

# 前提

---

Python3を実行できる環境で最新の `matplotlib`, `numpy`, `Pillow` をインストールしていること。

## 課題

---

### 1

---

差分画像は負の値も含むため、ヒストグラム表示の関数に通常の画像を読み込む処理をまとめると不都合が生じるため、画像の読み込みとヒストグラムの表示は別とした。

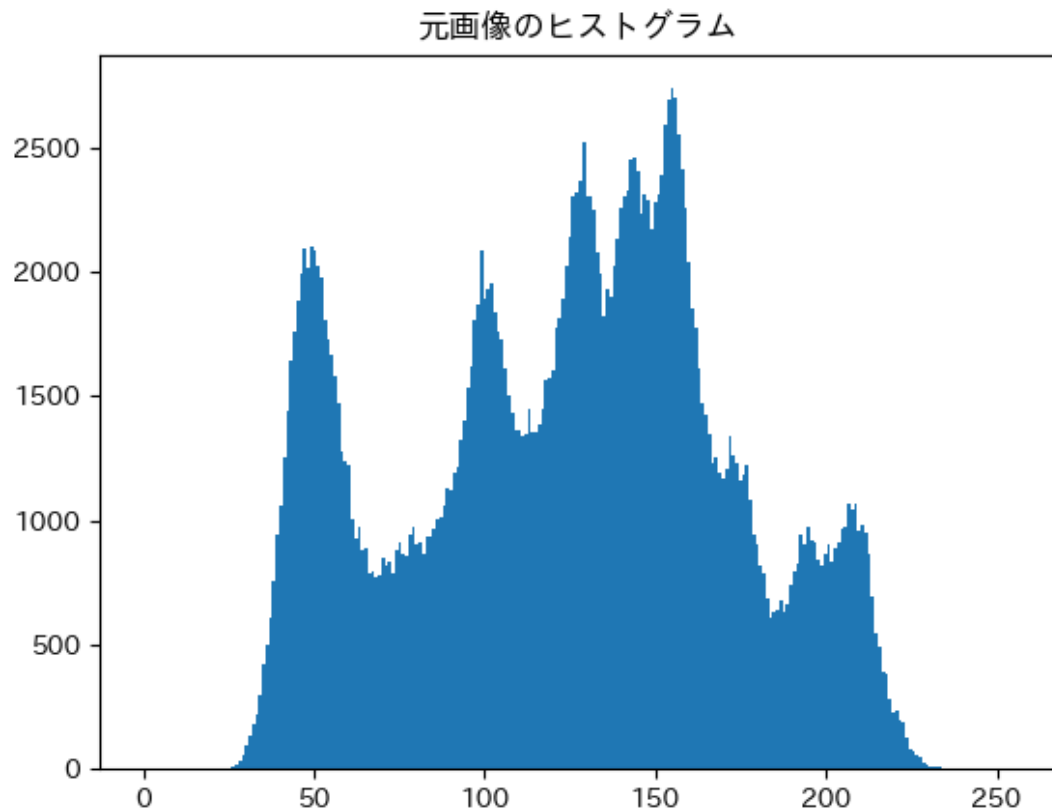
```
import japanize_matplotlib
import numpy as np
from matplotlib import pyplot
from PIL import Image

def read_img(filename: str) -> np.ndarray:
    return np.asarray(Image.open(filename))

def show_image_hist(image: np.ndarray, title: str, option: dict):
    pyplot.hist(image.flatten(), **option)
    pyplot.title(title)
    pyplot.show()
    pyplot.close()
    print('分散:', round(image.var(), 1))

# 実行例
img = read_img('LENA_GRAY.png')
show_image_hist(img, '元画像のヒストグラム', {'bins': 256, 'range': (0, 255)})
```

