PREPAR-VIE ASSURANCE

Notice d'utilisation : Automatisation de l'approche par transparence

Ecrit par

Benjamin BENISTI

Supervisé par : Meryem EL GHARIB

Dernière édition de la notice : 3 septembre 2025

Table des matières

1 Contexte **3**

2 Choix de programmation **4**
2.1 Sheets_Manip 4
2.1.1 conv_from_excel 4
2.1.2 normalized 5
2.1.3 conv_to_excel 5
2.1.4 input_SAS 5
2.1.5 transpa_type 6
2.2 Transpa_Tests 6
2.2.1 VM 6
2.2.2 coupon_rate 7
2.2.3 storage 7
2.2.4 maturity 7
2.2.5 coupon_freq 8
2.2.6 delt 8
2.2.7 ratio_nom_vm 8
2.2.8 not_cred, covered, empt_country, empt_under, infra . . 9

3 Mise en fonctionnement de la transparisation **10**
3.1 Procédure d'installation 10
3.2 Lancement du programme 11
3.2.1 Fichiers nécessaires à la pré-transparisation 11
3.2.2 Fichiers nécessaires à la transparisation 11

4 Erreurs récurrentes du programme et solutions **12**
4.1 Non-reconnaissance du chemin du fichier 12
4.2 ValueError : Length mismatch 13
4.2.1 Détection du fichier où se situe l'erreur 13
4.2.2 Correction de l'erreur 13
4.3 [Errno 2] No such file or directory : 14

4.4 zipfile.BadZipFile : File is not a zip file 14
4.5 ValueError : Impossible de détecter automatiquement la ligne d'en-tête. 15
4.6 Single positional indexer is out-of-bounds 16
4.7 Autre type d'erreur d'import 16

5 Programme d'automatisation de l'envoi des mails 17
5.1 Présentation du sous-dossier *Code - Mail* 17
5.2 Mise en fonctionnement du programme 17

1. Contexte

La programmation de l'automatisation de l'approche par transparence ayant sollicité un grand nombre de sous-fonctions, ce document a pour but de permettre à n'importe quel futur utilisateur de pouvoir repérer et modifier facilement ces dernières.

Afin de faciliter au mieux l'attribution des tâches et la relecture du programme, ce dernier a été construit sous la forme d'un dossier comportant plusieurs éléments.

Tout d'abord, le sous-dossier `Test_2024` contient les fichiers de test pouvant être utilisés lors de la phase de programmation. Ensuite, le sous-dossier `Code - Transpa` renferme les cinq fichiers de code que nous détaillerons par la suite : `Sheets_Manip`, `Transpa_Tests`, `Main`, `creer_executable_et_raccourci` et `Interface`.

Le fichier `Sheets_Manip` est dédié à la manipulation des fichiers TPT. Il gère notamment :

- la conversion des fichiers Excel en DataFrame (et inversement),
- la concaténation de plusieurs fichiers TPT,
- la normalisation des noms de colonnes après transparisation,
- et d'autres opérations de traitement de données.

`Transpa_Tests` constitue quant à lui la partie analytique de cette approche. En effet, c'est dans ce fichier que nous pourrons retrouver l'ensemble des fonctions utiles à la vérification des données.

De plus, le fichier `Main` est l'endroit permettant d'appeler les différentes fonctions des fichiers précédents manuellement ainsi que celui dans lequel pourra être effectuée la pré-transparisation et la transparisation.

Enfin et comme son nom l'indique, le fichier Interface permet, en l'exécutant, de lancer le pilotage simplifié de la [pré-]transparisation et/ou de l'envoi des mails et de choisir le paramétrage des tests qui en ont besoin.

2. Choix de programmation

2.1 Sheets_Manip

2.1.1 conv_from_excel

Parmi l'ensemble des fonctions développées, conv_from_excel constitue sans doute l'une des plus essentielles du programme. Elle permet d'importer des fichiers Excel et de les convertir en DataFrame, format exploitable pour les traitements ultérieurs.

