin_th : un vecteur contenant les chargements sur encours theoriques de l'annee, evalues sur la seule base de la revalorisation minimale des PM (nul en cas de typage [RetraiteEuroRest](#)).

base_enc_th : un vecteur contenant l'assiette de calcul des chargements sur encours de l'annee (nul en cas de typage [RetraiteEuroRest](#)).

soc_stock : un vecteur contenant le prelevements sociaux de l'annee (nul en cas de typage [RetraiteEuroRest](#)).

it_tech_stock : un vecteur contenant les interets techniques sur stock de l'annee (nul en cas de typage [RetraiteEuroRest](#)).

it_tech : un vecteur contenant les interets techniques sur stock et sur prestations de l'annee (nul en cas de typage [RetraiteEuroRest](#)).

bes_tx_cible : un vecteur contenant le besoin de financement de l'annee pour atteindre le taux cible de chaque assure.

Le format de la liste stock est :

pm_deb : un vecteur contenant le montant de PM en debut d'annee

pm_fin : un vecteur contenant le montant de PM en fin d'annee, avec revalorisation au taux minimum

pm_moy : un vecteur contenant le montant de PM moyenne sur l'annee.

Author(s)

Prim'Act

See Also

[calc_primes](#), [calc_prest](#), [calc_tx_cible](#), [calc_tx_min](#).

`calc_pmv1`*Mets a jour les valeurs des plus ou moins-values actions et immobilier.*

Description

`calc_pmv1` est une methode permettant de calculer les valeurs des plus ou moins values latentes sur actions et immobilier. Met a jour la valeur des attributs d'un objet [PortFin](#).

Usage

```
calc_pmv1(x)
```

Arguments

`x` objet de la classe [PortFin](#).

Value

L'objet `x` dont les attributs en PVL et en MVL a ete mise a jour.

Author(s)

Prim'Act

`calc_pmv1_action`*Calcule les plus et moins-values action.*

Description

`calc_pmv1_action` est une methode permettant de calculer les plus et moins-values du portefeuille action.

Usage

```
calc_pmv1_action(x)
```

Arguments

`x` objet de la classe [Action](#) (decrivant le portefeuille d'action).

Value

`pvl` correspondant a la somme des plus-values latentes actions.

`mv1` correspondant a la somme des moins-values latentes actions.

Author(s)

Prim'Act

calc_pmv1_immo	<i>Calcule les plus et moins-values immobilier.</i>
----------------	---

Description

calc_pmv1_immo est une methode permettant de calculer les plus et moins-values du portefeuille immobilier.

Usage

```
calc_pmv1_immo(x)
```

Arguments

x objet de la classe [Immo](#) (decrivant le portefeuille d'immobilier).

Value

pvl correspondant a la somme des plus-values latentes immobilier.

mv1 correspondant a la somme des moins-values latentes immobilier.

Author(s)

Prim'Act

calc_pmv1_oblig	<i>Calcule les plus et moins-values obligataires.</i>
-----------------	---

Description

calc_pmv1_oblig est une methode permettant de calculer les plus et moins-values du portefeuille obligataire.

Usage

```
calc_pmv1_oblig(x)
```

Arguments

x objet de la classe [Oblig](#) (decrivant le portefeuille d'obligations).

Value

pvl correspondant a la somme des plus-values latentes obligataires.

mv1 correspondant a la somme des moins-values latentes obligataires.

Author(s)

Prim'Act

`calc_PRE`*Calcul de la PRE.*

Description

`calc_PRE` est une methode permettant de calculer le montant de PRE.

Usage

```
calc_PRE(x, pmvl_action_immo)
```

Arguments

`x` objet de la classe [PRE](#), necessaire pour connaitre le stock de PRE initial.

`pmvl_action_immo` est un `numeric` correspondant au montant global de plus ou moins values latentes des actifs actions et immobiliers. En cas de moins value latente, la PRE est abondee. En cas de plus value latente, la PRE est integralement reprise.

Value

Le format de la liste renvoyee est :

`pre_courante` : valeur de la pre courante calculee a partir des inputs transmis

`var_pre` : variation de la pre courante

Author(s)

Prim'Act

`calc_prest`*Calcul les flux de prestations pour des contrats epargne en euros ou retraite euros en phases de restitution.*

Description

`calc_prest` est une methode permettant de calculer les flux de prestations, les chargements sur encours relatifs a ces prestations et les nombres de sorties sur une periode.

Usage

```
calc_prest(x, method, an, y)
```

Arguments

x	un objet de la classe EpEuroInd ou de la classe RetraiteEuroRest contenant les model points epargne euros, ou les model points retraite euro en phase de restitution.
method	un character prenant pour valeur normal pour le calcul des flux avec application de la revalorisation au titre de la participation aux benefices, et la valeur gar pour le calcul avec uniquement les flux garanti (calcul de la FDB).
an	une valeur integer correspondant a l'annee du calcul des prestations.
y	une liste a remplir uniquement si x est de type EpEuroInd contenant les parametres : proba_dyn : une liste contenant le taux de rachats dynamiques (totaux et partiels) par model points. tx_min : une liste contenant le taux de revalorisation minimum associes a chaque ligne de model points. Le format de cette liste correspond a la sortie de la methode calc_tx_min . tx_soc : est une valeur numeric correspondant au taux de prelevements sociaux. choc_lapse_mass : est une valeur numeric correspondant au choc de rachat massif.

Details

En epargne, cette methode permet de calculer les flux de sortie en echeance, les flux de rachat totaux et partiels et les flux de deces d'un contrat epargne en euros. Ces prestations font l'objet d'une revalorisation au taux minimum contractuel. Les nombres de sortie sont egalement produits. Il est possible de realiser un choc de rachat massif si an. Dans ce cas, les prestations de rachats massifs sortent en debut d'annee et ne sont pas revalorisees. Des chargements sont appliques sur flux de rachats. Des prelevements sur encours sont appliques sur les prestations revalorises au taux minimum contractuel. Cette methode permet de gerer les contrats a taux de revalorisation net negatif. Pour un contrat de retraite, elle permet de sortir les flux de rente et les nombres de sorties.

Value

Une liste contenant :

method : la valeur de l'argument method
 flux : une liste comprenant les flux de l'annee
 stock : une liste comprenant les nombres de sorties

Le format de la liste flux est :

ech : un vecteur contenant les flux de sortie en echeance de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rach_tot : un vecteur contenant les flux de rachat totaux de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

dc : un vecteur contenant les flux de deces de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rach_part : un vecteur contenant les flux de rachat partiel de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rente : le flux annuel de rente par model point : nul si l'objet est de type [EpEuroInd](#).

prest : un vecteur contenant les flux prestations de l'annee (renseigne que l'objet x soit de type [RetraiteEuroRest](#) ou [EpEuroInd](#)).

rev_ech : un vecteur contenant la revalorisation des echeances de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rev_rach_tot : un vecteur contenant la revalorisation des rachats totaux de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rev_dc : un vecteur contenant la revalorisation des deces de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rev_rach_part : un vecteur contenant la revalorisation des rachats partiels de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rev_prest : un vecteur contenant la revalorisation brute des prestations de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rev_prest_net : un vecteur contenant la revalorisation des prestations nette de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

enc_charg : un vecteur contenant les chargements sur l'encours de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

rach_charg : un vecteur contenant les chargements sur les rachats de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

soc_prest : un vecteur contenant les prelevements sociaux sur prestations de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

it_tech_prest : un vecteur contenant les interets techniques sur prestations de l'annee. : nul si l'objet est de type [RetraiteEuroRest](#).

arr_charg : un vecteur contenant les chargements sur arerages. : nul si l'objet est de type [EpEuroInd](#).

Le format de la liste stock est :

nb_ech : un vecteur contenant le nombre de sorties en echeance de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

nb_rach_tot : un vecteur contenant le nombre de rachats totaux de l'annee : nul si l'objet est de type [RetraiteEuroRest](#).

nb_dc : un vecteur contenant le nombre de deces de l'annee

nb_sortie : un vecteur contenant le nombre de sorties de l'annee

nb_contr_fin : un vecteur contenant le nombre de contrats en cours en fin d'annee

nb_contr_moy : un vecteur contenant la moyenne du nombre de contrats sur l'annee.

Author(s)

Prim'Act

See Also

[calc_proba_flux](#), [calc_tx_min](#).

calc_primes	<i>Calcule les flux de primes pour des contrats epargne en euros et des retraites en phase de restitution.</i>
-------------	--

Description

calc_primes est une methode permettant de calculer les flux de primes sur une periode.

Usage

```
calc_primes(x)
```

Arguments

x	un objet de la classe EpEuroInd ou de la classe RetraiteEuroRest contenant les model points epargne euros ou de retraite en phase de restitution.
---	---

Details

Cette fonction permet de projeter uniquement des primes periodiques de contrats epargne en euros. Pour la retraite en phase de restitution, il n'y a pas de prime et la methode renvoie des valeurs nulles.

Value

stock : une liste contenant le nombre de versements nb_verse associe a chaque model point.
 flux : une liste contenant pour chaque model point les montants de primes brutes pri_brut, les montants de primes nettes pri_net et les chargements sur primes pri_chgt.

Author(s)

Prim'Act

calc_proba_deces	<i>Calcule le vecteur des probabilites cumulees de deces</i>
------------------	--

Description

calc_proba_deces est une methode permettant de calculer le vecteur des probabilites cumulees de deces.

Usage

```
calc_proba_deces(table_mort, age, gen, n_periodes)
```

Arguments

table_mort	un objet de la classe ParamTableMort contenant la table de mortalite.
age	une valeur integer correspondant a l'age.
gen	une valeur integer correspondant a la generation.
n_periodes	une valeur integer correspondant a la borne de sommation.

Value

La valeur du taux de decés calcule.

Author(s)

Prim'Act

calc_proba_dyn	<i>Calcul les probabilités dynamiques de mouvement de flux pour des contrats épargne en euros et de retraite.</i>
----------------	---

Description

calc_proba_dyn est une méthode permettant de calculer les différents taux de sortie dynamique sur une période.

Usage

```
calc_proba_dyn(x, ht)
```

Arguments

x	un objet de la classe EpEuroInd contenant les model points épargne euros.
ht	un objet de la classe HypTech contenant différentes tables de mortalité et différentes lois de rachat.

Value

Une matrice contenant pour chaque model points en ligne :

qx_rach_tot_dyn : un vecteur contenant les taux de rachats totaux dynamiques

qx_rach_part_dyn : un vecteur contenant les taux de rachats partiels dynamiques.

Author(s)

Prim'Act

See Also

La récupération des taux de rachat dynamique : [get_rach_dyn](#).

calc_proba_flux	<i>Calcul les probabilites de mouvement de flux pour des contrats epargne en euros et de retraite.</i>
-----------------	--

Description

calc_proba_flux est une methode permettant de calculer les differents taux de sortie sur une periode.

Usage

```
calc_proba_flux(x, ht)
```

Arguments

x	un objet de la classe EpEuroInd ou de la classe RetraiteEuroRest contenant les model points epargne euros.
ht	un objet de la classe HypTech contenant differentes tables de mortalite et differentes lois de rachat.

Value

Pour un objet [EpEuroInd](#), renvoie une liste contenant pour chaque model points en ligne :

qx_rach_tot : un vecteur contenant les taux de rachats totaux

qx_dc : un vecteur contenant les taux de deces

qx_rach_part : un vecteur contenant les taux de rachats partiels

Pour un objet [RetraiteEuroRest](#), renvoie une liste contenant pour chaque model points en ligne :

proba_sortie_retraite : un vecteur contenant les probabilites de deces

proba_survie_un_an : un vecteur contenant les probabilites de survie

Author(s)

Prim'Act

See Also

La recuperation des taux de rachat structurel : [get_qx_rach](#). La recuperation des taux de rachat dynamique : [get_rach_dyn](#). La recuperation des taux de deces : [get_qx_mort](#).

calc_proba_survie	<i>Calcule le vecteur des probabilites cumulees de survies.</i>
-------------------	---

Description

calc_proba_survie est une methode permettant de calculer le vecteur des probabilites cumulees de survie.

Usage

```
calc_proba_survie(table_mort, age, gen, n_periodes)
```

Arguments

table_mort	un objet de la classe ParamTableMort contenant la table de mortalite.
age	une valeur integer correspondant a l'age.
gen	une valeur integer correspondant a la generation.
n_periodes	une valeur integer correspondant a la borne de sommation.

Value

Le vecteur des taux de deces calcules.

Author(s)

Prim'Act

calc_qx	<i>Calcule le taux de deces.</i>
---------	----------------------------------

Description

calc_qx est une methode permettant de calculer le taux de deces.

Usage

```
calc_qx(table_mort, age, gen)
```

Arguments

table_mort	un objet de la classe ParamTableMort contenant la table de mortalite.
age	une valeur integer correspondant a l'age.
gen	une valeur integer correspondant a la generation.

Value

La valeur du taux de deces calcule.

Author(s)

Prim'Act

calc_rach	<i>Calcule le taux de rachat.</i>
-----------	-----------------------------------

Description

calc_rach est une methode permettant de calculer le taux de rachat.

Usage

```
calc_rach(table_rach, age, anc)
```

Arguments

table_rach	un objet de la classe ParamTableRach contenant la table de rachat.
age	une valeur integer correspondant a l'age.
anc	une valeur integer correspondant a l'anciennete.

Value

La valeur du taux de rachat calcule.

Author(s)

Prim'Act

calc_rach_dyn	<i>Calcule la composante rachats dynamique.</i>
---------------	---

Description

calc_rach_dyn est une methode permettant de calculer la composante rachat dynamique selon la methodologie transmise dans le ONC de l'ACPR de 2013.

Arguments

p	un objet de la classe ParamRachDyn contenant les parametres de rachats dynamiques.
tx_cible	une valeur numeric correspondant au taux de revalorisation cible.
tx_serv	une valeur numeric correspondant au taux de revalorisation servi.

Value

La valeur du taux rachat.

Author(s)

Prim'Act

calc_RC

*Calcule de la RC.***Description**

calc_RC est une methode permettant de calculer le montant de Reserve de Capitalisation.

Usage

```
calc_RC(x, pmvr_oblig)
```

Arguments

x	objet de la classe RC , necessaire pour connaitre le stock de RC initial.
pmvr_oblig	est un numeric correspondant au montant global annuel de plus ou moins values realisees sur des actifs obligataires.

Value

Le format de la liste renvoyee est :

RC_courante : valeur de la RC courante initiale augmentee des plus ou moins values annuelles realisees

var_RC : variation de la RC courante.

Author(s)

Prim'Act

calc_rdt

*Calcule les rendements action, immobilier et de la tresorerie.***Description**

calc_rdt est une methode permettant de calculer les rendements des portefeuilles [Action](#) et [Immo](#) d'un objet [PortFin](#). Le rendement de la [Treso](#) est egalement fourni.

Usage

```
calc_rdt(x, mp_ESG)
```

Arguments

x	objet de la classe PortFin .
mp_ESG	objet de la classe ModelPointESG decrivant les conditions de l'annee n (ainsi que l'annee (n-1) pour les indices Actions et Immobilier).

Value

rdt_action un data.frame compose de deux colonnes et autant de lignes que le portefeuille Action.

rdt_immo un data.frame compose de deux colonnes et autant de lignes que le portefeuille Immobilier.

rdt_treso une valeur numeric correspondant au taux de rendement de la tresorerie..

Author(s)

Prim'Act

calc_rdt_marche_ref	<i>Calcul du taux de rendement de reference au niveau du marche</i>
---------------------	---

Description

calc_rdt_marche_ref est une methode permettant de calculer un taux cible.

Usage

```
calc_rdt_marche_ref(x, mp_esg)
```

Arguments

x un objet de la classe [ParamComport](#).

mp_esg est un objet de type [ModelPointESG](#), qui represente la situation courante en annee et simulations des valeurs de l'ESG.

Value

Une liste contenant les rendements de reference du marche.

Author(s)

Prim'Act

calc_reprise_ppb	<i>Reprend sur la valeur de la PPB</i>
------------------	--

Description

calc_reprise_ppb est une methode permettant de reprendre sur la PPB. La reprise est effectuee si les limites de reprise de la PPB sur l'annee ne sont pas atteintes. La valeur de cette limite est mise a jour suite a la reprise

Usage

```
calc_reprise_ppb(x, montant)
```

Arguments

x	un objet de la classe Ppb.
montant	la valeur numeric de la reprise.

Value

ppb l'objet x mis a jour
reprise le montant de la reprise effectuee.

Author(s)

Prim'Act

calc_resultat_fin	<i>Calcule le resultat financier.</i>
-------------------	---------------------------------------

Description

calc_resultat_fin est une methode permettant de calculer le resultat financier du portefeuille.

Usage

```
calc_resultat_fin(revenu, produit, frais_fin, var_rc)
```

Arguments

revenu	est un objet de type numeric, qui fournit les revenus du portefeuille financier.
produit	est un objet de type numeric, qui fournit le produit (ou la perte) des cessions.
frais_fin	est un objet de type numeric, qui fournit le montant des frais financiers.
var_rc	est un objet de typenumeric, donnant la variation de la reserve de capitalisation.

Value

La valeur du result financier.

Author(s)

Prim'Act

calc_result_technique *calcule le resultat technique*

Description

calc_result_technique est une methode permettant de calculer le resultat technique avant attribution de participation aux benefices.

Usage

```
calc_result_technique(passif_av_pb, var_pre)
```

Arguments

passif_av_pb	est une liste produite par la methode viellissement_av_pb appliquee a un portefeuille de passif.
var_pre	est une valeur numeric correspondant a la variation de PRE.

Value

Le resultat technique.

Author(s)

Prim'Act

See Also

[PRE](#), [viellissement_av_pb](#).

calc_result_technique_ap_pb

calcule le resultat technique apres prise en compte de la participation aux benefices.

Description

calc_result_technique_ap_pb est une methode permettant de calculer le resultat technique apres attribution de participation aux benefices.

Usage

```
calc_result_technique_ap_pb(passif_av_pb, passif_ap_pb, ppb, var_pre)
```

Arguments

passif_av_pb	est une liste produit par la methode viellissement_av_pb .
passif_ap_pb	est une liste produit par la methode viellissement_ap_pb .
ppb	est un objet de la classe Ppb qui renvoie l'etat courant de la PPB.
var_pre	est une valeur numeric correspondant a la variation de PRE.

Value

Le resultat technique apres participation aux benefices.

calc_revalo	<i>Applique la politique de revalorisation d'un canton.</i>
-------------	---

Description

calc_revalo est une methode permettant de d'appliquer l'ensemble de la politique de revalorisation d'un assureur.

Usage

```
calc_revalo(x, passif_av_pb, tra, plac_moy_vnc, result_tech, annee)
```

Arguments

x	un objet de la classe Canton .
passif_av_pb	est une liste produit par la methode viellissement_av_pb appliquee a un portefeuille de passif.
tra	est la valeur numeric du taux de rendement de l'actif.
plac_moy_vnc	est la valeur numeric moyenne des actifs en valeur nette comptable.
result_tech	est la valeur numeric du resultat technique prise en compte avant distribution de la PB.
annee	est la valeur integer correspondant a l'annee de projection.

Value

add_rev_netto_stock un vecteur numeric avec la valeur de la revalorisation nette servie par produit au titre de la participation aux benefices.

pmv1_liq le montant de plus-values latentes en actions a realiser.

ppb un objet [Ppb](#) correspondant a la PPB mise a jour.

ppb8_ind un vecteur numeric avec la valeur de la PPB devant sortir au titre de la regle des 8 ans, alloue par produit.

conso_ppb_init un vecteur numeric avec la valeur de la PPB initiale consommee servant au calcul du BEG.

tx_pb un vecteur reprenant les taux de PB par produit renseigne dans l'objet x.

tx_enc_moy un vecteur reprenant les taux de chargement sur encours theoriques moyens par produit.

