SAÉ Analyse de données, Reporting et Datavisualisation

Manuel Technique :

Outil de Gestion et Analyse des Accidents de la Vie Courante en France

Table des matières

Introduction :

Ce manuel technique vise à fournir une explication détaillée des fonctionnalités de l'outil de gestion et d'analyse des accidents de la vie courante en France. Cet outil, développé en Python avec l'utilisation, notamment, de la bibliothèque tkinter pour l'interface graphique et pandas pour la manipulation des données, permet d'importer, visualiser et analyser des données liées aux accidents domestiques. Il inclut également des fonctionnalités pour ouvrir des fichiers externes, comme une visualisation Power BI et des tutoriels, pour offrir une expérience utilisateur complète et interactive. Ce guide couvre chaque fonctionnalité de l'outil en détaillant son objectif, son fonctionnement et les messages d'erreur associés pour aider les utilisateurs à comprendre et à utiliser efficacement l'outil.

Explication du code Python de l'outil dynamique :

Chargement des données Excel

```python
import tkinter as tk
from tkinter import filedialog, messagebox
import pandas as pd
import json
import webbrowser
import os
from scipy.stats import chi2_contingency

# dictionnaire pour stocker les dataframes
data_frames = {}

#fonction de chargement des données
def charger_excel_data():
    global data_frames

    messagebox.showwarning(title="Avertissement", message="Veuillez à bien sélectionner le fichier Excel contenant les feuilles 'Accident' et 'BD_3quest'")
    file_path = filedialog.askopenfilename(
        title="Sélectionnez un fichier Excel",
        filetypes=[("Excel files", "*.xlsx;*.xls"), ("CSV files", "*.csv")]
    )

    if file_path:
        try:
            data_frames = pd.read_excel(file_path, sheet_name=None)
            creer_btn_feuilles()

        except Exception as e:
            messagebox.showerror("Erreur", f"Une erreur s'est produite lors de la lecture du fichier Excel:\n{e}")
```

Cette partie de notre outil permet à l'utilisateur de sélectionner un fichier Excel contenant des informations sur les accidents. Une fois le fichier sélectionné, notre outil lit les données de ce fichier et les prépare pour une utilisation ultérieure. Si le fichier est valide, les données sont chargées correctement. Si le fichier est invalide ou corrompu, un message d'erreur s'affiche pour indiquer que le fichier ne peut pas être lu.

Affichage des données

```python
#fonction d'affichage des données dans une zone de texte
def afficher_donnees(sheet_name):
    global data_frames

    if sheet_name in data_frames:
        df = data_frames[sheet_name]
        output_text = df.to_string(index=False)

        num_lignes = df.shape[0]
        num_colonnes = df.shape[1]

        titre = f"Données de la feuille : {sheet_name}"
        output_text = f"{titre}\n\n{output_text}"

        txt_output.config(state=tk.NORMAL)
        txt_output.delete('1.0', tk.END)
        txt_output.insert(tk.END, output_text)
        txt_output.config(state=tk.DISABLED)

        info_text.set(f"Feuille : {sheet_name}\nNombre de lignes : {num_lignes}\nNombre de colonnes : {num_colonnes}")
    else:
        messagebox.showwarning("Avertissement", f"La feuille '{sheet_name}' n'a pas été trouvée dans le fichier.")
```

Une fois les données importées, notre outil permet de les afficher dans une zone de texte pour que l'utilisateur puisse les consulter. En sélectionnant une feuille spécifique, les données correspondantes sont affichées avec des informations sur le nombre de lignes et de colonnes. Si la feuille spécifiée n'existe pas, un message d'avertissement s'affiche pour indiquer que la feuille n'a pas été trouvée.