### 実行結果



分散: 2290.2

## 2

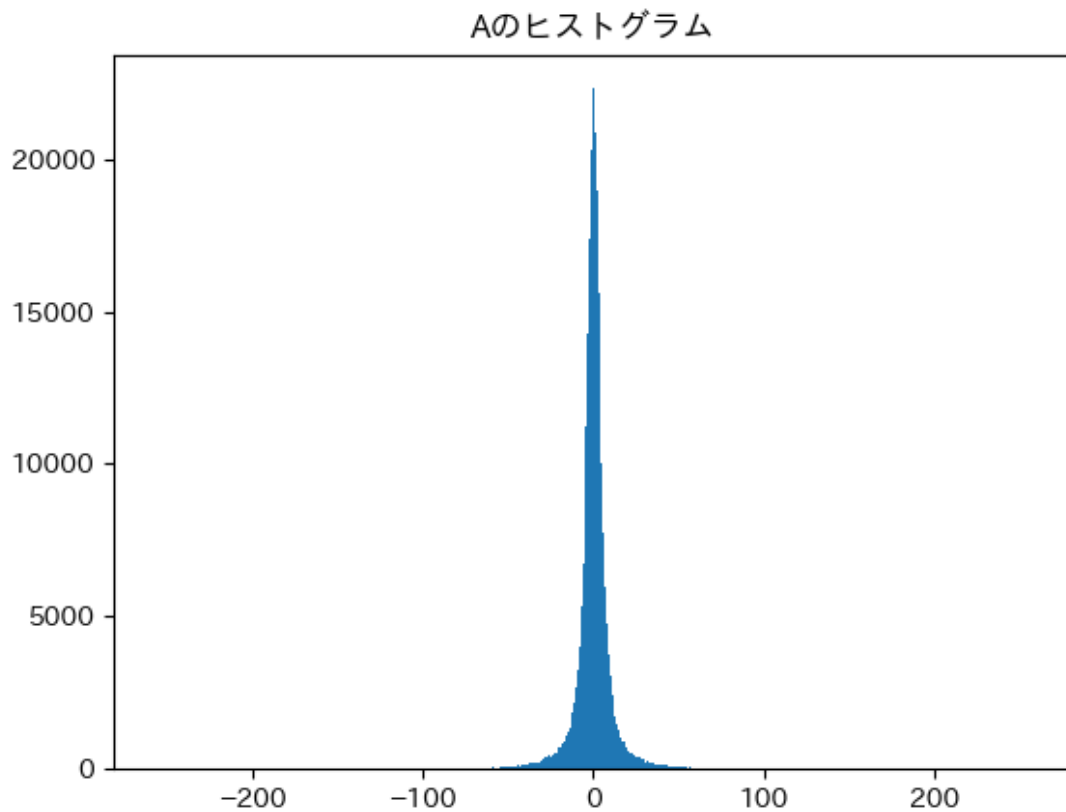
---

### A

```
def a(image: np.ndarray) -> np.ndarray:
    new_image = image.copy().astype(np.int16)
    new_image[:, 1:] -= image[:, :-1]
    return new_image

# 実行例
img = read_img('LENA_GRAY.png')
a_img = a(img)
show_image_hist(a_img, 'Aのヒストグラム', {'bins': 512, 'range': (-256, 255)})
```

