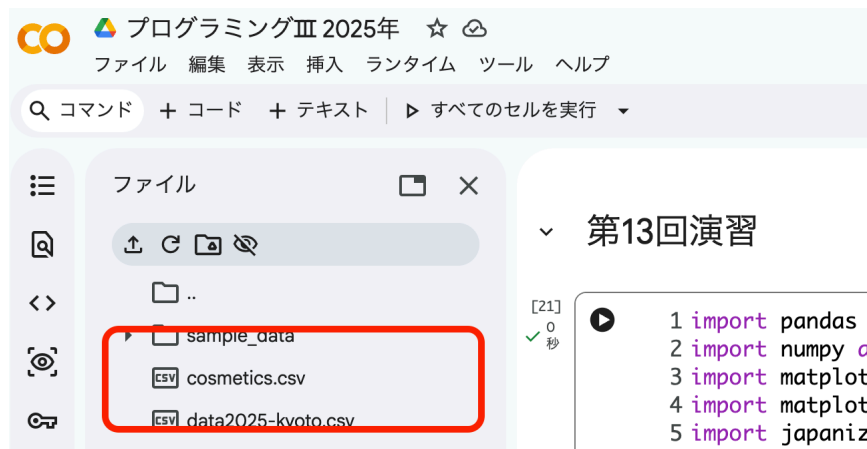


1 セットアップ

データファイル「cosmetics.csv」を以下の場所に保存する。



演習で利用するモジュールとデータファイルを読み込む。

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib
5 import japanize_matplotlib
6
7 pd.set_option('display.max_rows', 20)
8 pd.set_option('display.max_columns', None)
9
10 CSV_PATH = 'cosmetics.csv'
```

2 データの読み込みと整形

pandas の read_csv() メソッドを用い、csv ファイルからデータを読み込む。

```
1 # 読み込み (不正行はスキップ)
2 raw = pd.read_csv(CSV_PATH, engine='python', on_bad_lines='skip')
```

読み込んだデータを整形する。

```
1 # 列名を扱いやすくリネーム
2 rename_map = {
3     'Label': 'カテゴリ',
4     'Brand': 'ブランド',
5     'Name': '製品名',
6     'Price': '価格',
7     'Rank': '評価',
```

```
8     'Ingredients': '成分',
9     'Combination': '混合肌',
10    'Dry': '乾燥肌',
11    'Normal': '普通肌',
12    'Oily': '脂性肌',
13    'Sensitive': '敏感肌'
14 }
15
16 # 実際の列に存在するキーのみ置換
17 usable_map = {k: v for k, v in rename_map.items() if k in raw.columns}
18 df = raw.rename(columns=usable_map).copy()
19
20 # 数値列の型変換(安全なcoerce)
21 for col in ['価格', '評価']:
22     if col in df.columns:
23         df[col] = pd.to_numeric(df[col], errors='coerce')
24
25 # 肌タイプ列を数値/ブール化
26 skin_cols = [c for c in ['混合肌', '乾燥肌', '普通肌', '脂性肌', '敏感肌'] if c in df.columns]
27 for c in skin_cols:
28     df[c] = pd.to_numeric(df[c], errors='coerce').fillna(0).astype(int)
29
30 # 空白のトリム(代表列)
31 for c in [x for x in ['カテゴリ', 'ブランド', '製品名'] if x in df.columns]:
32     df[c] = df[c].astype(str).str.strip()
```

2.1 提出課題

以下のコードの実行結果を記録する。記録は画面のスクリーンショットでも良い。

```
1 # 列名確認
2 print('--- 原データ 列 ---')
3 print(list(raw.columns))
4
5 # 整形後の先頭 5行を表示
6 print('--- 整形後の先頭 5行 ---')
7 df.head()
```

3 データの品質確認

読み込んだデータの欠損や重複を確認する。

```
1 print('--- DataFrame.info ---')
2 df.info()
3
4 print('--- 欠損件数 ---')
5 print(df[['価格', '評価']].isna().sum())
6
7 print('--- 重複件数(ブランド+製品名) ---')
8 if set(['ブランド', '製品名']).issubset(df.columns):
9     dup = df.duplicated(subset=['ブランド', '製品名']).sum()
10    print('重複:', dup)
11 else:
12    print('重複: 判定用列が足りません')
```

4 基礎的な統計量の確認とヒストグラムによる可視化

価格と評価の記述統計 (describe) を表示し、ヒストグラムで可視化する。

```
1 print('--- 記述統計(価格・評価) ---')
2 print(df[['価格', '評価']].describe().round(2))
3
4 # ヒストグラム
5 ax = df['価格'].dropna().plot.hist(bins=30, alpha=0.7, title='価格の分布')
6 ax.set_xlabel('価格'); ax.set_ylabel('件数')
7 plt.show()
8 ax = df['評価'].dropna().plot.hist(bins=30, alpha=0.7, title='評価の分
   布', color='tab:orange')
9 ax.set_xlabel('評価'); ax.set_ylabel('件数')
10 plt.show()
```

4.1 提出課題

上記コードの実行結果を記録する。記録は画面のスクリーンショットでも良い。

5 カテゴリ別、ブランド別にデータを集計

カテゴリ別の件数と平均価格、平均評価を集計し、ブランド別の件数トップ 10 を棒グラフとして表示する。

