voici un code python +l'explication de fonctionnement+ le fichier excel base de travail, analyse ce code et dis moi si t’as tout compris

Explication Détaillée de la Fonction

"Générer/MàJ Matrice"

La fonction generate\_matrix est le cœur de l'application Python fournie. Son rôle principal

est de lire un ﬁchier Excel modèle, d'appliquer des calculs basés sur les paramètres d'étude

fournis par l'utilisateur, puis de sauvegarder la matrice mise à jour dans un nouveau ﬁchier

Excel.

Voici une décomposition de son fonctionnement:

1. Validation des Entrées Utilisateur

Avant tout traitement, la fonction commence par appeler self.validate\_inputs() . Cette étape

cruciale assure que tous les paramètres saisis par l'utilisateur (Niveau d'étude, Nombre de

patients, Nombre de visites, Type de centre, Durée d'étude, Nombre d'avenants, Nombre de

visites de monitoring, Nombre de pages CRF) sont valides et cohérents. Si des erreurs sont

détectées (par exemple, champs vides, valeurs non numériques, nombres négatifs ou

nombre de visites insuﬃsant), un message d'erreur est aﬃché à l'utilisateur, et la fonction

s'arrête, empêchant ainsi des calculs incorrects ou des erreurs d'exécution.

2. Sélection des Fichiers Excel

La fonction invite ensuite l'utilisateur à sélectionner deux ﬁchiers :

•

Fichier modèle Excel (.xlsm) : C'est le ﬁchier Excel de base qui contient la structure de

la matrice et les désignations de coûts. Le code utilise ﬁledialog.askopenﬁlename pour

permettre cette sélection.

•

Fichier de sortie Excel (.xlsm) : L'utilisateur est invité à spéciﬁer le nom et

l'emplacement du ﬁchier où la matrice remplie sera sauvegardée.

ﬁledialog.asksaveasﬁlename est utilisé pour cela, avec un nom de ﬁchier initial suggéré

( matrice\_remplie.xlsm ).

Si l'utilisateur annule l'une de ces sélections, la fonction se termine.

3. Chargement du Fichier Excel et Récupération des

Paramètres

Une fois les ﬁchiers sélectionnés, le code charge le ﬁchier modèle Excel à l'aide de

openpyxl.load\_workbook(source\_ﬁle, keep\_vba=True) . L'argument keep\_vba=True est important

car il préserve les macros VBA potentiellement présentes dans le ﬁchier .xlsm .

Les valeurs des paramètres d'étude saisis par l'utilisateur dans l'interface graphique sont

ensuite récupérées et converties dans les types de données appropriés (entiers, chaînes de

caractères, booléens) :

* studyLevel (Niveau de l'étude)
* numPatients (Nombre de patients)
* numVisits (Nombre total de visites/patient)
* centerType (Type de centre : Coordonnateur ou Associé)
* studyDuration (Durée de l'étude en années)
* numAvenants (Nombre d'avenants)
* numVisitesMonitoring (Nombre de visites de monitoring)
* personnelExterieur (Personnel extérieur, booléen)
* numPagesCRF (Nombre de pages CRF)

Un calcul clé est eﬀectué à ce stade : numVisitsSurSite = max(0, numVisits - 2) . Cela détermine le

nombre de visites sur site en soustrayant la visite de screening et la visite ﬁnale du nombre

total de visites, en s'assurant que le résultat ne soit pas négatif.

4. Préparation des Couleurs de Remplissage

Deux objets PatternFill sont créés pour le surlignage des cellules :

* highlight\_ﬁll\_default (bleu clair) : Utilisé pour le remplissage par défaut des cellules

calculées.

* highlight\_ﬁll\_level (rouge clair) : Utilisé pour les lignes dont le calcul est spéciﬁque au

niveau d'étude ou au type de centre, aﬁn de les distinguer visuellement.

5. Mise à Jour de la Cellule B10 (Nombre de Patients)

La valeur du nombre de patients ( numPatients ) est écrite dans la cellule B10 de la feuille

Excel et cette cellule est surlignée en bleu clair. C'est un point de départ visuel pour

l'utilisateur.

6. Détermination de la Plage de Données

La fonction self.ﬁnd\_data\_rows(sheet) est appelée pour identiﬁer la plage de lignes

pertinentes à traiter dans la feuille Excel. Cette fonction recherche :

* ﬁrstRow : La première ligne contenant une désignation (non vide), à partir de

START\_ROW (ligne 18).