Étant donné l'hétérogénéité des formats utilisés par les fonds d'investissement pour structurer les en-têtes des fichiers TPT, il a été décidé d'imposer, dès l'importation, une nomenclature standardisée pour les colonnes. Cette nomenclature est basée sur celle utilisée dans le résultat de la transparisation 2024. Par ailleurs, une colonne nommée Agregated_name_product, bien que présente dans les résultats finaux, n'apparaît jamais dans les fichiers TPT d'origine. Sa création est donc automatisée dès l'importation des données.

Il est important de souligner, comme le rappellent les notes de programmation, que cette fonction ne prend en charge que les fichiers au format .xls ou .xlsx. Ainsi, si un fichier TPT est reçu au format .csv, .zip ou .xlsb il devra être converti au préalable dans un format compatible. Bien que la détection des feuilles 'parasites' ait été codé, il est tout de même fortement recommandé de les supprimer manuellement afin de s'assurer de ne pas gêner l'exécution du programme.

2.1.2 normalized

Les fichiers TPT transmis par les fonds présentent souvent des irrégularités : suppression de colonnes, modification des en-têtes, ou encore remplissages incohérents et difficilement explicables. Pour pallier ces problèmes, la fonction `conv_from_excel` a été associée à une fonction de validation : `normalized`.

Cette fonction a pour objectif de s'assurer que le fichier importé respecte scrupuleusement le format attendu. Elle constitue un point de contrôle essentiel pour garantir la fiabilité du traitement des données. En cas de découverte de nouvelles anomalies dans les TPT reçus, c'est cette fonction qu'il conviendra de modifier afin d'adapter le programme aux évolutions des formats.

Cependant, avant toute tentative de modification de `normalized`, il est fortement recommandé d'analyser en profondeur la structure du code. En effet, plus d'une centaine de fichiers tests ont été utilisés pour concevoir cette fonction, et des dizaines de cas particuliers ont déjà été traités. Une modification non maîtrisée pourrait compromettre la stabilité de l'ensemble du processus.

2.1.3 conv_to_excel

Puisque l'objectif de ce programme est de faciliter le processus de transparisation, il était nécessaire de pouvoir obtenir les résultats sous une forme exploitable pour les calculs futurs. Ainsi, la fonction `conv_to_excel` permet de convertir un DataFrame en fichier Excel, sauvegardé à l'emplacement et sous le nom souhaités par l'utilisateur.

Il est à noter que certaines dates peuvent apparaître accompagnées d'une heure (par exemple : `2027-12-31 00:00:00`). Ce comportement provient de l'interprétation des dates par la bibliothèque `datetime` de Python, et il est tout à fait normal.

2.1.4 input_SAS

En raison du fait que l'input de R3S utilisé suite à la transparisation n'est pas le même que le fichier général de transparisation, il était nécessaire de coder une fonction capable d'effectuer automatiquement les re-traitements pour

générer ce fichier. Le fichier de référence utilisé afin de définir les contenus de chaque colonne est situé dans le répertoire suivant : `K:\_Bureautique\Corporate finance\Solvency II\SCR _202303\1_DonnÃ¯es Sources\V _SAS\Actif \01_Inputs SAS\BTranspa \BTranspa _202303.xlsm`. On y retrouvera toutes les formules Excel qui ont été retranscrites sur Python dans cette fonction.

2.1.5 transpa_type

La colonne `'Transpa_Type'` du fichier d'input bénéficiant d'un traitement particulier, il était nécessaire de recoder la formule un tant soit peu complexe qui était utilisée pour générer cette dernière. Ainsi, en s'aidant du contenu de la feuille `Corresp chocs CIC` présente dans tous les fichiers `03_Transpa`, nous avons crée de nombreux dictionnaire stockant les informations utiles, permettant ainsi de nécessiter d'un input en moins pour effectuer la transparisation.

