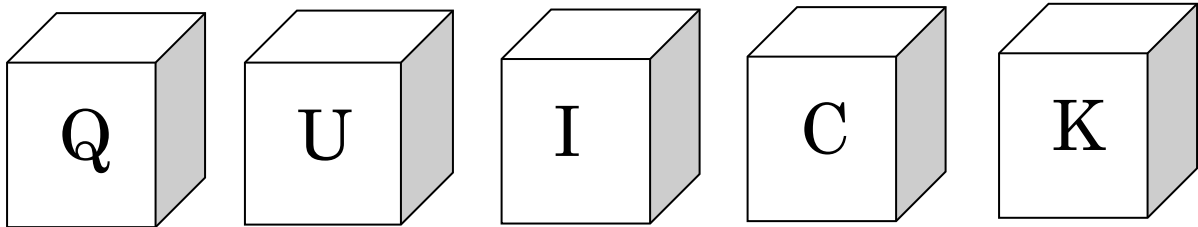


Quick Check

▶ Grammer & Usage



>> PySimpleGUI で GUI ツール作成
～Plotly で月間支出グラフを出力してみよう

Python を使ったグラフ表示の方法として、matplotlib やフレームワークなどを使ったことがある方もいらっしゃると思います。データを視覚化することで現状の課題に対するフィードバックがしやすくなる為、前向きにデータ活用していきたいものですね。(小さい規模での) 導入作業を繰り返し行うことで、周囲に合意を得られる形で情報が共有され、実際に導入実績につながるということもあるのかもしれません。

今回は、GUI ツールで「取り込んだ Excel (csv) ファイルを操作し、PDF ファイルでグラフ表示する」方法について確認していきます。ライブラリには

GUI : PySimpleGUI / グラフ表示 : Plotly / PDF 出力 : kaleido を使用します。

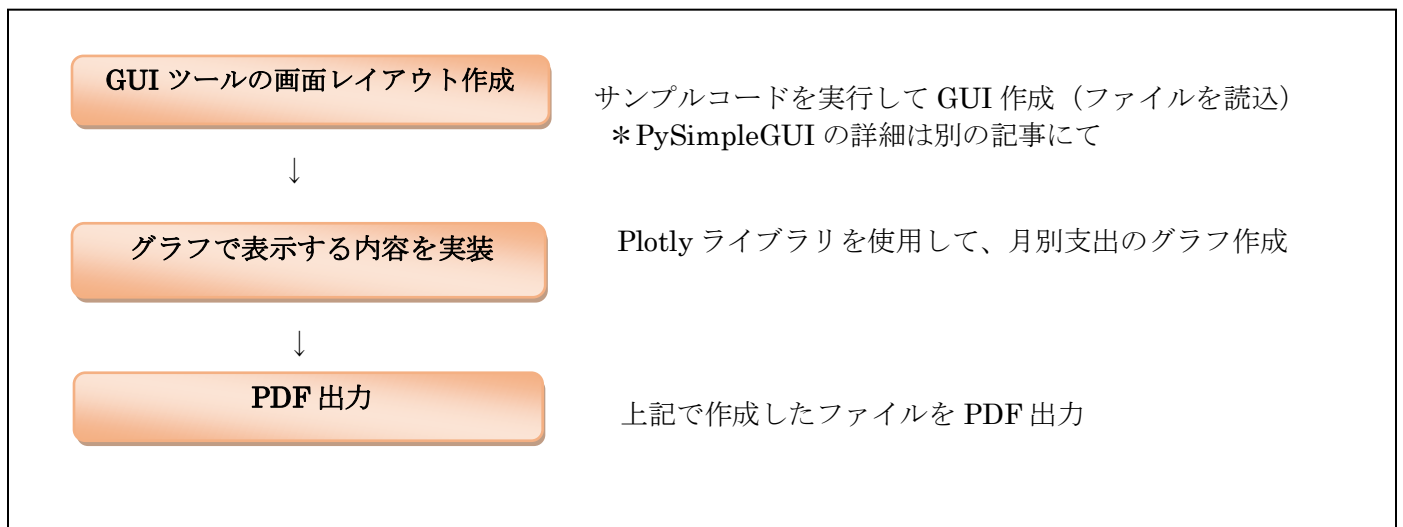
ライブラリを用いてデータを読み書きしたり、グラフを出力したり、PDF 出力するといった具合に、使用用途に合わせて組み合わせて実装していけると良いですね。

