レシピ61 SATの加重平均点を州ごとにapplyで計算（P171）

- `groupby`メソッドには、関数を使って各グループで計算を行うメソッドが4種類あり、戻り値は以下

メソッド	戻り値
agg	スカラー値
filter	Boolean
tranform	渡されるグループと同じ長さのSeries
apply	スカラー値、Series、DataFrame **柔軟に使える！**

- `apply`は、グループに対して1回しか呼び出せないので、グループ分けしていないカラムで呼び出される`transform`や`agg`と対照的

- このレシピでは、数学と言語能力のSAT点数の加重平均を、大学データセットを使い、州ごとに求める
- 学校ごとの学部学生の人数で点数に重みを与える

（1）collegeデータセットを読み込み、UGDS、SATMTMID、SATVRMIDカラムの**欠損値がある行を削除する**（欠損なしデータが必須）

- 'UGDS'・・・学部学生数
- 'SATMTMID'・・・SAT Math Median（大学進学のための標準テスト（Standardized Test）の、言語能力の点数・中央値）
- 'SATVRMID'・・・SAT Verbal Median（SATの数学の点数・中央値）

In [112…
```
college = pd.read_csv('college.csv')
subset = ['UGDS', 'SATMTMID', 'SATVRMID']
college2 = college.dropna(subset=subset)

print(f'college.shape:{college.shape}')
print(f'college2.shape:{college2.shape}')
```

```
college.shape:(7535, 27)
college2.shape:(1184, 27)
```

In [113…
```
print('▼college2.sample\n')
college2.head(2)
```

```
▼college2.sample
```

Out[113…

	INSTNM	CITY	STABBR	HBCU	MENONLY	WOMENONLY	RELAFFIL	SATVRMID	SATMTMID	DISTANCEONLY	…	UGDS_2MOR	UGDS_NRA	UGDS_UNKN	PPTUG_EF	CURROPER	PCTPELL	PCTFLOAN	UG25ABV	MD_EARN_WNE_P10	GRAD_DEBT_MDN_SUPP
0	Alabama A & M University	Normal	AL	1.0	0.0	0.0	0	424.0	420.0	0.0	…	0.0000	0.0059	0.0138	0.0656	1	0.7356	0.8284	0.1049	30300	33888
1	University of Alabama at Birmingham	Birmingham	AL	0.0	0.0	0.0	0	570.0	565.0	0.0	…	0.0368	0.0179	0.0100	0.2607	1	0.3460	0.5214	0.2422	39700	21941.5

2 rows × 27 columns

（2）次にSATの数学だけ加重平均を取る関数を作成する

- 加重平均とは、{学生数＊平均点}の合計　÷学生数の合計

In [114…
```
def weighted_math_average(df):
    weighted_math = df['UGDS'] * df['SATMTMID']
    return int(weighted_math.sum() / df['UGDS'].sum())
```

（3）州でグループ化して、applyメソッドにこの関数を渡すと、加重平均のスカラー値が得られる

In [115…
```
college2.groupby('STABBR').apply(weighted_math_average).head()
```

Out[115…
```
STABBR
AK    503
AL    536
AR    529
AZ    569
CA    564
dtype: int64
```

（4）参考までに`agg`メソッドに同じ関数を渡すとどうなるか

In [116…
```
# college2.groupby('STABBR').agg(weighted_math_average).head()
# これはエラーになるため、割愛（本と同じようにならない）
# KeyError:'UGDS'
```

（5）

- `apply`のよいところは、Seriesを返して複数の新たなカラムを作れること
- 返されたSeriesのインデックスは新たなカラム名になる
- ユーザー定義関数を、2つのSAT点の加重平均および算術平均を各グループの大学数とともに計算するように修正する
- この5つの値をSeriesで返す

- Pythonの`OrderedDict`
  - 順序付き辞書
  - Ptyhonの通常の辞書は要素の順番を保持しない（Python3.7以降は順序付きらしい？）
  - collectionsモジュールに順番が保持された辞書として`OrderDict`が用意されている

In [117…
```
from collections import OrderedDict

def weighted_average(df):

    data = OrderedDict()

    # 加重平均（学生数*SAT数学or言語で戻した点数 ÷ 全学生数）
    weight_m = df['UGDS'] * df['SATMTMID']
    weight_v = df['UGDS'] * df['SATVRMID']
    wm_avg = weight_m.sum() / df['UGDS'].sum()
    wv_avg = weight_v.sum() / df['UGDS'].sum()

    # 加重平均と算術平均の列を作成
    data['weighted_math_avg'] = wm_avg
    data['weighted_verbal_avg'] = wv_avg
    data['math_avg'] = df['SATMTMID'].mean()
    data['verbal_avg'] = df['SATVRMID'].mean()
    data['count'] = len(df)

    return pd.Series(data)

    # return pd.Series(data, dtype='int')
    # 書籍ではdtype='int'になっているが、これだと以下エラーになる
    # ValueError: Trying to coerce float values to integers
```

In [118…
```
college2.groupby('STABBR').apply(weighted_average).head(10)
```

Out[118…

STABBR	weighted_math_avg	weighted_verbal_avg	math_avg	verbal_avg	count
AK	503.000000	555.000000	503.000000	555.000000	1.0
AL	536.137917	533.383387	504.285714	508.476190	21.0
AR	529.112332	504.876157	515.937500	491.875000	16.0
AZ	569.313985	557.303350	536.666667	538.333333	6.0
CA	564.945420	539.316605	562.902778	549.083333	72.0
CO	553.123820	547.033996	540.214286	537.714286	14.0
CT	545.341834	533.417563	522.500000	517.857143	14.0
DC	621.905104	623.514036	588.333333	589.166667	6.0
DE	569.954949	553.534560	495.000000	486.666667	3.0
FL	565.324731	565.815873	521.842105	529.289474	38.0

（補足）割愛（P175）

- Seriesではなく、DataFrameでの返し方