2.2 Transpa_Tests

2.2.1 VM

Dans cette méthode, il s'agit de s'assurer que la valeur de marché (VM) d'un fonds correspond bien à la somme des VM des lignes qui le composent. Cette vérification permet de détecter d'éventuelles incohérences dans les données fournies.

Cependant, en l'absence de documentation précisant l'intervalle d'écart acceptable entre ces deux valeurs, un seuil de tolérance large a été défini : un écart maximal de 0,1%. Ce choix vise à éviter les faux positifs tout en conservant une certaine rigueur dans la validation.

Si l'utilisateur souhaite modifier cette tolérance, il lui suffit de modifier la case correspondante dans la partie paramétrage de la transparisation.

2.2.2 coupon_rate

Cette méthode permet de vérifier que les taux de coupon ne sont ni anormalement élevés, ni négatifs. Une fois encore, l'absence de documentation précisant ce qu'est un écart « anormalement élevé » nous a conduit à faire des choix arbitraires, fondés sur l'analyse des résultats précédents de transparisation.

Trois critères ont ainsi été définis :

— **Critère de rejet** : par défaut à 15 %. Tout taux de coupon dépassant cette valeur est considéré comme absurde est notifié à l'utilisateur.
— **Critère d'alerte** : par défaut à 9 %. Tout taux de coupon compris entre 9 % et 15 % est notifié, mais conservé dans les données.
— **Critère de positivité** Tout taux de coupon négatif est notifié à l'utilisateur.

Si ces critères ne conviennent pas, ils peuvent être facilement modifiés dans la partie paramétrage de l'interface.

2.2.3 storage

Cette méthode s'assure que tous les instruments possèdent le même code NACE et le même code CIC. Cependant, concernant le code NACE, puisque ce dernier permet de désigner des secteurs économiques très précis, une légère variation de ce dernier peut-être relevé par le programme (qui ne s'assure que de l'égalité stricte des codes au sein d'un même groupe) sans pour autant que cela soit une véritable erreur de la part du fond. Il peut être alors judicieux de redéfinir (ou même mettre sous silence) ce test afin de rendre la log plus lisible.

2.2.4 maturity

Cette fonction va regarder l'ensemble des cases de la colonne `39_Maturity date` et réagir de deux façons (entraînant dans les deux cas une notification à l'utilisateur le cas échéant) :

— **S'il n'y a aucune date :** on rajoute la date `1900-01-01` dans la

case.

— **Si la date de maturité est avant le 31/12 de l'année renseignée :** on la modifie en ajoutant un an à cette dernière.

2.2.5 coupon_freq

La méthode `maturity` va traiter la colonne `12_CIC Code of the instrument` ligne par ligne et analyser le troisième caractère afin de repérer les obligations (codées `1` ou `2` à cet emplacement). Une fois la détection effectuée, on s'assure que la colonne `38_Coupon payment frequency` de la ligne associée n'est pas vide. Dans le cas contraire, une notification est envoyée à l'utilisateur.

2.2.6 delt

Le choix de programmation de cette méthode n'est, pour une fois, pas arbitraire, mais découle directement du lexique des colonnes à notre disposition. En effet, le delta mentionné dans les fichiers TPT est défini comme la dérivée du prix de l'option par rapport au prix de l'actif sous-jacent. Par conséquent, sa plage de valeurs dépend du type d'option concerné.

Pour les options *put*, le delta est compris entre 0 et -1, tandis que pour les options *call*, il varie entre 0 et 1. Ces bornes sont donc utilisées comme critères de validation dans cette méthode.

Cependant, si l'utilisateur souhaite appliquer des critères différents, ceux-ci peuvent être facilement modifiés dans le passage suivant.

```
if pd.notna(value):
    if value > 1 or value < -1:
        wrong_deltas.append(row.iloc[-1])
```

2.2.7 ratio_nom_vm

Comme son nom l'indique, cette fonction s'assure que le ratio VM / Nominal ne dépasse pas 5. Si l'utilisateur souhaite modifier ce paramètre, il lui

suffira de modifier la case associée dans la partie paramétrage de l'interface de transparisation.