[添付資料のダウンロードはこちらから](#)

添付資料にある「ExpenseList.csv」を基にグラフ表示してみたいと思います。

ExpenseList.csv

	A	B	C	D	E	F
1	AccountDate	ExpenseItem	Memo	WithdrawalAmount		
2	2022/4/3	食費	サントリー天然水 1L 9本	1,200		
3	2022/4/3	食費	アサヒ十六茶 630ml 24本	2000		
4	2022/4/3	服飾費	zootie 白ズボン	2500		
5	2022/4/11	美容費	美容液はなこ	4000		
6	2022/4/15	服飾費	スカート秋冬プリーツ (カーキ)	4000		
7	2022/4/27	食費	主にoisixで購入	25,000		
8	2022/4/27	通信費	3月の携帯代	4,000		
9	2022/5/10	雑費	ボールペン黒5本セット	500		
10	2022/5/12	食費	スーパーで食材購入	6,680		
11	2022/5/13	書籍費	書籍を購入	2,260		
12	2022/5/14	食費	アサヒ十六茶 630ml 24本	2000		
13	2022/5/27	食費	主にoisixで購入	27,500		
14	2022/5/27	通信費	4月の携帯代	4,000		
15	2022/5/27	サブスク費	定額サービス利用料	7,000		



○ PySimpleGUI の画面レイアウト作成

それではまず、PySimpleGUI ライブラリを使って GUI ツールの画面レイアウトを作成していきましょう。

問題：サンプルコードを実行して画面レイアウトを作成

以下のサンプルのように（PySimpleGUI の基礎解説の参考サイトや他の Web サイトを参考にしながら）csv ファイルを読み込む画面レイアウトを作成してみましょう。



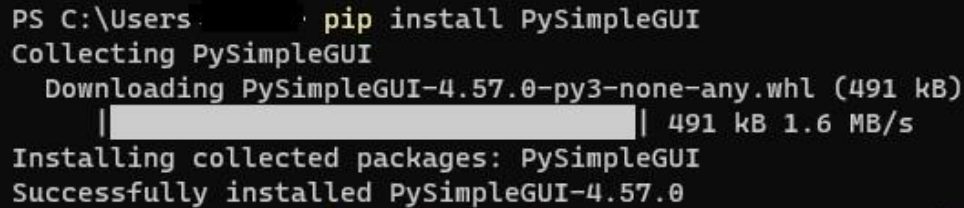
【参考サイト】：PySimpleGUI の使い方についての基礎解説
http://www.k-techlabo.org/blog2/?page_id=1481

- ・インストール

pip という Python のパッケージ管理ツールを使って、「PySimpleGUI」をインストールします。

※ Python インストール時に標準で入っているパッケージ以外をインストールする場合、
「**pip install パッケージ名**」のコマンドで、ローカル環境の PowerShell にてインストールをします。

pip install PySimpleGUI



```
PS C:\Users\...> pip install PySimpleGUI
Collecting PySimpleGUI
  Downloading PySimpleGUI-4.57.0-py3-none-any.whl (491 kB)
    | 491 kB 1.6 MB/s
Installing collected packages: PySimpleGUI
Successfully installed PySimpleGUI-4.57.0
```

▶ サンプルコード（画面レイアウト作成）： layout_PySimpleGUI.py

```
import PySimpleGUI as sg

layout = [
    [sg.Text("ファイル"), sg.InputText(), sg.FileBrowse('ファイル選択', key="inputFilePath")],
    [sg.Text("↓↓PDFにてグラフを表示:"),
    [sg.Button('グラフ出力')]]
]

window = sg.Window("ファイル読込", layout)

# イベントループ
while True:
    event, values = window.read()
    if event == sg.WIN_CLOSED:
        break
    elif event == 'グラフ出力':
        # 処理の内容を記載
        sg.popup('処理内容の実装前')

window.close()
```

○Python でファイル読み込み処理を実装する

■ CSVデータをpandasのデータフレームを利用して読込

問題

先程実行したGUIのソースにpandasのデータフレームを利用して読み込み、CSVファイルをデータ表示させて内容を確認してみましょう。
＊このとき、実行ファイルと同じディレクトリに「ExpenseList.csv」を追加すること

