

# Package ‘SimBEL’

March 23, 2017

**Type** Package

**Title** Un package de calcul du best estimate epargne sous Solvabilite II.

**Version** 0.3.0

**Author** Prim'Act

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

**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 francais en euros. Plusieurs chocs de la formule standard peuvent etre effectues.

**License** GPL-2

**LazyData** TRUE

**RoxygenNote** 5.0.1

**Collate** 'Action\_class.R' 'Action\_buy.R' 'Action\_calc\_pmv1.R'  
'Action\_calc\_vm.R' 'Action\_internal.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'  
'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' 'Be-run\_be.R'  
'Be-run\_be\_simu.R' 'Be\_class.R' 'Be-write\_results.R'  
'Be\_internal.R' 'Canton\_class.R' 'Canton\_calc\_fin\_proj.R'  
'Canton\_calc\_result\_technique\_ap\_pb.R' 'Canton\_internal.R'  
'Canton\_proj\_an.R' 'ChocSolvabilite2\_class.R'  
'ChocSolvabilite2\_do\_choc\_action\_type1.R'  
'ChocSolvabilite2\_do\_choc\_action\_type2.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\_up.R'  
 'ChocSolvabilite2\_do\_choc\_spread.R'  
 'ChocSolvabilite2\_do\_choc\_spread\_unitaire.R'  
 'ChocSolvabilite2\_do\_choc\_taux.R' 'ChocSolvabilite2\_internal.R'  
 'ChocSolvabilite2\_load.R' 'ESG\_class.R'  
 'ESG-get\_choc\_inflation\_frais.R' 'ESG\_chargement.R'  
 'ESG\_extract\_mp.R' 'ESG\_internal.R' 'EpEuroInd-class.R'  
 'EpEuroInd-calc\_pm.R' 'EpEuroInd-calc\_prest.R'  
 'EpEuroInd-calc\_primes.R' 'EpEuroInd-calc\_revalo\_pm.R'  
 'HypTech-class.R' 'EpEuroInd-calc\_tx\_cible.R'  
 'EpEuroInd-calc\_tx\_min.R' 'EpEuroInd-calc\_tx\_sortie.R'  
 'EpEuroInd-internal.R' 'EpEuroInd-vieilli\_mp.R'  
 'FraisFin\_class.R' 'FraisFin\_calc.R' 'FraisFin\_internal.R'  
 'FraisFin\_load.R' 'FraisPassif-class.R'  
 'FraisPassif-calc\_frais.R' 'FraisPassif-internal.R'  
 'FraisPassif-load.R' 'HypCanton\_class.R' 'HypCanton\_internal.R'  
 'HypCanton\_load.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\_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\_set\_architecture.R' 'ModelPointESG\_class.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\_echeancier.R'  
 'Oblig\_flux\_annee.R' 'Oblig\_internal.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\_class.R' 'PRE\_calc.R'  
 'PRE\_do\_update\_val\_courante.R' 'PRE\_do\_update\_val\_debut.R'  
 'PRE\_internal.R' 'PRE\_load.R' 'ParamAlmEngine\_class.R'  
 'ParamAlmEngine\_internal.R' 'ParamAlmEngine\_load.R'  
 'ParamBe\_class.R' 'ParamBe\_internal.R' 'ParamChocMket\_class.R'  
 'ParamChocMket\_internal.R' 'ParamChocSousc-class.R'  
 'ParamChocSousc-internal.R' 'ParamComport-class.R'  
 'ParamComport-calc\_tx\_cible.R' 'ParamComport-internal.R'  
 'ParamRachDyn-class.R' 'ParamRachDyn-calc\_rach\_dyn.R'  
 'ParamRachDyn-internal.R' 'ParamRevaloEngine\_class.R'  
 'ParamRevaloEngine\_internal.R' 'ParamRevaloEngine\_load.R'  
 'ParamTableMort-class.R' 'ParamTableMort-calc\_qx.R'  
 'ParamTableMort-internal.R' 'ParamTableRach-class.R'  
 'ParamTableRach-calc\_rach.R' 'ParamTableRach-internal.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-class.R'  
 'PortPassif-internal.R' 'PortPassif-load.R'  
 'PortPassif-proj\_annee\_av\_pb.R'  
 'PortPassif-vieillissement\_ap\_pb.R'  
 'PortPassif-vieillissement\_av\_pb.R' 'Ppb\_class.R'  
 'Ppb\_dotation\_reprise.R' 'Ppb\_init\_debut.R' 'Ppb\_internal.R'  
 'Ppb\_load.R' 'RC\_class.R' 'RC\_calc.R'  
 'RC\_do\_update\_val\_courante.R' 'RC\_do\_update\_val\_debut.R'  
 'RC\_internal.R' 'RC\_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' 'TabEpEuroInd-class.R'  
 'TabEpEuroInd-internal.R' 'TauxPB-class.R' 'TauxPB-internal.R'  
 'Treso\_class.R' 'Treso\_calc\_vm.R' 'Treso\_internal.R'  
 'Treso\_revalo.R' 'Treso\_revenu.R' 'Treso\_update.R'  
 'taux\_period-function.R'

## R topics documented:

Action . . . . .	7
AlmEngine . . . . .	7
AutresPassifs . . . . .	8
AutresReserves . . . . .	8
autres_passif_load . . . . .	9
autres_reserves_load . . . . .	9
base_prod_fin . . . . .	10
Be . . . . .	11
buy_action . . . . .	11
buy_immo . . . . .	12
buy_oblig . . . . .	12
calc_coupon . . . . .	13
calc_dotation_ppb . . . . .	13
calc_fin_proj . . . . .	14
calc_flux_annee . . . . .	15
calc_frais . . . . .	15
calc_frais_fin . . . . .	16

calc_marge_fin . . . . .	17
calc_nominal . . . . .	17
calc_pm . . . . .	18
calc_pmv1 . . . . .	20
calc_pmv1_action . . . . .	20
calc_pmv1_immo . . . . .	21
calc_pmv1_oblig . . . . .	21
calc_PRE . . . . .	22
calc_prest . . . . .	22
calc_primes . . . . .	24
calc_qx . . . . .	25
calc_rach . . . . .	25
calc_rach_dyn . . . . .	26
calc_RC . . . . .	26
calc_rdt . . . . .	27
calc_rdt_marche_ref . . . . .	27
calc_reprise_ppb . . . . .	28
calc_result_technique . . . . .	28
calc_result_technique_ap_pb . . . . .	29
calc_revalo . . . . .	29
calc_revalo_pm . . . . .	30
calc_sur_dec . . . . .	31
calc_tra . . . . .	32
calc_tx_cible . . . . .	32
calc_tx_cible_ref_marche . . . . .	33
calc_tx_min . . . . .	34
calc_tx_sortie . . . . .	34
calc_vm_action . . . . .	35
calc_vm_immo . . . . .	36
calc_vm_oblig . . . . .	36
calc_vm_treso . . . . .	37
calc_vnc . . . . .	37
calc_z_spread . . . . .	38
Canton . . . . .	39
changement_choc . . . . .	39
changement_ESG . . . . .	40
changement_PortFin . . . . .	41
changement_PortFin_reference . . . . .	41
ChocSolvabilite2 . . . . .	42
create_ptf_bought_action . . . . .	43
create_ptf_bought_immo . . . . .	43
create_ptf_bought_oblig . . . . .	44
do_calc_nb_sold_action . . . . .	45
do_calc_nb_sold_immo . . . . .	45
do_calc_nb_sold_oblig . . . . .	46
do_choc_action_type1 . . . . .	47
do_choc_action_type2 . . . . .	47
do_choc_frais . . . . .	48
do_choc_immo . . . . .	49
do_choc_longevite . . . . .	49
do_choc_mortalite . . . . .	50
do_choc_rachat_down . . . . .	51

do_choc_rachat_up	51
do_choc_spread	52
do_choc_spread_unitaire	53
do_choc_taux	53
do_update_pmv1	54
do_update_PRE_val_courante	54
do_update_PRE_val_debut	55
do_update_RC_val_courante	56
do_update_RC_val_debut	56
do_update_vm_vnc_precedent	57
duration_sensi	57
echeancier	58
EpEuroInd	59
ESG	59
extract_ESG	60
finance_cible_pmv1	60
finance_cible_ppb	61
finance_contrainte_legale	62
finance_tmg	63
FraisFin	64
FraisPassif	65
frais_fin_load	65
frais_passif_load	66
get_choc_inflation_frais	66
get_choc_rach	67
get_choc_table	68
get_comport	68
get_qx_mort	69
get_qx_rach	70
get_rach_dyn	70
HypCanton	71
HypTech	72
Immo	72
Initialisation	73
initialisation_load	74
init_create_folder	74
init_debut_pgg_psap	75
init_debut_ppb	75
init_scenario	76
init_SimBEL	76
load_ht	77
load_pp	78
ModelPointESG	78
Oblig	79
ParamAlmEngine	79
ParamBe	79
ParamChocMket	80
ParamChocSousc	80
ParamComport	80
ParamRachDyn	81
ParamRevaloEngine	82
ParamTableMort	82

param_alm_engine_load . . . . .	83
pb_contr . . . . .	83
PortFin . . . . .	84
PortPassif . . . . .	85
Ppb . . . . .	85
PRE . . . . .	86
pre_load . . . . .	86
print_alloc . . . . .	87
proj_an . . . . .	87
proj_annee_autres_passifs . . . . .	88
proj_annee_av_pb . . . . .	89
RC . . . . .	90
rc_load . . . . .	90
reallocate . . . . .	91
resultat_fin . . . . .	92
RevaloEngine . . . . .	92
revalo_action . . . . .	93
revalo_immo . . . . .	93
revalo_treso . . . . .	94
revenu_treso . . . . .	94
run_be . . . . .	95
run_be_simu . . . . .	95
sell_action . . . . .	96
sell_immo . . . . .	97
sell_oblig . . . . .	97
sell_pvl_action . . . . .	98
set_architecture . . . . .	98
TabEpEuroInd . . . . .	99
TauxPB . . . . .	99
Treso . . . . .	100
update_cc_oblig . . . . .	100
update_dur_det_action . . . . .	101
update_dur_det_immo . . . . .	101
update_dur_oblig . . . . .	102
update_mat_res . . . . .	102
update_PortFin . . . . .	103
update_PortFin_reference . . . . .	104
update_reserves . . . . .	104
update_sd_oblig . . . . .	105
update_treso . . . . .	106
update_vm_action . . . . .	106
update_vm_immo . . . . .	107
update_vm_oblig . . . . .	107
update_vnc_oblig . . . . .	108
update_zsp_oblig . . . . .	108
vieillissement_action_PortFin . . . . .	109
vieillissement_av_pb . . . . .	109
vieillissement_immo_PortFin . . . . .	110
vieillissement_oblig_PortFin . . . . .	111
vieillissement_treso_PortFin . . . . .	112
vieillissement_ap_pb . . . . .	112
vieilli_mp . . . . .	113

