

Package ‘SiALM’

March 24, 2018

Type Package

Title Evaluation des engagements Best-Estimate pour des contrats d'assurance-vie.

Version 0.1.0

Author Damien Tichit

Maintainer Damien Tichit <damien.tichit@sia-partners.com>

Description SiALM est un modele ALM permettant d'evaluer les engagements best-estimate pour une compagnie d'assurance.
Il permet de modeliser des contrats epargne.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends R (>= 3.4.4), doParallel(>= 1.0.11), dplyr(>= 0.7.4)

Collate 'HypALM-class.R' 'PPE-class.R' 'Immobilier-class.R'
'Obligation-class.R' 'Action-class.R' 'PTFCible-class.R'
'HypActif-class.R' 'Tresorerie-class.R' 'PTFActif-class.R'
'Actif-class.R' 'ReserveCapi-class.R' 'Provision-class.R'
'RachatConj-class.R' 'TabRachat-class.R' 'TabMorta-class.R'
'HypPassif-class.R' 'Epargne-class.R' 'PTFPassif-class.R'
'Passif-class.R' 'System-class.R' 'ALM-class.R'
'ALM-aggregagtion.R' 'Actif-proj_1an.R' 'Passif-proj_1an.R'
'System-proj_1an.R' 'ALM-calc_be.R' 'HypALM-load.R'
'PPE-load.R' 'ReserveCapi-load.R' 'Provision-load.R'
'TabMorta-load.R' 'HypPassif-load.R' 'Epargne-load.R'
'PTFPassif-load.R' 'Passif-load.R' 'Immobilier-load.R'
'Tresorerie-load.R' 'Obligation-load.R' 'Action-load.R'
'PTFActif-load.R' 'PTFCible-load.R' 'HypActif-load.R'
'Actif-load.R' 'System-load.R' 'ALM-load.R'
'Actif-aggregagtion.R' 'Actif-rebalancement.R'
'Action-aggregagtion.R' 'Action-eval_dividende.R'
'Action-load_cible.R' 'Action-rebalancement.R'
'Action-revalo.R' 'Epargne-aggregagtion.R' 'Epargne-proj_1an.R'
'Immobilier-aggregagtion.R' 'Immobilier-eval_loyer.R'
'Immobilier-load_cible.R' 'Immobilier-proj_1an.R'
'Immobilier-rebalancement.R' 'Immobilier-revalo.R'
'Obligation-aggregagtion.R' 'Obligation-eval_coupon.R'

'Obligation-load_cible.R' 'Obligation-rebalancement.R'
 'Obligation-revalo.R' 'Obligation-veillissement.R'
 'PPE-dotation.R' 'PPE-reprise.R' 'PPE-reprise_ppe_8ans.R'
 'PPE-veillissement.R' 'PTFActif-allocation.R'
 'PTFActif-calc_prod_fin.R' 'PTFActif-eval_frais_fin.R'
 'PTFActif-eval_prod_fin.R' 'PTFActif-revalo.R'
 'PTFActif-veillissement.R' 'PTFPassif-proj_1an.R'
 'Passif-aggregagtion.R' 'Passif-revalo.R' 'RachatConj-calc.R'
 'RachatConj-load.R' 'SiALM.R' 'System-calcul_pb.R'
 'TabMorta-calc_qx.R' 'TabRachat-calc_rx.R' 'TabRachat-load.R'
 'fonctions_utiles.R'

R topics documented:

Actif	4
Action	4
aggregation_actif	4
aggregation_action	5
aggregation_alm	5
aggregation_epargne	6
aggregation_immobilier	6
aggregation_obligation	7
aggregation_passif	7
allocation_ptf_actif	8
ALM	8
calcul_pb	9
calc_be	9
calc_prod_fin	10
calc_qx	10
calc_rachat_conj	11
calc_rx	11
dotation_ppe	12
Epargne	12
eval_coupon	13
eval_dividende	13
eval_frais_fin	14
eval_loyer	14
eval_prod_fin	15
HypActif	15
HypALM	15
HypPassif	16
Immobilier	16
load_actif	17
load_action	17
load_action_cible	18
load_alm	18
load_epargne	19
load_hyp_actif	19
load_hyp_alm	20
load_hyp_passif	20
load_immobilier	21
load_immobilier_cible	22

load_obligation	22
load_obligation_cible	23
load_passif	23
load_ppe	24
load_provision	24
load_ptf_actif	25
load_ptf_cible	26
load_ptf_passif	26
load_rachat_conj	27
load_reserve_capi	27
load_system	28
load_tab_morta	29
load_tab_rachat	29
load_tresorerie	30
Obligation	30
Passif	31
PPE	31
proj_1an_actif	31
proj_1an_epargne	32
proj_1an_immobilier	32
proj_1an_passif	33
proj_1an_ptf_passif	33
proj_1an_system	34
Provision	34
PTFActif	35
PTFCible	35
PTFPassif	35
RachatConj	36
rebalancement_actif	36
rebalancement_action	37
rebalancement_immobilier	37
rebalancement_obligation	38
reprise_ppe	38
reprise_ppe_8ans	39
ReserveCapi	39
revalo_action	40
revalo_immobilier	40
revalo_obligation	41
revalo_passif	41
revalo_ptf_actif	42
SiALM	42
System	42
TabMorta	43
TabRachat	43
Tresorerie	43
vieillissement_obligation	44
vieillissement_ppe	44
vieillissement_ptf_actif	45

Actif	<i>Classe Actif</i>
-------	---------------------

Description

Cette classe agregge l'ensemble des donnees relatives a l'actif de la compagnie d'assurance : portefeuille, hypotheses

Slots

ptf_actif est un objet de la classe [PTFActif](#) representant le portefeuille financier.

hyp_actif est un objet de la classe [HypActif](#) representant les hypotheses du portefeuille financier.