* lastRow : La dernière ligne à traiter. Elle est déterminée soit par la présence du

marqueur END\_ROW\_MARKER ("TOTAL GÉNÉRAL"), soit, si ce marqueur n'est pas trouvé,

par la dernière ligne de la feuille ( sheet.max\_row ).

* totalRow : La ligne où se trouve le "TOTAL GÉNÉRAL", utilisée pour mettre à jour la

somme ﬁnale.

Si la plage de données ne peut pas être déterminée, une erreur est levée.

7. Boucle de Traitement des Lignes

Le cœur de la fonction est une boucle qui itère sur chaque ligne de ﬁrstRow à lastRow

(inclus). Pour chaque ligne, les étapes suivantes sont eﬀectuées :

* **Récupération des données de la ligne** : La désignation (colonne A), la limite

d'occurrence (colonne B) et les consignes (colonne H) sont extraites. Les valeurs sont

converties en minuscules et les espaces superﬂus sont supprimés pour faciliter les

comparaisons.

* **Initialisation des variables** : quantity (quantité à calculer), is\_level\_speciﬁc ,

is\_center\_speciﬁc , is\_ﬁxed\_cost\_line , is\_special\_calculation sont initialisées. Ces

drapeaux ( ﬂags ) sont utilisés pour guider les calculs spéciﬁques à chaque type de

ligne.

* **Détection de Section** : Le code tente de détecter la section à laquelle appartient la

ligne (FORFAITS, TEMPS\_MEDICAL, TEMPS\_TEC, TEMPS\_INFIRMIER) en recherchant des

mots-clés dans la désignation. Bien que cette section soit détectée, elle n'est pas

utilisée de manière extensive dans la logique de calcul actuelle, sauf pour une condition

spéciﬁque ( consultation d'inclusion ).

8. Logique de Calcul par Désignation (Conditions if/elif )

C'est la partie la plus critique et la plus détaillée de la fonction. Le code utilise une série de

conditions if/elif pour identiﬁer la désignation de la ligne et appliquer la logique de calcul

correspondante. Voici les principales catégories de lignes traitées dans le code actuel :

**Forfaits Fixes Administratifs :**

frais administratifs : quantity = 1 (coût ﬁxe, spéciﬁque au centre).

frais supplémentaires pour l'élaboration d'un avenant : quantity = numAvenants (si > 0,

coût ﬁxe, spéciﬁque au centre).

mise en place de la recherche : quantity = 1 (coût ﬁxe, spéciﬁque au niveau).

**Forfaits Logistiques :**

* forfait de frais logistique : quantity = numPatients \* numVisits (si pas de personnel

extérieur) ou numPatients \* numVisits (si personnel extérieur et personnelExterieur est

vrai).

* forfait maintenance des appareils : quantity = studyDuration (coût ﬁxe).

**Temps Médical :**

* consultation d'inclusion (dans la section TEMPS\_MEDICAL) : quantity = numPatients

(spéciﬁque au niveau).

**Gestion des Avenants (Temps Médical/TEC) :**

* prise de connaissance de l'amendement ou prise de connaissance de l'addendum : quantity

= numAvenants \* (temps extrait de la désignation/consignes ou 0.5h/1h par défaut)

(coût ﬁxe).

**Temps TEC (Technicien d'Étude Clinique) :**

* temps tec formation : quantity = 5, 6 ou 8 heures selon le studyLevel (coût ﬁxe,

spéciﬁque au niveau).

* temps tec monitoring avec promoteur/cro : quantity = numVisitesMonitoring \* (2.5, 4 ou 5

heures selon le studyLevel ) (coût ﬁxe, spéciﬁque au niveau).

**Calculs Spéciaux pour Visites TEC (Screening, Sur Site, Finale) :**

Ces lignes ( temps tec visite de screening patient , temps tec visite sur site, de suivi patient ou

téléphonique , temps tec visite ﬁnale ou arrêt prématuré ) ont une logique de calcul plus

complexe qui prend en compte :

* Le TEMPS\_BASE (temps de base en heures) déﬁni dans les constantes du code pour

chaque type de visite et niveau d'étude.