Author(s)

Prim'Act

See Also

Le calcul du TRA : [calc_tra](#). Le vieillissement des passifs avant PB : [viellissement_av_pb](#). Le calcul du resultat technique avant PB : [calc_result_technique](#). Le calcul de la base de produits financiers : [base_prod_fin](#). Le calcul de la PB contractuelle : [pb_contr](#). Le financement des TMG par la PPB : [finance_tmg](#). Le financement du taux cible par la PPB : [finance_cible_ppb](#). Le financement du taux cible par la realisation plus-values latentes actions : [finance_cible_pmv1](#). Le financement du taux cible par la compression de la marge de l'assureur : [finance_cible_marge](#). Le calcul de la marge de l'assureur : [calc_marge_fin](#). L'application de la contrainte legale de participation aux benefices : [finance_contrainte_legale](#).

calc_revalo_pm	<i>Calcule et applique la revalorisation pour des PM pour des contrats epargne en euros et des retraites en phase de restitution.</i>
----------------	---

Description

calc_revalo_pm est une methode permettant de calculer la revalorisation des PM sur une annee.

Usage

```
calc_revalo_pm(x, y)
```

Arguments

x	un objet de la classe EpEuroInd ou de la classe RetraiteEuroRest contenant les model points epargne euros.
y	une liste contenant les parametres. rev_net_alloue : est une valeur numeric correspondant au montant de revalorisation a allouer. rev_brute_alloue_gar : est une valeur numeric correspondant au montant de revalorisation a allouer a la PM garantie. tx_soc : est une valeur numeric correspondant au taux de prelevement sociaux. A remplir uniquement si x est de type EpEuroInd .

Details

En epargne, cette methode permet de calculer les montants de PM de fin d'annee avec une revalorisation minimale et une revalorisation additionnelle au titre de la participation aux benefices de l'annee. Les chargements sur encours sont egalement calcules et preleves. Cette methode permet de gerer les contrats a taux de revalorisation net negatif. En retraite, elle permet de revaloriser le montant des rentes.

Value

Une liste contenant :

flux : une liste comprenant les flux de l'annee

stock : une liste comprenant les nombres de sorties

tx_rev_net : un vecteur correspondant au taux de revalorisation net appliques a chaque model point.

Le format de la liste flux est :

rev_stock_brut_ap_pb : un vecteur contenant la revalorisation brute de l'année appliquée au PM

rev_stock_netto_ap_pb : un vecteur contenant la revalorisation nette de l'année appliquée au PM. Elle peut être négative pour des contrats à taux négatif.

enc_charg_stock_ap_pb : un vecteur contenant les montants de chargement sur encours de l'année calculés pour le stock de PM

soc_stock_ap_pb : un vecteur contenant les prélèvements sociaux de l'année

Le format de la liste stock est : s

pm_fin_ap_pb : un vecteur contenant le montant de PM en fin d'année

Author(s)

Prim'Act

See Also

Le calcul des PM avec revalorisation minimale uniquement [calc_pm](#).

calc_sur_dec	<i>Calcul les surcotes/decotes de chaque composante d'un portefeuille obligataire.</i>
--------------	--

Description

calc_sur_dec est une méthode permettant de calculer les surcotes/decotes de chaque composante d'un portefeuille obligataire.

Usage

```
calc_sur_dec(x)
```

Arguments

x objet de la classe [Oblig](#) (décivant le portefeuille obligataire).

Value

Un vecteur contenant les surcotes decotes.

Author(s)

Prim'Act

calc_tra

*Calcul du taux de rendement financier***Description**

calc_tra est une methode permettant de calculer le taux de rendement financier du portefeuille.

Usage

```
calc_tra(plac_moy, res_fin)
```

Arguments

plac_moy est un objet de type numeric, qui fournit la valeur moyenne des placements de l'annee en valeur nette comptable.

res_fin est un objet de type numeric, qui fournit le resultat financier du portefeuille.

Value

La valeur du taux de rendement de l'actif.

Author(s)

Prim'Act

calc_tx_cible

*Calcul du taux cible pour des contrats epargne en euros et de retraite en phase de restitution.***Description**

calc_tx_cible est une methode permettant d'evaluer le taux de revalorisation cible de chaque model point.

Usage

```
calc_tx_cible(x, y)
```

Arguments

x un objet de la classe [EpEuroInd](#) ou de la classe [RetraiteEuroRest](#) contenant les model points epargne euros ou retraite euro en phases de restitution.

y une liste contenant les parametres.

list_rd : est un vecteur contenant les rendements de reference.

ht : est un objet de la classe [HypTech](#).

Le format de list_rd est :

le taux de rendement obligataire

le taux de rendement de l'indice action de reference

le taux de rendement de l'indice immobilier de reference

le taux de rendement de l'indice tresorerie de reference

Value

tx_cible_an : un vecteur contenant les taux cible de l'annee

tx_cible_se : un vecteur contenant les taux cible de l'annee sur base semestrielle

Note

Pour les besoins des calculs a mi-annee, des taux semestriels sont produits.

Author(s)

Prim'Act

See Also

La recuperation des taux cibles calcules : [get_comport](#).

calc_tx_cible_ref_marche

Calcule le taux de revalorisation cible.

Description

calc_tx_cible_ref_marche est une methode permettant de calculer le taux de revalorisation cible en evaluant le taux de rendement des assureurs sur le marche.

Arguments

param_comport un objet de la classe [ParamComport](#) contenant les parametres comportementaux.

list_rd une liste contenant les rendements de reference. Le format de cette liste est :

le taux de rendement obligataire

le taux de rendement de l'indice action de reference

le taux de rendement de l'indice immobilier de reference

le taux de rendement de l'indice tresorerie de reference

tx_cible_prec une valeur numeric correspondant au taux cible de la periode precedente.

Value

La valeur du taux cible.

Author(s)

Prim'Act

calc_tx_min	<i>Calcul le taux de revalorisation contractuel minimum pour des contrats epargne en euros.</i>
-------------	---

Description

calc_tx_min est une methode permettant de calculer les taux de revalorisation minimum sur une periode. La revalorisation minimum est le maximum entre le taux technique et le taux minimum garanti (TMG) du contrat.

Usage

```
calc_tx_min(x, an)
```

Arguments

x	un objet de la classe EpEuroInd contenant les model points epargne euros.
an	un numeric representant l'annee de projection courante.

Value

tx_tech_an : un vecteur contenant les taux de technique de l'annee
tx_tech_se : un vecteur contenant les taux de technique de l'annee sur base semestrielle
tx_an : un vecteur contenant les taux de revalorisation minimum de l'annee
x_se : un vecteur contenant les taux de revalorisation minimum de l'annee exprimes en semestriel.

Note

Pour les besoins des calculs a mi-annee, des taux semestriels sont produits.

Author(s)

Prim'Act

calc_vm_action	<i>Calcule les valeurs de marches du portefeuille action.</i>
----------------	---

Description

calc_vm_action est une methode permettant de calculer les valeurs de marche du portefeuille action.

Usage

```
calc_vm_action(x, rdt)
```

Arguments

x	objet de la classe Action (decrivant le portefeuille d'action).
rdt	vecteur de type <code>numeric</code> decrivant le rendement de chacune des lignes du portefeuille action de l'assureur. Contient autant d'elements que le portefeuille action a de lignes.

Value

Les valeurs de marche mises a jour.

Author(s)

Prim'Act

calc_vm_immo

Calcule les valeurs de marches du portefeuille immobilier.

Description

calc_vm_immo est une methode permettant de calculer les valeurs de marche du portefeuille immobilier.

Usage

```
calc_vm_immo(x, rdt)
```

Arguments

x	objet de la classe Immo (decrivant le portefeuille d'immobilier).
rdt	vecteur de type <code>numeric</code> decrivant le rendement de chacune des lignes du portefeuille immobilier de l'assureur. Contient autant d'elements que le portefeuille immobilier a de lignes.

Value

Les valeurs de marche mises a jour.

Author(s)

Prim'Act

calc_vm_oblig	<i>Calcul les valeurs de marches de chaque composante du portefeuille obligataires.</i>
---------------	---

Description

calc_vm_oblig est une methode permettant de calculer les valeurs de marche du portefeuille obligataires.

Usage

```
calc_vm_oblig(x, yield_curve)
```

Arguments

x	un objet de la classe Oblig (decrivant le portefeuille d'obligation).
yield_curve	un vecteur de type numeric contenant la courbe de taux (l'attribut yield_curve des objets de la classe ModelPointESG).

Value

Les valeurs de marche mises a jour.

Author(s)

Prim'Act

calc_vm_treso	<i>Calcule les valeurs de marches de chaque composante du portefeuille treso.</i>
---------------	---

Description

calc_vm_treso est une methode permettant de calculer les valeurs de marche de chaque ligne du portefeuille treso.

Usage

```
calc_vm_treso(x, rdt, flux_milieu, flux_fin)
```

Arguments

x	objet de la classe Treso (decrivant le portefeuille de treso).
rdt	vecteur decrivant le rendement de chacune des lignes treso du portefeuille. Contient autant d'elements que le portefeuille a de lignes.
flux_milieu	vecteur decrivant les flux de milieu d'annee (positif =entrants, negatifs = sortants).
flux_fin	vecteur decrivant les flux de fin d'annee (positif =entrants, negatifs = sortants).

Value

Les valeurs de marche mises a jour.

Author(s)

Prim'Act

calc_vnc	<i>Calcul les valeurs nettes comptables de chaque composante portefeuille obligataires.</i>
----------	---

Description

calc_vnc est une methode permettant de calculer les valeurs de marche.

Usage

```
calc_vnc(x, sd_unitaire)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille d'obligation).
sd_unitaire	vecteur de type <code>numeric</code> decrivant la surcote decote de chacune des lignes d'obligation du portefeuille obligation de l'assureur. Contient autant d'elements que le portefeuille a de lignes.

Value

L'objet x dont les valeurs nettes comptables ont ete mises a jour.

Author(s)

Prim'Act

calc_z_spread	<i>Calcul les zeros spreads de chaque composante d'un portefeuille obligataire.</i>
---------------	---

Description

calc_z_spread est une methode permettant de calculer les zeros spread de chaque composante d'un portefeuille obligataire.

Usage

```
calc_z_spread(x, yield_curve)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire).
yield_curve	vecteur decrivant la courbe de taux sans risque retenue.

Value

Un vecteur dont chaque element correspond a la valeur du zero spread de l'obligation du portefeuille obligataire. Ce vecteur a autant d'elements que le portefeuille obligataire a de lignes.

Author(s)

Prim'Act

Canton	<i>La classe Canton.</i>
--------	--------------------------

Description

Une classe pour le canton d'un assureur. Un objet de cette classe agrege un portefeuille financier, un portefeuille de passifs, l'ensemble des autres provisions ainsi que les parametres et donnees necessaires a la projection de la situation d'un l'assureur.

Slots

annee une valeur entiere correspondant a l'annee de projection.

ptf_fin est un objet de type [PortFin](#), qui represente le portefeuille d'investissement d'un canton.

ptf_passif est un objet de type [PortPassif](#), qui represente le portefeuille de passif d'un canton.

mp_esg est un objet de type [ModelPointESG](#), qui represente la situation courante deduite de l'ESG. Cet objet traduit la situation economique pour une annee donnee et une simulation donnee.

ppb est un objet de type [Ppb](#), qui represente la provision pour participation aux benefices (PPB).

hyp_canton est un objet de type [HypCanton](#), qui regroupe les hypotheses generales applicables au canton.

param_alm est un objet de type [ParamAlmEngine](#), qui contient les parametres utilises dans les methodes de gestion de l'allocation d'actifs.

param_revalo est un objet de type [ParamRevaloEngine](#), qui contient les parametres utilises dans les methodes de gestion de la revalorisation.

Author(s)

Prim'Act

See Also

La projection du Canton sur une annee : [proj_an](#). Le calcul du resultat technique : [calc_result_technique_ap_pb](#).
Le calcul des fins de projection : [calc_fin_proj](#).

Canton_DataSet	<i>Dataset Canton.</i>
----------------	------------------------

Description

Dataset de la classe [Canton](#) contenant un portefeuille de passif, d'actif ainsi que plusieurs hypothèses.

Usage

```
data(canton)
```

Format

Objet de la classe [Canton](#).

Note

Il s'agit de données simulées, elles ont été créées afin de tester les différentes fonctions du package.

Author(s)

Prim'Act

Examples

```
data("canton")
portefeuille_passif <- canton@ptf_passif
```

chargement_choc	<i>Permet de charger les paramètres de choc de la formule standard.</i>
-----------------	---

Description

chargement_choc est une méthode permettant de charger les paramètres l'ensemble des paramètres nécessaires à la bonne application des chocs de marche et de souscription au sens de la formule standard de la directive Solvabilité 2, tels que renseignés par l'utilisateur.

Usage

```
chargement_choc(x, folder_chocs_address)
```

Arguments

x objet de la classe [ChocSolvabilite2](#).

folder_chocs_address est un caractère. Cette chaîne de caractère est construite par la méthode [set_architecture](#) de la classe [Initialisation](#). Elle contient l'adresse du dossier contenant les fichiers de paramètres des chocs de la formule standard à appliquer. Ces derniers doivent être renseignés par l'utilisateur.

Value

x l'objet de la classe `ChocSolvabilite2` dont les attributs `scenario`, `param_choc_mket` et `param_choc_sousc` ont été mis à jour.

Author(s)

Prim'Act

See Also

La création de l'architecture de chargement des données et paramètres renseignés par l'utilisateur `set_architecture`, ainsi que les classes `ParamChocMket` et `ParamChocSousc`.

chargement_ESG	<i>Cette méthode charge les tables de simulations d'un ESG.</i>
----------------	---

Description

chargement_ESG est une méthode permettant de charger les trajectoires simulées par le générateur de scénarios économiques (ESG) de Prim'Act et d'alimenter un objet `ESG`.

Usage

```
chargement_ESG(folder_ESG_address, nb_simu, nb_annee_proj)
```

Arguments

folder_ESG_address	est un caractère. Il correspond au chemin de référence du dossier contenant les extractions de l'ESG Prim'Act.
nb_simu	est une valeur de type integer correspondant au nombre de trajectoire simulées par l'ESG Prim'Act.
nb_annee_proj	est une valeur de type integer correspondant au nombre d'années de projection des sorties de l'ESG Prim'Act.

Details

Les différentes adresses potentielles pour les différents ESG employés (central, hausse de taux, baisse de taux) sont construites par la fonction `set_architecture` de la classe `Initialisation`.

Value

x l'objet de la classe `ESG` construit.

Author(s)

Prim'Act

chargement_PortFin	<i>Charge le PortFin a partir des donnees renseignees par l'utilisateur.</i>
--------------------	--

Description

chargement_PortFin est une methode permettant de creer un objet [PortFin](#) a partir des donnees renseignees par l'utilisateur.

Usage

chargement_PortFin(folder_PortFin_address, mp_ESG)

Arguments

- folder_PortFin_address
est un chemin de type character, cf la methode [set_architecture](#)
- mp_ESG
est un objet de la classe [ModelPointESG](#), qui fournit le resultat financier du porfeuille.

Value

L'objet [PortFin](#) tel que precise par les donnees initiales et les parametres renseignes par l'utilisateur.

Author(s)

Prim'Act

chargement_PortFin_reference	<i>Charge le PortFin de reinvestissement a partir des donnees renseignees par l'utilisateur.</i>
------------------------------	--

Description

chargement_PortFin_reference est une methode permettant de creer un objet PortFin correspondant au portefeuille financer de reinvestissement a partir des donnees renseignees par l'utilisateur.

Usage

chargement_PortFin_reference(folder_PortFin_reference_address, mp_ESG)

Arguments

- folder_PortFin_reference_address
est un chemin de type character, cf la methode [set_architecture](#)
- mp_ESG
est un objet de la classe [ModelPointESG](#), qui fournit le resultat financier du porfeuille.

Value

L'objet `PortFin` correspondant au portefeuille financier de reinvestissement tel que precise par les donnees initiales et les parametres renseignes par l'utilisateur.

Author(s)

Prim'Act

chgt_period	<i>Fonction permettant de calculer le taux de chargement sur une periode.</i>
-------------	---

Description

chgt_period permet de calculer le taux de chargement sur une periode.

Usage

```
chgt_period(x, period)
```

Arguments

x	un vecteur de la classe <code>numeric</code> representant le taux annuel.
period	un character representant la peridocite souhaitee. Cette variable prend pour valeur : an : annuel ; se : semestriel ; trim : trimestriel ; mens : mensuel.

Value

Le taux periodique.

Author(s)

Prim'Act

ChocSolvabilite2*La classe ChocSolvabilite2.*

Description

La classe ChocSolvabilite2 permet de realiser les principaux des scenarios de choc initiaux au sens de la formule standard de la directive Solvabilite 2.

Details

Cette classe contient deux attributs qui contiennent respectivement l'ensemble des parametres necessaires a l'application des chocs Marche et Souscription. Cette classe contient aussi l'ensemble des methodes permettant d'appliquer chacun de ces chocs individuellement a un objet de la classe [Canton](#). Les chocs permis sont :

central : la situation centrale
taux_up : le choc de taux a la hausse
taux_down : le choc de taux a la baisse
action_type1 : le choc action de type 1
action_type2 : le choc action de type 2
immo : le choc immobilier
spread : le choc spread sur les obligations corporates
currency_up : les chocs sur devise a la hausse
currency_down : les chocs sur devise a la baisse
mortalite : le choc mortalite sur les tables de mortalite
longevite : le choc longevite sur les tables de mortalite
frais : le choc depenses sur le niveau des frais et l'inflation des frais
rachat_up : le choc de rachat a la hausse
rachat_down : le choc de rachat a la baisse
rachat_mass : le choc de rachat massif

Slots

scenario un vecteur de type character comprenant les noms des chocs a initialiser
param_choc_mket un objet de la classe [ParamChocMket](#).
param_choc_sousc un objet de la classe [ParamChocSousc](#).

Author(s)

Prim'Act

See Also

L'application des chocs de taux_up et taux_down : [do_choc_taux](#). L'application des chocs de action_type1 et action_type2 : [do_choc_action_type1](#), [do_choc_action_type2](#). L'application du choc de immo : [do_choc_immo](#). L'application du choc de spread : [do_choc_spread](#). L'application des chocs de currency_up et currency_down : [do_choc_currency](#). L'application du choc de mortalite : [do_choc_mortalite](#). L'application du choc de longevite : [do_choc_longevite](#). L'application du choc de frais : [do_choc_frais](#), [get_choc_inflation_frais](#). L'application des chocs de rachat_up, rachat_down et rachat_massif: [do_choc_rachat_up](#), [do_choc_rachat_down](#) et [do_choc_rachat_mass](#).

convert_table	<i>Transforme une table contenant des "l_x" en une table contenant des "q_x"</i>
---------------	--

Description

convert_table est une methode permettant de calculer le vecteur des probabilites cumulees de survie.

Usage

```
convert_table(table_mort, type)
```

Arguments

table_mort	un objet de la classe data.frame contenant la table de mortalite.
type	un objet de la classe character correspondant au type de la table de mortalite passee en parametre (qx ou lx).

Value

La table transformee.

Author(s)

Prim'Act

create_ptf_bought_action	<i>Ajuste les quantites d'actions a acheter.</i>
--------------------------	--

Description

create_ptf_bought_action est une methode permettant d'ajuster d'un coefficient les quantites d'actions a acheter. Cette methode est utilisee pour l'achat de nouvelles actions.

Usage

```
create_ptf_bought_action(x, coefficient)
```

Arguments

x	objet de la classe Action , correspondant au portefeuille actions de reinvestissement. Ce portefeuille est unitaire.
coefficient	un vecteur de type numeric qui a autant d'elements que le portefeuille de reinvestissement action a de lignes. Il correspond au coefficient a appliquer au portefeuille de reinvestissement action pour effectuer l'achat desire.

Value

x un objet de la classe [Action](#) correspondant a une proportion precise du portefeuille de reinvestissement action.

Author(s)

Prim'Act

See Also

La classe [Action](#).

create_ptf_bought_immo

Ajuste les quantites d'immobilier a acheter.

Description

create_ptf_bought_immo est une methode permettant d'ajuster d'un coefficient les quantites d'immobilier a acheter. Cette methode est utilisee pour l'achat de nouveaux titres immobilier.

Usage

```
create_ptf_bought_immo(x, coefficient)
```

Arguments

x	objet de la classe Immo , correspondant au portefeuille immobilier de reinvestissement. Ce portefeuille est unitaire.
coefficient	est un vecteur de type numeric qui a autant d'elements que le portefeuille de reinvestissement immo a de lignes. Il correspond au coefficient a appliquer au portefeuille de reinvestissement immo pour effectuer l'achat desire.

Value

x un objet de la classe [Immo](#) correspondant a une proportion precise du portefeuille de reinvestissement immo.