Création des boutons pour sélectionner les feuilles

```
#fonction de création de boutons servant à changer la feuille dont les données sont affichées dans la zone de texte
def creer_btn_feuilles():
    for widget in col_gauche.winfo_children():
        widget.grid_forget()

    load_button.grid(row=0, column=0, padx=(0, 10), pady=(0, 10))
    info_label.grid(row=1, column=0, padx=(0, 10), pady=(0, 10))

    btn_width1 = 10
    btn_height1 = 1
    feuilles_a_afficher = ["BD_3quest", "Accident"]
    l=len(feuilles_a_afficher)+1
    for i, sheet_name in enumerate(feuilles_a_afficher):
        btn_feuille = tk.Button(col_gauche, text=sheet_name, width=btn_width1, height=btn_height1, command=lambda s=sheet_name: changer_feuille(s))
        btn_feuille.grid(row=i+1, column=0, padx=10, pady=10)
    btn_pbi = tk.Button(col_gauche, text="Ouvrir la Dataviz Power BI", command=ouvrir_pbi, width=btn_width, height=btn_height, bg="#e12454", fg="#ffffff", padx=10, pady=5, font=("
    btn_pbi.grid(row=l, column=0, padx=10, pady=10)
    btn_pbi.bind("<Enter>", on_enter)
    btn_pbi.bind("<Leave>", on_leave)
```

Pour rendre l'outil plus interactif, des boutons sont créés pour chaque feuille de données disponible dans le fichier Excel. L'utilisateur peut cliquer sur ces boutons pour afficher les données de la feuille correspondante. Si les boutons sont créés correctement, ils permettent de changer la feuille affichée. Si aucun bouton n'est créé, cela signifie que les données n'ont pas été chargées ou qu'il y a eu un problème lors de la création des boutons.

Analyse statistique des données

```
#fonction d'application du traitement statistique au fichier contenant les données
def prog_stat():
    try:
        global data_frames
        df_accident = data_frames["Accident"]
        df_BD_3quest = data_frames["BD_3quest"]

        colonnes_a_supprimer = ['Texte_connaissance_etude', 'Date_remp_foyer', 'Nb_pers_foyer',
                                'Sexe1', 'Type_pers1', 'Annee_naiss1',
                                'Occup_log1', 'Inscrite1', 'Sexe2',
                                'Type_pers2', 'Annee_naiss2', 'Occup_log2',
                                'Inscrite2', 'Sexe3', 'Type_pers3',
                                'Annee_naiss3', 'Occup_log3','Inscrite3', 'Sexe3',
                                'Type_pers4',
                                'Annee_naiss4', 'Occup_log4','Inscrite4', 'Sexe4',
                                'Type_pers5',
                                'Annee_naiss5', 'Occup_log5','Inscrite5', 'Sexe5',
                                'Race_chien',
                                'Taille_com', 'Habitat_voisinage','Surface_exacte_log', 'Chauf',
                               'Abri_jardin',
                               'Plan_eau', 'Compo_foyer','Animaux_dom' ]

        df_BD_3quest = df_BD_3quest.drop(columns=colonnes_a_supprimer)
        df_accident_oui = df_accident.loc[df_accident['Acc'] == 'Oui']
        df_accident_non = df_accident.loc[df_accident['Acc'] == 'Non']

        df_BD_3quest['Age_actuel'] = df_BD_3quest['Age_actuel'].astype(int)
        df_BD_3quest.loc[df_BD_3quest['Age_actuel'] == 1, 'Age_actuel'] = 'Non renseigné'
```

