

Pythonで可視化の基本的な話

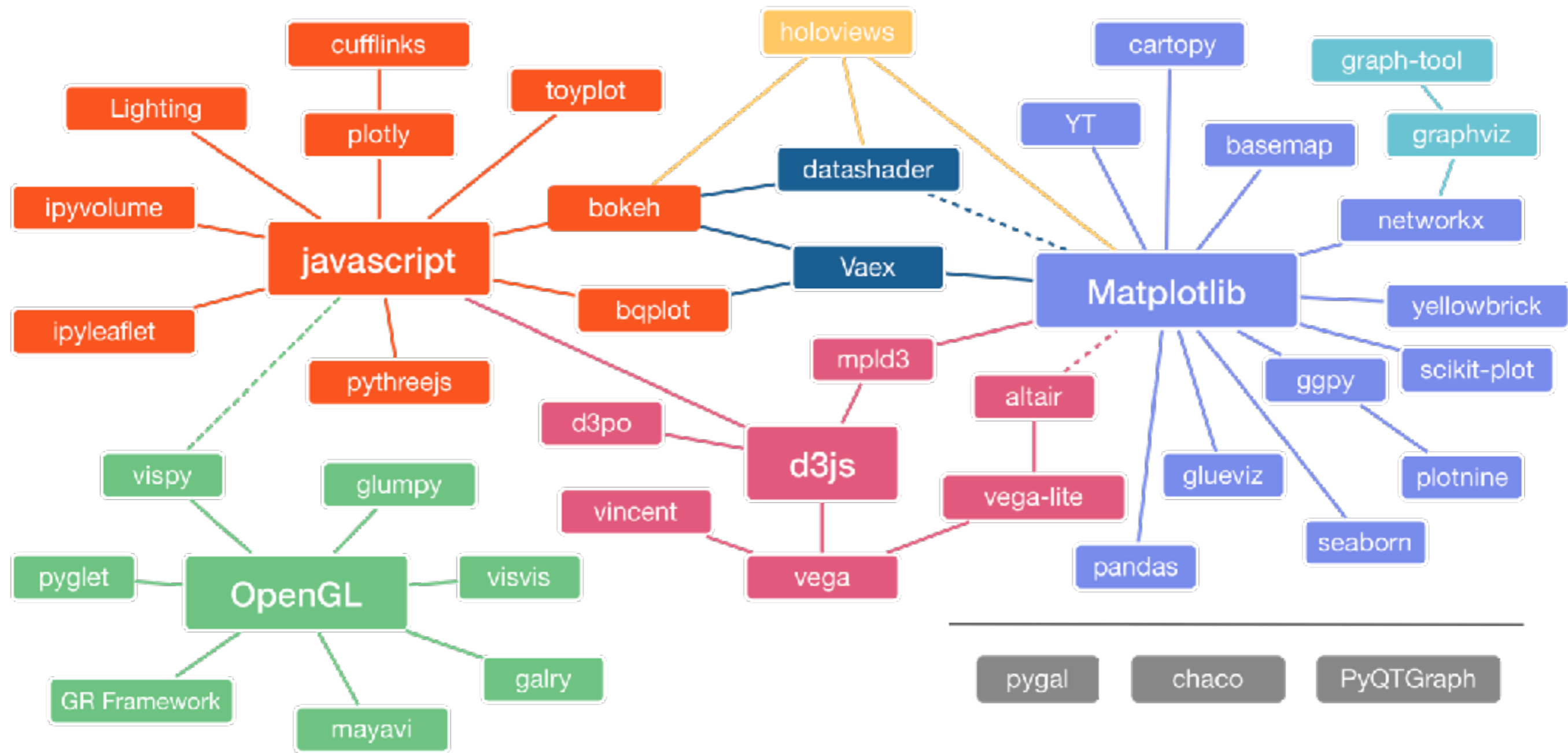
辻真吾 (@tsjshg)
Start Python Club
9/22, 2022