Author(s)

Prim'Act

See Also

La classe [Immo](#).

```
create_ptf_bought_oblig
```

Ajuste les quantites d'obligations a acheter.

Description

Cette methode permet d'ajuster d'un coefficient les quantites d'obligations a acheter. Cette methode est utilisee pour l'achat de nouveaux titres obligataires.

Usage

```
create_ptf_bought_oblig(x, coefficient)
```

Arguments

x	objet de la classe Oblig , correspondant au portefeuille obligataire de reinvestissement. Ce portefeuille est unitaire.
coefficient	est un vecteur de type numeric qui a autant d'elements que le portefeuille de reinvestissement obligataire a de lignes. Il correspond au coefficient a appliquer au portefeuille de reinvestissement obligataire pour effectuer l'achat desire.

Value

x un objet de la classe [Oblig](#) correspondant a une proportion precise du portefeuille de reinvestissement obligataire.

Author(s)

Prim'Act

See Also

La classe [Oblig](#).

```
DataBase
```

La classe DataBase.

Description

Une classe permettant de stocker les donnees d'output de la fonction [proj_an](#).

Slots

database une base de donnees SQLite

ecriture_base un logical qui indique si les output doivent etre ecrits dans la base SQLite.

Author(s)

Prim'Act

`do_calc_nb_sold_action`*Calcule le nombre d'actions a vendre.*

Description

Cette methode permet de calculer pour chaque ligne d'un portefeuille action d'un assureur le nombre d'unites a vendre afin de realiser un certain montant de vente en actions.

Usage

```
do_calc_nb_sold_action(x, montant_vente, method_vente)
```

Arguments

<code>x</code>	objet de la classe Action , correspondant au portefeuille action de l'assureur.
<code>montant_vente</code>	est un reel de type numeric correspondant a un montant de vente (en valeur de marche) totale d'action que l'assureur souhaite effectuer.
<code>method_vente</code>	est un element de type character correspondant a methode de vente retenue (seule la methode proportionnelle est implementee actuellement).

Value

data.frame contenant deux colonnes (`num_mp`, `nb_sold`) correspondant respectivement au numero de model point de chaque ligne action du portefeuille et du nombre d'unite a vendre pour chacune d'entre elles.

Author(s)

Prim'Act

See Also

[Action](#).

`do_calc_nb_sold_immo`*Calcule le nombre de titres immobilier a vendre.*

Description

Cette methode permet de calculer pour chaque ligne d'un portefeuille immobilier d'un assureur le nombre d'unites a vendre afin de realiser un certain montant de vente immo.

Usage

```
do_calc_nb_sold_immo(x, montant_vente, method_vente)
```

Arguments

x	objet de la classe Immo , correspondant au portefeuille immo de l'assureur.
montant_vente	est un reel de type numeric correspondant a la vente totale de vm immo que l'assureur souhaite effectuer.
method_vente	est un element de type character correspondant a methode de vente retenue (seule la methode proportionnelle est implementee actuellement).

Value

data.frame contenant deux colonnes (num_mp,nb_sold) correspondant respectivement au numero de model point de chaque ligne immo du portefeuille et du nombre d'unite a vendre pour chacune d'entre elles.

Author(s)

Prim'Act

See Also

La classe [Immo](#).

do_calc_nb_sold_oblig *Calcule le nombre d'obligations a vendre.*

Description

Cette methode permet de calculer pour chaque ligne d'un portefeuille obligataire d'un assureur le nombre d'unites a vendre afin de realiser un certain montant de vente obligataire.

Usage

```
do_calc_nb_sold_oblig(x, montant_vente, method_vente)
```

Arguments

x	objet de la classe Oblig , correspondant au portefeuille obligataire de l'assureur.
montant_vente	est un reel de type numeric correspondant a la vente totale de vm obligataire que l'assureur souhaite effectuer.
method_vente	est un element de type character correspondant a methode de vente retenue (seule la methode proportionnelle est implementee actuellement).

Value

data.frame contenant deux colonnes (num_mp,nb_sold) correspondant respectivement au numero de model point de chaque ligne obligataire du portefeuille et du nombre d'unite a vendre pour chacune d'entre elles.

Author(s)

Prim'Act

See Also

La classe [Oblig.](#)

do_choc_action_type1 *Permet a partir d'un canton initial de creer un canton choque action.*

Description

do_choc_action_type1 est une methode permettant d'appliquer le choc action type 1 de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_action_type1(x, canton)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.

Value

canton l'objet de la classe [Canton](#) correspondant au scenario choque action au sens de la formule standard Solvabilite 2.

Note

Il est possible d'appliquer des chocs actions distincts a chaque action selon l'index. Cette parametrisation est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_action_type2 *Permet a partir d'un canton initial de creer un canton choque action.*

Description

do_choc_action_type2 est une methode permettant d'appliquer le choc action type 2 de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_action_type2(x, canton)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	un objet de la classe Canton . Il correspond au canton non choqué (i.e. central) de l'assureur.

Value

canton l'objet de la classe [Canton](#) correspondant au scénario choqué action au sens de la formule standard Solvabilité 2.

Note

Il est possible d'appliquer des chocs actions distincts à chaque action selon l'index. Cette paramétrisation est effectuée dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_currency	<i>Permet à partir d'un canton initial de créer un canton choqué de devise.</i>
------------------	---

Description

do_choc_currency est une méthode permettant d'appliquer le choc currency de la formule standard Solvabilité 2 à un canton. Cette méthode s'applique uniquement aux obligations, actions et titres en immobilier.

Usage

```
do_choc_currency(x, nom_currency, direction, canton)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
nom_currency	objet caractère correspondant au nom de la devise du choc.
direction	objet caractère prenant pour valeur "up" pour le choc à la hausse et "down" pour le choc à la baisse.
canton	est un objet de la classe Canton . Il correspond au canton non choqué (i.e. central) de l'assureur.

Value

canton l'objet de la classe [Canton](#) correspondant au scénario choqué currency au sens de la formule standard Solvabilité 2. Cette paramétrisation est effectuée dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_frais	<i>Permet a partir d'un canton initial de creer un canton choque frais.</i>
---------------	---

Description

do_choc_frais est une methode permettant d'appliquer le choc frais de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_frais(x, canton, autres_passifs_choc)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.
autres_passifs_choc	est un objet de la classe AutresPassifs , il correspond au chargement des autres passifs choques. Ces derniers ont ete renseignes par l'utilisateur en donnees.

Value

canton l'objet de la classe [Canton](#) correspondant au scenario choque frais au sens de la formule standard Solvabilite 2.

Note

La parametrisation des chocs de frais est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_immo	<i>Permet a partir d'un canton initial de creer un canton choque immobilier.</i>
--------------	--

Description

do_choc_immo est une methode permettant d'appliquer le choc immobilier de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_immo(x, canton)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choqué (i.e. central) de l'assureur.

Value

canton l'objet de la classe [Canton](#) correspondant au scénario choqué immobilier au sens de la formule standard Solvabilité 2.

Note

Il est possible d'appliquer des chocs immobiliers distincts à chaque ligne immobilier présent en portefeuille selon l'index. Cette paramétrisation est effectuée dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_longevite	<i>Permet à partir d'un canton initial de créer un canton choqué longévité.</i>
-------------------	---

Description

do_choc_longevite est une méthode permettant d'appliquer le choc longévité de la formule standard Solvabilité 2 à un canton.

Usage

```
do_choc_longevite(x, canton, autres_passifs_choc)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choqué (i.e. central) de l'assureur.
autres_passifs_choc	est un objet de la classe AutresPassifs , il correspond au chargement des autres passifs choqués en longévité. Ces derniers ont été renseignés par l'utilisateur en données.

Value

canton l'objet de la classe [Canton](#) correspondant au scénario choqué longévité au sens de la formule standard Solvabilité 2.

Note

La paramétrisation des chocs de longévité est effectuée dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_mortalite	<i>Permet a partir d'un canton initial de creer un canton choque mortalite.</i>
-------------------	---

Description

do_choc_mortalite est une methode permettant d'appliquer le choc mortalite de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_mortalite(x, canton, autres_passifs_choc)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.
autres_passifs_choc	est un objet de la classe AutresPassifs , il correspond au chargement des autres passifs choques en mortalite. Ces derniers ont ete renseignes par l'utilisateur en donnees.

Value

canton l'objet de la classe canton correspondant au scenario choque mortalite au sens de la formule standard Solvabilite 2.

Note

La parametrisation des chocs de mortalite est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_rachat_down	<i>Permet a partir d'un canton initial de creer un canton dont les taux de rachat sont choques a la baisse.</i>
---------------------	---

Description

do_choc_rachat_down est une methode permettant d'appliquer le choc a la baisse des taux de rachat de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_rachat_down(x, canton, autres_passifs_choc)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.
autres_passifs_choc	est un objet de la classe AutresPassifs , il correspond au chargement des autres passifs choques en rachat a la baisse. Ces derniers ont ete renseignes par l'utilisateur en donnees.

Value

canton l'objet de la classe [Canton](#) correspondant au scenario de choc a la baisse des taux de rachats au sens de la formule standard Solvabilite 2.

Note

La parametrisation des chocs a la baisse des taux de rachat est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_rachat_mass	<i>Permet a partir d'un canton initial de creer un canton choque rachat massif.</i>
---------------------	---

Description

do_choc_rachat_mass est une methode permettant d'appliquer le choc rachat massif de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_rachat_mass(x, canton, autres_passifs_choc)
```


Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.
autres_passifs_choc	est un objet de la classe AutresPassifs , il correspond au chargement des autres passifs choques. Ces derniers ont ete renseignes par l'utilisateur en donnees.

Value

canton l'objet de la classe [Canton](#) correspondant a la mise a jour du choc rachat massif. au sens de la formule standard Solvabilite 2.

Note

La parametrisation des chocs de rachat massif est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_rachat_up	<i>Permet a partir d'un canton initial de creer un canton dont les taux de rachat sont choques a la hausse.</i>
-------------------	---

Description

do_choc_rachat_up est une methode permettant d'appliquer le choc a la hausse des taux de rachat de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_rachat_up(x, canton, autres_passifs_choc)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.
autres_passifs_choc	est un objet de la classe AutresPassifs , il correspond au chargement des autres passifs choques en rachat a la hausse. Ces derniers ont ete renseignes par l'utilisateur en donnees.

Value

canton l'objet de la classe [Canton](#) correspondant au scenario de choc a la hausse des taux de rachats au sens de la formule standard Solvabilite 2.

Note

La parametrisation des chocs a la hausse des taux de rachat est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

do_choc_spread	<i>Permet a partir d'un canton initial de creer un canton choque spread.</i>
----------------	--

Description

do_choc_spread est une methode permettant d'appliquer le choc spread de la formule standard Solvabilite 2 a un canton. Cette methode s'applique uniquement aux obligations de type corp.

Usage

```
do_choc_spread(x, canton)
```

Arguments

x	objet de la classe ChocSolvabilite2 .
canton	est un objet de la classe Canton . Il correspond au canton non choque (i.e. central) de l'assureur.

Value

canton l'objet de la classe [Canton](#) correspondant au scenario choque spread au sens de la formule standard Solvabilite 2.

Note

Il est possible d'appliquer des chocs de spreads distincts a chaque ligne du portefeuille obligataire selon le numero de rating et la duration de l'obligation. Cette parametrisation est effectuee dans les fichiers d'inputs utilisateurs.

Author(s)

Prim'Act

See Also

L'application du choc de spread a une ligne obligataire : [do_choc_spread_unitaire](#).

`do_choc_spread_unitaire`

Applique le choc spread de la formule standard Solvabilite 2 a une ligne obligataire.

Description

`do_choc_spread_unitaire` Permet a partir d'une table contenant les elements du choc de spread obligataire Solvabilite 2 et d'une ligne obligataire d'un element [Oblig](#) d'un portefeuille financier [PortFin](#) d'appliquer le choc de spread a cette ligne obligataire.

Usage

```
do_choc_spread_unitaire(table_choc_spread, ligne_oblig)
```

Arguments

`table_choc_spread`

un `data.frame` contenant la table de parametres avec les chocs de spreads.

`ligne_oblig`

un `data.frame`. Il correspond a une ligne obligataire d'un portefeuille [Oblig](#) d'un assureur.

Value

`vm_choquee` une valeur numerique correspondant a la valeur de marche de la ligne obligataire suite a l'application du choc de spread a cette ligne.

Author(s)

Prim'Act

See Also

La classe [PortFin](#).

`do_choc_taux`

Methode permettant d'appliquer le choc de taux a un Canton.

Description

`do_choc_taux` est une methode permettant d'appliquer le choc de taux de la formule standard Solvabilite 2 a un canton.

Usage

```
do_choc_taux(canton)
```

Arguments

`canton`

un objet de la classe [Canton](#), correspondant au canton auquel on souhaite appliquer le choc de taux.

Value

canton l'objet de la classe [Canton](#), mis a jour du choc de taux.

Author(s)

Prim'Act

do_update_pmv1	<i>Met a jour l'ensemble des attributs pvl et pml d'un portefeuille financier.</i>
----------------	--

Description

do_update_pmv1 est une methode permettant de mettre a jour les moins de plus ou moins-values latentes d'un objet [PortFin](#).

Usage

```
do_update_pmv1(x)
```

Arguments

x est un objet de la classe [PortFin](#),

Value

L'objet x de la classe [PortFin](#) dont les plus-values et moins-values latentes ont ete recalculées avec les elements du [PortFin](#) renseigne en input.

Author(s)

Prim'Act

See Also

Les methodes de calcul des plus ou moins-values latentes : [calc_pmv1_action](#), [calc_pmv1_immo](#), [calc_pmv1_oblig](#).

`do_update_PRE_val_courante`*Mise a jour de la valeur courante de PRE.*

Description

`do_update_PRE_val_courante` est une methode permettant de calculer le montant de PRE.

Usage

```
do_update_PRE_val_courante(x, val_courante)
```

Arguments

<code>x</code>	objet de la classe PRE correspondant a la PRE avant mise a jour.
<code>val_courante</code>	est un <code>numeric</code> correspondant au montant de PRE calcule par la fonction calc_PRE .

Value

L'objet [PRE](#) mis a jour de la nouvelle valeur courante de [PRE](#)

Author(s)

Prim'Act

See Also

La methode de calcul de la PRE [calc_PRE](#)

`do_update_PRE_val_debut`*Mise a jour de la valeur de debut de periode de la PRE*

Description

`do_update_PRE_val_debut` est une methode permettant de mettre a jour le montant de debut de periode de PRE.

Usage

```
do_update_PRE_val_debut(x, val_debut)
```

Arguments

<code>x</code>	objet de la classe PRE correspondant a la PRE avant mise a jour.
<code>val_debut</code>	est un <code>numeric</code> correspondant au montant de debut de periode de PRE.

Value

L'objet [PRE](#) mis a jour de la nouvelle valeur de debut de [PRE](#)

Author(s)

Prim'Act

See Also

La methode de calcul de la PRE [calc_PRE](#).

do_update_RC_val_courante

Mise a jour de la valeur courante de RC

Description

do_update_RC_val_courante est une methode permettant de calculer le montant de RC.

Usage

do_update_RC_val_courante(x, val_courante)

Arguments

x objet de la classe [RC](#) correspondant a la RC avant mise a jour.

val_courante est un numeric correspondant au montant de RC calcule par la fonction [calc_RC](#).

Value

L'objet [RC](#) mis a jour de la nouvelle valeur courante de [RC](#).

Author(s)

Prim'Act

See Also

La methode de calcul de la RC [calc_RC](#)

do_update_RC_val_debut

Mise a jour de la valeur initiale de RC

Description

do_update_RC_val_debut est une methode permettant de mettre a jour le montant de debut de periode de RC.

Usage

do_update_RC_val_debut(x, val_debut)

Arguments

- x objet de la classe [RC](#) correspondant a la RC avant mise a jour.
- val_debut est un numeric correspondant au montant de debut de periode de RC.

Value

L'objet x mis a jour de la nouvelle valeur de debut de [RC](#).

Author(s)

Prim'Act

See Also

La methode de calcul de la RC [calc_RC](#)

do_update_vm_vnc_precedent

Evalue et met a jour l'attribut vm_vnc_precedent d'un portefeuille financier.

Description

do_update_vm_vnc_precedent est une methode permettant de calculer et mettre a jour l'attribut vm_vnc_precedent d'un portefeuille financier.

Usage

do_update_vm_vnc_precedent(x)

Arguments

- x objet de la classe [PortFin](#), correspondant au portefeuille financier de l'assureur avant mise a jour de l'attribut vm_vnc_precedent.

Value

L'objet x de la classe [PortFin](#) renvoye correspond au portefeuille financier de l'assureur dont l'attribut vm_vnc_precedent a ete mis a jour.

Author(s)

Prim'Act

See Also

[print_alloc](#).

duration_sensi	<i>Calcule la duration de chaque composante d'un portefeuille obligataire.</i>
----------------	--

Description

duration_sensi est une methode permettant de calculer la duration de chaque composante d'un portefeuille obligataire.

Usage

```
duration_sensi(x)
```

Arguments

x objet de la classe `Oblig` (decrivant le portefeuille obligataire).

Value

Un data frame compose de deux colonnes : la premiere est composee de la duration de chacune des obligations du portefeuille obligataire. La seconde est composee de la sensibilite de chacune des obligations du portefeuille obligataire. Le dataframe de sortie a autant d'elements que le portefeuille obligataire a de lignes.

Author(s)

Prim'Act

echeancier	<i>Calcule les flux obligataires.</i>
------------	---------------------------------------

Description

echeancier est une methode permettant de calculer les flux jusqu'a maturite residuelle d'un model point ou d'un ensemble de model points obligataires.

Arguments

coupon	un vecteur contenant les taux de coupons de chaque obligation.
maturite	un vecteur d'entiers contenant les maturites residuelles de chaque obligation.
zspread	un vecteur contenant les zero-spreads de chaque obligation.
nominal	un vecteur contenant les valeurs nominales de chaque obligation.
yield	un vecteur contenant la courbe de taux consideree (peut-etre vide).

Value

Une matrice contenant :

`grid_flux` : la matrice d'écoulement des flux. Cette matrice a autant de colonnes que le max du vecteur de maturité résiduelle, et autant de lignes que les vecteurs d'input `coupon`, `maturite`, `zspread`, `nominal`. Chaque ligne décrit les flux annuels à venir pour l'actif obligataire dont les caractéristiques sont renseignées en input.

Author(s)

Prim'Act

EpEuroInd	<i>La classe EpEuroInd.</i>
-----------	-----------------------------

Description

Une classe pour les passifs épargne en euros.

Slots

`mp` un objet `data.frame` au format `fige` contenant l'ensemble de model points épargne en euros.
`tab` un objet de la classe [TabEpEuroInd](#) dédié au stockage de variables intermédiaires.
`tab_proba` un objet de la classe [TabProbaEpEuroInd](#) dédié au stockage des probabilités.

Author(s)

Prim'Act

See Also

Le calcul des primes, des prestations et des PM : [calc_primes](#), [calc_prest](#), [calc_pm](#). Le calcul du taux minimum et des taux cible de revalorisation : [calc_tx_min](#), [calc_tx_cible](#). La revalorisation des PM après participation aux bénéfices : [calc_revalo_pm](#). Le vieillissement des model points sur une période : [vieilli_mp](#).

ESG	<i>La classe ESG.</i>
-----	-----------------------

Description

Une classe de paramètres contenant les tables de simulation, générées par un générateur de scénarios économique.

Slots

`nb_simu` un entier (integer) correspondant au nombre de simulations.

`ind_action` une liste contenant les differents indices actions utilises. Chaque element de la liste contient `nb_simu` simulations de l'indice.

`ind_immo` une liste contenant les differents indices immobilier utilises. Chaque element de la liste contient `nb_simu` simulations de l'indice.

`ind_inflation` une liste contenant l'indice inflation utilise. L'element de la liste contient `nb_simu` simulations de l'indice.

`yield_curve` une liste contenant les courbes de taux simulees a chaque date de projection. Chaque element de la liste, correspondant a une annee de projection, contient `nb_simu` simulations de la courbe des taux.

`deflateur` une liste contenant le deflateur stochastique a utiliser. L'element de la liste contient `nb_simu` simulations du deflateur.

Author(s)

Prim'Act

See Also

Les methodes de chargement d'un ESG [chargement_ESG](#) et d'extraction d'un model point ESG [extract_ESG](#).

<code>extract_ESG</code>	<i>permet de construire et charger les trajectoires simulees par le Generateur de Scenarios Economiques de Prim'Act.</i>
--------------------------	--

Description

`extract_ESG` construit l'objet de classe [ModelPointESG](#) a partir d'un objet de la classe [ESG](#). Le `ModelPointESG` ainsi construit correspond a l'extraction de donnees de l'ESG pour une annee specifique et pour une simulation specifique.

Usage

