特徴数：　８×８＝64ピクセル　　クラス：　０～９の10クラス

８ピクセル

0	0	5	13	9	1	0	0
0	0	13	15	10	15	5	0
0	3	15	2	0	11	8	0
0	4	12	0	0	8	8	0
0	5	8	0	0	9	8	0
0	4	11	0	1	12	7	0
0	2	14	5	10	12	0	0
0	0	6	13	10	0	0	0

白：0
黒：1～15

８ピクセル

読み取り

(0, 0, 5, 13, 9, 1, 0, 0
0, 0, 13, 15, 10, 15, 5, 0
0, 3, 15, 2, 0, 11, 8, 0
0, 4, 12, 0, 0, 8, 8, 0
0, 5, 8, 0, 0, 9, 8, 0
0, 4, 11, 0, 1, 12, 7, 0
0, 2, 14, 5, 10, 12, 0, 0
0, 0, 6, 13, 10, 0, 0, 0)

```python
# -*- coding: utf-8 -*-
from sklearn import datasets
import numpy as np

# 手書き数字のデータをロードし、変数digitsに格納
digits = datasets.load_digits()

# 特徴量のセットを変数Xに、ターゲットを変数yに格納
X = digits.data
y = digits.target

# Xとyの次元を表示。それぞれ(1797, 64)と(1797,)となる。
# サンプル数1797, 特徴量の次元64, という意味
print(X.shape)
print(y.shape)

# サンプル数と特徴量の次元の取り出し方法
(n_samples, n_features) = X.shape
print('サンプル数: {0}'.format(n_samples))
print('特徴量の次元: {0}'.format(n_features))

# クラス数の取り出し方法
n_classes = len(np.unique(y))
print('クラス数: {0}'.format(n_classes))

# 0～9の全ての数字に対するそれぞれのサンプル数を表示
for i in range(n_classes):
    print('{0}のサンプル数:{1}'.format(i, len(X[y==i])))

```

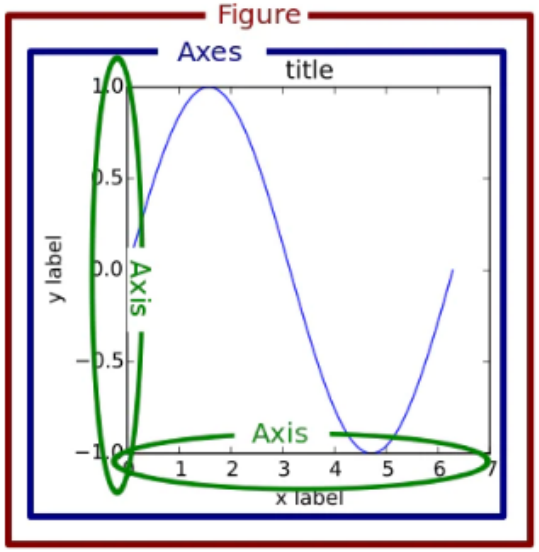
実行結果

```
(1797, 64)
(1797,)
サンプル数: 1797
特徴量の次元: 64
クラス数: 10
0のサンプル数:178
1のサンプル数:182
2のサンプル数:177
3のサンプル数:183
4のサンプル数:181
5のサンプル数:182
6のサンプル数:181
7のサンプル数:179
8のサンプル数:174
9のサンプル数:180
>>>
```