<i>Action</i>	7
write_be_results . . . . .	114
yield_to_maturity . . . . .	114
<b>Index</b>	<b>115</b>

---

Action	<i>La classe Action</i>
--------	-------------------------

---

## Description

Classe pour les actifs de type Action

## Slots

ptf\_action est un dataframe, 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	<i>La classe ALMEngine</i>
-----------	----------------------------

---

## 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	<i>La classe AutresPassifs</i>
---------------	--------------------------------

---

**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	<i>La classe AutreReserves</i>
----------------	--------------------------------

---

**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	<i>Methode permettant de charger la valeur initiale des autres passifs.</i>
--------------------	---

---

**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 renseignés 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 un objet 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

---

Be

*La classe Be*

---

### Description

Classe pour le BE d'un assureur

### Slots

param\_be est un objet de type ParamBe qui regroupe les parametres de base du best estimate.

canton est un objet de type Canton correspond au canton en date initiale.

esg est un objet de type ESG.

tab\_flux est une liste qui contient les flux moyens de best estimate.

tab\_be est une liste qui contient la valeur du BE et de ses composantes.

### Author(s)

Prim'Act

### See Also

Mettre le lien vers les methodes de la classe

---

buy\_action

*Mise a jour de chaque composante d'un portefeuille action suite a un achat d'un autre portefeuille action.*

---

### Description

buy\_action est une methode permettant de mettre a jour le portefeuille action suite a 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 immo suite a un achat d'un autre portefeuille immo.</i>
----------	---

---

**Description**

buy\_immo est une methode permettant de mettre a jour le portefeuille immo suite a l'achat d'un autre portefeuille immo. de chaque composante d'un portefeuille immo.

**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_coupon	<i>Calcul le coupon des models points constituant le portefeuille obligataire.</i>
-------------	--

---

**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.

**Value**

Un vecteur dont chaque element correspond a la valeur du coupon de l'obligation consideree : 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	<i>Dote la valeur de la PPB</i>
-------------------	---------------------------------

---

**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                valeur de la dotation.

**Value**

Une liste comprenant l'objet mis a jour et le montant de la dotation effectuee.

**Author(s)**

Prim'Act

---

calc_fin_proj	<i>calcule le flux et les resultats ajustes en fin de projection.</i>
---------------	---

---

### 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 <a href="#">Canton</a> .
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 financier 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.

### See Also

La classe [EpEuroInd](#) et ses methodes. La classe [FraisPassif](#) et ses methodes.

---

calc_flux_annee	<i>Calcul les flux perçus dans l'année du fait de la détention des obligations du portefeuille obligataire.</i>
-----------------	---

---

### Description

calc\_flux\_annee est une méthode permettant de calculer les valeurs de nominal de l'ensemble des obligations composant un portefeuille obligataire.

### Usage

```
calc_flux_annee(x)
```

### Arguments

x un objet de la classe Oblig.

### Value

Une liste composée de deux vecteurs:

tombée\_coupon : Chaque élément correspond aux tombées de coupon pour l'année à venir. Ce vecteur a autant d'éléments que le portefeuille obligataire d'inputs a de lignes.

tombée\_echeance : Chaque élément correspond aux tombées d'échéances pour l'année à venir. Ce vecteur a autant d'éléments que le portefeuille obligataire d'inputs a de lignes.

### Author(s)

Prim'Act

---

calc_frais	<i>Calcule des frais de passif.</i>
------------	-------------------------------------

---

### 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 <a href="#">FraisPassif</a> .
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 de Frais financier 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 de calculer la marge financiere de l'assureur apres attribution de PB contractuelle.

**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
contrib_tmg_prest	est une valeur numeric comprenant par produit la contribution de la PPB ayu financement des TMG sur prestations.
contrib_tmg_stock	est une valeur numeric comprenant par produit la contribution de la PPB ayu financement des TMG sur stock.
contrib_ppb_tx_cible	une valeur de type numeric comprenant par produit la contribution de la PPB au financement au taux cible sur stock.

**Value**

Le montant de la marge de l'assureur.

**Author(s)**

Prim'Act

---

calc_nominal	Calcul le nominal des models points constituant le portefeuille obligataire.
--------------	--

---

**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 :  $\text{parite} * \text{nominal} * \text{nb\_unit}$ . Le vecteur renvoie a autant d'elements que le portefeuille obligataire en input a de lignes.

**Author(s)**

Prim'Act

---

calc\_pm

*Calcul les PM pour des contrats epargne en euros.*

---

**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, tab_prime, tab_prest, tx_cible, tx_min, an, method, tx_soc)
```

**Arguments**

x	un objet de la classe <a href="#">EpEuroInd</a> contenant les model points epargne euros.
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 <a href="#">calc_primes</a> .
tab_prest	est une liste contenant les flux de prestations pour chaque ligne de model points. Le format de cette liste correspond a la sortie flux de la methode <a href="#">calc_prest</a> .
tx_cible	est une liste conteant les taux cible annuel et semestriel par model points. Le format de cette liste correspond a la sortie de la methode <a href="#">calc_tx_cible</a> .
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 <a href="#">calc_tx_min</a> .
an	une valeur numeric represantant l'annee de projection courante.
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).
tx_soc	est une valeur numeric correspondant au taux de prelevements sociaux.

**Details**

Cette methode permet de calculer les montants de PM de fin d'annee avec une revalorisation minimale. 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.

**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

rev\_stock\_nette : un vecteur contenant la revalorisation minimale nette de l'annee appliquee au PM

enc\_charg\_stock : un vecteur contenant les chargement sur encours de l'annee, calcules en prenant en compte la revalorisation minimale

enc\_charg\_base\_th : un vecteur contenant les chargements sur encours theoriques de l'annee, evalues sur la base de la PM non revalorisees

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

base\_enc\_th : un vecteur contenant l'assiette de calcul des chargements sur encours de l'annee

soc\_stock : un vecteur contenant le prelevements sociaux de l'annee

it\_tech\_stock : un vecteur contenant les interets techniques sur stock de l'annee

it\_tech : un vecteur contenant les interets techniques sur stock et sur prestations de l'annee

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	<i>Mets a jour les sous totaux de d'actions et immobilier en plus ou moins value latente.</i>
-----------	---

---

**Description**

calc\_pmv1 est une methode permettant de calculer les valeurs de marche.

**Usage**

calc\_pmv1(x)

**Arguments**

x                      objet de la classe PortFin.

**Value**

L'objet PortFin dont la somme des composantes en PVL et en MVL a ete mise a jour

**Author(s)**

Prim'Act

---

calc_pmv1_action	<i>Calcul les valeurs de marches de chaque composante du portefeuille action.</i>
------------------	---

---

**Description**

calc\_pmv1\_action est une methode permettant de calculer les valeurs de marche.

**Usage**

calc\_pmv1\_action(x)

**Arguments**

x                      objet de la classe Action (decrivant le portefeuille d'action).

**Value**

Une liste composee de deux elements (pv1, mv1) correspondant respectivement aux sommes des plus values latentes actions et aux somme des moins values latentes action.

**Author(s)**

Prim'Act

---

calc_pmv1_immo	<i>Calcul les valeurs de marches de chaque composante du portefeuille immobilier.</i>
----------------	---

---

**Description**

calc\_pmv1\_immo est une methode permettant de calculer les valeurs de marche.

**Usage**

calc\_pmv1\_immo(x)

**Arguments**

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

**Value**

Une liste composee de deux elements (pv1, mv1) correspondant respectivement aux sommes des plus values latentes immobiliers et aux somme des moins values latentes immobiliers.

**Author(s)**

Prim'Act

---

calc_pmv1_oblig	<i>Calcul les valeurs de marches de chaque composante du portefeuille d'obligations.</i>
-----------------	--

---

**Description**

calc\_pmv1\_oblig est une methode permettant de calculer les valeurs de marche.

**Usage**

calc\_pmv1\_oblig(x)

**Arguments**

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

**Value**

Une liste composee de deux elements (pv1, mv1) correspondant respectivement aux sommes des plus values latentes obligations et aux somme des moins values latentes obligations.

**Author(s)**

Prim'Act

---

calc_PRE	<i>Calcul de la PRE.</i>
----------	--------------------------

---

### 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 de 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 calculee a partir des inputs transmis

var\_pre : variation de la pre courante

### Author(s)

Prim'Act

---

calc_prest	<i>Calcul les flux de prestations pour des contrats epargne en euros.</i>
------------	---

---

### 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, tx\_sortie, tx\_min, an, method, tx\_soc)

## Arguments

x	un objet de la classe <a href="#">EpEuroInd</a> contenant les model points epargne euros.
tx_sortie	une matrice contenant les taux de sortie associes a chaque ligne de model points. Le format de cette matrice correspond a la sortie de la methode <a href="#">calc_tx_sortie</a> .
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 <a href="#">calc_tx_min</a> .
an	une valeur numeric represantant l'annee de projection courante.
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).
tx_soc	est une valeur numeric correspondant au taux de prelevements sociaux.

## Details

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. 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.

## 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  
 rach\_tot : un vecteur contenant les flux de rachat totaux de l'annee  
 dc : un vecteur contenant les flux de deces de l'annee  
 rach\_part : un vecteur contenant les flux de rachat partiel de l'annee  
 prest : un vecteur contenant les flux prestations de l'annee  
 rev\_ech : un vecteur contenant la revalorisation des echeances de l'annee  
 rev\_rach\_tot : un vecteur contenant la revalorisation des rachats totaux de l'annee  
 rev\_dc : un vecteur contenant la revalorisation des deces de l'annee  
 rev\_rach\_part : un vecteur contenant la revalorisation des rachats partiels de l'annee  
 rev\_prest : un vecteur contenant la revalorisation brute des prestations de l'annee  
 rev\_prest\_net : un vecteur contenant la revalorisation des prestations nette de l'annee  
 enc\_charg : un vecteur contenant les chargements sur l'encours de l'annee  
 rach\_charg : un vecteur contenant les chargements sur les rachats de l'annee  
 soc\_prest : un vecteur contenant les prelevements sociaux sur prestations de l'annee  
 it\_tech\_prest : un vecteur contenant les interets techniques sur prestations de l'annee.

Le format de la liste stock est :

nb\_ech : un vecteur contenant le nombre de sorties en echeance de l'annee

nb\_rach\_tot : un vecteur contenant le nombre de rachats totaux de l'annee

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\\_tx\\_sortie](#), [calc\\_tx\\_min](#).

---

calc\_primes

*Calcul les flux de primes pour des contrats epargne en euros.*

---

### 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](#) contenant les model points epargne euros.

### Details

Cette fonction permet de projeter uniquement des primes periodiques de contrats epargne en euros.

### 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 chargemenets sur primes pri\_chgt.

### 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 <a href="#">ParamTableMort</a> contenant la table de mortalite.
age	une valeur numeric correspondant a l'age.
gen	une valeur numeric 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 <a href="#">ParamTableRach</a> contenant la table de rachat.
age	une valeur numeric correspondant a l'age.
anc	une valeur numeric 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.

### Usage

