

1. 処理フロー

- 1.1 テーブルを「主要カテゴリ」「サブカテゴリ」「総額」の3つに分ける
- 1.2 ヘッダー加工
- 1.3 サブカテゴリに主要カテゴリ項目作成
- 1.4 結合
- 1.5 欠損埋め、ヘッダー整理

2. 参考コード

2.1 テーブルを「主要カテゴリ」「サブカテゴリ」「総額」の3つに分ける

```
# 「# 横持変換」の続きから
# 本来カラム名がすべて設定されていることが望ましい
concatenated_tables.rename(
    columns={concatenated_tables.columns[0]: "カテゴリ"}, inplace=True)

category_dict = {
    "衣料品": ["紳士服・洋品", "婦人服・洋品", "子供服・洋品", "その他衣料品"],
    "身のまわり品": None,
    "雑貨": ["化粧品", "美術・宝飾・貴金属", "その他雑貨"],
    "家庭用品": ["家具", "家電", "その他家庭用品"],
    "食料品": ["生鮮食品", "菓子", "惣菜", "その他食料品"],
    "食堂喫茶": None,
    "サービス": None,
    "その他": None
}

# 主要カテゴリ
category_keys = category_dict.keys()
df_main_category = concatenated_tables[
    concatenated_tables["カテゴリ"].isin(category_keys)]

# サブカテゴリ
category_values = [
    item for sublist in category_dict.values() for item in sublist]
df_sub_category = concatenated_tables[
    concatenated_tables["カテゴリ"].isin(category_values)]

# 総額
df_total = concatenated_tables[concatenated_tables["カテゴリ"] == "総額"]
```

2.2 ヘッダー加工

```
# 面倒なので無理やりカラム名変更
df_main_category.columns = [
    "主要カテゴリ", "主要カテゴリ売上高(千円)", "主要カテゴリ構成比(%)", "主要カテゴリ対前年増減(-)率(%)", "年"]
df_sub_category.columns = [
    "サブカテゴリ", "サブカテゴリ売上高(千円)", "サブカテゴリ構成比(%)", "サブカテゴリ対前年増減(-)率(%)", "年"]
df_total.drop(columns=["カテゴリ", "構成比(%)"], inplace=True)
df_total.columns = ["主要カテゴリ総額売上高(千円)", "主要カテゴリ総額対前年増減(-)率(%)", "年月"]
```

2.3 サブカテゴリに主要カテゴリ項目作成

```
# サブカテゴリに主要カテゴリ列追加
def add_main_category_column(df, category_dict):
    category_map = {v: k for k, values in category_dict.items()
                    for v in values}
    df["主要カテゴリ"] = df["サブカテゴリ"].map(category_map)
    return df

df_sub_category = add_main_category_column(df_sub_category, category_dict)
```

2.4 結合

```
# 結合
df_finaly = df_main_category.merge(
    df_sub_category, how="left", on=["主要カテゴリ", "年月"]) \
    .merge(df_total, how="left", on="年月")
```

2.5 欠損埋め、ヘッダー整理

```
# カラム順変更・欠損値補完
df_finaly = df_finaly[[
    col for col in df_finaly.columns if col != "年月"] + ["年月"]]
df_finaly.fillna("-", inplace=True)
```