### 実行結果



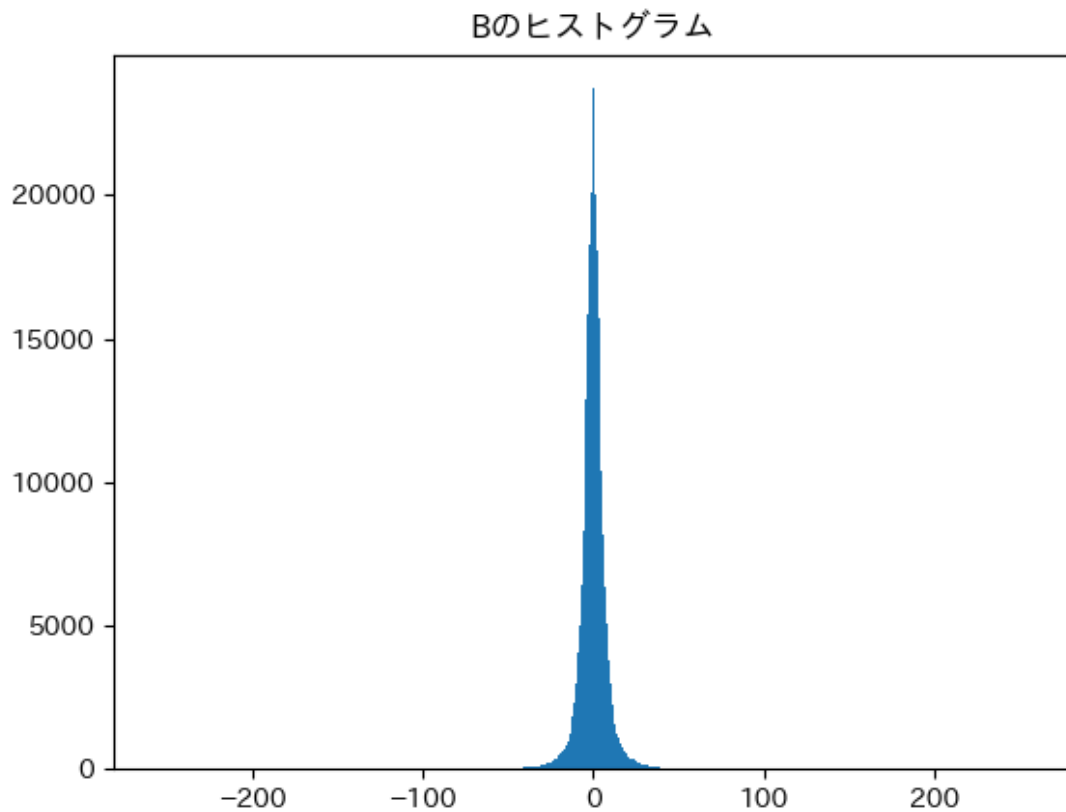
分散: 148.8

## B

```
def b(image: np.ndarray) -> np.ndarray:
    new_image = image.copy().astype(np.int16)
    new_image[1:, 1:] -= (new_image[1:, :-1] + new_image[:-1, 1:] - new_image[:-1, :-1])
    return new_image

# 実行例
img = read_img('LENA_GRAY.png')
b_img = b(img)
show_image_hist(b_img, 'Bのヒストグラム', {'bins': 512, 'range': (-256, 255)})
```

## 実行結果



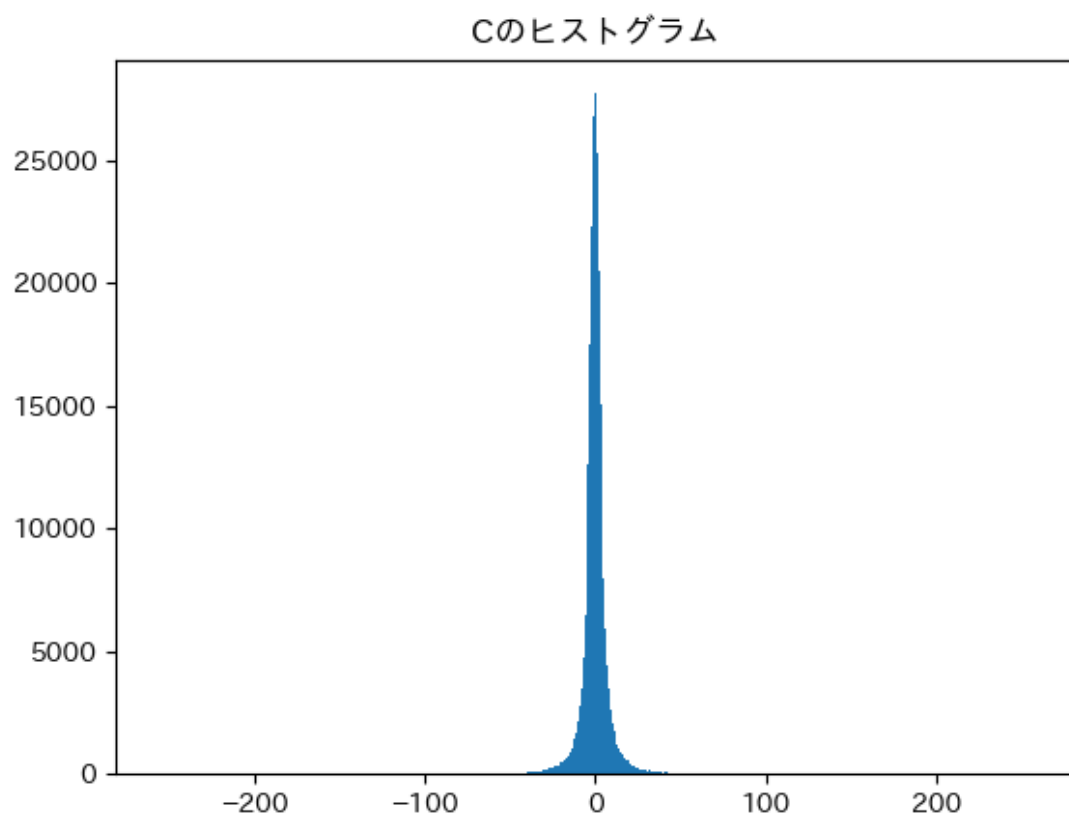
分散: 121.5

## C

```
def c(image: np.ndarray) -> np.ndarray:
    new_image = image.copy().astype(np.int16)
    # round前に0.1足すことで、0.5を1に切り上げるようにする
    new_image[1:, 1:] = np.round((new_image[1:, :-1] + new_image[:-1, 1:]) / 2 + 0.1)
    return new_image

# 実行例
img = read_img('LENA_GRAY.png')
c_img = c(img)
show_image_hist(c_img, 'Cのヒストグラム', {'bins': 512, 'range': (-256, 255)})
```

## 実行結果



分散: 120.4