```
calc_rach_dyn(p, tx_cible, tx_serv)
```

### Arguments

p	un objet de la classe <a href="#">ParamRachDyn</a> 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	<i>Calcul de la RC.</i>
---------	-------------------------

---

### Description

calc\_RC est une methode permettant de calculer le montant de RC.

### Usage

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

### Arguments

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

### Author(s)

Prim'Act

---

calc_rdt	<i>Calcul les rendements de chacune des composante des sous-portefeuilles action et immobilier du portefeuille PortFin.</i>
----------	---

---

**Description**

calc\_rdt est une methode permettant de calculer les rendements des portefeuilles Action et Immo d'un objet PortFin.

**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 & immo).

**Value**

Un data frame composÃ© de deux colonnes et autant de lignes que le portefeuille action a de lignes.

**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**

mp_esg	est un objet de type ModelPointESG, qui represente la situation courante en annee et simulations des valeurs de l'ESG.
param_comport	un objet de la classe ParamComport.

**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	objet de la classe Ppb.
montant	valeur de la reprise.

### Value

Une liste comprenant l'objet mis a jour et le montant de la reprise effectuee.

### Author(s)

Prim'Act

---

calc_result_technique	<i>calcule le resultat technique</i>
-----------------------	--------------------------------------

---

### 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 produit par la methode vieillissement_av_pb appliquee a un portefeuille de passif.
var_pre	est une valeur numeric correspondant a la variation de PRE.

### Value

Le resultat technique

---

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 <a href="#">viellissement_av_pb</a> .
passif_ap_pb	est une liste produit par la methode <a href="#">viellissement_ap_pb</a> .
ppb	est un objet de la classe <a href="#">Ppb</a> 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

*Applique la politique de revalorisation d'un canton.*

---

### 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)

### Arguments

x	un objet de la classe <a href="#">Canton</a> .
passif_av_pb	est une liste produit par la methode <a href="#">viellissement_av_pb</a> 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.
result_tech	est la valeur numeric du resultats techniques prise en compte de la PB.

**Value**

Une liste avec la valeur de la revalorisation additionnelle nette servie par produit, la ppb mise a jour, la PPB mise a jour et le resultat technique avant PB.

**Author(s)**

Prim'Act

---

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

---

**Description**

calc\_revalo\_pm est une methode permettant de calculer la revalorisation des PM sur une annee.

**Usage**

```
calc_revalo_pm(x, rev_net_alloue, tx_soc)
```

**Arguments**

x un objet de la classe [EpEuroInd](#) contenant les model points epargne euros.  
 rev\_net\_alloue une valeur de type numeric correspondant au montant de revalorisation a allouer.  
 tx\_soc est une valeur numeric correspondant au taux de prelevement sociaux.

**Details**

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.

**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'annee appliquee au PM

rev\_stock\_net\_ap\_pb : un vecteur contenant la revalorisation nette de l'annee appliquee au PM. Elle peut etre negative pour des contrats a taux negatif.

enc\_charg\_stock\_ap\_pb : un vecteur contenant les montants de chargement sur encours de l'annee calcules pour le stock de PM

soc\_stock\_ap\_pb : un vecteur contenant les prelevements sociaux de l'annee

Le format de la liste stock est : s

pm\_fin\_ap\_pb : un vecteur contenant le montant de PM en fin d'annee

### Author(s)

Prim'Act

### See Also

Le calcul des PM avec revalorisation minimale uniquement [calc\\_pm](#).

---

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

---

### Description

calc\_sur\_dec est une methode 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 (decrivant le portefeuille obligataire).

### Value

Un data.frame compose de deux colonnes : 1 ere colonne : surcotes decotes ; 2de colonne : valeurs nettes comptables.

### Author(s)

Prim'Act

---

calc_tra	<i>Calcul du taux de rendement financier</i>
----------	--

---

### 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	<i>Calcul du taux cible pour des contrats epargne en euros.</i>
---------------	---

---

### Description

calc\_tx\_cible est une methode permettant d'evaluer le taux de revalorisation cible de chaque model point.

### Usage

```
calc_tx_cible(x, ht, list_rd)
```

### Arguments

x	un objet de la classe <a href="#">EpEuroInd</a> contenant les model points epargne euros.
ht	un objet de la classe <a href="#">HypTech</a> contenant differentes lois de comportement.
list_rd	une liste contenant les rendements de reference. Le format de cette liste est : <b>le taux de rendement obligataire</b> <b>le taux de rendement de l'indice action de reference</b> <b>le taux de rendement de l'indice immobilier de reference</b> <b>le taux de rendement de l'indice tresorerie de reference</b>



**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.

**Usage**

```
calc_tx_cible_ref_marche(param_comport, list_rd, tx_cible_prec)
```

**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 <a href="#">EpEuroInd</a> 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_tx_sortie	<i>Calcul des taux de sortie pour des contrats epargne en euros.</i>
----------------	--

---

**Description**

calc\_tx\_sortie est une methode permettant de calculer les differents taux de sortie sur une periode.

**Usage**