```
extract_ESG(x, num_trajetoire, annee)
```

Arguments

<code>x</code>	un objet de la classe ESG .
<code>num_trajetoire</code>	une valeur de type integer correspondant a la trajectoire de simulation dont on souhaite obtenir les valeurs.
<code>annee</code>	une valeur de type integer correspondant a l'annee d'interet pour le model point (possibilite de selectionner les annees 0 a <code>nb_annee_proj</code>).

Value

`x` l'objet de la classe [ModelPointESG](#) construit.

Author(s)

Prim'Act

See Also

La classe [ModelPointESG](#).

finance_cible_marge	<i>Evalue le financement d'une revalorisation au taux cible par la marge de l'assureur</i>
---------------------	--

Description

finance_cible_marge est une methode permettant de determiner le financement d'une revalorisation au taux cible en comprimant la marge financiere de l'assureur

Usage

```
finance_cible_marge(marge_fin, bes_cible, rev_stocknette, marge_min)
```

Arguments

marge_fin	une valeur numeric donnant le montant courant de la marge financiere de l'assureur.
bes_cible	un vecteur numeric correspondant au besoin de financement necessaire pour atteindre le taux cible part produit.
rev_stocknette	un vecteur numeric comprenant par produit le montant de revalorisation nette au titre de le PB atteint.
marge_min	est une valeur numeric correspondant au montant minimum de marge financiere souhaite par l'assureur.

Value

rev_stocknette la valeur de la revalorisation nette servie par produit apres reduction de marge.
marge_fin le montant de marge de l'assureur apres reduction.

Author(s)

Prim'Act

finance_cible_pmv1	<i>Evalue le financement d'une revalorisation au taux cible par des cessions de plus-values latentes.</i>
--------------------	---

Description

finance_cible_pmv1 est une methode permettant de determiner le financement d'une revalorisation au taux cible par une cession de plus-values latentes en actions.

Usage

```
finance_cible_pmv1(bes_cible, rev_stock_nette, base_fin, seuil_pmv1, tx_pb)
```

Arguments

bes_cible	un vecteur numeric correspondant au besoin de financement necessaire pour atteindre le taux cible part produit.
rev_stock_nette	un vecteur numeric comprenant par produit le montant de revalorisation nette au titre de le PB atteint.
base_fin	un vecteur numeric comprenant par produit la base de produits financiers.
seuil_pmv1	une valeur numeric correspondant au montant de plus-values latentes qui peut etre liquidee. Ce montant doit etre exprime en tenant compte de l'abattement (mise a l'echelle) realise pour rapport aux plus-values latentes de l'actif general au passif.
tx_pb	un vecteur numeric comprenant par produit les taux de participation aux benefices contractuels.

Details

Lorsque la revalorisation nette est superieure au besoin de financement des taux cibles, on sert le taux cible et on partage le surplus. A l'inverse, les taux cible sont finances par les compensations entre produits lorsque certains prevoient une revalorisation superieure au taux cible, et par une liquidation de plus-values latentes.

Value

rev_stock_nette la valeur de la revalorisation nette servie par produit apres cession.

pmv1_liq le montant de plus-values a liquider, ramene a la valeur du passif, pour financer la revalorisation.

Author(s)

Prim'Act

finance_cible_ppb	<i>Evalue le financement d'une revalorisation au taux cible par une reprise de PPB. Cette methode permet aussi de redistribuer la PPB stockee 8 annees auparavant.</i>
-------------------	--

Description

finance_cible_ppb est une methode permettant de determiner le financement d'une revalorisation au taux cible par la reprise de provision pour participation aux benefices (PPB). Cette methode evalue egalement si une dotation est effectuee.

Usage

```
finance_cible_ppb(bes_cible, rev_stock_nette, ppb, ppb8_ind)
```

Arguments

bes_cible	un vecteur numeric correspondant au besoin de financement necessaire pour atteindre le taux cible par produit.
rev_stock_nette	un vecteur numeric comprenant par produit le montant de revalorisation nette au titre de le PB atteint.
ppb	un objet de la classe Ppb qui renvoie l'etat courant de la PPB.
ppb8_ind	une valeur numeric correspondant ? la ppb de l'annee t-8.

Details

Lorsque la revalorisation nette est superieure au besoin de financement des taux cibles, on sert le taux cible et on dote le reste a la PPB dans la limite du plafond de dotation annuel. A l'inverse, les taux cible sont finances par les compensations entre produits lorsque certains prevoient une revalorisation superieure au taux cible, puis par une reprise sur PPB.

Value

rev_stock_nette la valeur de la revalorisation nette servie apres une eventuelle reprise de PPB.
dotation le montant de dotation a la PPB.
reprise le montant de reprise sur la PPB.
ppb l'objet ppb mis a jour.

Author(s)

Prim'Act

finance_contrainte_legale

Applique la contrainte legale de participation aux benefices.

Description

finance_contrainte_legale est une methode permettant de calculer la contrainte legale de participation aux benefices et de l'appliquer si necessaire pour accroitre la revalorisation.

Usage

```
finance_contrainte_legale(base_fin, base_fin_etendu, result_tech, it_stock,  
    rev_stock_nette, rev_prest_nette, dot_ppb, marge_fin, ppb, param_revalo)
```

Arguments

base_fin	un vecteur numeric comprenant par produit la base de produits financiers.
base_fin_etendu	une valeur numeric comprenant la base totale de produits financiers (somme des produits modelise et des passifs non modelises).
result_tech	une valeur numeric comprenant le resultat technique.
it_stock	un vecteur numeric comprenant par produit les interets techniques affectes au stock.
rev_stock_nette	un vecteur de type numeric comprenant par produit la revalorisation nette affectee au stock.
rev_prest_nette	un vecteur de type numeric comprenant par produit la revalorisation nette affectee aux prestations.
dot_ppb	une valeur numeric comprenant la dotation de PPB financant la revalorisation sur stock.
marge_fin	une valeur numeric comprenant la marge financiere courante de l'assureur.
ppb	un objet de la classe Ppb qui renvoie l'etat courant de la PPB.
param_revalo	un objet de la classe ParamRevaloEngine . comprenant les parametres de revalorisation.

Details

Cette methode permet de calculer la contrainte de revalorisation imposee par la reglementation. Si cette contrainte est verifiee alors rien n'est fait, hormis la mise a jour eventuelle du solde negatif de PB. Sinon, la revalorisation additionnelle est dotee a la PPB, jusqu'au maximum de dotation possible, puis le reliquat est alloue entre les produits. La revalorisation additionnelle vient diminuer la marge financiere de l'assureur.

Value

rev_stock_nette la valeur de la revalorisation nette servie apres application de la contrainte legale.

marge_fin le montant de marge de l'assureur apres reduction.

ppb l'objet ppb mis a jour.

param_revalo l'objet param_revalo mis a jour (solde de PB reglementaire negatif).

Author(s)

Prim'Act

finance_tmg*Calcule la contribution de la PPB au financement des taux minimums garantis.*

Description

finance_tmg est une methode permettant d'évaluer le contribution de la PPB au financement des taux minimums garantis (TMG) sur prestations et sur stock.

Usage

```
finance_tmg(bes_tmg_prest, bes_tmg_stock, ppb)
```

Arguments

bes_tmg_prest	un vecteur numeric comprenant par produit le besoin de finance des TMG sur prestations.
bes_tmg_stock	un vecteur numeric comprenant par produit le besoin de finance des TMG sur le stock.
ppb	est un objet de la classe Ppb qui renvoie l'état courant de la PPB.

Details

Dans cette methode, il est considere que le PPB peut venir financer les TMG sur prestations et sur stock. Par convention, la PPB finance d'abord les TMG sur prestations, puis sur stock.

Value

contrib_tmg_prest la valeur de la contribution au financement des TMG sur prestations.
contrib_tmg_stock la valeur de la contribution au financement des TMG sur stock.
ppb l'objet ppb mis a jour.

Author(s)

Prim'Act

FraisFin*La classe FraisFin*

Description

Classe pour les parametres de frais financiers d'un assureur.

Slots

`tx_chargement` est une valeur numerique correspondant au taux de frais de gestion financiere.

`indicatrice_inflation` est un objet de type `logical`, qui permet d'indiquer si une inflation doit etre appliquee.

Lien a creer

NA

Author(s)

Prim'Act

See Also

Mettre le lien vers les methodes de la classe

FraisPassif*La classe FraisPassif*

Description

Une classe de parametres pour les frais des produits du portefeuille de passif.

Slots

`mp` un objet `data.frame` contenant les parametres de frais au passif par produit.

Author(s)

Prim'Act

See Also

Le calcul des frais de passif [calc_frais](#).

frais_fin_load	<i>Methode permettant de charger la valeur initiale des frais financiers dans un objet de type FraisFin.</i>
----------------	--

Description

frais_fin_load est une methode permettant de charger les frais financiers.

Usage

```
frais_fin_load(file_frais_fin_address)
```

Arguments

file_frais_fin_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner les Frais financier.

Value

L'objet de la classe FraisFin construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

frais_passif_load	<i>Methode permettant de charger la valeur des frais de passif.</i>
-------------------	---

Description

frais_passif_load est une methode permettant de charger les donnees associees a un objet de classe [FraisPassif](#).

Usage

```
frais_passif_load(file_frais_passif_address)
```

Arguments

file_frais_passif_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner un objet [FraisPassif](#).

Value

L'objet de la classe [FraisPassif](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

get_choc_inflation_frais

Applique le choc frais de la formule standard a la table de simulation pour l'indice inflation.

Description

get_choc_inflation_frais est une methode permettant d'appliquer le choc frais de la formule standard a la table de simulation pour l'indice inflation.

Usage

get_choc_inflation_frais(x, choc)

Arguments

x	un objet de la classe ESG .
choc	une valeur numeric correspondant au coefficient de choc a appliquer en additif au taux d'inflation.

Value

L'objet x mis a jour.

Note

L'inflation comprise dans l'ESG est suppose etre deja capitalise, i.e. $indice_{inflation} = (1 + tx_{inflation})^{annee}$. Il ne s'agit pas du taux d'inflation.

Author(s)

Prim'Act

get_choc_rach	<i>Applique les chocs de rachat de la formule standard.</i>
---------------	---

Description

get_choc_rach est une methode permettant d'appliquer a l'ensemble des lois de rachat structurelle d'un objet [HypTech](#) les chocs a la hausse ou a la baisse de la formule standard.

Usage

```
get_choc_rach(x, type_choc_rach, choc, choc_lim)
```

Arguments

x	un objet de la classe HypTech contenant differentes lois de rachat partielles et totales.
type_choc_rach	est un character renseignant le type de choc a applique : up pour le choc a la hausse, et down pour le choc a la baisse.
choc	une valeur numeric indiquant le taux de choc.
choc_lim	une valeur numeric indiquant la limite haute pour le choc a la hausse, ou une limite basse pour le choc a baisse.

Value

L'objet x apres choc.

get_choc_table	<i>Applique les chocs de mortalite et de longevite de la formule standard.</i>
----------------	--

Description

get_choc_table est une methode permettant d'appliquer a l'ensemble des table de mortalite d'un objet [HypTech](#) les chocs de mortalite ou de longevite de la formule standard.

Usage

```
get_choc_table(x, choc)
```

Arguments

x	un objet de la classe HypTech contenant differentes tables de mortalite.
choc	une valeur numeric indiquant le taux de choc.

Value

L'objet x apres choc.

Author(s)

Prim'Act

get_coef_rente	<i>Recupere le coefficient d'ax ou d'axy</i>
----------------	--

Description

get_coef_rente est une methode permettant d'obtenir le coefficient de survie esperee actualise au taux technique de rente.

Usage

```
get_coef_rente(x, ht)
```

Arguments

x	un objet de la classe RetraiteEuroRest .
ht	un objet de la classe HypTech .

Value

Le coefficient numeric d'ax ou axy selon le statut du model point dont les caracteristiques sont precisees en input.

Author(s)

Prim'Act

get_comport	<i>Recuperer les taux de revalorisation cible calcules.</i>
-------------	---

Description

get_comport est une methode permettant d'executer le calcul des taux de revalorisation cible.

Usage

```
get_comport(x, nom_table, list_rd, tx_cible_prec)
```

Arguments

x	un objet de la classe HypTech .
nom_table	un nom de la table de parametres de taux cible.
list_rd	une liste contenant les rendements de reference. Le format de cette liste est : le taux de rendement obligataire le taux de rendement de l'indice action de reference le taux de rendement de l'indice immobilier de reference le taux de rendement de l'indice tresorerie de reference
tx_cible_prec	une valeur numeric correspondant au taux cible de la periode precedente.

Value

La valeur du taux cible.

Author(s)

Prim'Act

See Also

Le calcul du taux cible [calc_tx_cible_ref_marche](#).

get_qx_mort

Recuperer les taux de deces calcules.

Description

get_qx_mort est une methode permettant d'executer le calcul des taux de deces.

Usage

```
get_qx_mort(x, nom_table, age, gen)
```

Arguments

x	un objet de la classe HypTech .
nom_table	un character designant le nom d'une table de mortalite.
age	est la valeur integer de l'age.
gen	est la valeur integer de la generation.

Value

Le taux de deces.

Author(s)

Prim'Act

See Also

Le calcul du taux de deces [calc_qx](#).

get_qx_rach	<i>Recuperer les taux de rachat calcules.</i>
-------------	---

Description

get_qx_rach est une methode permettant d'executer le calcul des taux de rachat structurel. Il peut s'agir soit de taux de rachat partiels, soit de taux de rachat totaux.

Usage

```
get_qx_rach(x, nom_table, age, anc)
```

Arguments

x	un objet de la classe HypTech .
nom_table	un character designant le nom d'une table de rachat.
age	est la valeur numeric de l'age.
anc	est la valeur numeric de l'anciennete du contrat.

Details

Selon le nom de la table nom_table, le resultat de cette fonction sera un taux de rachat partiel ou un taux de rachat total.

Value

Le taux de rachat.

Author(s)

Prim'Act

See Also

Le calcul du taux de rachat [calc_rach](#).

get_rach_dyn	<i>Recuperer les taux de rachat dynamiques calcules.</i>
--------------	--

Description

get_rach_dyn est une methode permettant d'executer le calcul des taux de rachat dynamique.

Usage

```
get_rach_dyn(x, nom_table, tx_cible, tx_serv)
```

Arguments

x	un objet de la classe HypTech .
nom_table	un nom de jeu de parametre de rachat dynamique.
tx_cible	est une valeur numeric correspondant taux de revalorisation cible.
tx_serv	est une valeur numeric correspondant taux de revalorisation servi.

Value

Le taux de rachat dynamique.

Author(s)

Prim'Act

See Also

Le calcul du taux de rachat dynamique [calc_rach_dyn](#).

HypCanton	<i>La classe HypCanton.</i>
-----------	-----------------------------

Description

Une class de parametres pour les parametres generaux du canton.

Slots

tx_soc une valeur numeric correspondant au taux de prelevements social.
tx_import une valeur numeric correspondant au taux d'impot sur le resultat.
method_taux_cible un character correspond au nom de la methode de calcul du taux cible.

Note

Dans la version courante, la valeur de method_taux_cible doit etre parametree a "Meth1".

Author(s)

Prim'Act

HypTech	<i>La classe HypTech.</i>
---------	---------------------------

Description

Une classe contenant les listes de tables de mortalite, de rachat, les parametres de rachat dynamique et les parametres comportementaux qui permettent de calculer les attentes en matiere de taux de revalorisation cible.

Details

Chaque element de ces listes doit avoir ete prealablement nomme.

Slots

tables_mort une liste contenant des tables de mortalite au format [ParamTableMort](#).
 tables_rach une liste contenant des tables de rachat (structurel) au format [ParamTableRach](#).
 param_rach_dyn une liste contenant des parametres de rachat dynamique au format [ParamRachDyn](#).
 param_comport une liste contenant des parametres comportementaux au format [ParamComport](#).

Author(s)

Prim'Act

See Also

Les classes de parametres contenues : [ParamTableMort](#), [ParamTableRach](#), [ParamRachDyn](#), [ParamComport](#).
 La methode pour l'application des chocs de mortalite et de longevite : [get_choc_table](#). La methode pour l'application des chocs de rachat haut et bas : [get_choc_rach](#). La methode pour la recuperation des parametres comportementaux : [get_comport](#). La methode pour la recuperation des taux de deces : [get_qx_mort](#). La methode pour la recuperation des taux de rachat structurel : [get_qx_rach](#). La methode pour la recuperation des taux de rachat dynamique : [get_rach_dyn](#).

hyp_canton_load	<i>Methode permettant de charger la valeur initiale des hypotheses du canton.</i>
-----------------	---

Description

hyp_canton_load est une methode permettant de charger les parametres associees a un objet de classe [HypCanton](#).

Usage

```
hyp_canton_load(file_hyp_canton_address)
```

Arguments

file_hyp_canton_address
 est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner un objet [HypCanton](#).

Value

L'objet de la classe [HypCanton](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

Immo

Classe pour les actifs de type immobilier.

Description

Classe pour les actifs de type immobilier.

Slots

`ptf_immo` est un dataframe, chaque ligne represente un actif immobilier du portefeuille d'immobilier.

Lien a creer

NA

Author(s)

Prim'Act

See Also

Les operations d'achat vente immo [buy_immo](#) et [sell_immo](#).

Initialisation

La classe Initialisation.

Description

Une classe permettant de gerer les parametres techniques necessaire a l'initialisation d'une etude.

Slots

`root_address` ce character doit correspondre a la racine du projet. C'est dans les sous dossiers de cet emplacement que l'ensemble des donnees, parametres et dossiers de sauvegarde doivent se situer, en respectant l'architecture etablie par Prim'Act.

`address` est une liste renseignee par la fonction [set_architecture](#) qui contient l'ensemble des adresses de l'architecture physique du projet (emplacement des donnees utilisateurs, emplacement des parametres utilisateurs, emplacement des sauvegardes temporaires et definitives).

`nb_simu` est un integer correspondant aux nombres de trajectoires simulees par le jeu de donnees de l'ESG Prim'Act.

`nb_annee_proj` est un integer correspondant au nombre d'annee de projection de la modelisation.

Note

Il est necessaire que l'attribut `nb_annee_proj` corresponde au nombre d'annee de projection des donnees de l'ESG Prim'Act.

Author(s)

Prim'Act

See Also

La mise en place de l'architecture de chargement des donnees et parametres renseignes par l'utilisateur [set_architecture](#), la creation et la sauvegarde du canton initial [init_SimBEL](#), la creation de l'architecture des scenarios central, de marche et de souscription de la formule standard ainsi que la creation des objets [Be](#) pour chacun de ces scenarios.

<code>initialisation_load</code>	<i>Chargement de certains attributs dans un objet Initialisation</i>
----------------------------------	--

Description

`initialisation_load` est la methode de chargement des attributs `nb_simu` et `nb_annee_proj` a partir des donnees de l'environnement utilisateur.

Usage

```
initialisation_load(x, file_lancement_address)
```

Arguments

<code>x</code>	un objet de la classe Initialisation .
<code>file_lancement_address</code>	nom complet (i.e. avec chemin d'accès et extension) du fichier contenant les parametres de lancement.

Value

Pas de sortie.

Note

Cette methode permet de creer l'objet [Canton](#) initial et de le sauvegarder dans le repertoire adequat de l'architecture.

Author(s)

Prim'Act

init_create_folder	<i>Creation de l'architecture de sauvegarde des scenarios et executions du code a partir de la racine renseignee.</i>
--------------------	---

Description

init_create_folder est une methode permettant de creer l'architecture de sauvegarde des scenarios et les executions du code a partir de la racine renseignee.

Usage