お前、誰よ？（自己紹介）

- 大学の研究所（東京大学先端科学技術研究センター）に勤めています
 - エネルギーシステムとバイオインフォマティクス
 - 数理工学を専門としていきたい
- 2010年に「Pythonスタートブック（初版）」を出版してから技術書を何冊か執筆しています
- 弟（修平）は画家をしています
- www.tsjshg.info

今日は可視化とWebアプリをまたぐ話をします

Pythonには可視化ライブラリがたくさんある



PyCon2017でのJake VanderPlasさんのトークで使われた図

(私が) よく使う方法

- Matplotlib系
 - そのまま使う (10年前はいつもこれ)
 - pandasから使う (だいたいDataFrameなので結局、便利)
 - seabornを使う (階層的クラスタリングの図などで)
- JavaScript系
 - Plotly (最近のおすすめ)

今日のサンプルデータ

```
from sklearn.datasets import load_iris
iris_data = load_iris()

import pandas as pd

iris = pd.DataFrame(data=iris_data.data, columns=iris_data.feature_names)
iris['species'] = [iris_data.target_names[i] for i in iris_data.target]
iris
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows x 5 columns

みんな大好きアヤメのデータ。最近ではペンギンが流行っているらしい

Matplotlib

- 最新のバージョンは3.6.0 (9/15, 2022)
- 老舗MathWorks社製の数値シミュレーションソフトMATLABで作られる出版品質の図を描けることを目指して開発された
- 描き方のスタイルが大きくわけて2つある
 - MATLABスタイル
 - オブジェクト指向スタイル
- pandasやseabornの可視化機能はMatplotlibを使って作られている
- なんでもできるが、微調整の泥沼にはまると時間が解ける危険あり

Plot types Examples Tutorials Reference User guide Develop Re

Examples

This page contains example plots. Click on any image to see the full in

For longer tutorials, see our tutorials page. You can also find external user guide.

Lines, bars and markers

Bar color demo

Bar Label Demo

Grouped bar chart with labels

Horizontal bar chart labels

CapStyle

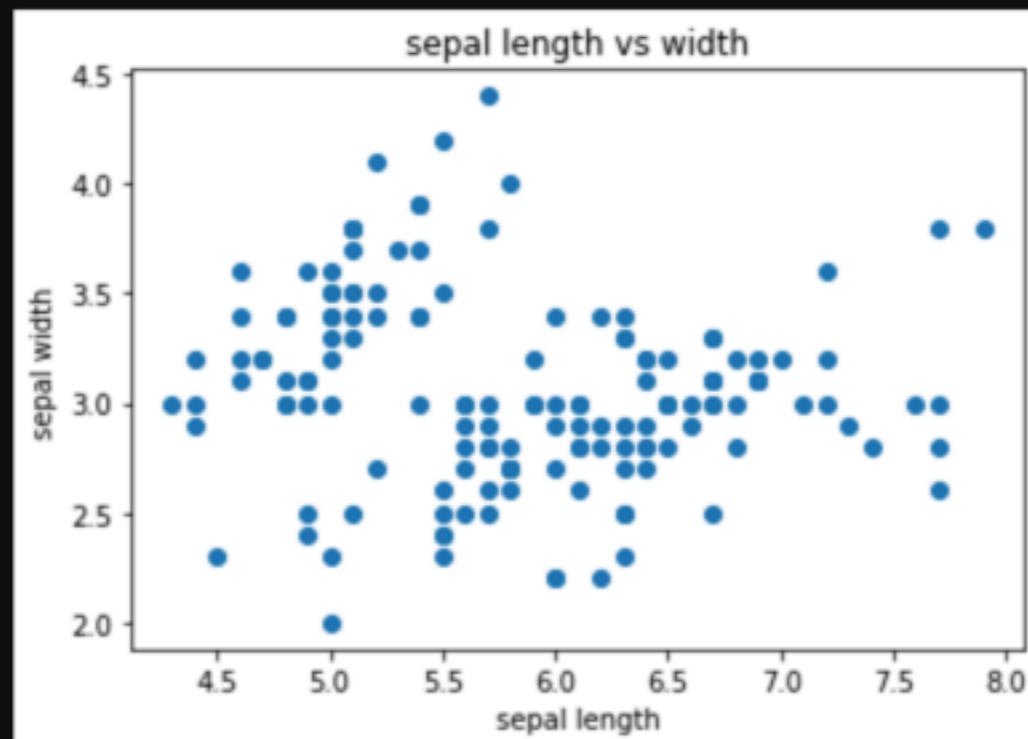
Plotting categorical variables

MATLABスタイル