```
calc_tx_sortie(x, ht)
```

**Arguments**

x	un objet de la classe <a href="#">EpEuroInd</a> contenant les model points epargne euros.
ht	un objet de la classe <a href="#">HypTech</a> contenant differentes tables de mortalite et differentes lois de rachat.

**Value**

Une matrice contenant pour chaque model points en ligne :

qx\_rach\_tot : un vecteur contenant les taux de rachats totaux  
 qx\_rach\_tot\_dyn : un vecteur contenant les taux de rachats totaux dynamiques  
 qx\_dc : un vecteur contenant les taux de deces  
 qx\_rach\_part : un vecteur contenant les taux de rachats partiels  
 qx\_rach\_part\_dyn : un vecteur contenant les taux de rachats partiels dynamiques.

**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_vm_action	<i>Calcul les valeurs de marches de chaque composante du portefeuille action.</i>
----------------	---

---

**Description**

calc\_vm\_action est une methode permettant de calculer les valeurs de marche.

**Usage**

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

**Arguments**

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

**Value**

L'objet x dont les valeurs de marche ont ete mises a jour.

**Author(s)**

Prim'Act

---

calc_vm_immo	<i>Calcul les valeurs de marches de chaque composante du portefeuille immobilier.</i>
--------------	---

---

**Description**

calc\_vm\_immo est une methode permettant de calculer les valeurs de marche.

**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 d'immobilier du portefeuille immobilier de l'assureur. Contient autant d'elements que le portefeuille a de lignes.

**Value**

L'objet x dont les valeurs de marche ont ete mises a jour.

**Author(s)**

Prim'Act

---

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

---

**Description**

calc\_vm\_oblig est une methode permettant de calculer les valeurs de marche.

**Usage**

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

**Arguments**

x	objet de la classe Oblig (decrivant le portefeuille d'obligation).
rdt	vecteur de type <code>numeric</code> decrivant le rendement 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 de marche ont ete mises a jour.

**Author(s)**

Prim'Act

---

calc_vm_treso	<i>Calcul 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 ptf. Contient autant d'elements que le portefeuille a de lignes.
flux_milieu	vecteur decrivant les flux (percus)entrants : positif, sortants : negatifs) en milieu d'annee, ventile selon chacune des lignes de cash.
flux_fin	vecteur decrivant les flux (entrants : positifs, sortants : negatifs) en fin d'annee, ventile selon chacune des lignes de cash.

**Value**

L'objet x dont les valeurs de marche ont ete mises a jour.

**Author(s)**

Prim'Act

---

calc_vnc	<i>Calcul les valeurs nettes comptables de chaque composante du portefeuille obligation.</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 numeric 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](#).

---

<code>chargement_choc</code>	<i>Permet de charger dans un objet de la classe ChocSolvabilite2 l'ensemble des parametres necessaires a la bonne application des chocs de marche et de souscription au sens de la formule standard de la directive Solvabilite 2.</i>
------------------------------	--

---

## Description

`chargement_choc` est une methode permettant de charger les parametres des differents chocs de marche et de souscription, tels que renseignes par l'utilisateur.

## Usage

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

**Arguments**

x objet de la classe ChocSolvabilite2.

folder\_choc\_address est un character. Cette chaine de caractere est construite par la fonction [set\\_architecture](#) de la classe [Initialisation](#), elle contient l'adresse du dossier contenant les fichiers de parametres des chocs de la formule standard a appliquer renseignes par l'utilisateur.

**Value**

x l'objet de la classe ChocSolvabilite2 dont les attributs param\_choc\_mket et param\_choc\_sousc ont ete mis a jour.

**Author(s)**

Prim'Act

**See Also**

La creation de l'architecture de chargement des donnees et parametres renseignes par l'utilisateur [set\\_architecture](#), ainsi que les classes [ParamChocMket](#) et [ParamChocSousc](#).

---

chargement_ESG	<i>Cette methode construit l'objet de classe ESG a partir des donnees .csv issues de l'ESG Prim'Act.</i>
----------------	--

---

**Description**

chargement\_ESG est une methode permettant de construire et charger les trajectoires simulees par le Generateur de Scenarios Economiques de Prim'Act.

**Usage**

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

**Arguments**

folder\_ESG\_address est un character. Il correspond au chemin de reference du dossier contenant les extractions de l'ESG Prim'Act. Les differentes adresses potentielles pour les differents ESG employes (central, hausse de taux, baisse de taux, avec ou sans VA) sont construites par la fonction [set\\_architecture](#) de la classe [Initialisation](#).

nb\_simu est une valeur de type integer correspondant au nombre de trajectoire simulees par l'ESG Prim'Act.

nb\_annee\_proj est une valeur de type integer correspondant au nombre d'annees de projection des sorties de l'ESG Prim'Act.

**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 <a href="#">set_architecture</a>
mp_ESG	est un objet de la classe ModelPointESG, qui fournit le resultat financier du portefeuille.

## Value

L'objet PortFin tel que precise par les donnees initiales et le sparametres 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 financier 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 <a href="#">set_architecture</a>
mp_ESG	est un objet de la classe ModelPointESG, qui fournit le resultat financier du portefeuille.

**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

---

ChocSolvabilite2

*La classe ChocSolvabilite2 permet d'implementer l'ensemble des scenarios de choc initiaux au sens de la formule standard de la directive Solvabilite 2. Cette classe contient deux attributs qui contiennent respectivement l'ensemble des parametres necessaire a l'application des chocs Marche et Souscription, au sens de la formule standard. Cette classe contient aussi l'ensemble des methodes permettant d'appliquer chacun de ces chocs individuellement a un objet de la classe canton.*

---

**Description**

La classe ChocSolvabilite2 permet d'implementer l'ensemble des scenarios de choc initiaux au sens de la formule standard de la directive Solvabilite 2. Cette classe contient deux attributs qui contiennent respectivement l'ensemble des parametres necessaire a l'application des chocs Marche et Souscription, au sens de la formule standard. Cette classe contient aussi l'ensemble des methodes permettant d'appliquer chacun de ces chocs individuellement a un objet de la classe canton.

**Slots**

param\_choc\_mket objet de la classe ParamChocMket

param\_choc\_sousc : objet de la classe ParamChocSousc

**Author(s)**

Prim'Act

**See Also**

Le calcul et la mise a jour des autres reserves [do\\_choc\\_action](#), [do\\_choc\\_immo](#), [do\\_choc\\_spread](#) et [do\\_choc\\_frais](#), [do\\_choc\\_mortalite](#), [do\\_choc\\_longevite](#), [do\\_choc\\_rachat\\_up](#), [do\\_choc\\_rachat\\_down](#).

---

`create_ptf_bought_action`*Ajuste les quantites d'actions a acheter.*

---

### 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

<code>x</code>	objet de la classe <a href="#">Action</a> , correspondant au portefeuille actions de reinvestissement. Ce portefeuille est unitaire.
<code>coefficient</code>	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 <a href="#">Immo</a> , 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**

create\_ptf\_bought\_oblig est une methode permettant 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 <a href="#">Oblig</a> , 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](#).

---

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

---

### Description

`do_calc_nb_sold_action` est une methode permettant 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 <a href="#">Action</a> , correspondant au portefeuille action de l'assureur.
<code>montant_vente</code>	est un reel de type numeric correspondant a la vente totale de vm 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

`do_calc_nb_sold_immo` est une methode permettant de calculer pour chaque ligne d'un portefeuille immo 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 <a href="#">Immo</a> , 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**

do\_calc\_nb\_sold\_oblig est une methode permettant 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 <a href="#">Oblig</a> , 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	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 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	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 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_frais	<i>Permet à partir d'un canton initial de créer un canton choqué frais.</i>
---------------	---

---

**Description**

do\_choc\_frais est une méthode permettant d'appliquer le choc frais de la formule standard Solvabilité 2 à 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 choqué (i.e. central) de l'assureur.

**Value**

canton l'objet de la classe canton correspondant au scénario choqué frais au sens de la formule standard Solvabilité 2.

**Note**

La paramétrisation des chocs de frais est effectuée 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 choque (i.e. central) de l'assureur.

**Value**

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

**Note**

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

**Author(s)**

Prim'Act

---

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

---

**Description**

do\_choc\_longevite est une methode permettant d'appliquer le choc longevite de la formule standard Solvabilite 2 a 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 choque (i.e. central) de l'assureur.

**Value**

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

**Note**

La parametrisation des chocs de longevite est effectuee 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.

**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.

**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_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.

**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 action.</i>
----------------	--

---

**Description**

do\_choc\_spread est une methode permettant d'appliquer le choc spread de la formule standard Solvabilite 2 a un canton.

**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 l'index de la ligne obligataire. Cette parametrisation est effectuee dans les fichiers d'inputs utilisateurs.

**Author(s)**

Prim'Act

---

do\_choc\_spread\_unitaire

*Permet a partir d'une table contenant les elements du choc de spread obligatoire S2 et d'une ligne obligatoire d'un element Oblig d'un portefeuille financier d'appliquer le choc de spread a cette ligne obligatoire.*

---

### Description

do\_choc\_spread\_unitaire est une methode permettant d'appliquer le choc spread de la formule standard Solvabilite 2 a une ligne obligatoire.

### Usage

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

### Arguments

table\_choc\_spread

data.frame contenant la table des chocs de spreads.

ligne\_oblig

est un data.frame. Il correspond a une ligne obligatoire d'un portefeuille Oblig d'un assureur.

### Value

vm\_choquee un numeric correspondant a la valeur de marche de la ligne obligatoire suite a l'application du choc de spread a cette ligne.

### Author(s)

Prim'Act

### See Also

La fonction d'application du choc de spread a un canton, faisant appel a cette fonction unitaire : [do\\_choc\\_spread](#).

---

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 objet PortFin</i>
----------------	--

---

**Description**

do\_update\_pmv1 est une methode permettant de calculer le taux de rendement financier du portefeuille.

**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 ont ete recalculées avec les elements du PortFin renseigne en input.

**Author(s)**

Prim'Act

---

do_update_PRE_val_courante	<i>Mise a jour de la valeur courante de PRE</i>
----------------------------	---

---

**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**

- x                      objet de la classe PRE correspondant a la PRE avant mise a jour.
- val\_courante        est un numeric 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 courante de PRE*

---

**Description**

do\_update\_PRE\_val\_debut est une methode permettant de calculer le montant de PRE.

**Usage**

do\_update\_PRE\_val\_debut(x, val\_debut)

**Arguments**

- x                      objet de la classe PRE correspondant a la PRE avant mise a jour.
- val\_courante        est un numeric correspondant au montant de PRE calcule par la fonction [calc\\_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**

<code>x</code>	objet de la classe RC correspondant a la RC avant mise a jour.
<code>val_courante</code>	est un <code>numeric</code> correspondant au montant de RC calcule par la fonction <a href="#">calc_RC</a> .

**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 courante de RC*

---

**Description**

`do_update_RC_val_debut` est une methode permettant de calculer le montant de RC.

**Usage**

```
do_update_RC_val_debut(x, val_debut)
```

**Arguments**

<code>x</code>	objet de la classe RC correspondant a la RC avant mise a jour.
<code>val_courante</code>	est un <code>numeric</code> correspondant au montant de RC calcule par la fonction <a href="#">calc_RC</a> .

**Value**

L'objet RC 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 les objets constituant un PortFin.*

---

**Description**

`do_update_vm_vnc_precedent` est une methode permettant de calculer et mettre a jour un portefeuille financier suite a un vieillissement.

**Usage**`do_update_vm_vnc_precedent(x)`**Arguments**

x	objet de la classe <code>PortFin</code> , correspondant au portefeuille financier de l'assureur avant mise a jour de l'attribut <code>vm_vnc_precedent</code> .
---	---

**Value**

L'objet 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

---

`duration_sensi`*Calcul la duration de chaque composante d'un portefeuille obligataire.*

---

**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 <code>Oblig</code> (decrivant le portefeuille obligataire).
---	--

**Value**

Un data frame compose de deux colonnes : la premiere est compose de la duration de chacune des obligations du portefeuille obligataire. La seconde est compose 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>Calcul les flux d'un model point ou d'un ensemble de odels points obligataires.</i>
------------	--

---

**Description**

echeancier est une methode permettant de calculer les flux jusqu'a maturite residuelle.

**Usage**

```
echeancier(coupon, maturite, zspread, nominal, yield)
```

**Arguments**

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

**Value**

Une matrice contenant :

`grid_flux` : la matrice d'ecoulement des flux. Cette matrice a autant de colonnes que le max du vecteur de maturite residuelle, et autant de lignes que les vecteurs d'input `coupon`, `maturite`, `zspread`, `nominal`. Chaque ligne decrit les flux annuels a venir pour l'actif obligataire de caracteristique renseigne en input.

**Author(s)**

Prim'Act

---

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

---

**Description**

Une classe pour les passifs epargne en euros.

**Slots**

mp un objet `data.frame` au format fige contenant l'ensemble de model points epargne en euros.

tab un objet de la classe [TabEpEuroInd](#) dedie au stockage de variables intermediaires.

**Author(s)**

Prim'Act

**See Also**

Le calcul des primes, des prestations et des PM : [calc\\_primes](#), [calc\\_prest](#), [calc\\_pm](#). Le calcul des taux de sortie, du taux minimum et des taux cible de revalorisation : [calc\\_tx\\_sortie](#), [calc\\_tx\\_min](#), [calc\\_tx\\_cible](#). La revalorisation des PM apres participation aux benefices : [calc\\_relavo\\_pm](#). Le vieillissement des model points sur une periode : [vieilli\\_mp](#).

---

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

---

**Description**

Classe pour les donnees globales d'actif

**Author(s)**

Prim'Act

**See Also**

Mettre le lien vers les methodes de la classe

---

extract_ESG	<i>Cette methode construit l'objet de classe ModelPoint_ESG a partir d'un objet de la classe ESG. Le ModelPoint ESG ainsi construit correspond a l'extraction de courbes de l'ESG pour une annee specifique et pour une simulation specifique.</i>
-------------	--

---

### Description

extract\_ESG est une methode permettant de construire et charger les trajectoires simulees par le Generateur de Scenarios Economiques de Prim'Act.

### Usage

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

### Arguments

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

### Value

x l'objet de la classe ModelPoint\_ESG construit.

### Author(s)

Prim'Act

### See Also

La classe [ModelPoint\\_ESG](#).

---

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

---

### Description

finance\_cible\_pmv1 est une methode permettant de determiner le financement de taux cibles par des cessions de plus-values latentes.

### Usage

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

**Arguments**

bes_cible	est un vecteur de type <code>numeric</code> comprenant par produit le besoin de financement aux taux cible.
rev_stocknette	est un vecteur de type <code>numeric</code> comprenant par produit le montant de revalorisation nette atteint.
base_fin	est un vecteur de type <code>numeric</code> comprenant par produit la base de produits financiers.
seuil_pmv1	est une valeur <code>numeric</code> comprenant au montant de plus-values latentes qui peut être liquidée. Ce montant doit être exprimé en tenant compte de l'abattement réalisé pour rapport les plus-values latentes de l'actif général au passif.
tx_pb	est un vecteur de type <code>numeric</code> comprenant par produit les taux de participation aux bénéfices contractuels.

**Details**

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

**Value**

Une liste avec la valeur de la revalorisation nette servie par produit et le montant de plus-values à liquider, ramené à la valeur du passif pour financer la revalorisation.

**Author(s)**

Prim'Act

---

finance_cible_ppb	<i>Evalue le financement d'un taux cible par la PPB.</i>
-------------------	--

---

**Description**

finance\_cible\_ppb est une méthode permettant de déterminer le financement de taux cibles par la provision pour participation aux bénéfices.

finance\_cible\_ppb est une méthode permettant de déterminer le financement de taux cibles par la provision pour participation aux bénéfices.

**Usage**

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