```
init_create_folder(x)
```

Arguments

x	objet de la classe Initialisation .
---	---

Value

En cas de bonne execution (i.e. l'ensemble des dossiers est cree ou ecrase) la methode renvoie un logical.

Note

Il est necessaire anterieurement a l'appel de cette fonction d'avoir dans un premier temps cree un objet [Initialisation](#) en lui ayant affecte une racine, puis dans un second temps d'avoir appele la methode [set_architecture](#) a ce meme objet.

Author(s)

Prim'Act

init_debut_pgg_psap	<i>Re-initialise un objet AutresReserves en debut d'annee.</i>
---------------------	--

Description

init_debut_pgg_psap est une methode permettant de re-initialiser les montants de PGG et de PSAP de debut de periode.

Usage

```
init_debut_pgg_psap(x)
```

Arguments

x	objet de la classe AutresReserves.
---	------------------------------------

Value

L'objet x reinitialise.

Author(s)

Prim'Act

init_scenario	<i>Initialisation des scenarios : central et de chocs d'un workspace.</i>
---------------	---

Description

init_scenario est la methode d'initialisation.

Usage

```
init_scenario(x)
```

Arguments

x	un objet de la classe Initialisation .
---	--

Value

Pas de sortie.

Note

Cette methode cree l'architecture, puis les objets [Be](#) correspondant a chacun des scenarios : central et de chocs de la formule standard.

Author(s)

Prim'Act

init_SimBEL	<i>Initialisation d'un workspace.</i>
-------------	---------------------------------------

Description

init_SimBEL est la methode d'initialisation d'un workspace.

Usage

```
init_SimBEL(x)
```

Arguments

x un objet de la classe [Initialisation](#).

Value

Pas de sortie.

Note

Cette methode permet de creer l'objet [Canton](#) initial et de le sauvegarder dans le repertoire adequat de l'architecture.

Author(s)

Prim'Act

init_tables	<i>Initialise les tables de la base de donnees. Cette fonction initialise les differentes tables de la base. Chacune des tables possede 4 colonnes : num_sim, num_an, prod et flux.</i>
-------------	---

Description

Initialise les tables de la base de donnees. Cette fonction initialise les differentes tables de la base. Chacune des tables possede 4 colonnes : num_sim, num_an, prod et flux.

Usage

```
init_tables(x)
```

Arguments

x est un objet de type [DataBase](#).

Author(s)

Prim'Act

insert_tables	<i>Fusionne et insere les donnees dans la DataBase</i>
---------------	--

Description

Fusionne et insere les donnees dans la [DataBase](#)

Usage

```
insert_tables(x, result_simu, ens_simu)
```

Arguments

x	est un objet de type DataBase .
result_simu	une liste contenant les tables a insérer dans la DataBase.
ens_simu	un vecteur de type numeric contenant les indices des elements de la liste a insérer dans la DataBase.

Author(s)

Prim'Act

load_action	<i>Methode permettant de charger le portefeuille action initial dans un objet de type Action.</i>
-------------	---

Description

load_action est une methode permettant de charger le portefeuille action.

Usage

```
load_action(file_action_address)
```

Arguments

file_action_address	est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner le portefeuille.
---------------------	--

Value

L'objet de la classe [Action](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [PortFin](#) et sa methode [chargement_PortFin](#).

load_be	<i>Recuperation de l'objet Be et affectation d'une base de donnees.</i>
---------	---

Description

load_be est la methode permettant de recuper un objet [Be](#).

Usage

```
load_be(x, choc_name)
```

Arguments

x	un objet de la classe Initialisation .
choc_name	un character indiquant le nom du choc.

Value

Pas de sortie.

Note

Cette methode permet de creer l'objet [Canton](#) initial et de le sauvegarder dans le repertoire adequat de l'architecture.

Author(s)

Prim'Act

load_epeuroind	<i>Methode permettant de charger un portefeuille epargne initital dans un objet de type EpEuroInd.</i>
----------------	--

Description

load_epeuroind est une methode permettant de charger un portefeuille epargne.

Usage

```
load_epeuroind(file_epeuroind_address)
```

Arguments

file_epeuroind_address	est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner le portefeuille.
------------------------	--

Value

L'objet de la classe [Action](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [PortPassif](#) et sa methode [load_pp](#).

load_ht

Methode permettant de charger la valeur des parametres techniques.

Description

load_ht est une methode permettant de charger les parametres associees a un objet de classe [HypTech](#).

Usage

load_ht(x)

Arguments

x est un objet de la classe [Initialisation](#) qui est utilise pour renseigner le chemin d'accès de tous les parametres techniques.

Value

L'objet de la classe [HypTech](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

load_immo

Methode permettant de charger le portefeuille action initial dans un objet de type [Immo](#).

Description

load_immo est une methode permettant de charger le portefeuille action.

Usage

load_immo(file_immo_address)

Arguments

file_immo_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner le portefeuille.

Value

L'objet de la classe [Immo](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [PortFin](#) et sa methode [chargement_PortFin](#).

load_oblig

Methode permettant de charger le portefeuille oblig initial dans un objet de type [Oblig](#).

Description

load_oblig est une methode permettant de charger le portefeuille oblig

Usage

```
load_oblig(file_oblig_address)
```

Arguments

file_oblig_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner le portefeuille.

Value

L'objet de la classe [Oblig](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [PortFin](#) et sa methode [chargement_PortFin](#).

load_pp	<i>Methode permettant de charger et d'instancier un portefeuille de passif.</i>
---------	---

Description

load_pp est une methode permettant de charger les parametres et les donnees associees a un objet de classe [PortPassif](#).

Usage

```
load_pp(x)
```

Arguments

x	est un objet de la classe Initialisation qui est utilise pour renseigner le chemin d'accès de tous les parametres et les donnees necessaires.
---	---

Value

L'objet de la classe [PortPassif](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

load_reteurorest	<i>Methode permettant de charger un portefeuille retraite initial dans un objet de type RetraiteEuroRest.</i>
------------------	---

Description

load_reteurorest est une methode permettant de charger un portefeuille retraite en phase de restitution.

Usage

```
load_reteurorest(file_reteurorest_address)
```

Arguments

file_reteurorest_address	est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner le portefeuille.
--------------------------	--

Value

L'objet de la classe [Action](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [PortPassif](#) et sa methode [load_pp](#).

load_treso	<i>Methode permettant de charger le portefeuille treso initital dans un objet de type Treso.</i>
------------	--

Description

load_treso est une methode permettant de charger le portefeuille treso

Usage

```
load_treso(file_treso_address)
```

Arguments

file_treso_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner le portefeuille.

Value

L'objet de la classe [Treso](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [PortFin](#) et sa methode [chargement_PortFin](#).

merge_actifs	<i>Aggregation des donnees relatives aux actifs.</i>
--------------	--

Description

Aggregation des donnees relatives aux actifs.

Usage

```
merge_actifs(action, immo, oblig, treso, sim)
```

Arguments

action	un data.frame contenant les donnees des actions a inserer dans la base..
immo	un data.frame contenant les donnees des immo a inserer dans la base.
oblig	un data.frame contenant les donnees des obligations a inserer dans la base.
treso	un data.frame contenant les donnees treso a inserer dans la base.
sim	un integer representant le numero de la simulation.

Author(s)

Prim'Act

merge_be	<i>Insertion des BE dans la table BE de la base de donnees.</i>
----------	---

Description

Insertion des BE dans la table BE de la base de donnees.

Usage

```
merge_be(be, sim, nom_produit)
```

Arguments

be	une matrix a inserer dans la base.
sim	un integer representant le numero de la simulation.
nom_produit	une liste de character contenant les noms des differents produits.

Author(s)

Prim'Act

merge_flux_fin	<i>Aggregation des donnees relatives aux flux financiers.</i>
----------------	---

Description

Aggregation des donnees relatives aux flux financiers.

Usage

```
merge_flux_fin(flux_fin, sim)
```

Arguments

flux_fin	est un objet de type matrix.
sim	un integer correspondant au numero de simulation.

Author(s)

Prim'Act

`merge_pb`*Aggregation des donnees relatives a la PB.*

Description

Aggregation des donnees relatives a la PB.

Usage

```
merge_pb(output_pb, sim)
```

Arguments

<code>output_pb</code>	est un objet de type <code>data.frame</code> .
<code>sim</code>	un integer correspondant au numero de simulation.

Author(s)

Prim'Act

`merge_table_be`*Jointure et insertion des donnees dans la base de donnees. Les donnees sont les ouputs (`output_be`) de la fonction [proj_an](#)*

Description

Jointure et insertion des donnees dans la base de donnees. Les donnees sont les ouputs (`output_be`) de la fonction [proj_an](#)

Usage

```
merge_table_be(prime, frais, prestation, prestation_fdb, sim, nom_produit)
```

Arguments

<code>prime</code>	une matrix a inserer dans la base.
<code>frais</code>	une matrix a inserer dans la base.
<code>prestation</code>	une matrix a inserer dans la base.
<code>prestation_fdb</code>	une matrix a inserer dans la base.
<code>sim</code>	un integer representant le numero de la simulation.
<code>nom_produit</code>	une liste de character contenant les noms des differents produits.

Author(s)

Prim'Act

<code>merge_table_produit</code>	<i>Jointure et insertion de donnees dans la DataBase. Les donnees sont les ouputs <code>output_produit</code> de la fonction <code>proj_an</code></i>
----------------------------------	---

Description

Jointure et insertion de donnees dans la DataBase. Les donnees sont les ouputs `output_produit` de la fonction `proj_an`

Usage

```
merge_table_produit(flux, stock, hors_model, fin, nb_annee, sim,
                    nom_produit)
```

Arguments

<code>flux</code>	une matrix a inserer dans la base.
<code>stock</code>	une matrix a inserer dans la base.
<code>hors_model</code>	un <code>data.frame</code> a inserer dans la base.
<code>fin</code>	un <code>data.frame</code> a inserer dans la base.
<code>nb_annee</code>	un integer indiquant le nombre d'annees de projection.
<code>sim</code>	un integer representant le numero de la simulation.
<code>nom_produit</code>	une liste de character contenant les noms des differents produits.

Author(s)

Prim'Act

ModelPointESG	<i>La classe ModelPointESG.</i>
---------------	---------------------------------

Description

Une classe pour une extraction de l'ESG pour une annee et une simulation particuliere.

Slots

`annee` une valeur integer correspondant a l'annee de projection.

`num_traj` une valeur integer correspondant au numero de simulation de l'ESG.

`indice_action` un `data.frame` contenant les valeurs prises par les indices actions pour l'annee et la simulation selectionnees.

`indice_immo` un `data.frame` contenant les valeurs prises par les indices immobiliers pour l'annee et la simulation selectionnees.

`indice_inflation` une valeur numeric correspondant a la valeur prise par l'indice inflation pour l'annee et la simulation selectionnees.

`yield_curve` un vecteur numeric contenant la structure par terme des taux d'interets spots pour l'annee et la simulation selectionnees. La courbe representee correspond aux valeurs des $R(k, k+i)$ ou i va de 1 au `nb_annee_proj`.

`deflateur` une valeur numeric correspondant a la valeur prise par le deflateur stochastique pour l'annee et la simulation selectionnees.

Author(s)

Prim'Act

See Also

Les methodes de chargement d'un ESG [chargement_ESG](#) et d'extraction d'un model point [extract_ESG](#).

Oblig*Classe pour les actifs de type obligation.*

Description

Classe pour les actifs de type obligation.

Slots

ptf_oblig est un dataframe, chaque ligne represente un actif obligation du portefeuille d'obligation.

Lien a creer

NA

Author(s)

Prim'Act

See Also

Les operations d'achat vente obligations [buy_oblig](#) et [sell_oblig](#).

ParamAlmEngine*La classe ParamAlmEngine.*

Description

Une classe pour le parametre ALM d'un canton.

Slots

ptf_reference est un objet de type [PortFin](#), qui represente le portefeuille d'investissement de reference d'un canton.

alloc_cible un vecteur de 4 elements rendant compte du pourcentage de l'actif composant respectivement les investissements: actions, immobiliers, obligataires et de tresorerie.

seuil_realisation_PVL une valeur numeric correspondant au pourcentage de plus-values actions qui peut etre liquidee chaque annee pour atteindre l'objectif de revalorisation cible des passifs.

Author(s)

Prim'Act

ParamBe	<i>La classe ParamBe.</i>
---------	---------------------------

Description

Une classe contenant le nombre d'annees de projection utilise pour le calcul du best estimate d'un assureur.

Slots

nb_annee un entier comprenant le nombre d'annees de projection.

Author(s)

Prim'Act

ParamChocMket	<i>La classe ParamChocMket.</i>
---------------	---------------------------------

Description

Une classe contenant les parametres des chocs de marche de la formule standard.

Slots

table_choc_action_type1 un data.frame contenant les parametres du choc action type 1.
 table_choc_action_type2 un data.frame contenant les parametres du choc action type 2.
 table_choc_immo un data.frame contenant les parametres du choc immobilier.
 table_choc_spread un data.frame contenant les parametres du choc de spread.
 table_choc_currency un data.frame contenant les parametres du choc currency.

Author(s)

Prim'Act

ParamChocSousc	<i>La classe ParamChocSousc.</i>
----------------	----------------------------------

Description

Une classe contenant les parametres des chocs souscription de la formule standard.

Slots

mp un data.frame contenant l'ensemble des parametres necessaires a l'application des chocs du module Souscription Vie.

Author(s)

Prim'Act

ParamComport

La classe de parametres de comportement ParamComport.

Description

Une classe pour les parametres de comportement.

Slots

mat_oblig une valeur numeric correspondant a la maturite du taux de rendement obligatoire pris en reference sur le marche.

alloc_mar un vecteur numeric correspondant a l'allocation pris en reference sur le marche. Le format de cette liste est :

le taux de rendement obligatoire

le taux de rendement de l'indice action de reference

le taux de rendement de l'indice immobilier de reference

le taux de rendement de l'indice tresorerie de reference.

w_n une valeur numeric correspondant au poids accorde au rendement de l'annee courante par rapport a l'annee precedente.

marge_mar une valeur numeric correspondant a la marge financiere pris en reference sur le marche.

ch_enc_mar une valeur numeric correspondant au niveau de chargement sur encours pris en reference sur le marche.

ind_ref_action une valeur numeric correspondant au numero de l'indice action pris en reference sur le marche.

ind_ref_immo une valeur numeric correspondant au numero de l'indice immobilier pris en reference sur le marche.

Author(s)

Prim'Act

See Also

Le calcul du taux cible [calc_tx_cible_ref_marche](#).

ParamRachDyn

La classe de parametres de rachat dynamique ParamRachDyn.

Description

Une classe pour les parametres de des lois de rachat dynamique.

Slots

vec_param un data frame contenant les parametres pour les rachats dynamiques.

Author(s)

Prim'Act

See Also

Le calcul du taux de rachat dynamique [calc_rach_dyn](#).

ParamRevaloEngine	<i>La classe ParamRevaloEngine. Une Classe pour les parametres utilises pour la gestion de la revalorisation.</i>
-------------------	---

Description

La classe ParamRevaloEngine. Une Classe pour les parametres utilises pour la gestion de la revalorisation.

Slots

taux_pb_fi une valeur numeric correspondant au taux de participation applique au resultat financier.

taux_pb_tech une valeur numeric correspondant au taux de participation applique au resultat technique.

tx_marge_min une valeur numeric correspondant au taux de marge minimal auquel s'attend l'assureur.

solde_pb_regl une valeur numeric correspondant au solde deficitaire de participation aux benefices reglementaire. Cette valeur doit etre negative.

Author(s)

Prim'Act

ParamTableMort	<i>La classe de parametres pour les tables de mortalite ParamTableMort.</i>
----------------	---

Description

Une classe de parametres pour les tables de mortalite.

Slots

age_min un entier correspondant a l'age minimal de la table.

age_max un entier correspondant a l'age maximal de la table.

gen_min un entier correspondant a la premiere generation de la table.

gen_max un entier correspondant a la derniere generation de la table.

table un data frame contenant la table de mortalite.

Note

Les tables de mortalite peuvent contenir des probabilites de deces par age (qx) ou le nombre de vivants par age (lx).

Author(s)

Prim'Act

See Also

Le calcul du taux de deces [calc_qx](#).

ParamTableRach

La classe de parametres pour les table de rachat ParamTableRach.

Description

Une classe de parametres pour les tables de rachat.

Details

Une table de rachat peut etre une table de rachat partiel ou une table de rachat total. Pour une table de rachat partiel, les taux de rachat sont exprimes en pourcentage de l'encours. Pour une table de rachat total, les taux de rachat sont exprimes en pourcentage du nombre de contrats.

Slots

age_min un entier correspondant a l'age minimal de la table.

age_max un entier correspondant a l'age maximal de la table.

anc_min un entier correspondant a la premiere anciennete de la table.

anc_max un entier correspondant a la derniere anciennete de la table.

table un `data.frame` contenant la table de rachat.

Author(s)

Prim'Act

See Also

Le calcul du taux de rachat [calc_rach](#).

param_alm_engine_load	<i>Chargement des attributs d'un objet ParamAlmEngine a partir des donnees utilisateurs.</i>
-----------------------	--

Description

param_alm_engine_load est la methode de chargement des attributs d'un objet [ParamAlmEngine](#) a partir des donnees de l'environnement utilisateur et d'un portefeuille financier de reference (charge par la fonction [chargement_PortFin_reference](#)).

Usage

```
param_alm_engine_load(file_alm_address, ptf_fin_ref)
```

Arguments

file_alm_address	un character contenant l'adresse exacte du fichier d'input utilisateur.
ptf_fin_ref	un objet de la classe PortFin correspondant au portefeuille de reinvestissement.

Value

L'objet de la classe [ParamAlmEngine](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

param_revalo_load	<i>Chargement des attributs d'un objet ParamRevaloEngine a partir des donnees utilisateurs.</i>
-------------------	---

Description

param_revalo_load est la methode de chargement des attributs d'un objet [ParamRevaloEngine](#) a partir des donnees de l'environnement utilisateur.

Usage

```
param_revalo_load(file_revalo_address)
```

Arguments

file_revalo_address	un character contenant l'adresse exacte du fichier d'input utilisateur.
---------------------	---

Value

L'objet de la classe [ParamRevaloEngine](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

pb_contr	<i>Calcule la PB contractuelle.</i>
----------	-------------------------------------

Description

pb_contr est une methode permettant de calculer la participation aux benefices contractuelle par produit.

Usage