2.2.8 not_cred, covered, empt_country, empt_under, infra

Ces fonctions s'assurent toutes de la bonne mise en forme des cases des colonnes `59_Credit quality step`, `55_Covered / not covered`, `52_Issuer country`, `131_Underlying Asset Category` et `132_Infrastructure_investment`. La plupart du temps, elles vérifient simplement que les colonnes ne sont pas vides ou qu'elles ne sont pas au format texte. Pour plus de détails, se référer au dossier de *contrôle général* de l'année précédente (fichier `transpa_03`) et consulter la description spécifique de chacune de ces fonctions.

3. Mise en fonctionnement de la transparisation

3.1 Procédure d'installation

Rendez-vous dans le dossier suivant : `K:\Bureautique\Corporate finance\Solvency II\SCR_202412\1_Données Sources\V2\Actif\1_Sources\Transparisation - Benjamin\Transparisation Code`.

Si c'est la première fois que vous vous apprêtez à utiliser ce programme et que vous n'avez jamais eu à préparer une quelconque procédure d'installation, veuillez suivre le passage suivant.

Tout d'abord, veuillez vous rendre dans le sous-dossier *Code Transpa* du dossier *Transparisation Code*. Une fois cela fait, faites clic droit dans une zone vide du dossier puis `GitBash Here`. Une fois la console ouverte, il vous suffira de recopier **une par une** les commandes suivantes :

1. `python -m pip install --trusted-host pypi.python.org --trusted-host files.pythonhosted.org --trusted-host pypi.org --upgrade pip`
2. `chmod +x launch.sh`
3. `pip install -r requirements.txt`

Explication sur ces formules :

1. `Permet de mettre à jour le module d'installation pip`
2. `Initialise la commande de lancement du programme`
3. `Télécharge les différentes bibliothèques utiles.`

3.2 Lancement du programme

Si vous avez déjà suivi la procédure d'installation précédente par le passé, il vous suffira de vous rendre de nouveau dans le sous-dossier *Code Transpa* du dossier *Transparisation Code*, d'ouvrir GitBash puis de taper la commande `./launch.sh`. Attendez ensuite quelques secondes puis l'interface apparaîtra.

3.2.1 Fichiers nécessaires à la pré-transparisation

Pour la pré-transparisation, vous devrez vous munir des fichiers suivants :

1. `Etat_Stock_OPCVM_(Année_en_cours)`
2. `03_Transpa_R3S_Niv1_(Année_précédente)`

3.2.2 Fichiers nécessaires à la transparisation

Pour la transparisation, vous devrez vous munir du fichier intitulé `01_SuiviTranspa` de l'année en cours.

4. Erreurs récurrentes du programme et solutions

Puisque les bibliothèques Python telles que pandas et os ont besoin d'inputs sous une forme bien particulière, un non-respect de cette dernière peut entraîner des erreurs récurrentes que nous tenteront de résumer dans ce passage.

4.1 Non-reconnaissance du chemin du fichier

Si vous tentez d'effectuer une conversion sur un fichier particulier sans passer par la méthode détaillée dans la partie 3, il est possible que le programme ne parvienne pas à ouvrir le fichier fourni en entrée. Deux causes principales peuvent expliquer ce comportement.

La première, la plus fréquente, est l'absence de double antislash (\\) dans le chemin du fichier. Dans ce cas, le programme ne parvient tout simplement pas à localiser le fichier.

La seconde cause possible est liée aux droits d'accès. Le programme peut ne pas être autorisé à accéder au fichier si celui-ci est encore ouvert dans un autre logiciel, ou s'il se trouve dans un répertoire nécessitant des permissions spéciales. Dans ce cas, si vous ne souhaitez pas déplacer le fichier, il est recommandé de fermer le logiciel d'exécution Python, puis de le relancer en tant qu'administrateur.