Author(s)

Damien Tichit pour Sia Partners

Action	<i>Classe Action</i>
--------	----------------------

Description

Cette classe represente le portefeuille des actions de la compagnie d'assurance.

Slots

ptf est un objet de type [data.frame](#) contenant les donnees relatives au portefeuille.

Author(s)

Damien Tichit pour Sia Partners

aggregation_actif	<i>Fonction aggregation_actif.</i>
-------------------	------------------------------------

Description

Cette fonction permet de faire appel aux différentes fonctions permettant d'aggreger les portefeuilles financiers d'une compagnie d'assurance.

Usage

```
aggregation_actif(actif)
```

Arguments

actif est un objet de type [Actif](#).

Author(s)

Damien Tichit pour Sia Partners

aggregation_action	<i>Fonction</i> aggregation_action.
--------------------	-------------------------------------

Description

Cette fonction permet d'aggrer les model-point pour un portefeuille d'actions.

Usage

```
aggregation_action(action)
```

Arguments

action est un objet de type [Action](#).

Author(s)

Damien Tichit pour Sia Partners

aggregation_alm	<i>Fonction</i> aggregation_alm.
-----------------	----------------------------------

Description

Cette fonction permet de faire appel aux différentes fonctions permettant d'aggrer les portefeuilles.

Usage

```
aggregation_alm(alm)
```

Arguments

alm est un objet de type [ALM](#).

Author(s)

Damien Tichit pour Sia Partners

aggregation_epargne	<i>Fonction</i> aggregation_epargne.
---------------------	--------------------------------------

Description

Cette fonction permet d'aggreger les model-point pour un portefeuille d'actions.

Usage

```
aggregation_epargne(epargne)
```

Arguments

epargne est un objet de type [Epargne](#).

Author(s)

Damien Tichit pour Sia Partners

aggregation_immobilier	<i>Fonction</i> aggregation_immobilier.
------------------------	---

Description

Cette fonction permet d'aggreger les model-point pour un portefeuille d'actions.

Usage

```
aggregation_immobilier(immobilier)
```

Arguments

immobilier est un objet de type [Immobilier](#).

Author(s)

Damien Tichit pour Sia Partners

`aggregation_obligation`*Fonction aggregation_obligation.*

Description

Cette fonction permet d'aggrer les model-point pour un portefeuille d'obligations.

Usage

```
aggregation_obligation(obligation)
```

Arguments

`obligation` est un objet de type [Obligation](#).

Author(s)

Damien Tichit pour Sia Partners

`aggregation_passif`*Fonction aggregation_passif.*

Description

Cette fonction permet de faire appel aux différentes fonctions permettant d'aggrer les portefeuilles passifs d'une compagnie d'assurance.

Usage

```
aggregation_passif(passif)
```

Arguments

`passif` est un objet de type [Passif](#).

Author(s)

Damien Tichit pour Sia Partners

allocation_ptf_actif	<i>Fonction allocation_ptf_actif</i>
----------------------	--------------------------------------

Description

Cette fonction permet de determiner l'allocation des differents produits financiers.

Usage

```
allocation_ptf_actif(ptf_actif)
```

Arguments

ptf_actif est un objet de type PTFActif.

Author(s)

Damien Tichit pour Sia Partners

ALM

<i>Classe ALM</i>

Description

Cette classe aggrege un objet [System](#) ainsi que l'ensemble des hypotheses du modele ALM.

Slots

system est un objet de la classe [System](#) qui aggrecent les actifs et les passifs ainsi que leurs hypotheses.

hyp_alm est un objet de la classe [HypALM](#) qui contient les differentes hypotheses du modele ALM.

Author(s)

Damien Tichit pour Sia Partners

calcul_pb	<i>Fonction calcul_pb.</i>
-----------	----------------------------

Description

Cette fonction permet de determiner la PB a distribuer.

Usage

```
calcul_pb(taux_pb, resultat_fin, resultat_tech)
```

Arguments

taux_pb	est une list contenant les deux taux de pb contractuels.
resultat_fin	est une list contenant les resultats financiers : PMVL, produits financiers.
resultat_tech	est une list contenant le resultat technique : chargements

Author(s)

Damien Tichit pour Sia Partners

calc_be	<i>Fonction calc_be.</i>
---------	--------------------------

Description

Cette fonction est une fonction centrale du package. Elle permet en effet de calculer un BEL.

Usage

```
calc_be(alm, parallel = FALSE, nb_core = 1L)
```

Arguments

alm	est un objet de type ALM contenant l'ensemble des donnees.
parallel	est une valeur logical. Lorsque cet argument est a TRUE, les calculs sont parallelises.
nb_core	est une valeur integer qui indique le nombre de coeurs utilises lorsque les calculs sont parallelises.

Details

C'est sur cette fonction que s'effectue les boucles sur le nombre de simulations ainsi que sur les annees.

Il est possible paralleliser les calculs afin d'accelerer le calcul d'un best-estimate.

Author(s)

Damien Tichit pour Sia Partners

See Also

Projection sur une annee d'un [System](#) : [proj_1an_system](#).

calc_prod_fin	<i>Fonction calc_prod_fin.</i>
---------------	--------------------------------

Description

Cette fonction permet de calculer les produits financiers pour un portefeuille financier : dividendes, coupons....

Usage

```
calc_prod_fin(ptf_actif)
```

Arguments

ptf_actif est un objet de type [PTFActif](#).

Author(s)

Damien Tichit pour Sia Partners

calc_qx	<i>Fonction calc_qx.</i>
---------	--------------------------

Description

Cette fonction permet de calculer des probas de deces pour un age et une table de mortalite donnees. Il est possible d'indiquer plusieurs ages differents sous forme de vecteur.