* Le COUT\_HORAIRE (coût horaire) déﬁni dans les constantes pour chaque type de

visite et niveau d'étude.

* Le numPagesCRF (nombre de pages CRF) pour calculer un additional\_time (temps

additionnel) en fonction du niveau d'étude (15 min par tranche de 10 pages pour

niveau 1, 15 min par tranche de 5 pages pour niveaux 2 et 3).

* Le numVisitsSurSite pour les visites sur site.

Si aucune des conditions if/elif ne correspond à la désignation de la ligne, la variable

quantity reste None , et la ligne n'est pas traitée par la suite (les colonnes E, F, G restent

vides).

9. Calcul et Remplissage des Cellules (E, F, G)

Si une quantity a été déterminée (c'est-à-dire si la ligne correspond à une logique de

traitement) :

* **Montant Unitaire (Colonne D)** : Le code tente d'extraire le montant unitaire de la

colonne D. Il vériﬁe d'abord si le montant est spéciﬁque au type de centre

( is\_center\_speciﬁc ) ou au niveau d'étude ( is\_level\_speciﬁc ) en utilisant les fonctions

extract\_montants\_par\_centre et extract\_montants\_par\_niveau . Si aucun montant

spéciﬁque n'est trouvé, il utilise la valeur directe de la cellule D.

* **Couleur de Remplissage** : La cellule est surlignée en rouge clair ( highlight\_ﬁll\_level ) si

elle est spéciﬁque au niveau ou au type de centre, sinon en bleu clair

( highlight\_ﬁll\_default ).

* **Calcul du Total Ligne (Colonne F)** : Pour les lignes avec calcul spécial (screening,

visites TEC), le total\_ligne est calculé selon la logique complexe décrite ci-dessus. Pour

les autres lignes, total\_ligne = quantity \* montant\_unitaire .

* **Calcul du Total Centre (Colonne G) :**
* Si la limite d'occurrence (colonne B) contient "centre", "etablissement", "protocole"

ou si la ligne est marquée comme is\_ﬁxed\_cost\_line , alors total\_centre = total\_ligne (le

coût est ﬁxe pour le centre, non multiplié par le nombre de patients).

* Sinon, total\_centre = total\_ligne \* numPatients .
* **Mise à jour des Cellules :** Les valeurs calculées pour quantity (colonne E), total\_ligne

(colonne F) et total\_centre (colonne G) sont écrites dans les cellules correspondantes,

et les cellules sont remplies avec la couleur appropriée.

* **Agrégation pour le Total Général :** La valeur du total\_centre est ajoutée à une liste

modiﬁed\_rows\_data pour le calcul ultérieur du total général.

10. Calcul et Mise à Jour du Total Général

Après avoir traité toutes les lignes, la fonction calcule la somme de tous les total\_centre

agrégés dans modiﬁed\_rows\_data . Si la ligne du total général ( totalRow ) a été trouvée, cette

somme est écrite dans la cellule correspondante de la colonne G et surlignée en bleu. Si la

ligne du total général n'est pas trouvée, la somme est quand même calculée et aﬃchée dans

un message d'avertissement.

11. Sauvegarde et Message de Succès

Enﬁn, le classeur Excel modiﬁé est sauvegardé dans le ﬁchier de sortie spéciﬁé par

l'utilisateur ( workbook.save(output\_ﬁle) ). Un message de succès est aﬃché, récapitulant le

chemin du ﬁchier de sortie, le nombre de lignes remplies, la répartition des visites et le total

général calculé.

12. Gestion des Erreurs

La fonction est enveloppée dans un bloc try-except pour gérer les erreurs potentielles,

telles que :

* KeyError : Si la feuille Excel spéciﬁée ( Annexe 2.1+MO-autorisation24 ) est introuvable.
* ValueError : Si la plage de données ne peut pas être déterminée.
* Exception générale : Pour capturer toute autre erreur inattendue et aﬃcher un message

d'erreur générique à l'utilisateur.

En résumé, la fonction generate\_matrix automatise le remplissage d'une matrice de coûts

Excel en appliquant une logique de calcul conditionnelle basée sur les désignations de

lignes et les paramètres d'étude fournis par l'utilisateur. Son eﬃcacité dépend directement

de la complétude de sa logique interne pour couvrir toutes les désignations de lignes

pertinentes dans le ﬁchier Excel modèle.