Conseil général : si vous constatez que l'icône de votre fichier ne correspond pas à celui d'un fichier Excel standard, il est fortement recommandé de le réenregistrer explicitement au format Excel afin d'éviter tout type d'erreur.

Symbole du format recommandé.

4.2 ValueError : Length mismatch

Si vous tentez de transpariser ou de convertir un fichier Excel en data frame et que l'erreur suivante s'affiche : `ValueError: Length mismatch: Expected axis has X elements, new values have 143 éléments`, avec `X` un nombre quelconque, cela provient du fait qu'un de vos fichiers en *input* n'a pas le bon nombre de colonnes et que le programme ne parvient pas à le corriger. Nous allons vous détailler précisément comment détecter de quel fichier il s'agit et comment corriger l'erreur.

4.2.1 Détection du fichier où se situe l'erreur

Si vous exécutez depuis l'interface : lisez le dernier fichier dont le traitement a commencé mais n'est pas terminé dans la console d'affichage.

Si vous exécutez manuellement : allez directement dans le `Terminal` puis regardez le dernier fichier dont le début du traitement a été annoncé mais pas sa fin. Ensuite, mettez le passage de commande de la transpiration entre `""" ... """` afin que votre code ne soit plus commandé par rien.

4.2.2 Correction de l'erreur

Une fois le fichier problématique détecté, vous pouvez aller vérifier si la colonne 62 ne manque pas. Le cas échéant, rajoutez-la manuellement sur Excel et retentez l'exécution du programme.

S'il n'y a pas de soucis à ce niveau, recopiez le passage suivant dans le

fichier `Main` puis regardez les colonnes une par une pour tenter de détecter une erreur à ce niveau. Une fois l'anomalie détectée, si vous vous sentez à l'aise en `Python`, vous pouvez modifier `normalized` pour qu'elle soit capable de traiter l'erreur de mise en forme détectée. Sinon, corrigez l'erreur manuellement dans le fichier Excel.

```
test = pd.read_excel('CHEMIN DU FICHIER')
df = SM.normalized(test, 'CHEMIN DU FICHIER')[0]
L = list(df.columns)
for i in range(len(L)):
    print(L[i])
```

Remplacez simplement le chemin du fichier par celui repéré dans le Terminal dans la partie 4.2.1.

4.3 [Errno 2] No such file or directory :

Cette erreur est en réalité très simple à traiter : elle provient probablement du fait que le chemin vers le fichier que vous tentez de transpariser est trop long pour être lu par Python. Il vous suffira alors de ré-enregistrer votre fichier avec un nom plus court ou de déplacer votre dossier de TPT vers un chemin moins profond afin de corriger cette erreur.

4.4 zipfile.BadZipFile : File is not a zip file

Si cette erreur s'affiche, c'est sans doute que vous avez tenté d'importer un fichier `.xlsx` qui n'en est pas réellement un. En effet, même si vous parvenez facilement à l'ouvrir via Excel, puisque ce format provient d'une version antérieure du logiciel (1997–2003), la conversion via Python peut se faire plus difficilement. Pour régler ce problème, il suffira que vous réenregistriez le fichier au format Excel et l'erreur devrait ne plus se produire.

4.5 ValueError : Impossible de détecter automatiquement la ligne d'en-tête.

Cette erreur provient d'une alerte de non-détection que nous avons nous-mêmes implémentée dans la fonction `normalized` du fichier `Sheets_Manip`. Deux sources d'erreurs principales ont étés recensées. Elle peut soit provenir du fait que certains fichiers Excel viennent avec plusieurs feuilles dont une seule seulement est utile soit du fait que la première case d'en-tête ne commence pas par '1_'. Dans le second cas, modifiez simplement l'en-tête en rajoutant ce terme et relancez la transparisation.

ATTENTION : Les fonctions de manipulations des colonnes ont étés codées de sorte à ce qu'elles détectent la présence de ces dernières via le numéro associé. Si ce numéro est absent de l'en-tête d'une colonne, le programme ne fonctionnera pas.