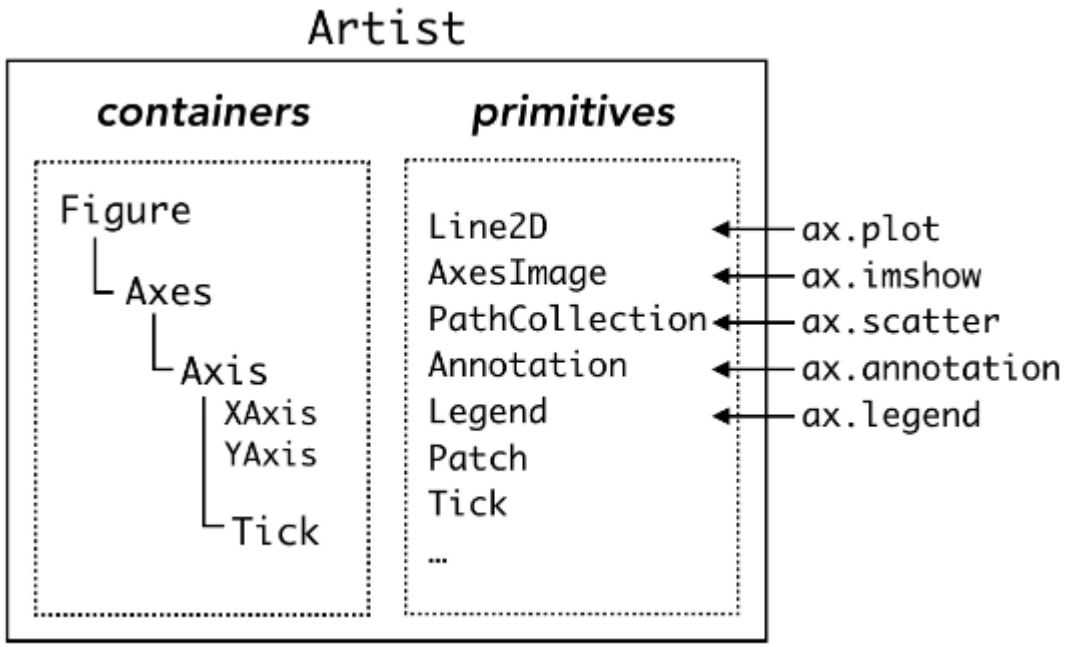
```python
# -*- coding: utf-8 -*-
from sklearn import datasets
import numpy as np
import matplotlib.pyplot as plt

# 手書き数字のデータをロードし、変数digitsに格納
digits = datasets.load_digits()

# 特徴量のセットを変数Xに、ターゲットを変数yに格納
X = digits.data
y = digits.target

# 手書き数字の画像表現を変数imagesに格納
images = digits.images

# 表示エリアの背景をシルバーにセット
fig = plt.figure()
fig.patch.set_facecolor('silver')

# 0～9の10枚の画像をそれぞれ3枚ずつ、計30枚描画
for i in range(10):
    for j in range(3):
        # 数字iの画像のうち、j枚目を取り出す
        img = images[y==i][j]
        #ランダムに取り出したい場合は下記を有効に
        #img = images[y==i][np.random.randint(0, len(images[y==i]))]
        # 縦5x横6の画像表示エリアのうち、3*i+j+1番目に描画
        plt.subplot(5, 6, 3*i + j + 1)
        # グラフとしての軸は描画しない
        plt.axis('off')
        # 白黒を反転した状態で描画
        plt.imshow(img, cmap=plt.cm.gray_r, interpolation='nearest')
        # 各画像にタイトルを描画
        plt.title('Data {0}'.format(i))

# 画像間に余裕をもたせて描画
plt.tight_layout()

# 描画した内容を画面表示
plt.show()
```

