

SheetChange Automator

progress 100%

Purpose

The SheetChange Automator streamlines and simplifies the process of comparing and tracking changes in dynamic Excel sheets, saving users time and reducing the risk of errors. It empowers data professionals and spreadsheet users to make informed decisions based on accurate and up-to-date data with ease.

Notes for self:

- CMD + SHFT + V (Markdown Preview)
- [Markdown CheatSheet](#)
- [Markdown Emoji CheatSheet](#)

Motivation 🚀

A common request for a **data analyst** is to calculate the difference for 2 excel workbooks where data may change on a month to month or year to year basis. Especially if the format in excel will be consistent. I decided to finally challenge myself and attempt to recreate the script from scratch to automate this task much faster!



Project Goals 🎯

- ☒ Import 2 distinct excel workbooks as 2 separate data frames in Python.
- ☒ Import the data frames into a new excel workbook using `openpyxl` & `xlsxwriter`.
- ☒ Copy over the common descriptive column, both price columns of 2 distinct years into the new excel workbook.
- ☒ Create a new column of the difference in the prices and save.

Overview of Project

From the [USDA site](#), I downloaded 2 datasets on the **average retail price** of Apples from 2013 and 2020. Then in Python, imported the datasets into dataframes, copied relevant data over into a new workbook and created a new column calculating the difference in price between the years.

How to Use the SheetChange Automator:

Step 1: Install Required Libraries Before using the code, ensure that the following Python libraries are installed on your computer:

- `xlsxwriter`
- `pandas`
- `openpyxl`

You can install these libraries using the pip command. For example:

```
pip install xlsxwriter pandas openpyxl
```

Step 2: Prepare Your Excel Files Make sure you have two Excel files that you want to compare and analyze. For this example, let's call them "file1.xlsx" and "file2.xlsx." Your files should contain data that you want to compare, and they should be placed in the same directory as the Python script.

Step 3: Modify the Code for Your Files Edit the Python script to specify the names of your Excel files. Replace the following lines with the names of your files:

```
df1 = pd.read_excel('apples_2013.xlsx')
df2 = pd.read_excel('apples_2020.xlsx')
```

Replace `'apples_2013.xlsx'` and `'apples_2020.xlsx'` with the names of your Excel files:

```
df1 = pd.read_excel('file1.xlsx')
df2 = pd.read_excel('file2.xlsx')
```

Step 4: Run the Code Save your modified Python script and run it by executing the script using the Python interpreter. You can do this from the command line or terminal:

```
python your_script_name.py
```

Step 5: Review the Results

Methods & Technology

- Data tools: [Python](#), [Excel](#)
- Data Source: [apples_2013.xlsx](#), [apples_2020.xlsx](#), [USDA Food Price Outlook](#)
- Software: Visual Studio Code 1.80.0, GitHub Desktop, Microsoft Excel, Github

Featured Deliverables

- [Apple Price Index.xlsx](#)

Contact

If you have any questions, DM me on [Linkedin!](#)