

▶ Section 1-1

そ う か へ い き ん

相加平均

英 名 Arithmetic mean

別 名 算術平均

レベル 小学5年

標本 統計の対象とするデータの集まりから取り出した一部のデータ

$$x = \{x_1, x_2, \dots, x_n\} \quad \text{..... 標本}$$

 x_i 標本 x の i 番目の要素

$$\text{相加平均: } \mu = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{..... 式 a}$$

n 標本の数是一定とは限らないため、変数 n を使って標本数を表す

Σ シグマ (Σ 、 σ) はギリシャ文字の1つで、ラテン文字の「エス (S 、 s)」に相当する。総和 (英語で summation) を意味する記号として使われる

$$\text{相加平均: } \mu = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad \text{..... 式 b}$$

μ ミュー (M 、 μ) はギリシャ文字の1つで、ラテン文字の「エム (M 、 m)」に相当する。平均 (英語で mean) を意味する記号として使われる

相加平均とは

日常生活で使う平均のことです。具体的に言えば、3人がテストを受けてそれぞれ55点、40点、70点だった場合、 $\frac{55+40+70}{3} = 55$ で出すアレです。「算術平均」とも言います。

小学校のころに習った人が多いと思いますが*1、これを数式に直すと、式aのようにになります。Σのような文字に慣れていないと「なんだこれ?」と思ってしまいそうな記述ですね。内容自体は小学5年生でもできる計算なので、覚えてしまえば難しくありません。

本書の序盤ではこうした簡単な問題を取り扱い、数式をコードに変換する作業に慣れていきます。

たしかめよう

初回なので、簡単なデータを使用します。元SMAPメンバー5人の身長の「相加平均」を求めてみましょう。

■ 標本データを用意する

元SMAPメンバー5人の身長を配列に格納します。公表されているデータに基づくと、以下のようになります。

```
x = [170, 170, 176, 176, 183] # 単位はcm
```

この5人の身長の相加平均を求めるために(暗算でも求められそうですが)数式を読んでコードに直していきましょう。

■ 式a をコードに直す

Σ(シグマ)は「総和」を表すのでした。

Σの下に書かれている $i = 1$ が「初期値」です。Σの上に書かれている n が「終了値」です。初期値から終了値まで i を1ずつ増やし、Σの右に書かれた式にあてはめた結果を足し算していきます。Pythonで書くと、次のようになります。

```
avg = 0

for i in range(1, n + 1):
    avg += # ここにΣの右に書かれた処理を書く
```

x_i は標本 x の i 番目の要素を意味します。Pythonで書くと、次のようになります。

*1 今の学習カリキュラムでは小学5年生で学ぶそうです。

```
x[i]
```

ここまでの内容を踏まえて、総和 $\sum_{i=1}^n x_i$ をPythonで書くと、次のようになります。

```
avg = 0

for i in range(1, n + 1):
    avg += x[i]
```

この総和に $\frac{1}{n}$ を掛ければ相加平均を求められます。 n は標本の数、つまり「配列 `x` の長さ」です。

従って、「式a」をPythonで書くと、次のようになります。なお、配列の添字（インデックス）は1番目ではなく0番目から始めるので、`range(1, n + 1)` を `range(n)`^{*2} に修正しています。

```
x = [170, 170, 176, 176, 183]
n = len(x)
avg = 0

for i in range(n):
    avg += x[i]

(1 / n) * avg
```

上記のコードを実行すると、元SMAPメンバーの身長 of 相加平均は175.0cmだと分かります。

```
175.0
```

Σ を使った数式はこの先たくさん出てきます。ここで使い方を覚えておきましょう。

■ NumPyを使う

ライブラリNumPyに含まれている `mean` 関数を使えば、かんたんに相加平均を求めることができます。

*2 `range(0, n)` と等価。

```
x = [170, 170, 176, 176, 183]
np.array(x).mean()
```

```
175.0
```

ためしてみよう①

Σ が1つの場合は `for` 文に直せば良いとわかりました。しかし、 Σ は単体ではなく2つ連続で出てくることがあります。これは `for` ループの入れ子になります。

下記は変数 i と変数 j を1～9まで入れ子でループして、値をかけ合せて合算する数式になります。

$$\sum_{i=1}^9 \sum_{j=1}^9 ij$$

式c

式cは何を計算しているのでしょうか。また、この数式をPythonコードに直して、値を計算してみましょう。

■ 正解

「掛け算の九九の総和」を計算しています。

```
x = 0

for i in range(1, 10):
    for j in range(1, 10):
        x += i * j

x
```

実行すると答えを得られます。

```
2025
```

ためしてみよう②

先ほどの数式は、次のように変数 i の総和と変数 j の総和をかけ合わせる形に変形することができます。

$$\sum_{i=1}^9 \sum_{j=1}^9 ij = \sum_{i=1}^9 i \sum_{j=1}^9 j$$

式 d

このように数式を変形すると、`for` ループを入れ子にしないで実装することで、計算量を削減できます (具体的には $O(n^2)$ から $O(n)$ に削減できます)。

右辺の数式を `for` ループが入れ子にならないように実装して、左辺の数式と同じ値になることをたしかめてみましょう。

正解

```
x = 0

for i in range(1, 10):
    x += i

y = 0

for j in range(1, 10):
    y += j

x * y
```

「ためしてみよう①」で得られた答えと同じ答えを得られます。

2025

覚えておきたいポイント

- ・ μ (ミュー) は「平均」を意味します。
- ・ Σ (シグマ) は「総和」を意味します。 Σ は `for` 文で加算していくコードとひも付けて覚えましょう。

これら2つは今後も頻出するので覚えておきましょう。