```
1 # カテゴリ別集計
2 if 'カテゴリ' in df.columns:
3     grp = df.groupby('カテゴリ').agg(
4         件数=('製品名', 'count') if '製品名' in df.columns else ('価格', 'count'),
5         平均価格=('価格', 'mean'),
```

```

6         中央価格=('価格','median'),
7         平均評価=('評価','mean'),
8         中央評価=('評価','median')
9     ).round(2)
10    print('--- カテゴリ別 集計 ---')
11    display(grp)
12
13    # 可視化: 平均価格・平均評価
14    ax = grp['平均価格'].plot.bar(title='カテゴリ別 平均価格', color='tab:blue')
15    ax.set_xlabel('カテゴリ'); ax.set_ylabel('価格'); plt.tight_layout(); plt.show()
16    ax = grp['平均評価'].plot.bar(title='カテゴリ別 平均評価', color='tab:orange')
17    ax.set_xlabel('カテゴリ'); ax.set_ylabel('評価'); plt.tight_layout(); plt.show()
18
19    if 'ブランド' in df.columns:
20        brand_counts = df['ブランド'].value_counts().head(10)
21        display(brand_counts)
22        ax = brand_counts.plot.bar(title='ブランド別 件数 トップ10', color='tab:green')
23        ax.set_xlabel('ブランド'); ax.set_ylabel('件数'); plt.tight_layout(); plt.show()

```

5.1 提出課題

上記コードの実行結果を記録する。記録は画面のスクリーンショットでも良い。

6 ヒートマップによる可視化

ブランドを行、カテゴリを列とした行列を作成しセル値として価格の中央値（または評価の中央値）を計算しヒートマップとして可視化する。

```

1    if set(['ブランド','カテゴリ']).issubset(df.columns):
2        brand_order = df['ブランド'].value_counts().head(12).index
3        sub = df[df['ブランド'].isin(brand_order)].copy()
4        pivot_price = sub.pivot_table(index='ブランド', columns='カテゴリ', values='価格', aggfunc='median')
5        pivot_rank = sub.pivot_table(index='ブランド', columns='カテゴリ', values='評価', aggfunc='median')
6
7        display(pivot_price); display(pivot_rank)
8
9        # ヒートマップはMatplotlibのimshowを使い、ラベルはpandasのindex/columnsを利用
10       for mat, title, cmap in [(pivot_price, '×カテゴリ Brand 価格中央値(ヒートマップ)', 'YlGnBu'),
11                                (pivot_rank, '×カテゴリ Brand 評価中央値(ヒートマップ)', 'OrRd')]:
12           arr = mat.values

```

```
13     ax = plt.imshow(arr, cmap=cmap, aspect='auto')
14     plt.colorbar(ax)
15     plt.xticks(range(mat.shape[1]), mat.columns, rotation=90)
16     plt.yticks(range(mat.shape[0]), mat.index)
17     plt.title(title)
18     plt.tight_layout(); plt.show()
```

6.1 提出課題

上記コードの実行結果を記録する。記録は画面のスクリーンショットでも良い。

7 各ブランドの肌タイプごとの適合率を集計

各ブランドが、どの肌タイプ（混合／乾燥／普通／脂性／敏感）にどれだけ適合しているか、比率（適合率）を集計し、上位 10 ブランドの結果を棒グラフとして表示する。

```
1  if skin_cols:
2      brand_order = df['ブランド'].value_counts().head(10).index if 'ブラン
3      sub = df[df['ブラン
4      rate = sub.groupby('ブランド')[skin_cols].mean().round(2)
5      display(rate)
6
7      # pandas の plot で棒グラフ（複数列）
8      ax = rate.plot(kind='bar', figsize=(12,5), title='ブランド別 肌タイプ適合率(上位 10)')
9      ax.set_ylabel('適合率'); ax.set_xlabel('ブランド'); plt.tight_layout(); plt.show()
```

7.1 提出課題

上記コードの実行結果を記録する。記録は画面のスクリーンショットでも良い。

8 各製品のコスパを比較

製品ごとのコスパ（評価 / 価格）を新しい指標として作り、カテゴリ別に上位 5 製品を抽出する。抽出結果を表で確認し、代表カテゴリで棒グラフにして比較する。

```
1  if set(['価格', '評価']).issubset(df.columns):
2      df['value_score'] = (df['評価'] / df['価格']).replace([np.inf, -np.inf], np.nan)
3      if 'カテゴリ' in df.columns:
4          top_per_cat = []
5          for cat, g in df.dropna(subset=['value_score']).groupby('カテゴリ'):
6              g2 = g.sort_values('value_score', ascending=False).head(5)
```

```
7         top_per_cat.append((cat, g2[['ブランド', '製品名', '価格', '評価', 'value_score']]))
8     for cat, tbl in top_per_cat:
9         print(f'--- コスパ上位(カテゴリ: {cat}) ---')
10        display(tbl)
11    if top_per_cat:
12        major_cat = max(top_per_cat, key=lambda x: x[1].shape[0])
13        tbl = major_cat[1].copy()
14        tbl.index = tbl['製品名'].astype(str)
15        ax = tbl['value_score'].plot.bar(title=f'コスパ上位
16        5({major_cat[0]}', color='tab:purple')
17        ax.set_ylabel('value_score (評価/価格)'); ax.set_xlabel('製品
18        名'); plt.tight_layout(); plt.show()
```

8.1 提出課題

Moisturizer の上位 10 製品の結果（棒グラフ）を記録する。