Usage

```
calc_qx(tab_morta, age)
```

Arguments

tab_morta est un objet de type [TabMorta](#).
age est un integer.

Author(s)

Damien Tichit pour Sia Partners

calc_rachat_conj	<i>Fonction calc_rachat_conj.</i>
------------------	-----------------------------------

Description

Cette fonction permet de calculer les taux de rachats conjoncturels.

Usage

```
calc_rachat_conj(rachat_conj, tx_cible, tx_serv)
```

Arguments

rachat_conj	est un objet de type RachatConj .
tx_cible	est un numeric.
tx_serv	est un numeric.

Author(s)

Damien Tichit pour Sia Partners

calc_rx	<i>Fonction calc_rx.</i>
---------	--------------------------

Description

Cette fonction permet de calculer des probas de rachat pour une anciennete et une table de rachat donnees. Il est possible d'indiquer plusieurs ages differents sous forme de vecteur.

Usage

```
calc_rx(tab_rachat, anc)
```

Arguments

tab_rachat	est un objet de type TabRachat .
anc	est un integer.

Author(s)

Damien Tichit pour Sia Partners

dotation_ppe	<i>Fonction</i> dotation_ppe.
--------------	-------------------------------

Description

Cette fonction permet de doter la PPE. Le montant est dote sur la 1ere annee de reserve.

Usage

```
dotation_ppe(ppe, montant)
```

Arguments

ppe	est un objet de type PPE .
montant	est un numeric representant le montant a doter.

Author(s)

Damien Tichit pour Sia Partners

Epargne	<i>Classe</i> Epargne
---------	-----------------------

Description

Cette classe represente le portefeuille des contrats epargne de la compagnie d'assurance.

Slots

ptf est un objet de type [data.frame](#) contenant les donnees relatives au portefeuille.

Author(s)

Damien Tichit pour Sia Partners

eval_coupon	<i>Fonction eval_coupon</i>
-------------	-----------------------------

Description

Cette fonction permet de calculer les coupons pour un portefeuille obligataire.

Usage

```
eval_coupon(obligation)
```

Arguments

obligation est un objet de type [Obligation](#).

Author(s)

Damien Tichit pour Sia Partners

eval_dividende	<i>Fonction eval_dividende</i>
----------------	--------------------------------

Description

Cette fonction permet de calculer les dividendes pour un portefeuille d'actions.

Usage

```
eval_dividende(action)
```

Arguments

action est un objet de type [Action](#).

Author(s)

Damien Tichit pour Sia Partners

eval_frais_fin	<i>Fonction eval_frais_fin</i>
----------------	--------------------------------

Description

Cette fonction permet d'évaluer les frais relatifs au portefeuille actif d'une compagnie d'assurance.

Usage

```
eval_frais_fin(ptf_actif, hyp_actif)
```

Arguments

ptf_actif est un objet de type [PTFActif](#).

hyp_actif est un objet de type [HypActif](#).

Author(s)

Damien Tichit pour Sia Partners

eval_loyer	<i>Fonction eval_loyer</i>
------------	----------------------------

Description

Cette fonction permet de calculer les loyers pour un portefeuille d'immobiliers.

Usage

```
eval_loyer(immobilier)
```

Arguments

immobilier est un objet de type [Immobilier](#).

Author(s)

Damien Tichit pour Sia Partners

eval_prod_fin	Fonction eval_prod_fin.
---------------	-------------------------

Description

Cette fonction permet d'évaluer les produits financiers pour les différentes classes d'actifs.

Usage

```
eval_prod_fin(ptf_actif)
```

Arguments

ptf_actif est un objet de type [PTFActif](#).

Author(s)

Damien Tichit pour Sia Partners

HypActif	Classe HypActif
----------	-----------------

Description

Cette classe agrège l'ensemble des hypothèses relatives au passif de la compagnie d'assurance : portefeuille cible

Slots

ptf_cible est un objet de la classe [PTFCible](#) représentant le portefeuille financier cible.

Author(s)

Damien Tichit pour Sia Partners

HypALM	Classe HypALM
--------	---------------

Description

Cette classe agrège l'ensemble des hypothèses relatives au modèle ALM.

Slots

nb_simu est un entier représentant le nombre de simulation souhaité pour calculer le BEL.

an_proj est un entier représentant le nombre d'années de projection.

tsr est un numeric contenant les taux sans risque avec Volatility Adjustment.

Author(s)

Damien Tichit pour Sia Partners

HypPassif	<i>Classe HypPassif</i>
-----------	-------------------------

Description

Cette classe agregge l'ensemble des hypotheses relatives au passif d'une compagnie d'assurance : tables de rachat, de mortalite...

Slots

tab_morta_h est un objet de la classe [TabMorta](#) contenant la table de mortalite pour les hommes.

tab_morta_f est un objet de la classe [TabMorta](#) contenant la table de mortalite pour les femmes.

tab_rachat_tot est un objet de la classe [TabRachat](#) contenant la table modelisant les rachats totaux.

tab_rachat_part est un objet de la classe [TabRachat](#) contenant la table modelisant les rachats partiels.

rachat_conj est un objet de la classe [RachatConj](#) contenant les parametres modelisant les rachats conjoncturels

Author(s)

Damien Tichit pour Sia Partners

Immobilier	<i>Classe Immobilier</i>
------------	--------------------------

Description

Cette classe represente le portefeuille des immobilier de la compagnie d'assurance.

Slots

ptf est un objet de type [data.frame](#) contenant les donnees relatives au portefeuille.