▶ コード

```
import pandas as pd
# pandasでcsvファイルの内容を読み込む
# csvファイルの、1列目にAccountDate, 2列目にExpenseItem, 3列目にMemo, 4列目にWithdrawalAmount
# が入っているとする
df = pd.read_csv('ExpenseList.csv', encoding='SHIFT_JIS')

print(df)
```

○ Pandas とは？

Python でデータ処理をするために作られた高機能なライブラリ。代表的な使い方として Series や DataFrame を使ったデータの処理方法があります。Pandas を利用するには、以下のようにインポートを行います。As キーワードを使用して pd で呼び出せるようにします。

In

```
import pandas as pd
```

▷使用例

```
# 1 次元データの利用
ser = pd.Series([10, 20, 30, 40])
ser

# 2 次元データの利用
df = pd.DataFrame([10, "a", True],
                  [20, "b", False],
                  [30, "c", False],
                  [40, "d", True])
df
```

csv や Excel のデータを読み込んだり、列や行を削除したり、フィルターをかけて抽出をしたりといった Excel やデータベース言語の SQL でできることが Pandas の機能にはあります。

○Python でグラフ表示～ Plotly ライブラリの基礎

次に、グラフ描画するにあたって、使用するライブラリの基礎を押さえておきましょう。

・Plotly とは？

インタラクティブな可視化ができるライブラリの中で、「Plotly express」という Plotly のラッパーを使うと、より簡潔な方法でデータを可視化することができます。

今回は、「Plotly express」を使って、折れ線グラフ、棒グラフ、円グラフを描画する例をみていきます。

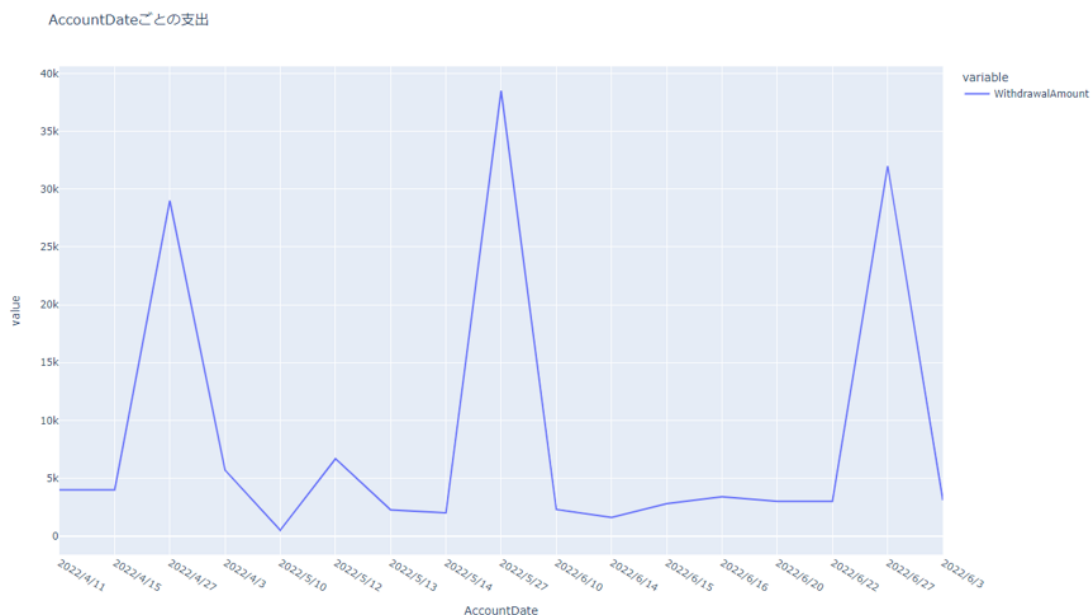
問題 : Plotly の基本①～折れ線グラフの書き方を理解する

ExpenseList.csv のに各月（AccountDate ごと）の支出合計額の推移を Plotly express を利用して、折れ線グラフで表示してください。

作業の際、PDF 出力する為に「pip install kaleido」を実施してください。

【書式】：折れ線グラフ（Plotly express）

```
import plotly.express as px
px.line(対象のデータフレーム, title='AccountDate ごとの支出')
```



