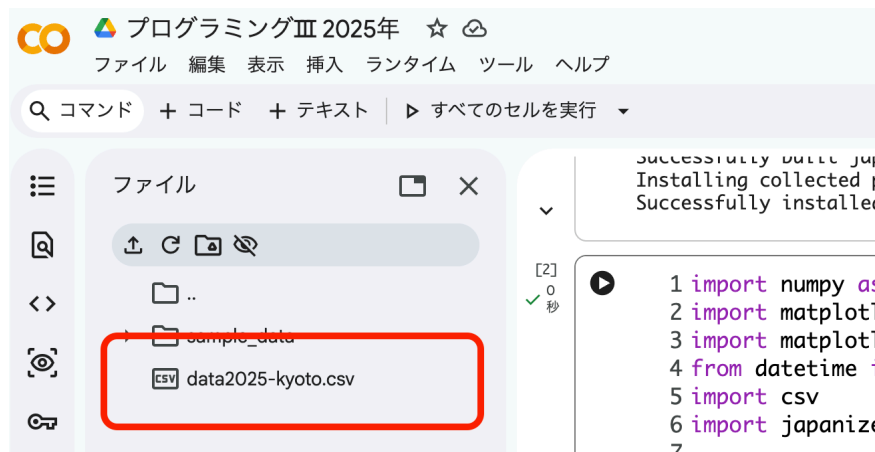


1 セットアップ

日本語フォントを利用するため、追加モジュールのインストールを行う。

```
1 !pip install japanize-matplotlib
```

データファイル「data2025-kyoto.csv」を以下の場所に保存する。



演習で利用するモジュールを読み込む。

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4 from datetime import datetime
5 import csv
6 import japanize_matplotlib # 日本語利用
7
8 csv_path = 'data2025-kyoto.csv' # ファイルを指定
```

2 CSV ファイルからデータの読み込み

csv モジュールでファイルからデータを読み込む。

```
1 # CSV 読み込み (ヘッダー行の取得)
2 with open(csv_path, 'r', encoding='utf-8') as f:
3     r = csv.reader(f)
4     header = next(r)
5     rows = [row for row in r]
6
7 # 列名 → インデックス
8 col_idx = {name: i for i, name in enumerate(header)}
```

つづいて、numpy の配列形式に変換する。

```
1  # 数値列をNumPy 配列へ
2  def to_float_array(col_name):
3      i = col_idx[col_name]
4      return np.array([float(r[i]) for r in rows], dtype=float)
5
6  def to_int_array(col_name):
7      i = col_idx[col_name]
8      return np.array([int(r[i]) for r in rows], dtype=int)
9
10 # 年月の処理
11 ym_str = np.array([r[col_idx['時間軸 (月次) ']] for r in rows])
12 years = np.array([int(s[:4]) for s in ym_str], dtype=int)
13 months = np.array([int(s[4:6]) for s in ym_str], dtype=int)
14
15 # 各列を格納
16 data = {
17     '年月文字列': ym_str,
18     '年': years,
19     '月': months,
20     '日数': to_int_array('日数'),
21     '食料円': to_float_array('1 食料【円】'),
22     'チョコ円': to_float_array('チョコ【円】'),
23     'アイス円': to_float_array('アイス【円】'),
24     '降水量合計 mm': to_float_array('降水量の合計 (mm)'),
25     '平均湿度%': to_float_array('平均湿度 (%)'),
26     '最高気温°C': to_float_array('日最高気温の平均 (°C)'),
27     '日照率%': to_float_array('日照率 (%)'),
28     '平均風速 m/s': to_float_array('平均風速 (m/s)')
29 }
```

2.1 提出課題

以下のコードの実行結果を記録する。

```
1  # ヘッダー確認
2  print('列名:', header)
3  print('行数:', len(rows))
4  print('期間サンプル:', rows[0][col_idx['時間軸 (月次) ']], '→', rows[-1][col_idx['時間軸 (月次) ']])
5
6  # 1日あたりの支出( 指標作成)
```

```
7 data['食料_1 日平均円'] = data['食料円'] / data['日数']
8 data['チョコ_1 日平均円'] = data['チョコ円'] / data['日数']
9 data['アイス_1 日平均円'] = data['アイス円'] / data['日数']
10
11 print('食料_1 日平均円 =', data['食料_1 日平均円'])
12 print('チョコ_1 日平均円 =', data['チョコ_1 日平均円'])
13 print('アイス_1 日平均円 =', data['アイス_1 日平均円'])
```

3 基礎統計の確認

平均値, 中央値, 最小値, 最大値, 標本標準偏差などの基礎統計量の計算を行う.