```
pb_contr(base_fin, tx_pb, rev_stock_brut, ch_enc_th, tx_enc_moy)
```

Arguments

base_fin	un vecteur numeric comprenant par produit la base de produits financiers.
tx_pb	un vecteur numeric comprenant par produit les taux de participation aux benefices contractuels.
rev_stock_brut	un vecteur de type numeric comprenant par produit la revalorisation appliquee sur le stock au taux minimum.
ch_enc_th	est un vecteur de type numeric comprenant par produit le montant total des chargements sur encours appliques au stock et revalorises au taux minimum. Il s'agit ici des chargements qui pourraient theoriquement etre preleves.
tx_enc_moy	un vecteur numeric comprenant par produit les taux de chargements sur encours moyens.

Details

Le montant des chargements ch_enc_th est theorique et peut conduire a l'application d'une revalorisation nette negative.

Value

ch_enc_ap_pb_contr un vecteur comprenant par produit les chargements sur encours appliques
rev_stock_nette_contr un vecteur comprenant par produit la revalorisation contractuelle nette.

Author(s)

Prim'Act

PortFin

*La classe PortFin.***Description**

Une Classe pour le portefeuille financier.

Slots

`annee` est un objet de type `integer` correspondant a l'annee courante.

`ptf_action` est un objet de type [Action](#), qui represente le portefeuille d'action d'un canton.

`ptf_immo` est un objet de type [Immo](#), qui represente le portefeuille immobilier d'un canton.

`ptf_oblig` est un objet de type [Oblig](#), qui represente le portefeuille obligataire d'un canton.

`ptf_treso` est un objet de type [Treso](#), qui represente le portefeuille monetaire d'un canton.

`pre` est un objet de type [PRE](#), qui represente la PRE d'un canton.

`rc` est un objet de type [RC](#), qui represente la RC d'un canton.

`frais_fin` est un objet de type [FraisFin](#), qui represente les frais financiers d'un canton.

`pvl_action` est un `numeric`, qui correspond a la somme des plus values latentes des actifs Actions qui sont en situation de plus values latentes.

`pvl_immo` est un `numeric`, qui correspond a la somme des plus values latentes des actifs Immo qui sont en situation de plus values latentes.

`pvl_oblig` est un `numeric`, qui correspond a la somme des plus values latentes des actifs Oblig qui sont en situation de plus values latentes.

`mvl_action` est un `numeric`, qui correspond a la somme des moins values latentes des actifs Actions qui sont en situation de moins values latentes.

`mvl_immo` est un `numeric`, qui correspond a la somme des moins values latentes des actifs Immos qui sont en situation de moins values latentes.

`mvl_oblig` est un `numeric`, qui correspond a la somme des moins values latentes des actifs Oblig qui sont en situation de moins values latentes.

`vm_vnc_precedent` est une liste composee de deux elements : la `vm_precedente` et la `vnc_precedente`, correspondant respectivement a la valeur de marche et a la valeur nette comptable en debut d'annee de l'objet PortFin.

Author(s)

Prim'Act

See Also

Le calcul des rendements : [calc_rdt](#). Le calcul des plus ou moins-values latentes : [calc_pmv1](#).
 Le chargement des donnees initiales : [chargement_PortFin](#), [chargement_PortFin_reference](#).
 Les methodes de mise a jour : [update_PortFin](#), [update_PortFin_reference](#), [do_update_pmv1](#),
[do_update_vm_vnc_precedent](#). L'allocation d'actifs : [print_alloc](#).

PortPassif

La classe PortPassif.

Description

Une classe regroupant l'ensemble des donnees de passifs et les hypotheses correspondantes.

Slots

annee une valeur entiere correspondant a l'annee de projection.

eei une liste d'objets de la classe [EpEuroInd](#) contenant l'ensemble des produits de type epargne en euros.

rer une liste d'objets de la classe [RetraiteEuroRest](#) contenant l'ensemble des produits de retraite en phase de restitution.

names_class_prod un vecteur character indiquant les noms de classes de produits.

ht un objet de classe [HypTech](#) contenant les hypotheses techniques.

fp un objet de classe [FraisPassif](#) contenant les hypotheses de frais de passif par produit.

tx_pb un objet de classe [TauxPB](#) contenant les taux contractuel de participation aux benefices par produit.

autres_passifs un objet de classe [AutresPassifs](#).

autres_reserves un objet de classe [AutresReserves](#).

calc_proba un logical indiquant si les probabilites ont ete calculees.

choc_lapse_mass un numeric comprenant la valeur du choc de rachat massif. Cet attribut doit etre a 0 lorsque aucun choc n'est souhaite. Sa valeur est comprise entre 0 et 1. Il ne s'applique que la premiere annee de projection.

choc_mort_cat un numeric comprenant la valeur du choc de mortalite catastrophique Cet attribut doit etre a 0 lorsque aucun choc n'est souhaite. Sa valeur est comprise entre 0 et 1. Il ne s'applique que la premiere annee de projection.

Author(s)

Prim'Act

See Also

La projection des produits sur l'annee avant attribution de participation aux benefices : [proj_annee_av_pb](#).
Le vieillissement des model points de passifs avant et apres attribution de participation aux benefices : [viellissement_av_pb](#), [vieillissement_ap_pb](#).

Ppb	<i>La classe Ppb.</i>
-----	-----------------------

Description

Classe pour la provision pour participation aux benefices (PPB)

Slots

hist_ppb est le vecteur contenant les valeurs courantes numeric prise par la PPB sur les huit dernieres annees.

valeur_ppb est la valeur courante numeric prise par la PPB.

ppb_debut est la valeur prise numeric par la PPB en debut d'annee.

seuil_rep est une valeur numeric correspondant a la proportion de PPB de debut d'annee que l'on peut reprendre sur une periode.

seuil_dot est une valeur numeric correspondant au montant maximal de dotation possible sur la PPB sur une periode, exprimee comme une fraction de la PPB de debut d'annee.

compte_rep est une valeur numeric qui totalise les montants de reprises effectuees sur une periode.

compte_dot est une valeur numeric qui totalise les montants de dotations effectuees sur une periode.

Author(s)

Prim'Act

See Also

La dotation et la reprise de PPB : [calc_dotation_ppb](#), [calc_reprise_ppb](#).

ppb_8ans	<i>Calcule la valeur de la PPB a distribue en appliquant la regle des 8 ans.</i>
----------	--

Description

ppb_8ans est une methode permettant d'appliquer la regle des 8 ans.

Usage

```
ppb_8ans(x)
```

Arguments

x un objet de la classe [Ppb](#).

Value

une liste contenant les parametres

ppb_8 : la valeur numeric correspondant au montant de la ppb de l'annee t-8.

ppb : un objet [Ppb](#) correspondant a la PPB mise a jour.

Author(s)

Prim'Act

ppb_load	<i>Methode permettant de charger les valeurs des hypotheses et des donnees de PPB</i>
----------	---

Description

ppb_load est une methode permettant de charger les parametres associees a un objet de classe [Ppb](#).

Usage

```
ppb_load(file_ppb_address)
```

Arguments

file_ppb_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner un objet [Ppb](#).

Value

L'objet de la classe [Ppb](#) construit a partir des inputs renseignes par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

PRE	<i>La classe PRE.</i>
-----	-----------------------

Description

Une classe pour la gestion de la provision pour risque d'exigibilite (PRE).

Slots

`val_debut` est une valeur numeric correspondant a la valeur de la PRE en debut d'annee.

`val_courante` est une valeur numeric correspondant a la valeur courante de la PRE.

Author(s)

Prim'Act

See Also

Les methodes de calcul de la PRE [calc_PRE](#), et de mises a jour des PRE initiales et courantes [do_update_PRE_val_courante](#), [do_update_PRE_val_debut](#).

pre_load	<i>Chargement de la valeur initiale de la PRE</i>
----------	---

Description

`pre_load` est une methode permettant de charger la valeur de PRE initiale dans un objet de type PRE.

Usage

```
pre_load(file_PRE_address)
```

Arguments

`file_PRE_address`
est un character correspondant a l'adresse du fichier d'input renseignant les donnees de PRE

Value

Un objet de la classe [PRE](#) charge a partir des donnees du fichier dont le nom est precise en input.

Author(s)

Prim'Act

print_alloc	<i>Calcul le poids de chaque composante du portefeuille.</i>
-------------	--

Description

print_alloc est une methode permettant de calculer l'allocation absolue et relative du portefeuille.

Usage

```
print_alloc(x)
```

Arguments

x objet de la classe [PortFin](#).

Value

Un data.frame compose de quatre colonnes et cinq lignes. La colonne

alloc_valeur : decrit le montant alloue en valeur de marche par poche d'actif.

alloc_proportion : decrit la proportion allouee en valeur de marche par poche d'actif.

alloc_valeur_vnc : decrit le montant alloue en valeur nette comptable par poche d'actif.

alloc_proportion_vnc : decrit la proportion allouee en valeur nette comptable par poche d'actif.

Les lignes correspondent aux classes d'actifs : (Action / Immobilier / Obligation / Tresorerie / Actifs cumules)

Author(s)

Prim'Act

proj_an	<i>Projetter un canton sur une periode.</i>
---------	---

Description

proj_an est une methode permettant de projeter un canton sur une annee. Cette methode calcule les flux de best estimate des passifs et fait vieillir d'une annee les elements du canton.

Usage

```
proj_an(x, annee_fin, pre_on)
```

Arguments

x est un objet de type [Canton](#).

annee_fin est une valeur integer correspondant a l'annee de fin de projection.

pre_on est une valeur logical qui lorsqu'elle vaut TRUE prend en compte la variation de PRE dans le resultat technique, utilisee pour le calcul de la participation aux benefices reglementaires.

Details

Cette methode est la procedure central du package SimBEL puisqu'elle cohorde les interactions entre les actifs et les passifs, declenche l'algorithme de revalorisation, calcule le resultat comptable et evalue les flux de best estimate.

Value

canton l'objet x vieilli d'une annee.

annee l'annee de projection.

nom_produit le nom des produits de passifs consideres.

output_produit une liste comprenant les variables de flux, les variables de stocks et les resultats des passifs non-modelises.

output_be une liste comprenant les flux utilises pour le calcul du best estimate par produit.

result_tech la valeur du resultat technique.

result_fin la valeur du resultat financier.

tra la valeur du taux de rendement de l'actif.

result_brut la valeur du resultat brut d'impot.

result_net la valeur du resultat net d'impot.

Author(s)

Prim'Act

See Also

Le vieillissement du portefeuille de passif avant PB : [viellissement_av_pb](#). Le vieillissement du portefeuille financier : [update_PortFin](#), [update_PortFin_reference](#). L'affiche de l'etat courant du portefeuille financier : [print_alloc](#). Le calcul des frais financier : [calc_frais_fin](#). La reallocation du portefeuille financier : [reallocate](#). Le calcul de la PRE : [calc_PRE](#). Le calcul du resultat technique : [calc_result_tech](#), [calc_result_tech_ap_pb](#). Le calcul du resultat financier et du TRA : [calc_resultat_fin](#), [calc_tra](#). L'application de l'algorithme d'attribution de la participation aux benefices : [calc_revalo](#). Le vieillissement du portefeuille de passif apres PB : [vieillissement_ap_pb](#). Les autres methodes de vieillissement des actifs et de passifs: [sell_pvl_action](#), [do_update_pmv1](#), [do_update_PRE_val_courante](#), [do_update_vm_vnc_precedent](#), [vieillissement_ppb](#), [do_update_RC_val_debut](#), [do_update_PRE_val_debut](#), [init_debut_pgg_psap](#). Le calcul des fins de projection : [calc_fin_proj](#).

proj_annee_autres_passifs

Extrait les flux et les PM des produits non modelises

Description

proj_annee_autres_passifs est une methode permettant de calculer les PM et les flux sur une annee pour des passif non modelises. Cette methode calcule applique une inflation au frais.

Usage

proj_annee_autres_passifs(an, x, coef_inf)

Arguments

an	est l'annee de projection.
x	un objet de la classe <code>AutresPassifs</code> contenant l'ensemble des donnees de passifs non modelises.
coef_inf	un numeric correpondant au coefficient d'inflation a appliquer sur les frais.

Value

Un `data.frame` contenant les flux des passifs de l'annee.

Author(s)

Prim'Act

proj_annee_av_pb	<i>Calcule les flux et les PM des produits modelises</i>
------------------	--

Description

proj_annee_av_pb est une methode permettant de calculer les PM et les flux sur une annee avant PB. Cette methode calcule egalement les frais sur flux et sur primes.

Usage

```
proj_annee_av_pb(an, x, tx_soc, coef_inf, list_rd)
```

Arguments

an	une valeur integer correspondant a l'annee de projection.
x	un objet de la classe <code>PortPassif</code> contenant l'ensemble des produits de passifs.
tx_soc	une valeur numeric correspondant au taux de charges sociales.
coef_inf	une valeur numeric correspondant au coefficient d'inflation considere pour le traitement des frais.
list_rd	un vecteur contenant les rendements de reference. Le format de cette liste est : le taux de rendement obligataire le taux de rendement de l'indice action de reference le taux de rendement de l'indice immobilier de reference le taux de rendement de l'indice tresorerie de reference

Details

L'annee de projection est utilisee pour gerer les produits dont les clauses dependent de l'annee. Cette methode calcule deux fois les prestations et les PM pour permettre de calculer le montant de FDB.

Value

x l'objet pour lequel les tableaux de resultats des objets [EpEuroInd](#) sont mis a jour.

nom_produit un vecteur de character contenant les noms des produits.

flux_agg une matrice contenant les flux agreges par produits.

stock_agg une matrice contenant les stocks agreges par produits.

Author(s)

Prim'Act

See Also

La classe [EpEuroInd](#) et ses methodes. La classe [FraisPassif](#) et ses methodes.

RC	<i>La classe RC.</i>
----	----------------------

Description

Une classe pour la gestion de la reserve de capitalisation (RC).

Slots

val_debut est une valeur numeric correspondant a la valeur de la RC en debut d'annee.

val_courante est une valeur numeric correspondant a la valeur courante de la RC.

Author(s)

Prim'Act

See Also

Les methodes de calcul de la RC [calc_RC](#), et de mises a jour des RC initiales et courantes [do_update_RC_val_courante](#), [do_update_RC_val_debut](#).

rc_load	<i>Chargement de la valeur initiale de la RC</i>
---------	--

Description

rc_load est une methode permettant de charger la valeur de RC initiale dans un objet de type RC.

Usage

```
rc_load(file_RC_address)
```

Arguments

file_RC_address

est un character correspondant a l'adresse du fichier d'input renseignant les donnees de RC

Value

Un objet de la classe [RC](#) charge a partir des donnees du fichier dont le nom est precise en input.

Author(s)

Prim'Act

reallocate	<i>Realise les operations d'achats ventes</i>
------------	---

Description

reallocate est une methode permettant d'ajuster l'allocation du [PortFin](#) de l'assureur.

Usage

```
reallocate(x, ptf_reference, alloc_cible)
```

Arguments

x objet de la classe [PortFin](#).

ptf_reference est le portefeuille de reinvestissement. C'est un objet de la classe [PortFin](#).

alloc_cible est un vecteur de type numeric constitue de 4 elements, il contient les proportions cibles d'allocations action, immobilier, obligataire et de tresorerie.

Value

portFin l'objet initial de la classe [PortFin](#) realloue a l'allocation cible.

pmvr le montant total des plus ou moins values realisees.

pmvr_oblig le montant des plus ou moins values obligataires realisees lors de la reallocation.

pmvr_action le montant des plus ou moins values action realisees lors de l'etape de reallocation.

pmvr_immo le montant des plus ou moins values immobilieres realisees lors de l'etape de reallocation.

var_rc la variation de la reserve de capitalisation induite par la reallocation.

var_pre la variation de la provision pour risque d'exigibilite induite par la reallocation.

plac_moy_vm la valeur de marche moyenne des placements de l'assureur au cours de l'operation de reallocation.

plac_moy_vnc la valeur nette comptable moyenne des placements de l'assureur au cours de l'operation de reallocation.

Note

Les operations d'achat/vente sont effectuees en termes de nombre d' unite d'achat/vente.

Author(s)

Prim'Act

See Also

La classe [PortFin](#).

RetraiteEuroRest

La classe RetraiteEuroRest.

Description

Une classe pour les passifs de retraite en euros en phase de restitution.

Slots

mp un objet `data.frame` au format `fige` contenant l'ensemble de model points retraite euros en phase de restitution.

tab un objet de la classe [TabRetEuroRest](#) dedie au stockage de variables intermediaires.

tab_proba un objet de la classe [TabProbaRetEuroRest](#) dedie au stockage des probabilites associees aux model point.

Author(s)

Prim'Act

See Also

Le calcul des prestations et des PM : [calc_prest](#), [calc_pm](#). Le calcul du taux minimum et des taux cible de revalorisation : [calc_tx_min](#), [calc_tx_cible](#). La revalorisation des PM apres participation aux benefices : [calc_revalo_pm](#). Le vieillissement des model points sur une periode : [vieilli_mp](#).

RevaloEngine

La classe RevaloEngine.

Description

Une classe comprenant les methodes pour l'application de la revalorisation des passifs.

Slots

param_revalo est objet de type [ParamRevaloEngine](#) comprenant les parametres utilises pour la revalorisation des contrats.

Author(s)

Prim'Act

revalo_action	<i>Revalorise les valeurs de marche du portefeuille action.</i>
---------------	---

Description

revalo_action est une methode permettant de revaloriser et de calculer les dividendes du portefeuille action sur une periode.

Usage

```
revalo_action(x, S, S_prev)
```

Arguments

x	un objet de la classe Action (decrivant le portefeuille d'action).
S	un vecteur numeric correspondant a la valeur des indices actions
S_prev	un vecteur numeric correspondant a la valeur des indices actions a la periode precedente.

Value

Un data frame compose de deux colonnes et autant de lignes que le portefeuille action a de lignes. La premiere colonne decrit de le rendement annuel de chacune des actions composants le portefeuille action. La seconde colonne decrit les dividendes annuelles perçues au titre de chacune des actions composants le portefeuille action.

Author(s)

Prim'Act

revalo_immo	<i>Revalorise les valeurs de marche du portefeuille immobilier.</i>
-------------	---

Description

revalo_immo est une methode permettant de revaloriser et de calculer les loyers du portefeuille immobilier sur une periode.

Usage

```
revalo_immo(x, S, S_prev)
```

Arguments

x	un objet de la classe Immo (decrivant le portefeuille d'immobilier).
S	un vecteur numeric correspondant a la valeur des indices immobiliers
S_prev	un vecteur numeric correspondant a la valeur des indices immobiliers a la periode precedente.

Value

Un data frame compose de deux colonnes et autant de lignes que le portefeuille immobilier a de lignes. La premiere colonne decrit de le rendement annuel de chacune des lignes d'immobilier composants le portefeuille immobilier. La seconde colonne decrit les loyers annuelles percues au titre de chacune des lignes d'immobilier composants le portefeuille immobilier.

Author(s)

Prim'Act

revalo_treso	<i>Calcul les valeurs de marches de chaque composante du portefeuille treso.</i>
--------------	--

Description

revalo_treso est une methode permettant de calculer les valeurs de marche.

Usage

```
revalo_treso(Rt, Rt_prev)
```

Arguments

Rt	valeur du taux zero coupon 1 an de l'annee.
Rt_prev	valeur du taux zero coupon 0 an de l'annee.

Value

Le taux de rendement.

Author(s)

Prim'Act

revenu_treso	<i>Calcule le revenu tresorerie.</i>
--------------	--------------------------------------

Description

revenu_treso est une methode permettant de calculer le revenu de la tresorerie.

Usage

```
revenu_treso(x, rdt, flux_milieu)
```

Arguments

x	est un objet de la classe Treso en debut d'annee.
rdt	est le rendement de la classe Treso au cours de l'annee (i.e. en fin d'annee)
flux_milieu	est le flux du milieu de l'annee en cours.

Value

Le montant du revenu.

Author(s)

Prim'Act

run_be	<i>Calcul d'un BE.</i>
--------	------------------------

Description

run_be est une methode permettant de calculer un best estimate pour un canton.

Usage

```
run_be(x, pre_on, parallel, nb_coeur = 0L)
```

Arguments

x	un objet de type Be .
pre_on	une valeur logical qui lorsqu'elle vaut TRUE prend en compte la variation de PRE dans le resultat technique utilisee pour le calcul de la participation aux benefices reglementaires.
parallel	une valeur logical qui indique si les calculs seront parallelises.
nb_coeur	une valeur integer qui indique le nombre de coeurs utilises dans le cas ou les calculs sont parallelises. Par default cette valeur est egale a 0.

Details

Il s'agit de la methode principale du package SimBEL. Cette methode requiert le chargement d'un objet [Be](#) deja parametre et alimente en donnees. La methode [init_scenario](#) permet d'alimenter un objet [Be](#) dans la situation "centrale" de la formule standard et en situation de choc.

Value

be l'objet x mis a jour : l'attribut tab_be contient le best estimate et sa decomposition, l'attribut tab_flux contient les flux moyens du best estimate et ses composantes.

err_simu un vecteur contenant la liste des simulations qui ont generees des erreurs et qui n'ont pu etre utilisees pour le calcul du best estimate.

Author(s)

Prim'Act

See Also

Le calcul du best estimate pour une simulation : [run_be_simu](#). L'initialisation d'un best estimate : [init_scenario](#). La classe [Be](#). La sortie des resultats au format ".csv" : [write_be_results](#).

run_be_simu	<i>Calcul d'un BE par une simulation.</i>
-------------	---

Description

run_be_simu est une methode permettant de calculer un best estimate pour une simulation donnee.

Usage

```
run_be_simu(x, i, pre_on)
```

Arguments

x	un objet de type Be .
i	un entier (integer) correspondant au numero de la simulation.
pre_on	une valeur logical qui lorsqu'elle vaut TRUE prend en compte la variation de PRE dans le resultat technique utilisee pour le calcul de la participation aux benefices reglementaires.

Details

Pour une simulation donnee, cette methode projette un [Canton](#) jusqu'au terme, parametre dans l'objet x.

Value

resultats une liste dont le format est le suivant :

nom_produit : un vecteur contenant le liste des noms de produits..

prime : une matrice contenant les flux de primes par produit.

prestation : une matrice contenant les flux de prestations par produit.

prestation_fdb : une matrice contenant les flux de prestations discretionnaires par produit.

frais : une matrice contenant les flux de frais par produit.

flux_be : une matrice contenant les flux de best estimate par produit.

prime_actu : une matrice contenant la valeur des primes actualisees par produit.

prestation_actu : une matrice contenant la valeur des prestations actualisees par produit.

prestation_fdb_actu : une matrice contenant la valeur des prestations discretionnaires actualisees par produit.

frais_actu : une matrice contenant la valeur des frais actualisees par produit.

be : une matrice contenant la valeur du best estimate par produit.

canton un objet de type [Canton](#).

Author(s)

Prim'Act

See Also

La methode de projection d'un [Canton](#) : [proj_an](#). L'extraction d'une simulation de l'[ESG](#) : [extract_ESG](#).

La classe [Be](#).

sell_action	<i>Mise a jour de chaque composante du portefeuille action suite a la vente de tout ou partie de ce portefeuille.</i>
-------------	---

Description

sell_action est une methode permettant de mettre a jour chaque composante d'un portefeuille action suite a la vente de tout ou partie de ce portefeuille.

Usage

```
sell_action(x, num_sold, nb_sold)
```

Arguments

x	objet de la classe Action (decrivant le portefeuille action en detention).
num_sold	vecteur de type numeric contenant le numero de model point action du portefeuille que l'on souhaite vendre.
nb_sold	vecteur de type numeric contenant le nombre d'unite que l'on souhaite vendre (a autant de ligne que le vecteur num_sold).

Value

action l'objet x mis a jour de l'operation de vente (suppression des lignes vendues).
pmvr le montant des plus ou moins-values realisees.

Author(s)

Prim'Act

sell_immo	<i>Mise a jour de chaque composante du portefeuille immobilier suite a la vente de tout ou partie de ce portefeuille.</i>
-----------	---

Description

sell_immo est une methode permettant de mettre a jour chaque composante d'un portefeuille immobilier suite a la vente de tout ou partie de ce portefeuille.

Usage

```
sell_immo(x, num_sold, nb_sold)
```

Arguments

x	objet de la classe Immo (decrivant le portefeuille immobilier en detention).
num_sold	vecteur de type numeric contenant le numero de model point immobilier du portefeuille que l'on souhaite vendre.
nb_sold	vecteur de type numeric contenant le nombre d'unite que l'on souhaite vendre (a autant de ligne que le vecteur num_sold).

Value

immo l'objet x mis a jour de l'operation de vente (suppression des lignes vendues).
pmvr le montant des plus ou moins-values realisees.

Author(s)

Prim'Act

sell_oblig	<i>Mise a jour de chaque composante du portefeuille obligation suite a la vente de tout ou partie de ce portefeuille.</i>
------------	---

Description

sell_oblig est une methode permettant de mettre a jour chaque composante d'un portefeuille obligation suite a la vente de tout ou partie de ce portefeuille.

Usage

```
sell_oblig(x, num_sold, nb_sold)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligation en detention).
num_sold	vecteur de type numeric contenant le numero de model point obligation du portefeuille que l'on souhaite vendre.
nb_sold	vecteur de type numeric contenant le nombre d'unite que l'on souhaite vendre (a autant de ligne que le vecteur num_sold).