▷ コード

```
import PySimpleGUI as sg
import pandas as pd
import plotly.express as px

layout = [
    [sg.Text("ファイル"), sg.InputText(key='-file-'), sg.FileBrowse('ファイル選択',
key="inputFilePath", target='-file-')],
    [sg.Text("↓↓ PDF にてグラフを表示:"),
    [sg.Button('グラフ出力')
]

window = sg.Window("ファイル読込", layout)

# イベントループ
while True:
    event, values = window.read()
    if event == sg.WIN_CLOSED:
        break
    elif event == 'グラフ出力':
        # ファイルパスを取得
        f = values['-file-']

        # pandas で csv ファイルの内容を読み込む
        # csv ファイルの、1 列目に AccountDate, 2 列目に ExpenseItem, 3 列目に Memo,
4 列目に WithdrawalAmount が入っているとする
        df = pd.read_csv(f, encoding='SHIFT_JIS')
        # WithdrawalAmount 列を数値変換する
        df['WithdrawalAmount'] = df['WithdrawalAmount'].apply(lambda x: x.replace(',', '')).astype('int')

        #作成したデータフレームを基にグラフを作成
        df2 = df[['AccountDate', 'WithdrawalAmount']].groupby('AccountDate').sum(numeric_only=True)
        print(df2)

        #折れ線グラフを作成
        fig = px.line(df2, title='AccountDate ごとの支出')
        fig.show()
        #出力内容を PDF でグラフ表示 (kaleido ライブラリ)
        fig.write_image("./test.pdf")

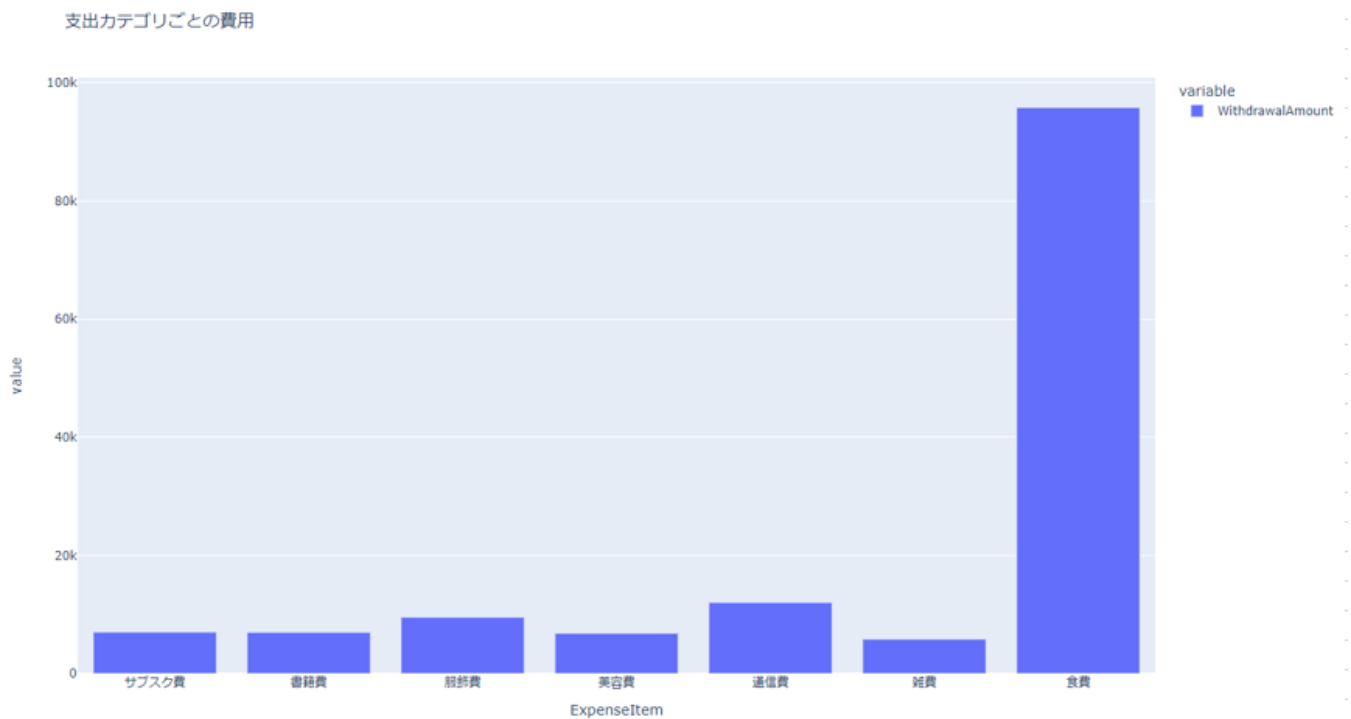
window.close()
```