```
1 # 基礎統計を計算する関数
2 def describe(arr):
3     return {
4         'count': arr.size,
5         'mean': float(np.mean(arr)),
6         'median': float(np.median(arr)),
7         'min': float(np.min(arr)),
8         'max': float(np.max(arr)),
9         'std': float(np.std(arr, ddof=1)) # 標本標準偏差
10    }
11
12 numeric_keys = ['食料円', 'チョコ円', 'アイス円', '食料_1 日平均円', 'チョコ_1 日平均円', 'アイ
13               ス_1 日平均円', '降水量合計 mm', '平均湿度%', '最高気温°C', '日照率%', '平均風速 m_s']
14
15 for k in numeric_keys:
16     stats = describe(data[k])
17     print(f'[{k}]', stats)
```

3.1 提出課題

基礎統計を記録する.

4 折れ線グラフによる可視化

アイスの支出と最高気温の折れ線グラフを重ねて描画する.

```
1 # 横軸ラベルの作成( YYYY-MM 形式)
2 labels = np.array([f"{y}-{m:02d}" for y, m in zip(data['年'], data['月'])])
3
4 fig, ax1 = plt.subplots(figsize=(12, 4))
```

```
5 ax1.plot(labels, data['アイス円'], color='tab:blue', label='アイス円')
6 ax1.set_ylabel('円')
7 ax1.tick_params(axis='x', rotation=90)
8
9 ax2 = ax1.twinx()
10 ax2.plot(labels, data['最高気温°C'], color='tab:red', label='最高気温°C')
11 ax2.set_ylabel('°C')
12
13 ax1.set_title('アイス円 と 最高気温°C の推移')
14 fig.legend(loc='upper left')
15 plt.tight_layout()
16 plt.show()
```

4.1 提出課題

チョコの支出と最低気温の折れ線グラフを重ねて描画したものを記録する.

5 各データの相関を可視化

支出データと気象データの相関をヒートマップで描画する.

```
1 # 行列の作成( 列方向=特徴量)
2 features = ['食料円', 'チョコ円', 'アイス円', '降水量合計 mm', '平均湿度%', '最高気温°C', '日照率%', '平均風速 m/s']
3 X = np.vstack([data[k] for k in features]) # shape: (n_features, n_samples)
4
5 # 相関行列( 各行同士の相関)
6 C = np.corrcoef(X)
7
8 # ヒートマップ
9 plt.figure(figsize=(8,6))
10 plt.imshow(C, cmap='RdBu', vmin=-1, vmax=1)
11 plt.colorbar(label='相関係数')
12 plt.xticks(np.arange(len(features)), features, rotation=90)
13 plt.yticks(np.arange(len(features)), features)
14 plt.title('支出・気象の相関行列')
15 plt.tight_layout()
16 plt.show()
```

5.1 提出課題

上記コードで描画されるヒートマップを記録する.

6 条件抽出と比較

夏（7月から9月）と冬（12月から2月）のアイスの支出平均を計算する。また、降水量上位 25% でのチョコの支出平均を計算する。

```
1 is_summer = (data['月'] >= 7) & (data['月'] <= 9)
2 is_winter = (data['月'] == 12) | (data['月'] <= 2)
3
4 print('アイス円 平均(夏) : ', round(np.mean(data['アイス円'][is_summer]),1))
5 print('アイス円 平均(冬) : ', round(np.mean(data['アイス円'][is_winter]),1))
6
7 q75 = np.quantile(data['降水量合計 mm'], 0.75)
8 high_rain = data['降水量合計 mm'] >= q75
9 print('チョコ円 平均(降水量 上位 25%) : ', round(np.mean(data['チョコ円'][high_rain]),1))
10 print('チョコ円 平均(それ以外) : ', round(np.mean(data['チョコ円'][~high_rain]),1))
```

6.1 提出課題

夏（7月から9月）と冬（12月から2月）のチョコの支出平均を計算し記録する。また、最高気温 25% でのアイスの支出平均を計算し記録する。

7 各支出の季節性

アイスの支出金額について季節指数（月平均/全体平均）の棒グラフを描画する。

```
1 # 季節指数: 月別平均 / 全体平均
2 month_means = np.array([np.mean(data['アイス円'][data['月']==m]) for m in range(1,13)])
3 seasonal_index = month_means / np.mean(data['アイス円'])
4
5 plt.figure(figsize=(8,4))
6 plt.bar(np.arange(1,13), seasonal_index)
7 plt.xticks(np.arange(1,13))
8 plt.ylabel('季節指数')
9 plt.title('アイス円の季節指数（月別）')
10 plt.tight_layout()
11 plt.show()
```

7.1 提出課題

チョコの支出金額について季節指数の棒グラフを記録する。