Author(s)

Damien Tichit pour Sia Partners

load_actif	<i>Fonction load_actif.</i>
------------	-----------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [Actif](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_actif(address)
```

Arguments

address	est un objet de type <code>character</code> indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	---

Details

La creation d'un objet [Actif](#) necessite la creation de deux autres objets : [PTFActif](#), [HypActif](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [PTFActif](#) : [load_ptf_actif](#).

Construction d'un objet de type [HypActif](#) : [load_hyp_actif](#).

load_action	<i>Fonction load_action.</i>
-------------	------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [Action](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_action(address)
```

Arguments

address	est un objet de type <code>character</code> indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	---

Details

La creation d'un objet [Action](#) necessite des donnees presentes dans un fichier nomme `Actions.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_action_cible	<i>Fonction load_action_cible.</i>
-------------------	------------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [Action](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM. Cette fonction sera utilisee pour le chargement du portefeuille cible.

Usage

```
load_action_cible(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [Action](#) necessite des donnees presentes dans un fichier nomme `Actions.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_alm	<i>Fonction load_alm.</i>
----------	---------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [ALM](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_alm(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaire pour la construction de l'objet.
---------	---

Details

La creation d'un objet [ALM](#) necessite la creation de deux autres objets : [System](#) et [HypALM](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [System](#) : [load_system](#).

Construction d'un objet de type [HypALM](#) : [load_hyp_alm](#).

load_epargne	<i>Fonction</i> load_epargne.
--------------	-------------------------------

Description

Cette fonction permet de charger les données pour un objet de type [Epargne](#). Les données auront été préalablement disposées dans une architecture propre à SiALM.

Usage

```
load_epargne(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des données nécessaires pour la construction de l'objet.
---------	--

Details

La création d'un objet [Epargne](#) nécessite des données présentes dans un fichier nommé `Epargne.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_hyp_actif	<i>Fonction</i> load_hyp_actif.
----------------	---------------------------------

Description

Cette fonction permet de charger les données pour un objet de type [HypActif](#). Les données auront été préalablement disposées dans une architecture propre à SiALM.

Usage

```
load_hyp_actif(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des données nécessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [HypActif](#) necessite la creation d'un autre objet : [PTFCible](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [PTFCible](#) : [load_ptf_cible](#).

load_hyp_alm	<i>Fonction</i> load_hyp_alm.
--------------	-------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [HypALM](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_hyp_alm(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [HypALM](#) necessite plusieurs parametres presents dans un fichier nomme Hypotheses_ALM.csv.

Author(s)

Damien Tichit pour Sia Partners

load_hyp_passif	<i>Fonction</i> load_hyp_passif.
-----------------	----------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [HypPassif](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_hyp_passif(address)
```

Arguments

address est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.

Details

La creation d'un objet [HypPassif](#) necessite la creation de deux autres objets differents : [TabMorta](#) et [TabRachat](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [TabMorta](#) : [load_tab_morta](#).

Construction d'un objet de type [TabRachat](#) : [load_tab_rachat](#).

load_immobilier	<i>Fonction load_immobilier.</i>
-----------------	----------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [Immobilier](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_immobilier(address)
```

Arguments

address est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.

Details

La creation d'un objet [Immobilier](#) necessite des donnees presentes dans un fichier nomme `Immobilier.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_immobilier_cible *Fonction load_immobilier_cible.*

Description

Cette fonction permet de charger les donnees pour un objet de type [Immobilier](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_immobilier_cible(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [Immobilier](#) necessite des donnees presentes dans un fichier nomme `Immobilier.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_obligation *Fonction load_obligation.*

Description

Cette fonction permet de charger les donnees pour un objet de type [Obligation](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_obligation(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [Obligation](#) necessite des donnees presentes dans un fichier nomme `Obligations.csv`.
Cette fonction permet egalement d'aggreger les donnees : 1 obligation par maturite residuelle.

Author(s)

Damien Tichit pour Sia Partners

load_obligation_cible *Fonction load_obligation_cible.*

Description

Cette fonction permet de charger les données pour un objet de type [Obligation](#). Les données auront été préalablement disposées dans une architecture propre à SiALM. Cette fonction sera utilisée pour le chargement du portefeuille cible.

Usage

```
load_obligation_cible(address)
```

Arguments

address	est un objet de type <code>character</code> indiquant le dossier dans lequel se situe l'ensemble des données nécessaires pour la construction de l'objet.
---------	---

Details

La création d'un objet [Obligation](#) nécessite des données présentes dans un fichier nommé `Obligations.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_passif *Fonction load_passif.*

Description

Cette fonction permet de charger les données pour un objet de type [Passif](#). Les données auront été préalablement disposées dans une architecture propre à SiALM.

Usage