問題 : Plotly の基本②～棒グラフの書き方を理解する

前の問題のグラフ表示方法を参考にして
ExpenseList.csv の支出カテゴリごとの合計額を
Plotly express を利用して、棒グラフで表示してください。

【書式】: 棒グラフ (Plotly express)

```
import plotly.express as px
px.bar(対象のデータフレーム, title='支出カテゴリごとの費用')
```



問題 : Plotly の基本③～円グラフの書き方を理解する

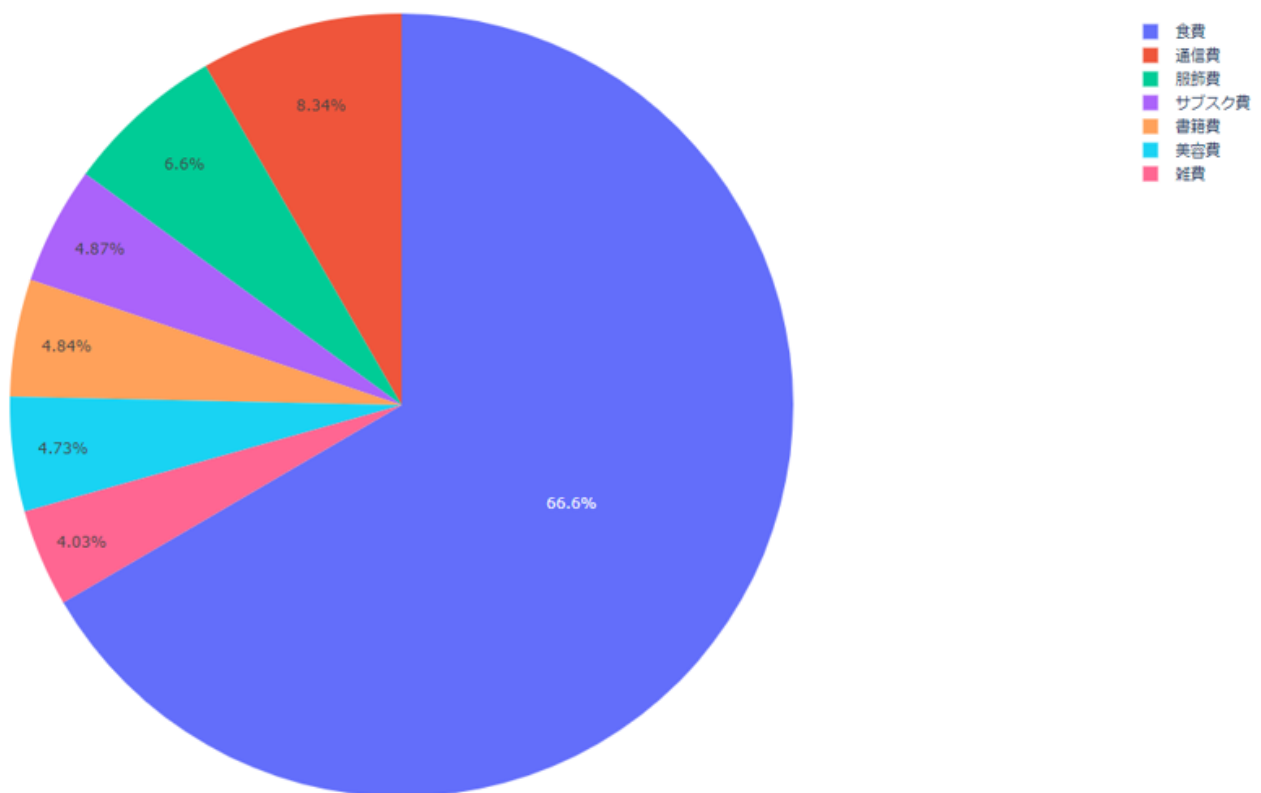
前の問題のグラフ表示方法を参考にして

ExpenseList.csv の支出カテゴリごとの合計費用の割合について

Plotly express を利用して、円グラフで表示してください。

【書式】：棒グラフ（Plotly express）

```
import plotly.express as px
px.pie(対象のデータフレーム, values='WithdrawalAmount', names='ExpenseItem')
```



=====
業務で Python を使ってツールを作る機会がある方もいらっしゃると思います。自身の PC で Visual Studio Code を使って簡単に今回のサンプルを試すことができますので、この機会に Python に触れる時間を作っていただけると嬉しい限りです。

以上となります。