...

```
    df_accident_oui['Type_acc'] = df_accident_oui['Type_acc'].apply(remplacer_type_accident)
    df_accident_oui['Heure_acc'] = df_accident_oui['Heure_acc'].astype(int)

    df_accident_oui.loc[df_accident_oui['Heure_acc'] < 12, 'Heure_cat'] = 'Matin'
    df_accident_oui.loc[(df_accident_oui['Heure_acc'] >= 12) & (df_accident_oui['Heure_acc'] < 18), 'Heure_cat'] = 'Après-midi'
    df_accident_oui.loc[df_accident_oui['Heure_acc'] >= 18, 'Heure_cat'] = 'Soir'

    df_accident_f = pd.concat([df_accident_oui, df_accident_non], ignore_index=True)
    df_final = pd.merge(df_accident_f, df_BD_3quest, on='Id_volontaire', how='outer')

    df_accident_subset = df_accident_oui[['Fatigue', 'Heure_cat']]
    contingency_table = pd.crosstab(df_accident_subset['Fatigue'], df_accident_subset['Heure_cat'])
    chi2, p, dof, expected = chi2_contingency(contingency_table)
    if p <= 0.05:
        p1 = 1
    elif p <= 0.1:
        p1 = 2
    else:
        p1 = 0
```

Extrait de la fonction "prog_stat"

L'outil sert également à appliquer un traitement statistique aux données lors de leur import. Par exemple, il peut analyser les relations entre les différentes variables et tester si ces relations sont statistiquement significatives. Les résultats de ces analyses sont ensuite sauvegardés dans un nouveau fichier Excel. Si tout se passe bien, les données sont traitées et sauvegardées correctement. En cas de problème, un message d'erreur est affiché pour indiquer l'échec de l'analyse ou de la sauvegarde des données.

Lancement de la fonction principale

```
#fonction de lancement des fonctions principales
def exe_donnees():
    charger_excel_data()
    prog_stat()
    afficher_donnees("Accident")
```

Cette fonctionnalité essentielle permet de lancer les principales opérations de l'outil. Elle charge d'abord les données, effectue les analyses statistiques, puis affiche les données traitées. Cette fonction déclenche le reste des opérations nécessaires pour l'utilisation de l'outil. Dans cet exemple, nous avons choisi d'afficher une certaine feuille à son exécution, mais cela pourrait être n'importe quelle feuille selon les besoins. Si toutes les opérations se déroulent correctement, les données sont prêtes à être utilisées et affichées.

Ouverture de fichiers externes

```
#fonction d'ouverture du fichier de Dataviz sur PowerBI
def ouvrir_pbi():
    os.startfile("powerbi_grp4.pbix")

#fonction pour ouvrir le tutoriel PDF ou vidéo
def ouvrir_tuto():
    choix = messagebox.askquestion("Ouvrir Tutoriel", "Voulez-vous ouvrir le manuel utilisateur (vidéo YouTube) ou le manuel technique (PDF)?",
                                   icon='question', type='yesnocancel',
                                   default='cancel',
                                   detail="Cliquez 'Oui' pour la vidéo ou 'Non' pour le PDF.")

    if choix == 'yes':
        youtube_url = "https://www.youtube.com/watch?v=dQw4w9WgXcQ"
        try:
            webbrowser.open(youtube_url)
        except Exception as e:
            messagebox.showerror("Erreur", f"Impossible d'ouvrir la vidéo YouTube:\n{e}")
    elif choix == 'no':
        file_path = "tutoriel.pdf"
        try:
            webbrowser.open(file_path)
        except Exception as e:
            messagebox.showerror("Erreur", f"Impossible d'ouvrir le fichier PDF:\n{e}")
```

Cette fonctionnalité permet à l'utilisateur d'ouvrir des fichiers externes importants pour l'analyse des données et l'utilisation de l'outil. En cliquant sur le bouton approprié, le fichier Power BI contenant des visualisations de données s'ouvre automatiquement. De même, l'utilisateur peut ouvrir le tutoriel contenant le manuel technique au format PDF et le manuel utilisateur au format vidéo Youtube, pour l'utilisation de l'outil.

Création et positionnement des éléments graphiques

La dernière section décrit comment les différents éléments de l'interface graphique sont créés et positionnés dans la fenêtre principale. L'interface est composée de plusieurs cadres (frames) pour organiser les éléments visuellement de manière claire et structurée. Les labels, boutons et zones de texte sont placés dans ces cadres pour créer une interface utilisateur intuitive.

Architecture du projet