```
load_passif(address)
```

Arguments

address	est un objet de type <code>character</code> indiquant le dossier dans lequel se situe l'ensemble des données nécessaires pour la construction de l'objet.
---------	---

Details

La création d'un objet [Passif](#) nécessite la création de deux autres objets : [Passif](#), [Provision](#) et [HypPassif](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [PTFPassif](#) : [load_ptf_passif](#).

Construction d'un objet de type [HypPassif](#) : [load_hyp_passif](#).

Construction d'un objet de type [Provision](#) : [load_provision](#).

load_ppe

Fonction load_ppe.

Description

Cette fonction permet de charger les donnees pour un objet de type [PPE](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

load_ppe(address)

Arguments

address est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.

Details

La creation d'un objet [PPE](#) necessite des donnees presentes dans les fichiers nommes PPE.csv et TM-H.csv.

Author(s)

Damien Tichit pour Sia Partners

load_provision

Fonction load_provision.

Description

Cette fonction permet de charger les donnees pour un objet de type [Provision](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

load_provision(address)

Arguments

address est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.

Details

La creation d'un objet [Provision](#) necessite la creation de deux autres objets : [PPE](#) et [ReserveCapi](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [PPE](#) : [load_ppe](#).

Construction d'un objet de type [ReserveCapi](#) : [load_reserve_capi](#).

load_ptf_actif	<i>Fonction</i> load_ptf_actif.
----------------	---------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [Actif](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_ptf_actif(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [Actif](#) necessite la creation de trois autres objets : [Action](#), [Obligation](#), [Immobilier](#) et [Tresorerie](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [Action](#) : [load_action](#).

Construction d'un objet de type [Obligation](#) : [load_obligation](#).

Construction d'un objet de type [Tresorerie](#) : [load_tresorerie](#).

Construction d'un objet de type [Immobilier](#) : [load_immobilier](#).

load_ptf_cible	<i>Fonction load_ptf_cible.</i>
----------------	---------------------------------

Description

Cette fonction permet de charger les données pour un objet de type [PTFCible](#). Les données auront été préalablement disposées dans une architecture propre à SiALM.

Usage

```
load_ptf_cible(address)
```

Arguments

address	est un objet de type <code>character</code> indiquant le dossier dans lequel se situe l'ensemble des données nécessaires pour la construction de l'objet.
---------	---

Details

La création d'un objet [PTFCible](#) nécessite la création de trois autres objets : [Action](#), [Obligation](#) et [Immobilier](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [Action](#) : [load_action_cible](#).

Construction d'un objet de type [Obligation](#) : [load_obligation_cible](#).

Construction d'un objet de type [Immobilier](#) : [load_immobilier_cible](#).

load_ptf_passif	<i>Fonction load_ptf_passif.</i>
-----------------	----------------------------------

Description

Cette fonction permet de charger les données pour un objet de type [PTFPassif](#). Les données auront été préalablement disposées dans une architecture propre à SiALM.

Usage

```
load_ptf_passif(address)
```

Arguments

address	est un objet de type <code>character</code> indiquant le dossier dans lequel se situe l'ensemble des données nécessaires pour la construction de l'objet.
---------	---

Details

La creation d'un objet [PTFPassif](#) necessite la creation d'un autre objet : [Epargne](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [Epargne](#) : [load_epargne](#).

load_rachat_conj	<i>Fonction</i> load_rachat_conj.
------------------	-----------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [RachatConj](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_rachat_conj(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [RachatConj](#) necessite des donnees presentes dans un fichier nommes Rachats_conjoncturels.c

Author(s)

Damien Tichit pour Sia Partners

load_reserve_capi	<i>Fonction</i> load_reserve_capi.
-------------------	------------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [ReserveCapi](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_reserve_capi(address)
```

Arguments

address est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.

Details

La creation d'un objet [ReserveCapi](#) necessite des donnees presentes dans les fichiers nommes `reserve_capi.csv`.

Author(s)

Damien Tichit pour Sia Partners

load_system	<i>Fonction</i> load_system.
-------------	------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [System](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_system(address)
```

Arguments

address est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.

Details

La creation d'un objet [System](#) necessite la creation de deux autres objets : [Actif](#) et [Passif](#).

Author(s)

Damien Tichit pour Sia Partners

See Also

Construction d'un objet de type [Actif](#) : [load_actif](#).

Construction d'un objet de type [Passif](#) : [load_passif](#).

load_tab_morta	<i>Fonction load_tab_morta.</i>
----------------	---------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [TabMorta](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_tab_morta(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [TabMorta](#) necessite des donnees presentes dans les fichiers nommes TM-F . csv et TM-H . csv.

Author(s)

Damien Tichit pour Sia Partners

load_tab_rachat	<i>Fonction load_tab_rachat.</i>
-----------------	----------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [TabRachat](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_tab_rachat(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [TabRachat](#) necessite des donnees presentes dans un fichier nomme Rachat . csv.

Author(s)

Damien Tichit pour Sia Partners

load_tresorerie	<i>Fonction</i> load_tresorerie.
-----------------	----------------------------------

Description

Cette fonction permet de charger les donnees pour un objet de type [Tresorerie](#). Les donnees auront ete prealablement disposees dans une architecture propre a SiALM.

Usage

```
load_tresorerie(address)
```

Arguments

address	est un objet de type character indiquant le dossier dans lequel se situe l'ensemble des donnees necessaires pour la construction de l'objet.
---------	--

Details

La creation d'un objet [Tresorerie](#) necessite des donnees presentes dans un fichier nomme Tresorerie.csv.

Author(s)

Damien Tichit pour Sia Partners

Obligation	<i>Classe</i> Obligation
------------	--------------------------

Description

Cette classe represente le portefeuille des obligations de la compagnie d'assurance.

Slots

ptf est un objet de type [data.frame](#) contenant les donnees relatives au portefeuille.

Author(s)

Damien Tichit pour Sia Partners

Passif	<i>Classe</i> Passif
--------	----------------------

Description

Cette classe aggrege l'ensemble des donnees relatives au passif de la compagnie d'assurance : hypotheses, portefeuille, provisions

Slots

ptf_passif est un objet de la classe [PTFPassif](#) representant le portefeuille passif.

hyp_passif est un objet de la classe [HypPassif](#) contenant l'ensemble des hypotheses du passif.

provision est un objet de la classe [Provision](#).

Author(s)

Damien Tichit pour Sia Partners

PPE	<i>Classe</i> PPE
-----	-------------------

Description

Cette classe represente le Provision pour Participation aux Excedents.

Slots

ppe est un [numeric](#) contenant les montants dotes sur les huites dernieres annees.

Author(s)

Damien Tichit pour Sia Partners

proj_1an_actif	<i>Fonction</i> proj_1an_actif.
----------------	---------------------------------

Description

Cette fonction permet de projeter horizon 1 an l'actif d'une compagnie d'assurance.

Usage