```
finance_cible_ppb(bes_cible, rev_stocknette, ppb)
```

**Arguments**

marge_fin	est valeur numeric donnant le montant courant de la marge financiere de l'assureur.
bes_cible	est un vecteur de type numeric comprenant par produit le besoin de financement aux taux cible.
rev_stock_nette	est un vecteur de type numeric comprenant par produit le montant de revalorisation nette atteint.
marge_min	est une valeur de type numeric correspondant au montant minimum de marge financiere souhaite par l'assureur.
ppb	est un objet de la classe Ppb qui renvoie l'etat courant de la PPB.
bes_cible	est un vecteur de type numeric comprenant par produit le besoin de financement aux taux cible.
rev_stock_nette	est un vecteur de type numeric comprenant par produit le montant de revalorisation nette atteint.

**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, et par une reprise sur PPB.

**Value**

Une liste avec la valeur de la revalorisation nette servie par produit et le montant de marge de l'assureur.

Une liste avec la valeur de la revalorisation nette servie par produit, les montants de dotation et reprise sur PPB et la PPB mise à jour.

**Author(s)**

Prim'Act

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	est un vecteur de type numeric comprenant par produit la base de produits financiers.
result_tech	est une valeur de type numeric comprenant le resultat technique.
it_stock	est un vecteur de type numeric comprenant par produit les interets techniques affectes au stock.
rev_stocknette	est un vecteur de type numeric comprenant par produit la revalorisation nette affectee au stock.
rev_prestnette	est un vecteur de type numeric comprenant par produit la revalorisation nette affectee aux prestations.
dot_ppb	est une valeur de type numeric comprenant la dotation de PPB financant la revalorisation sur stock, hors TMG.
marge_fin	est une valeur de type numeric comprenant la marge financiere courante de l'assureur.
ppb	est un objet de la classe Ppb qui renvoie l'etat courant de la PPB.
param_revalo	est un objet de la classe ParamRevaloEngine comprenant les parametres de revalorisation.
base_fin_entendu	est une valeur numeric comprenant la base totale de produits financiers, y compris passifs non modelises.

**Value**

Une liste avec la revalorisation nette affectee au stock la marge financiere de l'assureur apres prise en compte de la contrainte legale, la PPB et l'objet param\_revalo qui comprend l'etat du solde de participation aux benefices reglementaire.

**Author(s)**

Prim'Act

---

finance_tmg	<i>Calcule la contribution de la PPB au financement des taux minimum garanti.</i>
-------------	---

---

**Description**

finance\_tmg est une methode permettant d'evaluer le contribution de la PPB au financement des taux minimum garanti sur prestations et sur stock.

**Usage**

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

**Arguments**

- `bes_tmg_prest` est un vecteur de type `numeric` comprenant par produit le besoin de finance des TMG sur prestations. @param `bes_tmg_stock` est un vecteur de type `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**

Cette fonction priorise le financement des TMG prestations. Elle met à jour la PPB des reprises effectuees.

**Value**

Une liste comprenant la valeur de la contribution au financement des TMG sur prestations et sur stock. Cette liste comprend également la PPB mise à 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 `numeric` 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 applique.

**Author(s)**

Prim'Act

**See Also**

Mettre le lien vers les methodes de la classe



---

FraisPassif	<i>La classe FraisPassif</i>
-------------	------------------------------

---

**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 calculer 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 <a href="#">FraisPassif</a> .
---------------------------	--

### 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	<i>Applique le choc dépenses de la formule standard a l'inflation la table de simulation pour l'indice inflation</i>
--------------------------	--

---

### Description

get\_choc\_inflation\_frais est une methode permettant d'appliquer pplique le choc dépenses 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 <a href="#">ESG</a> .
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 d'inflation compris 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	Applique les chocs de rachat de la formule standard.
---------------	--

---

**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**

type_choc_rach	est un caracter 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.
ht	un objet de la classe <a href="#">HypTech</a> contenant differentes lois de rachat partielles et totales.

**Value**

L'objet ht 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

choc	une valeur numeric indiquant le taux de choc.
ht	un objet de la classe <a href="#">HypTech</a> contenant differentes tables de mortalite.

### Value

L'objet ht apres choc.

### 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 <a href="#">HypTech</a> .
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 : <b>le taux de rendement obligataire</b> <b>le taux de rendement de l'indice action de reference</b> <b>le taux de rendement de l'indice immobilier de reference</b> <b>le taux de rendement de l'indice tresorerie de reference</b>
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	<i>Recuperer les taux de deces calcules.</i>
-------------	--

---

**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 <a href="#">HypTech</a> .
nom_table	un nom de la table de mortalite.
age	est la valeur numeric de l'age.
gen	est la valeur numeric 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 <a href="#">HypTech</a> .
nom_table	un nom de la 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 <a href="#">HypTech</a> .
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

*La classe HypCanton.*

---

**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

*La classe HypTech.***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 elements de ces liste doit avoir prealablement ete 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 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](#).

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.

**Author(s)**

Prim'Act



**See Also**

Mettre le lien vers les methodes de la classe

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.

---

initialisation_load	<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**

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_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 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 creer 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_debut_ppb	<i>Re-initialise un objet Ppb en debut d'annee</i>
----------------	--

---

**Description**

init\_debut\_ppb est une methode permettant de re-initialiser les montants de dotation ou de reprise cumules sur l'annee et de re-initialiser le montant de PPB de debut de periode.

**Usage**

```
init_debut_ppb(x)
```

**Arguments**

x	objet de la classe Ppb.
---	-------------------------

**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

---

load_ht	<i>Methode permettant de charger la valeur des parametres techniques.</i>
---------	---

---

**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_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 [PortPassifs](#).

### 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.

---

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

---

### Description

Classe pour une extraction de l'ESG.

### Slots

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

num\_traj est une valeur integer correspondant au numero de simulation de l'ESG.

yield\_curve Courbe des  $R(k, k+i)$  ou  $i$  va de 1 au nb\_annee\_proj.

### Author(s)

Prim'Act

### See Also

Mettre le lien vers les methodes de la classe

---

Oblig	<i>Classe pour les actifs de type obligation.</i>
-------	---

---

**Description**

Classe pour les actifs de type obligation.

**Slots**

ptf\_oblig est un dataframe, chaque ligne represente un actif obligation du portefeuille d'obligation.

**Author(s)**

Prim'Act

**See Also**

Mettre le lien vers les methodes de la classe

Les operations d'achat vente obligations [buy\\_oblig](#) et [sell\\_oblig](#).

---

ParamAlmEngine	<i>La classe ParamAlmEngine</i>
----------------	---------------------------------

---

**Description**

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

**Author(s)**

Prim'Act

**See Also**

Methode principale : [param\\_alm\\_engine\\_load](#)

---

ParamBe	<i>Classe pour ParamBe</i>
---------	----------------------------

---

**Description**

Classe pour ParamBe

---

ParamChocMket

*La classe ESG*

---

**Description**

Classe pour les donnees globales d'actif

**Author(s)**

Prim' Act

**See Also**

Mettre le lien vers les methodes de la classe

---

ParamChocSousc

*La classe ParamChocSousc*

---

**Description**

Classe pour pour les parametres de comportement

**Slots**

mp : model point contenant l'ensemble des parametres pour faire les differents chocs

**Author(s)**

Prim' Act

**See Also**

Mettre le lien vers les methodes de la classe

---

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 obligataire 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 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.**

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>Classe pour ParamRevaloEngine</i>
-------------------	--------------------------------------

---

### Description

Classe pour ParamRevaloEngine

---

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

---

### Description

Une classe de parametres pour les tables de mortalite.

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.

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.

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.

### Note

Les tables de mortalite doivent contenir des effectifs sous risque par age ( $L_x$ ).

### Author(s)

Prim'Act

Prim'Act

### See Also

Le calcul du taux de deces [calc\\_qx](#).

Le calcul du taux de rachat [calc\\_rach](#).

---

param\_alm\_engine\_load    *Chargement des attributs d'un objet ParamAlmEngine a partir des donnees utilisateurs.*

---

### 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\\_ref](#)).

### Usage

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

### Arguments

file_alm_address	un character renvoyant .
ptf_fin_ref	un objet de la classe PortFin correspondant au portefeuille de reinvestissement.

### Value

Pas de sortie.

### Author(s)

Prim'Act

---

pb\_contr                      *Calcule la PB contractuelle.*

---

### 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	est un vecteur de type numeric comprenant par produit la base de produits financiers.
tx_pb	est un vecteur de type numeric comprenant par produit les taux de participation aux benefices contractuels.
rev_stock_brut	est 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 stock, revalorise au taux minimum.
tx_enc_moy	est un vecteur de type 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**

Une liste composee de deux vecteurs comprenant par produit les chargements sur encours appliques et la revalorisation contractuelle nette.

**Author(s)**

Prim'Act

---

PortFin	<i>La classe PortFin</i>
---------	--------------------------

---

**Description**

Classe pour le portefeuille global d'actif

**Slots**

`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 Obligs qui sont en situation de plus values latentes.

`mv1_action` est un `numeric`, qui correspond a la somme des moins values latentes des actifs Actions qui sont en situation de moins values latentes.

`mv1_immo` est un `numeric`, qui correspond a la somme des moins values latentes des actifs Immos qui sont en situation de moins values latentes.

`mv1_oblig` est un `numeric`, qui correspond a la somme des moins values latentes des actifs Obligs 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

---

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

---

### 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.

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](#).

### 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 : [vieillissement\\_av\\_pb](#), [vieillissement\\_ap\\_pb](#).

---

Ppb	<i>La classe Ppb</i>
-----	----------------------

---

### Description

Classe pour la provision pour participation aux benefices (PPB)

### Slots

valeur\_ppb est la valeur courante prise par la PPB.

ppb\_debut est la valeur prise par la PPB en debut d'annee.

seuil\_rep est une valeur numeric correspond a la proportion de PPB de debut d'annee que l'on peut reprendre sur une periode. sur la PPB sur une periode.

seuil\_dot est une valeur numeric correspond a la 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 reprise effectuee sur une periode.

compte\_dot est une valeur numeric qui totalise les montants de dotation effectuee sur une periode.

**Author(s)**

Prim'Act

**See Also**

Mettre le lien vers les methodes de la classe

---

PRE	<i>La classe PRE</i>
-----	----------------------

---

**Description**

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**

Mettre le lien vers les methodes de la classe

---

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

*Calcul le poids de chaque composante du portefeuille action.*


---

### 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

*Projetter un canton sur une periode.*


---

### 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 numeric 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 : [viellissement\\_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](#), [init\\_debut\\_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.

**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 numeric 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	une liste contenant les rendements de reference. Le format de cette liste est : <b>le taux de rendement obligataire</b> <b>le taux de rendement de l'indice action de reference</b> <b>le taux de rendement de l'indice immobilier de reference</b> <b>le taux de rendement de l'indice tresorerie de reference</b>

**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**

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**

Mettre le lien vers les methodes de la classe

---

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 RCcise en input.

**Author(s)**

Prim'Act

---

**reallocate***Realise les operations d'achats ventes*

---

**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 <a href="#">PortFin</a> .
ptf_reference	est le portefeuille de reinvestissement. C'est un objet de la classe <a href="#">PortFin</a> .
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](#).

---

resultat_fin	<i>Calcul de 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

---

RevaloEngine	<i>La classe RevaloEngine</i>
--------------	-------------------------------

---

**Description**

Classe comprenant les methodes pour l'application de la revalorisation des passifs

**Slots**

param\_revalo est objet de type ParamRevalo comprenant les parametres utilises pour la revalorisation des contrats.

**Author(s)**

Prim'Act

**See Also**

Mettre le lien vers les methodes de la classe

---

revalo_action	<i>Calcul les valeurs de marches de chaque composante du portefeuille action.</i>
---------------	---

---

**Description**

revalo\_action est une methode permettant de calculer les valeurs de marche.

**Usage**