ATTENTION 2 : Ne recopiez pas simplement le nom correct des en-têtes dans la ligne posant problème dans le TPT. Les fond d'investissement ayant la merveilleuse idée de ne pas livrer les TPT avec toutes les colonnes, vous risquerez d'associer des données aux mauvaises colonnes. Dans le cas où cette erreur survient, vous n'avez d'autre choix que de rajouter manuellement les chiffres convenant à chaque colonne présente. (Attention aux colonnes telles que 8_, 17_, etc.)

Dans le cas où plusieurs feuilles sont présentes,et bien que le programme ait été pensé pour éliminer ce genre de feuilles parasites, il est possible que l'élimination des feuilles qui ne nous intéressent pas ne soit pas complète. Ainsi, si cette erreur apparaît, il est fortement conseillé de modifier le fichier problématique pour n'y laisser que la feuille d'intérêt pour la transparisation.

On a aussi constaté que l'encodage automatique des cellules de la part d'Excel peut aussi poser problème. En effet, puisque Excel décide automatiquement du type de données que vous avez imputé dans chaque cellule, il se peut que lors de la conversion en Python, une cellule d'en-tête n'ait pas le type `str` (qui correspond à une chaîne de caractères). Pour pallier ce problème, il suffit de sélectionner l'ensemble des en-têtes, de faire clic droit, *Format de cellules*, *Texte*.

4.6 Single positional indexer is out-of-bounds

Il a été constaté, lorsqu'un fichier TPT possède plusieurs feuilles utiles à la transparisation l'apparition de cette erreur. Bien que la tentative de lecture feuille par feuille puis concaténation des données utiles a été codée, il se peut que vous fassiez face à cette erreur. Le cas échéant, il ets conseillé de regrouper les données sur une seule et unique feuille et de retenter la transparisation.

4.7 Autre type d'erreur d'import

Si vous constatez une erreur qui n'est d'aucun des types détaillés dans cette partie, et si le problème se produit lors de l'import du fichier, vous pouvez facilement repérer celui qui pose un problème à la transparisation en allant dans le *Terminal*. En effet, le programme a été modifié afin que vous puissiez suivre la progression du traitement des fichiers. Ainsi, si une erreur non répertoriée survient (ce qui est fortement probable puisque les fonds semblent innovants pour faire en sorte que leur TPT soit différent de ce qui est attendu), lisez le nom du fichier problématique et vérifiez manuellement que tout est en norme.

5. Programme d'automatisation de l'envoi des mails

5.1 Présentation du sous-dossier *Code - Mail*

Afin de faciliter le contact des fonds pour recevoir les TPT, il a été ajouté au dossier *Transparisation Code* un sous-dossier *Code - Mail* dans lequel vous retrouverez deux fichiers : *interface_envoi_mail* qui correspond au code Python permettant d'afficher manuellement l'interface d'envoi automatique et *Template - A copier* qui est le fichier Excel qui vous servira de suivi de contact pour l'envoi des mails.

5.2 Mise en fonctionnement du programme

Après avoir copié, rempli et renommé le fichier *Template - A copier*, deux options s'offrent à vous. Soit vous utilisez l'interface général dont l'utilisation a été détaillée auparavant et dans ce cas il vous suffira de vous rendre dans l'onglet d'envoi des mails, soit vous souhaitez faire apparaître manuellement l'interface d'envoi. Pour cela, il vous suffira d'exécuter le programme *interface_envoi_mail* et de suivre les indications de fonctionnement. Vous pourrez alors choisir le message ainsi que l'objet du mail que vous enverrez à tous les fonds n'ayant pas encore envoyés leur TPT.

`A noter` : Le programme modifie lui-même l'objet du mail lorsqu'il détecte que le fonds qu'il tente de joindre a déjà été contacté par le passé. Il rajoute alors "[RELANCE]" devant ce dernier, permettant de transmettre le caractère important de ce mail.