```
proj_1an_actif(actif)
```

Arguments

actif est un objet de type [Actif](#).

Author(s)

Damien Tichit pour Sia Partners

proj_1an_epargne	<i>Fonction</i> proj_1an_epargne.
------------------	-----------------------------------

Description

Cette fonction permet de projeter horizon 1 an un portefeuille de contrats epragnes.

Usage

```
proj_1an_epargne(epargne, hyp_passif)
```

Arguments

epargne	est un objet de type Epargne .
hyp_passif	est un objet de type HypPassif .

Author(s)

Damien Tichit pour Sia Partners

proj_1an_immobilier	<i>Fonction</i> proj_1an_immobilier
---------------------	-------------------------------------

Description

Cette fonction permet de projeter horizon 1 an un portefeuille d'immobiliers. Elle calcule notamment les loyers.

Usage

```
proj_1an_immobilier(immobilier)
```

Arguments

immobilier	est un objet de type Immobilier .
------------	---

Author(s)

Damien Tichit pour Sia Partners

proj_1an_passif	<i>Fonction</i> proj_1an_passif
-----------------	---------------------------------

Description

Cette fonction permet de projeter horizon 1 an le passif d'une compagnie d'assurance.

Usage

```
proj_1an_passif(passif)
```

Arguments

passif est un objet de type [Passif](#).

Author(s)

Damien Tichit pour Sia Partners

proj_1an_ptf_passif	<i>Fonction</i> proj_1an_ptf_passif
---------------------	-------------------------------------

Description

Cette fonction permet de projeter horizon 1 an le portefeuille passif : gestion des differents passifs

Usage

```
proj_1an_ptf_passif(ptf_passif, hyp_passif)
```

Arguments

ptf_passif est un objet de type [PTFPassif](#).

hyp_passif est un objet de type [HypPassif](#).

Author(s)

Damien Tichit pour Sia Partners

proj_1an_system	<i>Fonction</i> proj_1an_system.
-----------------	----------------------------------

Description

Cette fonction permet de projeter a horizon 1 an un objet [System](#).

Usage

```
proj_1an_system(system, an)
```

Arguments

system	est un objet de type System.
an	est un integer.

Author(s)

Damien Tichit pour Sia Partners

See Also

Projection des passifs : [proj_1an_passif](#)

Projection des actifs : [proj_1an_actif](#)

Provision	<i>Classe</i> Provision
-----------	-------------------------

Description

Cette classe aggere les differentes provisions relatives au passif d'une compagnie d'assurance : PPE, Reserve de Capitalisation

Slots

ppe est un objet de la classe [PPE](#).
reserve_capi est un objet de la classe [ReserveCapi](#).

Author(s)

Damien Tichit pour Sia Partners

PTFActif	Classe PTFActif
----------	-----------------

Description

Cette classe represente le portefeuille financier de la compagnie d'assurance.

Slots

action est un objet de la classe [Action](#) representant le portefeuille action.

obligation est un objet de la classe [Obligation](#) representant le portefeuille obligation.

tresorerie est un objet de la classe [Tresorerie](#) representant le portefeuille tresorerie.

immobilier est un objet de la classe [Immobilier](#) representant le portefeuille tresorerie.

Author(s)

Damien Tichit pour Sia Partners

PTFCible	Classe PTFCible
----------	-----------------

Description

Cette classe represente le portefeuille financier cible de la compagnie d'assurance dans le cadre de

Slots

action est un objet de la classe [Action](#) representant le portefeuille cible action.

obligation est un objet de la classe [Obligation](#) representant le portefeuille cible obligation.

immobilier est un objet de la classe [Immobilier](#) representant le portefeuille cible Immobilier.

alloc_cible est un objet data.frame indiquant l'allocation cible pour chaque produit.

Author(s)

Damien Tichit pour Sia Partners

PTFPassif	Classe PTFPassif
-----------	------------------

Description

Cette classe represente le portefeuille passif de la compagnie d'assurance.

Slots

epargne est un objet de la classe [Epargne](#) representant le portefeuille epargne.

Author(s)

Damien Tichit pour Sia Partners

RachatConj	<i>Classe</i> RachatConj
------------	--------------------------

Description

Cette classe permet de modeliser les rachats conjoncturels.

Slots

alpha est un numeric
beta est un numeric
gamma est un numeric
delta est un numeric
RCmin est un numeric
RCmax est un numeric

Author(s)

Damien Tichit pour Sia Partners

rebalancement_actif	<i>Fonction</i> rebalancement_actif
---------------------	-------------------------------------

Description

Cette fonction permet de rebalancer le portefeuille d'actif vers le portefeuille cible.

Usage

```
rebalancement_actif(actif)
```

Arguments

actif est un objet de type Actif.

Author(s)

Damien Tichit pour Sia Partners

rebalancement_action *Fonction* rebalancement_action

Description

Cette fonction permet de rebalancer le portefeuille d'action vers un portefeuille cible. Le montant total cible, en valeur de marche, du portefeuille cible est renseigné dans le paramètre alloc_cible.

Usage

```
rebalancement_action(action, alloc_cible)
```

Arguments

action	est un objet de type Action . Ce paramètre représente le ptf actuel de la compagnie.
alloc_cible	est un numeric. Ce paramètre indique l'allocation cible à atteindre (en valeur de marche).

Author(s)

Damien Tichit pour Sia Partners

rebalancement_immobilier
 Fonction rebalancement_immobilier

Description

Cette fonction permet de rebalancer le portefeuille immobilier vers un portefeuille cible. Le montant total, en valeur de marche, du portefeuille cible est renseigné dans le paramètre alloc_cible.

Usage