```
revalo_action(x, S, S_prev)
```

**Arguments**

x	objet de la classe Action (decrivant le portefeuille d'action).
S	vecteur numeric de valeur de chaque stock du ptf en milieu d'annee N (date de versement des dividendes)
S_prev	vecteur numeric de valeur de chaque stock du ptf en milieu d'annee N-1.

**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 percues au titre de chacune des actions composants le portefeuille action.

**Author(s)**

Prim'Act

---

revalo_immo	<i>Calcul les valeurs de marches de chaque composante du portefeuille immobilier.</i>
-------------	---

---

**Description**

revalo\_immo est une methode permettant de calculer les valeurs de marche.

**Usage**

```
revalo_immo(x, S, S_prev)
```

**Arguments**

x	objet de la classe Immo (decrivant le portefeuille d'immobilier).
S	vecteur numeric de valeur de chaque stock du ptf en milieu d'annee N (date de versement des dividendes)
S_prev	vecteur numeric de valeur de chaque stock du ptf en milieu d'annee N-1.

**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 dividendes 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**

S	vecteur de valeur de chaque ligne du ptf en milieu d'annee N (date de calcul des flux).
S_prev	vecteur de valeur de chaque ligne du ptf en milieu d'annee N-1.

**Value**

Un vecteur ayant autant d elements que les vecteurs inputs. Chaque element correspondant au rendement annuel d'une lige de tresorerie.

**Author(s)**

Prim'Act

---

revenu_treso	<i>Calcul le revenu tresorerie.</i>
--------------	-------------------------------------

---

**Description**

revenu\_treso est une methode permettant de calculer les valeurs de marche.

**Usage**

```
revenu_treso(x, rdt, flux_milieu)
```

**Arguments**

x	est un objet de la classe Tresor en debut d'annee
rdt	est le rendement de la classe Tresor au cours de l'annee (i.e. en fin d'annee)
flux_milieu	est le flux du milieu de l'annee en cours (i.e. ulterieur a l'objet Tresor renseigne)

**Author(s)**

Prim'Act

---

run_be	<i>Calcul d'un BE</i>
--------	-----------------------

---

**Description**

run\_be est une methode permettant de calcul un best estimate pour un canton.

**Usage**

```
run_be(x, pre_on)
```

**Arguments**

x	est un objet de type Be.
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.

**Value**

L'objet BE avec les tables de resultats mise a jour et la liste des simulations ignorees en cas d'erreur.

**Author(s)**

Prim'Act

---

run_be_simu	<i>Calcul d'un BE par simulation</i>
-------------	--------------------------------------

---

**Description**

run\_be\_simu est une methode intermediaire permettant de calcul un best estimate pour une simulation donnee.

**Usage**

```
run_be_simu(x, i, pre_on)
```

**Arguments**

x	est un objet de type Be.
i	est un entier (integer) correspondant au numero de la simulation.
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.

**Value**

Une liste contenant les noms de produits calcules, les flux par produits et par annee pour une simulation sous forme de matrice et les flux actualises par produits.

**Author(s)**

Prim'Act

---

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**

L'objet x mis a jour de l'operation de vente (suppression des lignes vendues).

**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**

L'objet x mis a jour de l'operation de vente (suppression des lignes vendues).

**Author(s)**

Prim'Act

---

selloblig	<i>Mise a jour de chaque composante du portefeuille obligation suite a la vente de tout ou partie de ce portefeuille.</i>
-----------	---

---

**Description**

selloblig 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**

```
selloblig(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**

L'objet x mis a jour de l'operation de vente (suppression des lignes vendues).

**Author(s)**

Prim'Act

---

sell_pvl_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\_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	objet de la classe Action (decrivant le portefeuille action en detention).
montant	reel de type numeric contient le montant de plus value latente que l'on souhaite realiser.

**Value**

L'objet x mis a jour de l'operation de vente (suppression des lignes vendues) et pmvr le montant de plus value realisees.

**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 <a href="#">Initialisation</a> .
---	--

**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

---

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

---

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

---

Treso	<i>La classe Tresor</i>
-------	-------------------------

---

**Description**

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**

Mettre le lien vers les methodes de la classe

---

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 Obligation.

**Value**

L'objet x dont les coupons courus ont ete mis a jour

**Author(s)**

Prim'Act

---

update\_dur\_det\_action *Mise a jour de chaque composante du portefeuille action suite a la vente de tout ou partie de ce portefeuille.*

---

**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 de chaque composante du portefeuille immobilier suite a la vente de tout ou partie de ce portefeuille.*

---

**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 duration 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 Obligation.

**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).
---	---

**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 suite a un vieillissement.

## Usage

```
update_PortFin(an, x, new_mp_ESG, flux_milieu, flux_fin)
```

## Arguments

an	numeric 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

Le format de la liste renvoyee est :

ptf : un vecteur contenant les flux de sortie en echeance de l'annee

revenu\_fin :

var\_vnc\_oblig :

## Author(s)

Prim'Act

## See Also

La fonction 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 suite a un vieillissement.

### 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 de la classe PortFin 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 Obligation.

**Value**

L'objet x dont les surcotes decotes ont ete mises a jour.

**Author(s)**

Prim'Act

---

update_treso	<i>Permet d'intégrer un flux (entrant ou sortant) au compte de trésorerie d'un Portefeuille financier.</i>
--------------	--

---

**Description**

update\_treso est une methode permettant d'integrer un flux au compte de trésorerie.

**Usage**

```
update_treso(x, flux)
```

**Arguments**

x	objet de la classe Tresor, correspondant a l'actif Trésorerie d'un assureur antérieur a integration d'un flux.
flux	est une numeric correspondant a un flux. Si il est positif, le flux est entrant. Si il est negatif, le flux est sortant.

**Value**

L'objet Tresor mis a jour du flux precise en input.

**Author(s)**

Prim'Act

---

update_vm_action	<i>Mise a jour de chaque composante du portefeuille action suite a la vente de tout ou partie de ce portefeuille.</i>
------------------	---

---

**Description**

update\_vm\_action est une methode permettant de mettre a jour la duree de detention 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).
---	--

**Value**

L'objet x mis a jour du vieillissement de la duree de detention.

**Author(s)**

Prim'Act

---

update_vm_immo	<i>Mise a jour de chaque composante du portefeuille immobilier suite a la vente de tout ou partie de ce portefeuille.</i>
----------------	---

---

**Description**

update\_vm\_immo est une methode permettant de mettre a jour la duree de detention 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).
---	--

**Value**

L'objet x mis a jour du vieillissement de la duree de detention.

**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 Obligation.

**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 Obligation.

### 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 zspreads d'un portefeuille obligataire.</i>
------------------	--

---

### Description

update\_zsp\_oblig est une methode permettant de mettre a jour les zspreads 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

---

vieillessement\_action\_PortFin

*Effectue le vieillissement/la projection du portefeuille action d'un portefeuille financier.*


---

### Description

vieillessement\_action\_PortFin est une methode permettant de projeter la composante action d'un portefeuille financier. suite a un vieillissement.

### Usage

```
vieillessement_action_PortFin(x, table_rdt)
```

### Arguments

x	objet de la classe PortFin, correspondant au portefeuille financier de l'assureur avant l'etape de vieillissement de son attribut ptf_action de la classe Action.
table_rdt	est une liste, construite par la fonction <a href="#">calc_rdt</a> . 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.