Value

oblig l'objet x mis a jour de l'operation de vente (suppression des lignes vendues).
pmvr le montant des plus ou moins-values realisees.

Author(s)

Prim'Act

sell_pvl_action	<i>Mise a jour du portefeuille action suite a une realisation de plus-values latentes action.</i>
-----------------	---

Description

sell_pvl_action est une methode permettant de mettre a jour chaque composante d'un portefeuille action suite a la vente de tout ou partie de ce portefeuille afin de realiser un montant de plus-values latentes.

Usage

```
sell_pvl_action(x, montant)
```

Arguments

x	un objet de la classe Action (decrivant le portefeuille action en detention).
montant	un montant numeric de plus-values latentes que l'on souhaite realiser.

Value

action l'objet x mis a jour de l'operation de vente (suppression des lignes vendues).
pmvr le montant des plus ou moins-values realisees.

Note

Les cessions sont realisees au prorata des plus-values latentes actions.

Author(s)

Prim'Act

set_architecture	<i>Definition de l'architecture d'un workspace.</i>
------------------	---

Description

set_architecture.

Usage

```
set_architecture(x)
```

Arguments

x	un objet de la classe Initialisation .
---	--

Value

Objet mis a jour de l'ensemble des chemins du workspace, ceux ci sont stockes sous forme de liste dans l'attribut address de l'objet [Initialisation](#) renseigne en input.

Author(s)

Prim'Act

SimBEL	<i>SimBEL: Un package de calcul du best estimate epargne sous Solvabilite 2.</i>
--------	--

Description

SimBEL fourni un ensemble de fonctionnalites pour permettre l'évaluation d'un best estimate epargne sous Solvabilite 2. L'utilisation de ce package necessite au prealable de disposer de donnees stockees dans un repertoire dont le format est predetermine par la societe Prim'Act. Ce package est developpe a partir d'objet de type S4.

Details

Ce package comprends :

- une modelisation d'un canton auquel est relie un portefeuille d'actifs et un portefeuille de passif. SimBEL gere les interactions entre ces deux objets.
- une modelisation du best estimate pour des produits d'epargne en euros.
- d'appliquer les principaux chocs de la formule standard.

TabEpEuroInd	<i>La classe TabEpEuroInd.</i>
--------------	--------------------------------

Description

Une classe pour le stockage en memoire de variable de calcul au niveau du model point [EpEuroInd](#).

Slots

tab un objet list au format fige contenant l'ensemble des variables stockees.

Author(s)

Prim'Act

TabProbaEpEuroInd	<i>La classe TabProbaEpEuroInd.</i>
-------------------	-------------------------------------

Description

Une classe pour le stockage en memoire des differentes probabilites invariantes au niveau du model point [EpEuroInd](#).

Slots

rachat_tot est un data.frame contenant les taux de rachat totaux pour chaque model point.

rachat_part est un data.frame contenant les taux de rachat partiels pour chaque model point.

qx_mort est un data.frame contenant les probabilite des deces pour chaque model point.

Author(s)

Prim'Act

TabProbaRetEuroRest	<i>La classe TabProbaRetEuroRest.</i>
---------------------	---------------------------------------

Description

Une classe pour le stockage en memoire des differentes probabilites invariantes au niveau du model point [RetraiteEuroRest](#).

Slots

ax est un data.frame contenant les valeurs des coefficients actuariels pour chaque model point.

sortie_retraite est un data.frame contenant les la probabilite de sortie d'un contrat retraite pour chaque model point.

survie_un_an est un data.frame contenant les valeurs des coefficients actuariels pour chaque model point.

Author(s)

Prim'Act

TabRetEuroRest	<i>La classe TabRetEuroRest.</i>
----------------	----------------------------------

Description

Une classe pour le stockage en memoire de variable de calcul au niveau du model point de retraite euro en phase de restitution [TabRetEuroRest](#).

Slots

tab un objet list au format fige contenant l'ensemble des variables stockees.

Author(s)

Prim'Act

TauxPB	<i>La classe TauxPB.</i>
--------	--------------------------

Description

Une classe pour le stockage des parametres de taux de participation contractuelle par produit.

Slots

mp un data frame contenant les parametres des taux de participation contractuelle par produit.

Author(s)

Prim'Act

tauxpb_load	<i>Methode permettant de charger les valeurs des hypotheses et des donnees de TauXPB</i>
-------------	--

Description

tauxpb_load est une methode permettant de charger les parametres associees a un objet de classe [TauxPB](#).

Usage

```
tauxpb_load(file_tauxpb_address)
```

Arguments

file_tauxpb_address

est un character contenant l'adresse exacte du fichier d'input utilisateur permettant de renseigner un objet [TauxPB](#).

Value

L'objet de la classe [TauxPB](#) construit a partir des inputs renseignés par l'utilisateur.

Author(s)

Prim'Act

See Also

La classe [Initialisation](#) et sa methode [set_architecture](#) pour renseigner l'input.

taux_period	<i>Fonction permettant de calculer le taux d'interet sur une periode.</i>
-------------	---

Description

taux_period permet de calculer le taux d'interet sur une periode

Usage

```
taux_period(x, period)
```

Arguments

x	un vecteur de la classe <code>numeric</code> representant le taux annuel.
period	un character representant la peridocite souhaitee. Cette variable prend pour valeur : an : annuel ; se : semestriel ; trim : trimestriel ; mens : mensuel.

Value

Le taux periodique.

Author(s)

Prim'Act

Treso	<i>La classe Tresor.</i>
-------	--------------------------

Description

Une classe pour les actifs de type Tresorerie.

Slots

ptf_treso est un dataframe, chaque ligne represente un actif de tresorerie du portefeuille de monetaire.

Author(s)

Prim'Act

See Also

Les methodes de calcul des valeurs [calc_vm_treso](#), de calcul des revenus de la tresorerie [revenu_treso](#), de calcul de la revalorisation de la tresorerie [revalo_treso](#), de mise a jour de la tresorerie [update_treso](#).

update_cc_oblig	<i>Mise a jour des coupons courus d'un portefeuille obligataire.</i>
-----------------	--

Description

update_cc_oblig est une methode permettant de mettre a jour les coupons courus des composantes d'un portefeuille obligataire.

Usage

```
update_cc_oblig(x, coupon)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
coupon	un vecteur de numeric a assigner a l'objet Oblig .

Value

L'objet x dont les coupons courus ont ete mis a jour

Author(s)

Prim'Act

update_dur_det_action *Mise a jour des durees de detention d'un portefeuille action.*

Description

update_dur_det_action est une methode permettant de mettre a jour la duree de detention des composantes d'un portefeuille Action.

Usage

```
update_dur_det_action(x)
```

Arguments

x objet de la classe [Action](#) (decrivant le portefeuille action en detention).

Value

L'objet x mis a jour du vieillissement de la duree de detention.

Author(s)

Prim'Act

update_dur_det_immo *Mise a jour des durees de detention d'un portefeuille immobilier.*

Description

update_dur_det_immo est une methode permettant de mettre a jour la duree de detention des composantes d'un portefeuille immobilier.

Usage

```
update_dur_det_immo(x)
```

Arguments

x objet de la classe [Immo](#) (decrivant le portefeuille immo en detention).

Value

L'objet x mis a jour du vieillissement de la duree de detention.

Author(s)

Prim'Act

update_dur_oblig	<i>Mise a jour des durations d'un portefeuille obligataire.</i>
------------------	---

Description

update_dur_oblig est une methode permettant de mettre a jour la duration des composantes d'un portefeuille obligataire.

Usage

```
update_dur_oblig(x, duration)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
duration	un vecteur de numeric a assigner a l'objet Oblig .

Value

L'objet x dont les durations ont ete mises a jour.

Author(s)

Prim'Act

update_mat_res	<i>Mise a jour de la maturite residuelle et de la duree de detention de chaque composante d'un portefeuille obligataire.</i>
----------------	--

Description

update_mat_res est une methode permettant de mettre a jour la maturite residuelle et la duree de detention de chaque composante d'un portefeuille obligataire.

Usage

```
update_mat_res(x)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire).
---	---

Value

L'objet x dont

mat_res : est diminuee d'une unite (une unite correspond a un an)

dur_det : est augmentee d'une unite (une unite correspond a un an)

Author(s)

Prim'Act

update_PortFin	<i>Evalue et met a jour les objets constituant un PortFin.</i>
----------------	--

Description

update_PortFin est une methode permettant de calculer et mettre a jour un portefeuille financier pour le faire vieillir d'une annee.

Usage

```
update_PortFin(an, x, new_mp_ESG, flux_milieu, flux_fin)
```

Arguments

an	integer correspond a l'annee de projection du portefeuille financier.
x	objet de la classe PortFin , correspondant au portefeuille financier de l'assureur avant l'etape de vieillissement.
new_mp_ESG	est un objet de la classe ModelPointESG , decrivant les conditions economiques permettant d'effectuer le vieillissement du portefeuille financier.
flux_milieu	est une valeur numeric correspondant a la somme des flux percus en milieu d'annee (coupons des obligations, loyers immobiliers, dividendes des actions, revenus de la tresorerie).
flux_fin	est une valeur numeric correspondant a la somme des flux percus en fin d'annee (tombee d'echeance d'obligation).

Value

ptf un vecteur contenant les flux de sortie en echeance de l'annee

revenu_fin les revenus realises au cours de la periode (coupons, tombees d'echeance, dividendes et loyers).

revenu_fin_det les revenus detaille pour chaque actif realises au cours de la periode.

var_vnc_oblig la variation de valeur nette comptable obligataire.

Author(s)

Prim'Act

See Also

Le vieillissement des classes d'actifs : [vieillissement_action_PortFin](#), [vieillissement_immo_PortFin](#), [vieillissement_oblig_PortFin](#), [vieillissement_treso_PortFin](#). Le calcul des rendements : [calc_rdt](#). La methode de mise a jour specifique au portefeuille de reinvestissement [update_PortFin_reference](#).

update_PortFin_reference

Evalue et met a jour les objets constituant un PortFin_reference.

Description

update_PortFin_reference est une methode permettant de calculer et mettre a jour un portefeuille financier de reinvestissement pour le faire vieillir d'une annee.

Usage

update_PortFin_reference(an, x, mp_ESG)

Arguments

an	numeric correspond a l'annee de projection du portefeuille financier de reinvestissement.
x	objet de la classe PortFin , correspondant au portefeuille financier de reinvestissement avant l'etape de vieillissement.
mp_ESG	est un objet de la classe ModelPointESG , decrivant les conditions economiques permettant d'effectuer le vieillissement du portefeuille financier de reinvestissement.

Value

L'objet x renvoye correspond au portefeuille financier de reinvestissement veilli d'une annee.

Author(s)

Prim'Act

See Also

La fonction de mise a jour specifique au portefeuille [update_PortFin](#).

update_reserves

Evalue et met a jour la valeur des autres reserves.

Description

update_reserves est une methode permettant de calculer la valeur de la nouvelle PGG et de la nouvelle PSAP et les met a jour.

Usage

update_reserves(x, prest_ep, prest_autres, pm_ep, pm_autres)

Arguments

x	objet de la classe AutresReserves .
prest_ep	est une valeur numeric correspondant a la somme des prestations nettes de chargement et de charges sociales sur epargne.
prest_autres	est une valeur numeric correspondant a la somme des prestations nettes de chargements et de charges sociales sur autres passifs.
pm_ep	est une valeur numeric correspondant a la somme des PM nettes de chargements et de charges sociales sur epargne.
pm_autres	est une valeur numeric correspondant a la somme des PM nettes de chargement et de charges sociales sur autres passifs.

Value

x l'objet [AutresReserves](#) mis a jour.
var_psap une valeur numeric correspondant a la variation de PSAP.
var_gg une valeur numeric correspondant a la variation de PGG.

Note

Il s'agit d'une methode simplifiee.

Author(s)

Prim'Act

update_sd_oblig	<i>Mise a jour des surcotes decotes d'un portefeuille obligataire.</i>
-----------------	--

Description

update_sd_oblig est une methode permettant de mettre a jour la surcotes decotes des composantes d'un portefeuille obligataire.

Usage

```
update_sd_oblig(x, sd)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
sd	un vecteur de numeric a assigner a l'objet Oblig .

Value

L'objet x dont les surcotes decotes ont ete mises a jour.

Author(s)

Prim'Act

`update_table_output_pb`*Mise a jour de la table des PB. Ajoute une colonne "diff_pb".*

Description

Mise a jour de la table des PB. Ajoute une colonne "diff_pb".

Usage

```
update_table_output_pb(table)
```

Arguments

`table` est un objet de type `data.frame`.

Value

La table mise a jour.

Author(s)

Prim'Act

`update_table_output_produit`*Mise a jour de la table OUTPUT_PRODUIT. Ajoute plusieurs colonnes : chgt, delta_pm, frais, credit, debit, resultat.*

Description

Mise a jour de la table OUTPUT_PRODUIT. Ajoute plusieurs colonnes : `chgt`, `delta_pm`, `frais`, `credit`, `debit`, `resultat`.

Usage

```
update_table_output_produit(table)
```

Arguments

`table` est un objet de type `data.frame`.

Value

La table mise a jour.

Author(s)

Prim'Act

update_tab_proba	<i>Mise a jour des tables de probas pour les contrats d'epargne et de retraite. Insertion d'une nouvelle colonne avec les probabilites.</i>
------------------	---

Description

update_tab_proba est une methode permettant de calculer les differents taux de sortie sur une periode.

Usage

```
update_tab_proba(x, an, y)
```

Arguments

x	un objet de la classe TabProbaEpEuroInd ou de la classe TabProbaRetEuroRest a mettre a jour.
an	une valeur de type integer correspondant a l'annee des probabilites.
y	une liste contenant les probabilites a inserer dans la table : proba_flux : probabilites de mouvement de flux pour les contrats d'epargne en euros et de retraite. Le format de cette liste correspond a la sortie de la methode calc_proba_flux coef_rente : coefficients actuariels, uniquement pour le cas des retraites. le taux de rendement de l'indice action de reference le taux de rendement de l'indice immobilier de reference le taux de rendement de l'indice tresorerie de reference

Author(s)

Prim'Act

See Also

Le calcul des probabilites : [calc_proba_flux](#).

update_treso	<i>Permet d'integrer un flux (entrant ou sortant) au compte de tresorerie d'un portefeuille financier.</i>
--------------	--

Description

update_treso est une methode permettant d'integrer un flux au compte de tresorerie.

Usage

```
update_treso(x, flux)
```

Arguments

x	objet de la classe Treso , correspondant a l'actif Tresorerie d'un assureur anterieur a integration d'un flux.
flux	est un numeric correspondant a un flux. S'il est positif, le flux est entrant. S'il est negatif, le flux est sortant.

Value

L'objet x mis a jour du flux precise en input.

Author(s)

Prim'Act

update_vm_action	<i>Mise a jour de valeurs de marche d'un portefeuille action.</i>
------------------	---

Description

update_vm_action est une methode permettant de mettre a jour la valeur de marche des composantes d'un portefeuille Action.

Usage

```
update_vm_action(x, vm)
```

Arguments

x	objet de la classe Action (decrivant le portefeuille action en detention).
vm	un vecteur de numeric ayant la meme longueur que le portefeuille action a de lignes et correspondant aux nouvelles valeurs de marche du portefeuille action.

Value

L'objet x mis a jour.

Author(s)

Prim'Act

update_vm_immo	<i>Mise a jour des valeurs de marche d'un portefeuille immobilier.</i>
----------------	--

Description

update_vm_immo est une methode permettant de mettre a jour les valeurs de marche des composantes d'un portefeuille immobilier.

Usage

```
update_vm_immo(x, vm)
```

Arguments

x	objet de la classe Immo (decrivant le portefeuille immobilier en detention).
vm	un vecteur de numeric ayant la meme longueur que le portefeuille immobilier a de lignes et correspondant aux nouvelles valeurs de marche du portefeuille immobilier.

Value

L'objet x mis a jour.

Author(s)

Prim'Act

update_vm_oblig	<i>Mise a jour des valeurs de marche d'un portefeuille obligataire.</i>
-----------------	---

Description

update_vm_oblig est une methode permettant de mettre a jour les valeurs de marche des composantes d'un portefeuille obligataire.

Usage

```
update_vm_oblig(x, vm)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
vm	un vecteur de numeric a assigner a l'objet Oblig .

Value

L'objet x dont les valeurs de marche ont ete mises a jour.

Author(s)

Prim'Act

update_vnc_oblig	<i>Mise a jour des valeurs nettes comptables d'un portefeuille obligataire.</i>
------------------	---

Description

update_vnc_oblig est une methode permettant de mettre a jour les valeurs nettes comptables des composantes d'un portefeuille obligataire.

Usage

```
update_vnc_oblig(x, vnc)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
vnc	un vecteur de numeric a assigner a l'objet Oblig .

Value

L'objet x dont les valeurs nettes comptables ont ete mis a jour

Author(s)

Prim'Act

update_zsp_oblig	<i>Mise a jour des z-spreads d'un portefeuille obligataire.</i>
------------------	---

Description

update_zsp_oblig est une methode permettant de mettre a jour les z-spreads des composantes d'un portefeuille obligataire.