```
rebalancement_immobilier(immo, alloc_cible)
```

Arguments

immo	est un objet de type Immobilier . Ce paramètre représente le ptf actuel de la compagnie.
alloc_cible	est un numeric. Ce paramètre indique l'allocation cible à atteindre.
immo_cible	est un objet de type Immobilier . Ce paramètre représente le ptf cible.

Author(s)

Damien Tichit pour Sia Partners

`rebalancement_obligation`*Fonction rebalancement_obligation*

Description

Cette fonction permet de rebalancer le portefeuille d'obligation vers un portefeuille cible. Le montant total, en valeur de marche, du portefeuille cible est renseigné dans le paramètre `alloc_cible`.

Usage

```
rebalancement_obligation(oblig, oblig_cible, alloc_cible)
```

Arguments

<code>oblig</code>	est un objet de type Obligation . Ce paramètre représente le ptf actuel de la compagnie.
<code>oblig_cible</code>	est un objet de type Obligation . Ce paramètre représente le ptf cible.
<code>alloc_cible</code>	est un <code>numeric</code> . Ce paramètre indique l'allocation cible à atteindre.

Author(s)

Damien Tichit pour Sia Partners

`reprise_ppe`*Fonction reprise_ppe.*

Description

Cette fonction permet de reprendre un montant sur la PPE. Le montant est prioritairement repris sur les plus vieilles dotations.

Usage

```
reprise_ppe(ppe, montant)
```

Arguments

<code>ppe</code>	est un objet de type PPE .
<code>montant</code>	est un <code>numeric</code> représentant le montant à reprendre.

Author(s)

Damien Tichit pour Sia Partners

reprise_ppe_8ans	<i>Fonction</i> reprise_ppe_8ans.
------------------	-----------------------------------

Description

Cette fonction permet de reprendre le montant dotee 8 annees auparavant. Elle met egalement a 0 l'element du vecteur correspondant a la PPE dotee 8 ans auparavant.

Usage

```
reprise_ppe_8ans(ppe)
```

Arguments

ppe est un objet de type [PPE](#).

Author(s)

Damien Tichit pour Sia Partners

ReserveCapi	<i>Classe</i> ReserveCapi
-------------	---------------------------

Description

Cette classe represente la Reserve de Capitalisation.

Slots

montant est un numeric representant le capital present dans la reserve.

Author(s)

Damien Tichit pour Sia Partners

revalo_action	<i>Fonction revalo_action</i>
---------------	-------------------------------

Description

Cette fonction permet de revaloriser le portefeuille action. Calcule également les plus ou moins values latentes.

Usage

```
revalo_action(action)
```

Arguments

action est un objet de type [Action](#).

Author(s)

Damien Tichit pour Sia Partners

revalo_immobilier	<i>Fonction revalo_immobilier</i>
-------------------	-----------------------------------

Description

Cette fonction permet de revaloriser le portefeuille immobilier.

Usage

```
revalo_immobilier(immobilier)
```

Arguments

immobilier est un objet de type [Immobilier](#).

Author(s)

Damien Tichit pour Sia Partners

revalo_obligation	<i>Fonction revalo_obligation</i>
-------------------	-----------------------------------

Description

Cette fonction permet de revaloriser les différentes obligations d'un portefeuille obligataire. Elle calcule également les plus ou moins value latentes (PMVL) engendrées.

Usage

```
revalo_obligation(obligation, yield_curve)
```

Arguments

obligation	est un objet de type Obligation .
yield_curve	est un <code>numeric</code> contenant les prix zero-coupon.

Author(s)

Damien Tichit pour Sia Partners

revalo_passif	<i>Fonction revalo_passif</i>
---------------	-------------------------------

Description

Cette fonction permet de revaloriser le passif d'une compagnie d'assurance.

Usage

```
revalo_passif(passif, pb)
```

Arguments

passif	est un objet de type Passif .
pb	est un <code>numeric</code> représentant le montant de PB à distribuer.

Author(s)

Damien Tichit pour Sia Partners

revalo_ptf_actif	<i>Fonction revalo_ptf_actif</i>
------------------	----------------------------------

Description

Cette fonction permet de projeter de revaloriser le portefeuille financier. Elle calcule également les plus ou moins value latentes (PMVL) engendrees.

Usage

```
revalo_ptf_actif(ptf_actif)
```

Arguments

ptf_actif est un objet de type [PTFActif](#).

Author(s)

Damien Tichit pour Sia Partners

SiALM	<i>SiALM est un package permettant de calculer un best-estimate pour une compagnie d'assurance proposant les prosuits suivants : contrats d'epargne.</i>
-------	--

Description

SiALM est un package permettant de calculer un best-estimate pour une compagnie d'assurance proposant les prosuits suivants : contrats d'epargne.

System	<i>Classe System</i>
--------	----------------------

Description

Cette classe regroupe les actifs et les passifs d'une compagnie d'assurance.

Slots

passif est un objet de type [Passif](#).

actif est un objet de type [Actif](#).

taux_pb est une list contenant les taux de pb contractuels.

Author(s)

Damien Tichit pour Sia Partners

TabMorta	<i>Classe TabMorta</i>
----------	------------------------

Description

Cette classe represente une table de mortalite.

Slots

table est un objet de type `data.frame` contenant la table de mortalite.

Author(s)

Damien Tichit pour Sia Partners

TabRachat	<i>Classe TabRachat</i>
-----------	-------------------------

Description

Cette classe represente une table de rachat

Slots

table est un objet de type `data.frame` contenant la table de rachat

Author(s)

Damien Tichit pour Sia Partners

Tresorerie	<i>Classe Tresorerie</i>
------------	--------------------------

Description

Cette classe represente la tresorerie de la compagnie d'assurance.

Slots

ptf est un objet de type `data.frame` contenant les donnees relatives au portefeuille.

Author(s)

Damien Tichit pour Sia Partners

`vieillissement_obligation`*Fonction vieillissement_obligation*

Description

Cette fonction permet de vieillir un portefeuille obligataire : mise a jour de la maturite residuelle et vente des obligations arrivees a maturite.

Usage`vieillissement_obligation(obligation)`**Arguments**

`obligation` est un objet de type [Obligation](#).

Author(s)

Damien Tichit pour Sia Partners

`vieillissement_ppe`*Fonction vieillissement_ppe.*

Description

Cette fonction permet de vieillir d'une annee l'objet [PPE](#)

Usage`vieillissement_ppe(ppe)`**Arguments**

`ppe` est un objet de type [PPE](#).

Author(s)

Damien Tichit pour Sia Partners

`vieillissement_ptf_actif`*Fonction vieillissement_ptf_actif*

Description

Cette fonction permet de vieillir les differents composants d'un portefeuille d'actif.

Usage