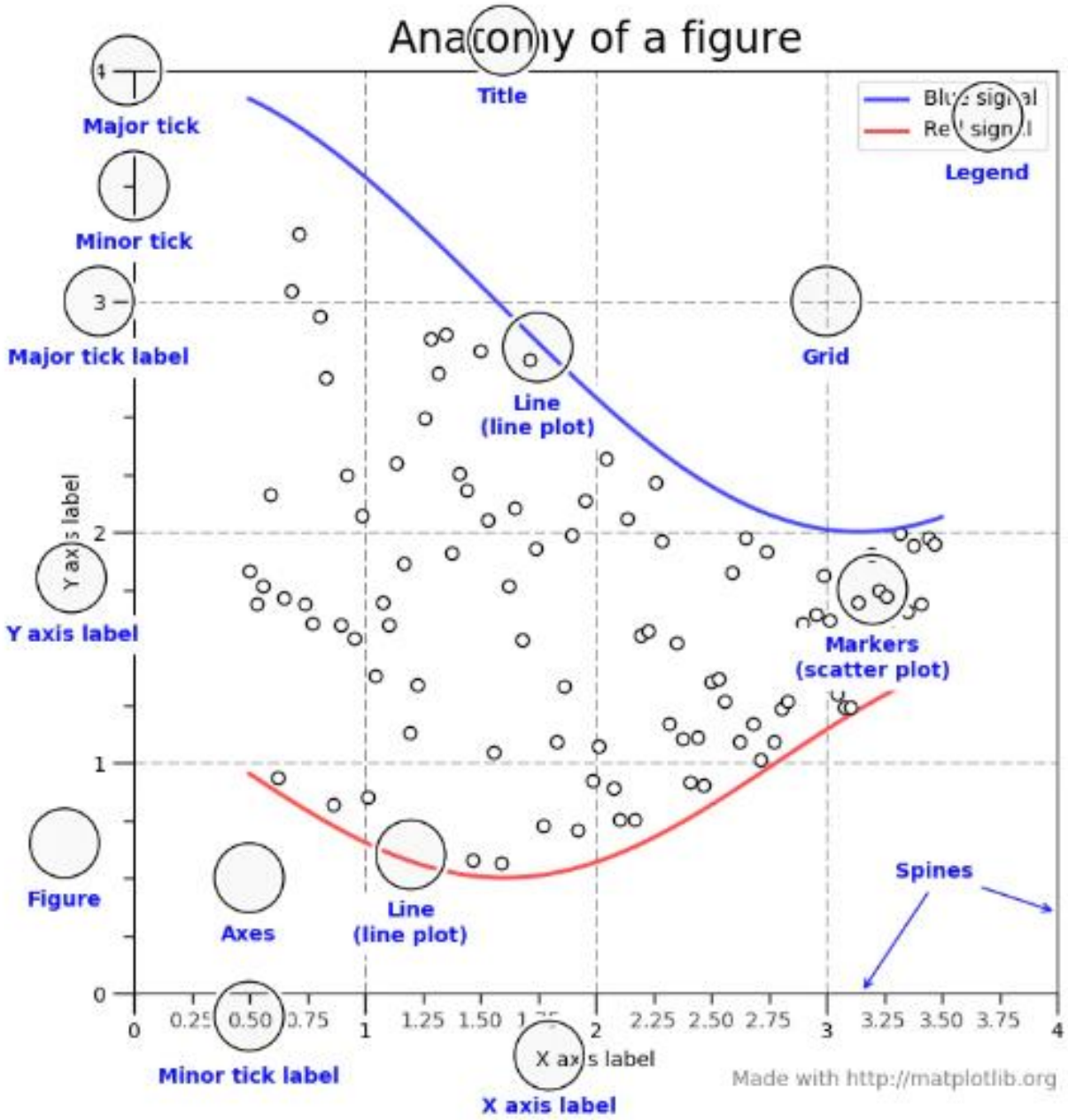
パッケージ	モジュール	機能
matplotlib	pyplot	図形や軸を作成するインターフェース

コンポーネント	機能
figure()	グラフや文字が描かれる台紙
subplot(縦分割、横分割、番地)	グラフを描く位置を設定
parch	塗り潰し
imshow	画像表示
title(‘〇〇’)	〇〇をタイトルとして表示