Usage

```
update_zsp_oblig(x, zspread)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire en detention).
zspread	un vecteur de numeric a assigner a l'objet Obligation .

Value

L'objet x dont les zspreads ont ete mis a jour

Author(s)

Prim'Act

`vieillissement_action_PortFin`*Effectue le vieillissement du portefeuille action d'un portefeuille financier.*

Description

`vieillissement_action_PortFin` est une methode permettant de projeter la composante action d'un portefeuille financier.

Usage

```
vieillissement_action_PortFin(x, table_rdt)
```

Arguments

<code>x</code>	objet de la classe PortFin , correspondant au portefeuille financier de l'assureur avant l'etape de vieillissement.
<code>table_rdt</code>	est une liste, construite par la fonction calc_rdt . Cette table contient les tables d'evolution des cours et rendements sur l'annee consideree de chacune des classes d'actif. Les tables sont constuities a partir des extractions du Generateur de Scenario Economique de Prim'Act.

Value

`portFin` le portefeuille financier dont l'attribut `ptf_action` a ete vieilli d'une annee.
`dividende` le montant de dividende percu en milieu d'annee suite au vieillissement du portefeuille action.

Author(s)

Prim'Act

See Also

La fonction de calcul des rendements des actifs [calc_rdt](#).

`vieillissement_immo_PortFin`*Effectue le vieillissement du portefeuille immo d'un portefeuille financier.*

Description

`vieillissement_immo_PortFin` est une methode permettant de projeter la composante immobilier d'un portefeuille financier.

Usage

```
vieillissement_immo_PortFin(x, table_rdt)
```

Arguments

x	objet de la classe PortFin , correspondant au portefeuille financier de l'assureur avant l'etape de vieillissement.
table_rdt	est une liste, construite par la fonction calc_rdt . Cette table contient les tables d'evolution des cours et rendements sur l'annee consideree de chacune des classes d'actif. Les tables sont constuities a partir des extractions du Generateur de Scenario Economique de Prim'Act.

Value

codeportFin le portefeuille financier dont l'attribut ptf_immo a ete vieilli d'une annee.
 loyer le montant de loyer percu en milieu d'annee suite au vieillissement du portefeuille immobilier.

Author(s)

Prim'Act

See Also

La fonction de calcul des rendements des actifs [calc_rdt](#).

vieillessement_oblig_PortFin

Effectue le vieillissement du portefeuille obligataire d'un portefeuille financier.

Description

vieillessement_oblig_PortFin est une methode permettant de projeter la composante obligataire d'un portefeuille financier.

Usage

```
vieillessement_oblig_PortFin(x, new_mp_ESG)
```

Arguments

x	objet de la classe PortFin , correspondant au portefeuille financier de l'assureur avant l'etape de vieillissement.
new_mp_ESG	est un objet de type ModelPointESG , correspondant aux conditions economiques de l'annee du vieillissement.

Value

portFin le portefeuille financier dont l'attribut ptf_oblig a ete vieilli d'une annee.
 coupon le montant des coupons percu en milieu d'annee suite au vieillissement du portefeuille obligataire.
 echeance le montant des echeance percu en fin d'annee suite au vieillissement du portefeuille obligataire.
 var_vnc_oblig la variation de VNC constatee sur les titres non arrives en echeance.

Author(s)

Prim'Act

See AlsoLa fonction de calcul des rendements des actifs [calc_rdt](#).

vieillissement_ppb*Vieillissement d'un an de la PPB.*

Description

vieillissement_ppb est une methode permettant de vieillir d'un an la PPB. Cette methode permet de reinitialiser les montants de dotation et de reprise cumules sur l'annee, de re-initialiser le montant de PPB de debut de periode, et de mettre a jour le vecteur historique de la PPB.

Usage

```
vieillissement_ppb(x)
```

Arguments

x objet de la classe [Ppb](#).

Value

L'objet x vieilli d'une annee.

Author(s)

Prim'Act

vieillissement_treso_PortFin*Effectue le vieillissement/la projection du portefeuille tresorerie d'un portefeuille financier.*

Description

vieillissement_treso_PortFin est une methode permettant de projeter la composante obligataire d'un portefeuille financier.

Usage

```
vieillissement_treso_PortFin(x, flux_milieu, flux_fin, table_rdt)
```

Arguments

x	objet de la classe PortFin, correspondant au portefeuille financier de l'assureur avant l'étape de vieillissement.
flux_milieu	est un numeric correspondant aux revenus perçus en milieu d'année (coupons obligataires, loyers, dividendes).
flux_fin	est un numeric correspondant aux revenus perçus en fin d'année (tombees d'échéances et revenus de trésorerie).
table_rdt	est une liste, construite par la fonction calc_rdt . Cette table contient les tables d'évolution des cours et rendements sur l'année considérée de chacune des classes d'actif. Les tables sont constituées à partir des extractions du Générateur de Scenario Economique de Prim'Act.

Value

L'objet renvoyé de la classe PortFin correspond au portefeuille financier initial dont l'attribut ptf_treso a été vieilli d'une année.

Author(s)

Prim'Act

See Also

La fonction de calcul des rendements des actifs [calc_rdt](#).

vieillissement_ap_pb	<i>Vieillessement du portefeuille sur l'année apres attribution de participation aux benefices.</i>
----------------------	---

Description

vieillissement_ap_pb est une methode permettant de calculer les PM et les flux sur une année apres PB. Cette methode vieilli le portefeuille de passifs apres attribution de PB.

Usage

```
vieillissement_ap_pb(x, rev_netten_alloue, rev_brute_alloue_gar, tx_soc)
```

Arguments

x	un objet de la classe PortPassif contenant l'ensemble des produits de passifs.
rev_netten_alloue	un vecteur numeric contenant par produit le supplement de revalorisation par rapport au taux minimum.
rev_brute_alloue_gar	un vecteur numeric contenant par produit le supplement de revalorisation au titre de la part garantie initialement (PPB initiale).
tx_soc	une valeur numeric correspondant au taux de charges sociales.

Value

x l'objet x mis a jour.

nom_produit un vecteur de character contenant les noms des produits.

flux_agg une matrice contenant les flux agreges par produits.

stock_agg une matrice contenant les stocks agreges par produits.

Author(s)

Prim'Act

See Also

L'attribution de la revalorisation par model point : [calc_revalo_pm](#) Le vieillissement des model points : [vieilli_mp](#).

vieilli_mp	<i>Veillissement d'un an des contrats epargne en euros.</i>
------------	---

Description

vieilli_mp est une methode permettant de vieillir les model points epargne en euros d'une periode.

Usage

```
vieilli_mp(x, pm_fin, pm_gar, tx_revalo)
```

Arguments

x	un objet de la classe EpEuroInd ou de la classe RetraiteEuroRest contenant les model points epargne euros.
pm_fin	un vecteur de type numeric contenant par model point les montants de PM revalorises apres participation aux benefices.
pm_gar	un vecteur de type numeric contenant par model point les montants de PM garanties revalorises apres participation aux benefices au titre de la PPB initiale.
tx_revalo	un vecteur de type numeric contenant par model point les taux de revalorisation nets appliques.

Value

l'objet x vieilli d'une periode.

Author(s)

Prim'Act

See Also

Calcul de la revalorisation des PM [calc_revalo_pm](#).

viellissement_av_pb	<i>Vieillissement du portefeuille sur l'annee avant attribution de participation aux benefices.</i>
---------------------	---

Description

viellissement_av_pb est une methode permettant de vieillir l'objet [PortPassif](#) sur l'annee avant attribution de participation aux benefices.

Usage

```
viellissement_av_pb(an, x, coef_inf, list_rd, tx_soc)
```

Arguments

an	une valeur numeric correspondant a l'annee de projection.
x	un objet de la classe PortPassif contenant l'ensemble des produits de passifs.
coef_inf	une valeur numeric correspondant au coefficient d'inflation considere pour le traitement des frais.
list_rd	une liste contenant les rendements de reference. Le format de cette liste est : le taux de rendement obligataire le taux de rendement de l'indice action de reference le taux de rendement de l'indice immobilier de reference le taux de rendement de l'indice tresorerie de reference
tx_soc	une valeur numeric correspondant au taux de charges sociales.

Value

Une liste comprenant :

ptf : Le portefeuille x mis a jour.

result_av_pb : Une liste dont le premier element designe les noms des produits, puis deux matrices de resultats agreges : une pour les flux et une pour le stock. Le format de cette sortie decoule de celui de la methode [proj_annee_av_pb](#).

result_autres_passifs : un vecteur contenant les resultats des passifs non modelises.

var_psap : la variation de PSAP sur l'annee.

var_pgg : la variation de PGG sur l'annee.

flux_milieu : les flux de milieu d'annee entrant en tresorerie en milieu de periode.

flux_fin : les flux de fin d'annee entrant en tresorerie en fin de periode.

Author(s)

Prim'Act

See Also

La projection des passifs sur un an avant PB : [proj_annee_av_pb](#). La projection des autres passifs : [proj_annee_autres_passifs](#). La mise a jour des autres reserves : [update_reserves](#).

write_be_results	<i>Enregistre les resultats d'une evaluation best estimate</i>
------------------	--

Description

write_be_results est une methode permettant d'enregistrer en .cvs les resultats d'une evaluation best estimate.

Usage

```
write_be_results(nom_run, path, x)
```

Arguments

nom_run	est un objet de type character utilise pour nommer le fichier de resultats.
path	est un objet de type character utilise pour indiquer le chemin d'enregistrement des resultats.
x	est un objet de type Be .

Author(s)

Prim'Act

yield_to_maturity	<i>Calcule les yield to maturity de chaque composante d'un portefeuille obligataire.</i>
-------------------	--

Description

yield_to_maturity est une methode permettant de calculer les yield to maturity de chaque composante d'un portefeuille obligataire.

Usage

```
yield_to_maturity(x)
```

Arguments

x	objet de la classe Oblig (decrivant le portefeuille obligataire).
---	---

Value

Un vecteur dont chaque element correspond au yield to maturity des obligations du portefeuille obligataire. Ce vecteur a autant d'elements que le portefeuille obligataire a de lignes.

Author(s)

Prim'Act

Index

* classes

Action, 8
AutresPassifs, 9
AutresReserves, 10
Be, 12
Canton, 48
DataBase, 56
EpEuroInd, 73
ESG, 73
FraisFin, 80
FraisPassif, 80
HypCanton, 87
HypTech, 88
Immo, 89
Initialisation, 89
ModelPointESG, 102
Oblig, 103
ParamAlmEngine, 103
ParamBe, 104
ParamChocMket, 104
ParamChocSousc, 104
ParamComport, 105
ParamRachDyn, 105
ParamRevaloEngine, 106
ParamTableMort, 106
ParamTableRach, 107
PortFin, 110
PortPassif, 111
Ppb, 112
PRE, 114
RC, 118
RetraiteEuroRest, 120
RevaloEngine, 120
TabEpEuroInd, 128
TabProbaEpEuroInd, 129
TabProbaRetEuroRest, 129
TabRetEuroRest, 130
TauxPB, 130
Treso, 132

* datasets

Be_DataSet, 13
Canton_DataSet, 49

Action, 8, 14, 25, 35, 45, 55, 57, 94, 95, 98,

110, 121, 125, 127, 133, 140

AlmEngine, 9
autres_passif_load, 10
autres_reserves_load, 11
AutresPassifs, 9, 10, 61–65, 111
AutresReserves, 10, 11, 111, 137

base_prod_fin, 12, 40
Be, 12, 13, 90, 92, 95, 123, 124, 149
be (Be_DataSet), 13
Be_DataSet, 13
buy_action, 8, 13
buy_immo, 14, 89
buy_oblig, 15, 103

calc_ax, 15
calc_axy, 16
calc_coupon, 17
calc_dotation_ppb, 17, 112
calc_fin_proj, 18, 48, 116
calc_flux_annee, 19
calc_frais, 19, 80
calc_frais_fin, 20, 116
calc_marge_fin, 21, 40
calc_nominal, 22
calc_pb_attrib, 22
calc_pm, 23, 41, 73, 120
calc_pmv1, 25, 110
calc_pmv1_action, 25, 68
calc_pmv1_immo, 26, 68
calc_pmv1_oblig, 26, 68
calc_PRE, 27, 69, 70, 114, 116
calc_prest, 23, 24, 27, 73, 120
calc_primes, 23, 24, 30, 73
calc_proba_decès, 30
calc_proba_dyn, 31
calc_proba_flux, 29, 32, 139
calc_proba_survie, 16, 33
calc_qx, 33, 85, 107
calc_rach, 34, 86, 107
calc_rach_dyn, 34, 87, 106
calc_RC, 35, 70, 71, 118
calc_rdt, 35, 110, 135, 143–146
calc_rdt_marche_ref, 36

- calc_reprise_ppb, 36, 112
- calc_result_technique, 38, 40, 116
- calc_result_technique_ap_pb, 38, 48, 116
- calc_resultat_fin, 37, 116
- calc_revalo, 39, 116
- calc_revalo_pm, 40, 73, 120, 147
- calc_sur_dec, 41
- calc_tra, 40, 42, 116
- calc_tx_cible, 23, 24, 42, 73, 120
- calc_tx_cible_ref_marche, 43, 85, 105
- calc_tx_min, 23, 24, 28, 29, 44, 73, 120
- calc_vm_action, 44
- calc_vm_immo, 45
- calc_vm_oblig, 46
- calc_vm_treso, 46, 132
- calc_vnc, 47
- calc_z_spread, 47
- Canton, 12, 13, 18, 39, 48, 49, 53, 59–68, 91, 93, 95, 115, 124
- canton (Canton_DataSet), 49
- Canton_DataSet, 49
- changement_choc, 49
- changement_ESG, 50, 74, 103
- changement_PortFin, 51, 94, 97, 99, 110
- changement_PortFin_reference, 51, 108, 110
- chgt_period, 52
- ChocSolvabilite2, 49, 50, 53, 59–66
- convert_table, 54
- create_ptf_bought_action, 54
- create_ptf_bought_immo, 55
- create_ptf_bought_oblig, 56
- DataBase, 12, 56, 93, 94
- do_calc_nb_sold_action, 57
- do_calc_nb_sold_immo, 57
- do_calc_nb_sold_oblig, 58
- do_choc_action_type1, 54, 59
- do_choc_action_type2, 54, 59
- do_choc_currency, 54, 60
- do_choc_frais, 54, 61
- do_choc_immo, 54, 61
- do_choc_longevite, 54, 62
- do_choc_mortalite, 54, 63
- do_choc_rachat_down, 54, 64
- do_choc_rachat_mass, 54, 64
- do_choc_rachat_up, 54, 65
- do_choc_spread, 54, 66
- do_choc_spread_unitaire, 66, 67
- do_choc_taux, 54, 67
- do_update_pmv1, 68, 110, 116
- do_update_PRE_val_courante, 69, 114, 116
- do_update_PRE_val_debut, 69, 114, 116
- do_update_RC_val_courante, 70, 118
- do_update_RC_val_debut, 70, 116, 118
- do_update_vm_vnc_precedent, 71, 110, 116
- duration_sensi, 72
- echeancier, 72
- EpEuroInd, 23, 28–32, 40, 42, 44, 73, 95, 111, 118, 128, 129, 147
- ESG, 12, 13, 50, 73, 74, 82, 124
- extract_ESG, 74, 74, 103, 124
- finance_cible_marge, 40, 75
- finance_cible_pmv1, 40, 76
- finance_cible_ppb, 40, 77
- finance_contrainte_legale, 40, 78
- finance_tmg, 40, 79
- frais_fin_load, 81
- frais_passif_load, 81
- FraisFin, 80, 110
- FraisPassif, 19, 80, 81, 111, 118
- get_choc_inflation_frais, 54, 82
- get_choc_rach, 83, 88
- get_choc_table, 83, 88
- get_coef_rente, 84
- get_comport, 43, 84, 88
- get_qx_mort, 32, 85, 88
- get_qx_rach, 32, 86, 88
- get_rach_dyn, 31, 32, 86, 88
- hyp_canton_load, 88
- HypCanton, 48, 87, 88, 89
- HypTech, 31, 32, 42, 83–87, 88, 96, 111
- Immo, 14, 26, 35, 45, 55, 58, 89, 96, 97, 110, 121, 125, 133, 141
- init_create_folder, 91
- init_debut_pgg_psap, 10, 92, 116
- init_scenario, 13, 92, 123
- init_SimBEL, 90, 93
- init_tables, 93
- Initialisation, 11, 49, 50, 81, 82, 89, 89, 90–93, 95, 96, 98, 113, 127, 131
- initialisation_load, 90
- insert_tables, 94
- load_action, 94
- load_be, 95
- load_epeuroind, 95
- load_ht, 96
- load_immo, 96
- load_oblig, 97
- load_pp, 96, 98, 99
- load_reteurorest, 98

- load_treso, 99
- merge_actifs, 99
- merge_be, 100
- merge_flux_fin, 100
- merge_pb, 101
- merge_table_be, 101
- merge_table_produit, 102
- ModelPointESG, 35, 36, 46, 48, 51, 74, 75, 102, 135, 136, 144
- Oblig, 15, 17, 19, 22, 26, 41, 46–48, 56, 58, 59, 67, 72, 97, 103, 110, 126, 132, 134, 137, 141, 142, 149
- param_alm_engine_load, 108
- param_revalo_load, 108
- ParamAlmEngine, 48, 103, 108
- ParamBe, 12, 13, 104
- ParamChocMket, 50, 53, 104
- ParamChocSousc, 50, 53, 104
- ParamComport, 36, 43, 88, 105
- ParamRachDyn, 34, 88, 105
- ParamRevaloEngine, 48, 78, 106, 108, 120
- ParamTableMort, 30, 33, 88, 106
- ParamTableRach, 34, 88, 107
- pb_contr, 40, 109
- PortFin, 25, 35, 48, 51, 52, 67, 68, 71, 94, 97, 99, 103, 108, 110, 115, 119, 120, 135, 136, 143, 144
- PortPassif, 48, 96, 98, 99, 111, 117, 146, 148
- Ppb, 12, 17, 22, 38, 39, 48, 77–79, 112, 112, 113, 145
- ppb_8ans, 112
- ppb_load, 113
- PRE, 27, 38, 69, 110, 114, 114
- pre_load, 114
- print_alloc, 71, 110, 115, 116
- proj_an, 48, 56, 101, 102, 115, 124
- proj_annee_autres_passifs, 9, 116, 148
- proj_annee_av_pb, 111, 117, 148
- RC, 35, 70, 71, 110, 118, 119
- rc_load, 118
- reallocate, 9, 116, 119
- RetraiteEuroRest, 23, 24, 28–30, 32, 40, 42, 84, 98, 111, 120, 129, 147
- revalo_action, 121
- revalo_immo, 121
- revalo_treso, 122, 132
- RevaloEngine, 120
- revenu_treso, 122, 132
- run_be, 13, 123
- run_be_simu, 13, 123, 124
- sell_action, 8, 125
- sell_immo, 89, 125
- sell_oblig, 103, 126
- sell_pvl_action, 8, 116, 127
- set_architecture, 11, 49–51, 81, 82, 89–91, 96, 98, 113, 127, 131
- SimBEL, 128
- SimBEL-package (SimBEL), 128
- TabEpEuroInd, 73, 128
- TabProbaEpEuroInd, 73, 129, 139
- TabProbaRetEuroRest, 120, 129, 139
- TabRetEuroRest, 120, 130, 130
- taux_period, 131
- TauxPB, 111, 130, 130, 131
- tauxpb_load, 130
- Treso, 35, 46, 99, 110, 122, 132, 140
- update_cc_oblig, 132
- update_dur_det_action, 133
- update_dur_det_immo, 133
- update_dur_oblig, 134
- update_mat_res, 134
- update_PortFin, 110, 116, 135, 136
- update_PortFin_reference, 110, 116, 135, 136
- update_reserves, 10, 136, 148
- update_sd_oblig, 137
- update_tab_proba, 139
- update_table_output_pb, 138
- update_table_output_produit, 138
- update_treso, 132, 139
- update_vm_action, 140
- update_vm_immo, 141
- update_vm_oblig, 141
- update_vnc_oblig, 142
- update_zsp_oblig, 142
- vieilli_mp, 73, 120, 147, 147
- vieillissement_action_PortFin, 135, 143
- vieillissement_immo_PortFin, 135, 143
- vieillissement_oblig_PortFin, 135, 144
- vieillissement_ppb, 116, 145
- vieillissement_treso_PortFin, 135, 145
- vieillissement_ap_pb, 38, 111, 116, 146
- vieillissement_av_pb, 38–40, 111, 116, 148
- write_be_results, 13, 123, 149
- yield_to_maturity, 149