```
vieillissement_ptf_actif(ptf_actif)
```

Arguments

`ptf_actif` est un objet de type [PTFActif](#).

Author(s)

Damien Tichit pour Sia Partners

Index

*Topic **classes**

Actif, 4
Action, 4
ALM, 8
Epargne, 12
HypActif, 15
HypALM, 15
HypPassif, 16
Immobilier, 16
Obligation, 30
Passif, 31
PPE, 31
Provision, 34
PTFActif, 35
PTFCible, 35
PTFPassif, 35
RachatConj, 36
ReserveCapi, 39
System, 42
TabMorta, 43
TabRachat, 43
Tresorerie, 43

Actif, 4, 4, 17, 25, 28, 31, 42
Action, 4, 5, 13, 17, 18, 25, 26, 35, 37, 40
aggregation_actif, 4
aggregation_action, 5
aggregation_alm, 5
aggregation_epargne, 6
aggregation_immobilier, 6
aggregation_obligation, 7
aggregation_passif, 7
allocation_ptf_actif, 8
ALM, 5, 8, 18

calc_be, 9
calc_prod_fin, 10
calc_qx, 10
calc_rachat_conj, 11
calc_rx, 11
calcul_pb, 9

data.frame, 4, 12, 16, 30, 43
dotation_ppe, 12

Epargne, 6, 12, 19, 27, 32, 35
eval_coupon, 13
eval_dividende, 13
eval_frais_fin, 14
eval_loyer, 14
eval_prod_fin, 15

HypActif, 4, 14, 15, 17, 19, 20
HypALM, 8, 15, 18–20
HypPassif, 16, 20, 21, 23, 24, 31–33

Immobilier, 6, 14, 16, 21, 22, 25, 26, 32, 35,
37, 40

load_actif, 17, 28
load_action, 17, 25
load_action_cible, 18, 26
load_alm, 18
load_epargne, 19, 27
load_hyp_actif, 17, 19
load_hyp_alm, 19, 20
load_hyp_passif, 20, 24
load_immobilier, 21, 25
load_immobilier_cible, 22, 26
load_obligation, 22, 25
load_obligation_cible, 23, 26
load_passif, 23, 28
load_ppe, 24, 25
load_provision, 24, 24
load_ptf_actif, 17, 25
load_ptf_cible, 20, 26
load_ptf_passif, 24, 26
load_rachat_conj, 27
load_reserve_capi, 25, 27
load_system, 19, 28
load_tab_morta, 21, 29
load_tab_rachat, 21, 29
load_tresorerie, 25, 30

numeric, 31

Obligation, 7, 13, 22, 23, 25, 26, 30, 35, 38,
41, 44

Passif, 7, 23, 28, 31, 33, 41, 42

PPE, [12](#), [24](#), [25](#), [31](#), [34](#), [38](#), [39](#), [44](#)
proj_lan_actif, [31](#), [34](#)
proj_lan_epargne, [32](#)
proj_lan_immobilier, [32](#)
proj_lan_passif, [33](#), [34](#)
proj_lan_ptf_passif, [33](#)
proj_lan_system, [10](#), [34](#)
Provision, [23–25](#), [31](#), [34](#)
PTFActif, [4](#), [10](#), [14](#), [15](#), [17](#), [35](#), [42](#), [45](#)
PTFCible, [15](#), [20](#), [26](#), [35](#)
PTFPassif, [24](#), [26](#), [27](#), [31](#), [33](#), [35](#)

RachatConj, [11](#), [16](#), [27](#), [36](#)
rebalancement_actif, [36](#)
rebalancement_action, [37](#)
rebalancement_immobilier, [37](#)
rebalancement_obligation, [38](#)
reprise_ppe, [38](#)
reprise_ppe_8ans, [39](#)
ReserveCapi, [25](#), [27](#), [28](#), [34](#), [39](#)
revalo_action, [40](#)
revalo_immobilier, [40](#)
revalo_obligation, [41](#)
revalo_passif, [41](#)
revalo_ptf_actif, [42](#)

SiALM, [42](#)
SiALM-package (SiALM), [42](#)
System, [8](#), [10](#), [18](#), [19](#), [28](#), [34](#), [42](#)

TabMorta, [10](#), [16](#), [21](#), [29](#), [43](#)
TabRachat, [11](#), [16](#), [21](#), [29](#), [43](#)
Tresorerie, [25](#), [30](#), [35](#), [43](#)

vieillissement_obligation, [44](#)
vieillissement_ppe, [44](#)
vieillissement_ptf_actif, [45](#)