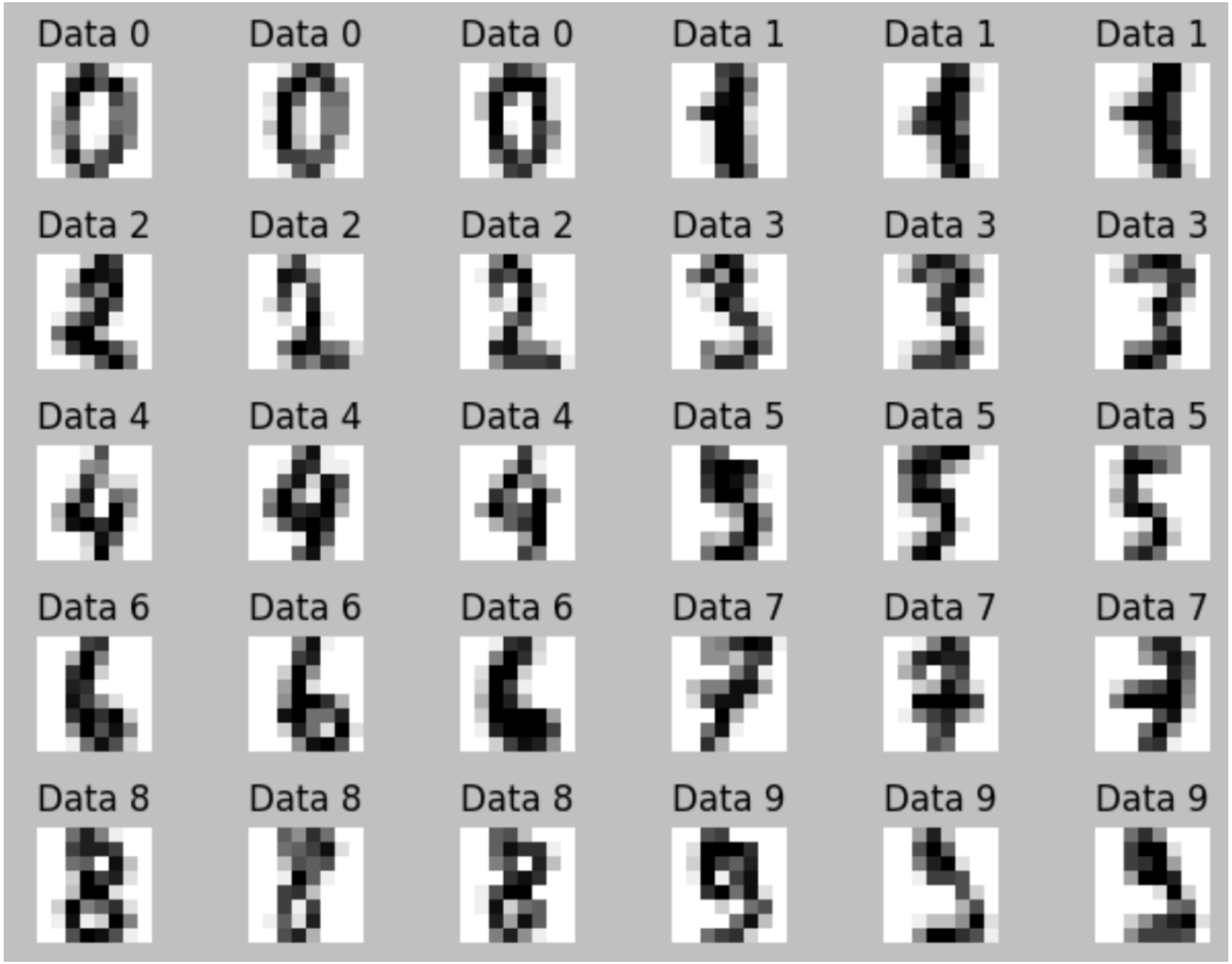
・containersという箱の中にprimitivesが入っている
・containersの中は、階層構造





出典：https://matplotlib.org/stable/tutorials/introductory/usage.html

実行結果


Data 0
Data 0
Data 0
Data 1
Data 1
Data 1
Data 2
Data 2
Data 2
Data 3
Data 3
Data 3
Data 4
Data 4
Data 4
Data 5
Data 5
Data 5
Data 6
Data 6
Data 6
Data 7
Data 7
Data 7
Data 8
Data 8
Data 8
Data 9
Data 9
Data 9