version 2022/10/19

# PG03: １変量データの統計量

## 1. CSV ファイルの読み込み (StatData01_1.csv)

Jupyter Notebook または Google Colab を起動する。

Google Colab の場合は、前もって Google drive のマウントを済ませておく。

```
from google.colab import drive
drive.mount('/content/drive')
```

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
Data=pd.read_csv('F:/2022_数理統計学概論/StatData/StatData01_1.csv',      # 読み出したいファイルのパス
                 skiprows=1,            # データファイルの最初の1行を飛ばす
                 names=['Weight'])      # カラム名を付ける
Data.head()
```

Out[2]:

| | Weight |
|---|---|
| 0 | 3110 |
| 1 | 3100 |
| 2 | 3140 |
| 3 | 3050 |
| 4 | 2480 |

ここまでは、前回やったとおり。

## 2. 平均値 = データの総和 / データの個数

In [3]:

```
len(Data) #行数（データの個数）
```

Out[3]:

100

In [4]:

```
sum(Data['Weight'])  # 総和
```

Out[4]:

316020

In [5]:

```
sum(Data['Weight']) / len(Data['Weight'])
```

Out[5]:

3160.2

後から引用するためには、名前をつけておくとよい。

In [6]:

```
size = len(Data)
size
```

Out[6]:

100

In [7]:

```
average = sum(Data['Weight']) / size
average
```

Out[7]:

3160.2

## 3. NumPy に備わっている関数で計算しよう

In [8]:

```
# 中央値
median = np.median(Data['Weight'])
median
```

Out[8]:

3160.0

In [9]:

```
# 平均値
mean = np.mean(Data['Weight'])
mean
```

Out[9]:

3160.2

In [10]:

```
# 分散
variance = np.var(Data['Weight'])
variance
```

Out[10]:

143911.96000000008

In [11]:

```
# 標準偏差
std = np.std(Data['Weight'])
std
```

Out[11]:

379.35729859856406

In [12]:

```
# 標準偏差の検算
np.sqrt(variance)
```

Out[12]:

379.35729859856406

分散には2種類あって、上に出てきた統計的データ処理で用いるものは「標本分散」と呼ばれる。 もう一つは推測統計で用いる「不偏分散」というものがある。

In [13]:

```
unbiased_variance = np.var(Data['Weight'], ddof=1)      # 不偏分散
unbiased_variance
```

Out[13]:

145365.61616161626

無駄に長い小数表示は、経過的には良いが最終的には適切に丸めること。 まず、桁数の先の方は計算誤差のため正しくない。 また、有効数字の観点から適切な桁数表示が求められる。

np.round(xxx, 2) とすれば, xxx を小数第3位を丸めて小数第2位までの表示にする。

In [14]:

```
np.round(std,2)  # 標準偏差
```

Out[14]:

379.36

# 4. 統計量のまとめ

DataFrame を使って、基本的な統計量を「表」の形でまとめてみよう。

表には StatSummary という名前を付けた。

In [15]:

```
StatSummary=pd.DataFrame([
              ['平均値', np.round(np.mean(Data['Weight']),2)],     # np.round(xxx,2) を使用
              ['分散', np.round(np.var(Data['Weight']),2)],
              ['標準偏差', np.round(np.std(Data['Weight']),2)],
              ['最小値', min(Data['Weight'])],
              ['最大値', max(Data['Weight'])],
              ['中央値', np.round(np.median(Data['Weight']),2)],
              ['サイズ', len(Data['Weight'])]
             ])
StatSummary
```

Out[15]:

| | 0 | 1 |
|---|---|---|
| 0 | 平均値 | 3160.20 |
| 1 | 分散 | 143911.96 |
| 2 | 標準偏差 | 379.36 |
| 3 | 最小値 | 2270.00 |
| 4 | 最大値 | 4180.00 |
| 5 | 中央値 | 3160.00 |
| 6 | サイズ | 100.00 |

デフォルトでカラム名に番号が付く。 項目名を付けたいので、カラム名を変更する。

In [16]:

```
StatSummary=StatSummary.rename(columns={0:'統計量', 1:'Weight'})
StatSummary
```

Out[16]:

| | 統計量 | Weight |
|---|---|---|
| 0 | 平均値 | 3160.20 |
| 1 | 分散 | 143911.96 |
| 2 | 標準偏差 | 379.36 |
| 3 | 最小値 | 2270.00 |
| 4 | 最大値 | 4180.00 |
| 5 | 中央値 | 3160.00 |
| 6 | サイズ | 100.00 |

## 5. 統計量の計算に pandas は非推奨

以上、NumPy によって統計量を計算した。 実は、pandas でも統計量を計算できるが、分散や標準偏差の扱いに錯誤を引き起こしやすいため非推奨である。 たとえば、8個の統計量を一括出力する .describe() があるが、標準偏差は不偏分散から計算しているので実用上注意を要する（標準偏差があっていないことに注意しよう）。

In [17]:

```
Data.describe()
```

Out[17]:

| | Weight |
|---|---|
| **count** | 100.000000 |
| **mean** | 3160.200000 |
| **std** | 381.268431 |
| **min** | 2270.000000 |
| **25%** | 2897.500000 |
| **50%** | 3160.000000 |
| **75%** | 3367.500000 |
| **max** | 4180.000000 |

In [ ]:

```

```