### Author(s)

Prim'Act

### See Also

La fonction de calcul des rendements des actifs [calc\\_rdt](#).

---

vieillessement\_av\_pb    *Vieillessement du portefeuille sur l'annee avant attribution de participation aux benefices.*


---

### 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 <a href="#">PortPassif</a> 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 : <b>le taux de rendement obligataire</b> <b>le taux de rendement de l'indice action de reference</b> <b>le taux de rendement de l'indice immobilier de reference</b> <b>le taux de rendement de l'indice tresorerie de reference</b>
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](#).

---

vieillissement\_immo\_PortFin

*Effectue le vieillissement/la projection 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 de son attribut ptf_immo de la classe Immo.
table_rdt	est une liste, construite par la fonction <a href="#">calc_rdt</a> . Cette table contient les tables d'evolution des cours et rendements sur l'annee consideree de chacune des classes d'actif. Les tables sont constituées a partir des extractions du Generateur de Scenario Economique de Prim'Act.

**Author(s)**

Prim'Act

**See Also**

La fonction de calcul des rendements des actifs [calc\\_rdt](#).

---

vieillissement\_oblig\_PortFin

*Effectue le vieillissement/la projection du portefeuille obligataire d'un portefeuille financier.*

---

**Description**

vieillissement\_oblig\_PortFin est une methode permettant de projeter la composante obligataire d'un portefeuille financier.

**Usage**

```
vieillissement_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 de son attribut ptf_oblig de la classe Oblig.
table_rdt	est une liste, construite par la fonction <a href="#">calc_rdt</a> . Cette table contient les tables d'evolution des cours et rendements sur l'annee consideree de chacune des classes d'actif. Les tables sont constituées a partir des extractions du Generateur de Scenario Economique de Prim'Act.

**Author(s)**

Prim'Act

**See Also**

La fonction de calcul des rendements des actifs [calc\\_rdt](#).

---

`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 obligatoire d'un portefeuille financier.

### Usage

`vieillissement_treso_PortFin(x, flux_milieu, flux_fin, table_rdt)`

### Arguments

<code>x</code>	objet de la classe <code>PortFin</code> , correspondant au portefeuille financier de l'assureur avant l'etape de vieillissement de son attribut <code>ptf_treso</code> de la classe <code>Treso</code> .
<code>table_rdt</code>	est une liste, construite par la fonction <a href="#">calc_rdt</a> . Cette table contient les tables d'evolution des cours et rendements sur l'annee consideree de chacune des classes d'actif. Les tables sont constiutes a partir des extractions du Generateur de Scenario Economique de Prim'Act.

### Author(s)

Prim'Act

### See Also

La fonction de calcul des rendements des actifs [calc\\_rdt](#).

---

`vieillissement_ap_pb`

*Vieillissement du portefeuille sur l'annee apres attribution de participation aux benefices.*

---

### Description

`vieillissement_ap_pb` est une methode permettant de calculer les PM et les flux sur une annee apres PB. Cette methode vieilli le portefeuille de passifs apres attribution de PB.

### Usage

`vieillissement_ap_pb(x, rev_nette_alloue, tx_soc)`

### Arguments

<code>x</code>	un objet de la classe <a href="#">PortPassif</a> contenant l'ensemble des produits de passifs.
<code>rev_nette_alloue</code>	un vecteur numeric contenant par produit le supplement de revalorisation par rapport au taux minimum.
<code>tx_soc</code>	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 peridoe.

**Usage**

```
vieilli_mp(x, pm_fin_ap_pb, tx_revalo)
```

**Arguments**

x	un objet de la classe <a href="#">EpEuroInd</a> contenant les model points epargne euros.
pm_fin_ap_pb	un vecteur de type numeric contenant par model point les montants de PM revalorises apres participation aux benefices.
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](#).

---

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>Calcul 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 de l'obligation correspondante du portefeuille obligataire. Ce vecteur a autant d'elements que le portefeuille obligataire a de lignes.

### Author(s)

Prim'Act

# Index

## \*Topic **classes**

AutresPassifs, 8  
AutresReserves, 8  
Be, 11  
Canton, 39  
EpEuroInd, 59  
ESG, 59  
FraisFin, 64  
FraisPassif, 65  
HypCanton, 71  
HypTech, 72  
Initialisation, 73  
ModelPointESG, 78  
ParamAlmEngine, 79  
ParamChocMket, 80  
ParamChocSousc, 80  
ParamComport, 80  
ParamRachDyn, 81  
ParamTableMort, 82  
PortFin, 84  
PortPassif, 85  
Ppb, 85  
PRE, 86  
RC, 90  
RevaloEngine, 92  
TabEpEuroInd, 99  
TauxPB, 99  
Treso, 100

Action, 7, 43, 45  
Action (buy\_action), 11  
Action (calc\_pmv1\_action), 20  
Action (calc\_vm\_action), 35  
Action (revalo\_action), 93  
Action (sell\_action), 96  
Action (sell\_pvl\_action), 98  
Action (update\_dur\_det\_action), 101  
Action (update\_vm\_action), 106  
AlmEngine, 7  
AlmEngine (create\_ptf\_bought\_action), 43  
AlmEngine (create\_ptf\_bought\_immo), 43  
AlmEngine (create\_ptf\_bought\_oblig), 44  
AlmEngine (do\_calc\_nb\_sold\_action), 45  
AlmEngine (do\_calc\_nb\_sold\_immo), 45  
AlmEngine (do\_calc\_nb\_sold\_oblig), 46  
AlmEngine (reallocate), 91  
autres\_passif\_load, 9  
autres\_reserves\_load, 9  
AutresPassifs, 8, 9, 85  
AutresPassifs (autres\_passif\_load), 9  
AutresPassifs  
    (proj\_annee\_autres\_passifs), 88  
AutresReserves, 8, 9, 10, 85  
AutresReserves (autres\_reserves\_load), 9  
AutresReserves (init\_debut\_pgg\_psap), 75  
AutresReserves (update\_reserves), 104

base\_prod\_fin, 10  
Be, 11, 73, 76  
Be (run\_be\_simu), 95  
Be (run\_be), 95  
Be (write\_be\_results), 114  
buy\_action, 7, 11  
buy\_immo, 12, 73  
buy\_oblig, 12, 79

calc\_coupon, 13  
calc\_dotation\_ppb, 13  
calc\_fin\_proj, 14, 39, 88  
calc\_flux\_annee, 15  
calc\_frais, 15, 65  
calc\_frais\_fin, 16, 88  
calc\_marge\_fin, 17  
calc\_nominal, 17  
calc\_pm, 18, 31, 59  
calc\_pmv1, 20  
calc\_pmv1\_action, 20  
calc\_pmv1\_immo, 21  
calc\_pmv1\_oblig, 21  
calc\_PRE, 22, 55, 88  
calc\_prest, 18, 19, 22, 59  
calc\_primes, 18, 19, 24, 59  
calc\_qx, 25, 69, 82  
calc\_rach, 25, 70, 82  
calc\_rach\_dyn, 26, 71, 81  
calc\_RC, 26, 56, 57  
calc\_rdt, 27, 109, 111, 112  
calc\_rdt\_marche\_ref, 27

- calc\_relavo\_pm, [59](#)
- calc\_reprise\_ppb, [28](#)
- calc\_result\_technique, [28](#), [88](#)
- calc\_result\_technique\_ap\_pb, [29](#), [39](#), [88](#)
- calc\_resultat\_fin, [88](#)
- calc\_revalo, [29](#), [88](#)
- calc\_revalo\_pm, [30](#), [113](#)
- calc\_sur\_dec, [31](#)
- calc\_tra, [32](#), [88](#)
- calc\_tx\_cible, [18](#), [19](#), [32](#), [59](#)
- calc\_tx\_cible\_ref\_marche, [33](#), [69](#), [81](#)
- calc\_tx\_min, [18](#), [19](#), [23](#), [24](#), [34](#), [59](#)
- calc\_tx\_sortie, [23](#), [24](#), [34](#), [59](#)
- calc\_vm\_action, [35](#)
- calc\_vm\_immo, [36](#)
- calc\_vmoblig, [36](#)
- calc\_vm\_treso, [37](#)
- calc\_vnc, [37](#)
- calc\_z\_spread, [38](#)
- Canton, [14](#), [29](#), [39](#), [74](#), [77](#), [87](#)
- Canton (calc\_fin\_proj), [14](#)
- Canton (calc\_result\_technique\_ap\_pb), [29](#)
- Canton (proj\_an), [87](#)
- changement\_choc, [39](#)
- changement\_ESG, [40](#)
- changement\_PortFin, [41](#)
- changement\_PortFin\_ref, [83](#)
- changement\_PortFin\_reference, [41](#)
- ChocSolvabilite2, [42](#)
- ChocSolvabilite2 (changement\_choc), [39](#)
- ChocSolvabilite2
  - (do\_choc\_action\_type1), [47](#)
- ChocSolvabilite2
  - (do\_choc\_action\_type2), [47](#)
- ChocSolvabilite2 (do\_choc\_frais), [48](#)
- ChocSolvabilite2 (do\_choc\_immo), [49](#)
- ChocSolvabilite2 (do\_choc\_longevite), [49](#)
- ChocSolvabilite2 (do\_choc\_mortalite), [50](#)
- ChocSolvabilite2 (do\_choc\_rachat\_down), [51](#)
- ChocSolvabilite2 (do\_choc\_rachat\_up), [51](#)
- ChocSolvabilite2
  - (do\_choc\_spread\_unitaire), [53](#)
- ChocSolvabilite2 (do\_choc\_spread), [52](#)
- ChocSolvabilite2 (do\_choc\_taux), [53](#)
- create\_ptf\_bought\_action, [43](#)
- create\_ptf\_bought\_immo, [43](#)
- create\_ptf\_boughtoblig, [44](#)
- do\_calc\_nb\_sold\_action, [45](#)
- do\_calc\_nb\_sold\_immo, [45](#)
- do\_calc\_nb\_soldoblig, [46](#)
- do\_choc\_action, [42](#)
- do\_choc\_action\_type1, [47](#)
- do\_choc\_action\_type2, [47](#)
- do\_choc\_frais, [42](#), [48](#)
- do\_choc\_immo, [42](#), [49](#)
- do\_choc\_longevite, [42](#), [49](#)
- do\_choc\_mortalite, [42](#), [50](#)
- do\_choc\_rachat\_down, [42](#), [51](#)
- do\_choc\_rachat\_up, [42](#), [51](#)
- do\_choc\_spread, [42](#), [52](#), [53](#)
- do\_choc\_spread\_unitaire, [53](#)
- do\_choc\_taux, [53](#)
- do\_update\_pmv1, [54](#), [88](#)
- do\_update\_PRE\_val\_courante, [54](#), [88](#)
- do\_update\_PRE\_val\_debut, [55](#), [88](#)
- do\_update\_RC\_val\_courante, [56](#)
- do\_update\_RC\_val\_debut, [56](#), [88](#)
- do\_update\_vm\_vnc\_precedent, [57](#), [88](#)
- duration\_sensi, [57](#)
- echeancier, [58](#)
- EpEuroInd, [14](#), [18](#), [23](#), [24](#), [30](#), [32](#), [34](#), [35](#), [59](#), [85](#), [89](#), [90](#), [99](#), [113](#)
- EpEuroInd (calc\_pm), [18](#)
- EpEuroInd (calc\_prest), [22](#)
- EpEuroInd (calc\_primes), [24](#)
- EpEuroInd (calc\_revalo\_pm), [30](#)
- EpEuroInd (calc\_tx\_cible), [32](#)
- EpEuroInd (calc\_tx\_min), [34](#)
- EpEuroInd (calc\_tx\_sortie), [34](#)
- EpEuroInd (vieilli\_mp), [113](#)
- ESG, [59](#), [66](#)
- ESG (changement\_ESG), [40](#)
- ESG (extract\_ESG), [60](#)
- ESG (get\_choc\_inflation\_frais), [66](#)
- extract\_ESG, [60](#)
- finance\_cible\_pmv1, [60](#)
- finance\_cible\_ppb, [61](#)
- finance\_contrainte\_legale, [62](#)
- finance\_tmg, [63](#)
- frais\_fin\_load, [65](#)
- frais\_passif\_load, [66](#)
- FraisFin, [64](#)
- FraisFin (calc\_frais\_fin), [16](#)
- FraisFin (frais\_fin\_load), [65](#)
- FraisPassif, [14](#), [15](#), [65](#), [66](#), [85](#), [90](#)
- FraisPassif (calc\_frais), [15](#)
- FraisPassif (frais\_passif\_load), [66](#)
- get\_choc\_inflation\_frais, [66](#)
- get\_choc\_rach, [67](#), [72](#)
- get\_choc\_table, [68](#), [72](#)
- get\_comport, [33](#), [68](#), [72](#)

- get\_qx\_mort, 35, 69, 72
- get\_qx\_rach, 35, 70, 72
- get\_rach\_dyn, 35, 70, 72
  
- HypCanton, 39, 71
- HypTech, 32, 35, 67–71, 72, 77, 85
- HypTech (get\_choc\_rach), 67
- HypTech (get\_choc\_table), 68
- HypTech (get\_comport), 68
- HypTech (get\_qx\_mort), 69
- HypTech (get\_qx\_rach), 70
- HypTech (get\_rach\_dyn), 70
- HypTech (load\_ht), 77
  
- Immo, 44, 46, 72
- Immo (buy\_immo), 12
- Immo (calc\_pmv1\_immo), 21
- Immo (calc\_vm\_immo), 36
- Immo (revalo\_immo), 93
- Immo (sell\_immo), 97
- Immo (update\_dur\_det\_immo), 101
- Immo (update\_vm\_immo), 107
- init\_create\_folder, 74
- init\_debut\_pgg\_psap, 8, 75, 88
- init\_debut\_ppb, 75, 88
- init\_scenario, 76
- init\_SimBEL, 73, 76
- Initialisation, 9, 10, 40, 65, 66, 73, 74–78, 98, 99
- Initialisation (init\_SimBEL), 76
- Initialisation (init\_create\_folder), 74
- Initialisation (init\_scenario), 76
- Initialisation (initialisation\_load), 74
- Initialisation (set\_architecture), 98
- initialisation\_load, 74
  
- load\_ht, 77
- load\_pp, 78
  
- ModelPoint\_ESG, 60
- ModelPointESG, 39, 78
  
- Oblig, 44, 46, 47, 79
- Oblig (buy\_oblig), 12
- Oblig (calc\_coupon), 13
- Oblig (calc\_flux\_annee), 15
- Oblig (calc\_nominal), 17
- Oblig (calc\_pmv1\_oblig), 21
- Oblig (calc\_sur\_dec), 31
- Oblig (calc\_vm\_oblig), 36
- Oblig (calc\_vnc), 37
- Oblig (calc\_z\_spread), 38
- Oblig (duration\_sensi), 57
- Oblig (echeancier), 58
- Oblig (sell\_oblig), 97
- Oblig (update\_cc\_oblig), 100
- Oblig (update\_dur\_oblig), 102
- Oblig (update\_mat\_res), 102
- Oblig (update\_sd\_oblig), 105
- Oblig (update\_vm\_oblig), 107
- Oblig (update\_vnc\_oblig), 108
- Oblig (update\_zsp\_oblig), 108
- Oblig (yield\_to\_maturity), 114
  
- param\_alm\_engine\_load, 79, 83
- ParamAlmEngine, 39, 79
- ParamAlmEngine (param\_alm\_engine\_load), 83
- ParamBe, 79
- ParamChocMket, 40, 80
- ParamChocSousc, 40, 80
- ParamComport, 33, 72, 80
- ParamComport (calc\_tx\_cible\_ref\_marche), 33
- ParamRachDyn, 26, 72, 81
- ParamRachDyn (calc\_rach\_dyn), 26
- ParamRevaloEngine, 39, 82
- ParamTableMort, 25, 72, 82
- ParamTableMort (calc\_qx), 25
- ParamTableRach, 25, 72
- ParamTableRach (calc\_rach), 25
- pb\_contr, 83
- PortFin, 39, 84, 91
- PortFin (calc\_pmv1), 20
- PortFin (calc\_rdt), 27
- PortFin (calc\_tra), 32
- PortFin (changement\_PortFin\_reference), 41
- PortFin (changement\_PortFin), 41
- PortFin (do\_update\_pmv1), 54
- PortFin (do\_update\_vm\_vnc\_precedent), 57
- PortFin (print\_alloc), 87
- PortFin (resultat\_fin), 92
- PortFin (update\_PortFin\_reference), 104
- PortFin (update\_PortFin), 103
- PortFin (vieillissement\_action\_PortFin), 109
- PortFin (vieillissement\_immo\_PortFin), 110
- PortFin (vieillissement\_oblig\_PortFin), 111
- PortFin (vieillissement\_treso\_PortFin), 112
- PortPassif, 39, 78, 85, 89, 109, 110, 112
- PortPassif (calc\_rdt\_marche\_ref), 27

- PortPassif (load\_pp), 78
- PortPassif (proj\_annee\_av\_pb), 89
- PortPassif (vieillissement\_av\_pb), 109
- PortPassif (vieillissement\_ap\_pb), 112
- PortPassifs, 78
- Ppb, 29, 39, 64, 85
- Ppb (calc\_dotation\_ppb), 13
- Ppb (calc\_reprise\_ppb), 28
- Ppb (init\_debut\_ppb), 75
- PRE, 86
- PRE (calc\_PRE), 22
- PRE (do\_update\_PRE\_val\_courante), 54
- PRE (do\_update\_PRE\_val\_debut), 55
- PRE (pre\_load), 86
- pre\_load, 86
- print\_alloc, 87, 88
- proj\_an, 39, 87
- proj\_annee\_autres\_passifs, 8, 88, 110
- proj\_annee\_av\_pb, 85, 89, 110
  
- RC, 90
- RC (calc\_RC), 26
- RC (do\_update\_RC\_val\_courante), 56
- RC (do\_update\_RC\_val\_debut), 56
- RC (rc\_load), 90
- rc\_load, 90
- reallocate, 7, 88, 91
- resultat\_fin, 92
- revalo\_action, 93
- revalo\_immo, 93
- revalo\_treso, 94
- RevaloEngine, 92
- RevaloEngine (base\_prod\_fin), 10
- RevaloEngine (calc\_marge\_fin), 17
- RevaloEngine (calc\_result\_technique), 28
- RevaloEngine (calc\_revalo), 29
- RevaloEngine (finance\_cible\_pmv1), 60
- RevaloEngine (finance\_cible\_ppb), 61
- RevaloEngine
  - (finance\_contrainte\_legale), 62
- RevaloEngine (finance\_tmng), 63
- RevaloEngine (pb\_contr), 83
- revenu\_treso, 94
- run\_be, 95
- run\_be\_simu, 95
  
- sell\_action, 7, 96
- sell\_immo, 73, 97
- sell\_oblig, 79, 97
- sell\_pvl\_action, 7, 88, 98
- set\_architecture, 9, 10, 40, 41, 65, 66, 73, 75, 77, 78, 98
  
- TabEpEuroInd, 59, 99
- TauxPB, 85, 99
- Treso, 100
- Treso (calc\_vm\_treso), 37
- Treso (revalo\_treso), 94
- Treso (revenu\_treso), 94
- Treso (update\_treso), 106
  
- update\_cc\_oblig, 100
- update\_dur\_det\_action, 101
- update\_dur\_det\_immo, 101
- update\_dur\_oblig, 102
- update\_mat\_res, 102
- update\_PortFin, 88, 103, 104
- update\_PortFin\_reference, 88, 103, 104
- update\_reserves, 8, 104, 110
- update\_sd\_oblig, 105
- update\_treso, 106
- update\_vm\_action, 106
- update\_vm\_immo, 107
- update\_vm\_oblig, 107
- update\_vnc\_oblig, 108
- update\_zsp\_oblig, 108
  
- vieilli\_mp, 59, 113, 113
- vieillissement\_action\_PortFin, 109
- vieillissement\_ap\_pb, 85
- vieillissement\_av\_pb, 85, 109
- vieillissement\_immo\_PortFin, 110
- vieillissement\_oblig\_PortFin, 111
- vieillissement\_treso\_PortFin, 112
- vieillissement\_ap\_pb, 112
- viellissement\_ap\_pb, 29, 88
- viellissement\_av\_pb, 29, 88
  
- write\_be\_results, 114
  
- yield\_to\_maturity, 114