```
import matplotlib.pyplot as plt

plt.scatter(iris['sepal length (cm)'], iris['sepal width (cm)'])
plt.title('sepal length vs width')
plt.xlabel('sepal length')
plt.ylabel('sepal width')

# Jupyter環境では省略可能
plt.show()
```



MATLABでの使い方と似た描き方（らしいです）

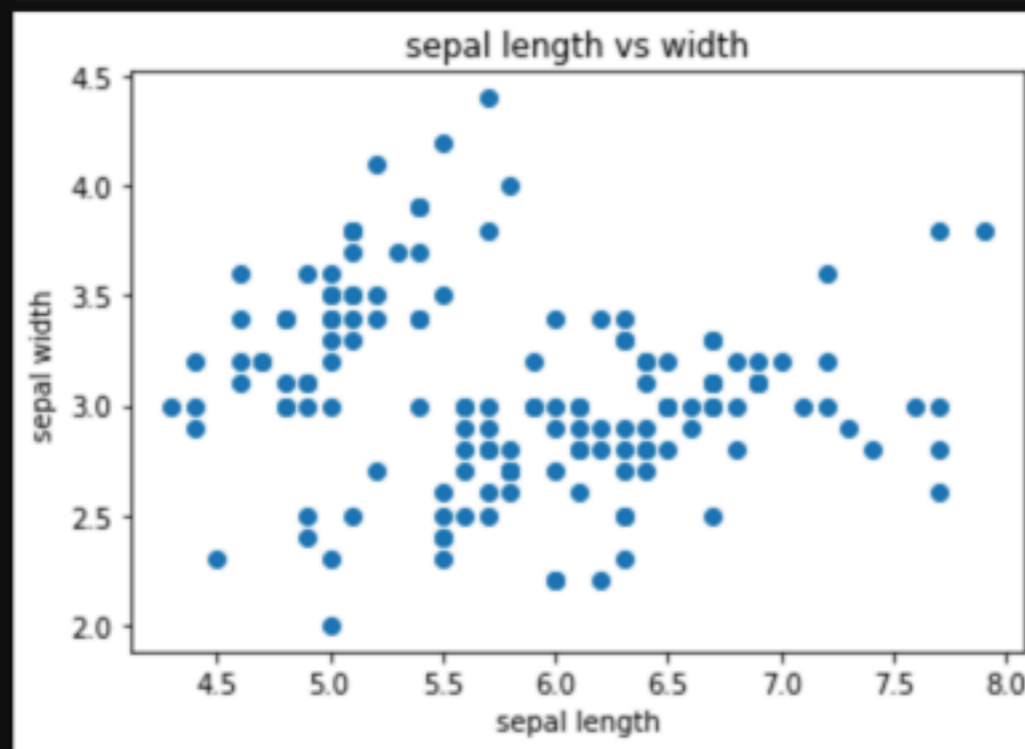
オブジェクト指向スタイル

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.scatter(iris['sepal length (cm)'], iris['sepal width (cm)'])
ax.set_title('sepal length vs width')
ax.set_xlabel('sepal length')
ax.set_ylabel('sepal width')

# Jupyter環境では省略可能
plt.show()
```



オブジェクトを作ってそこにデータやタイトルなどをセットしていくやり方

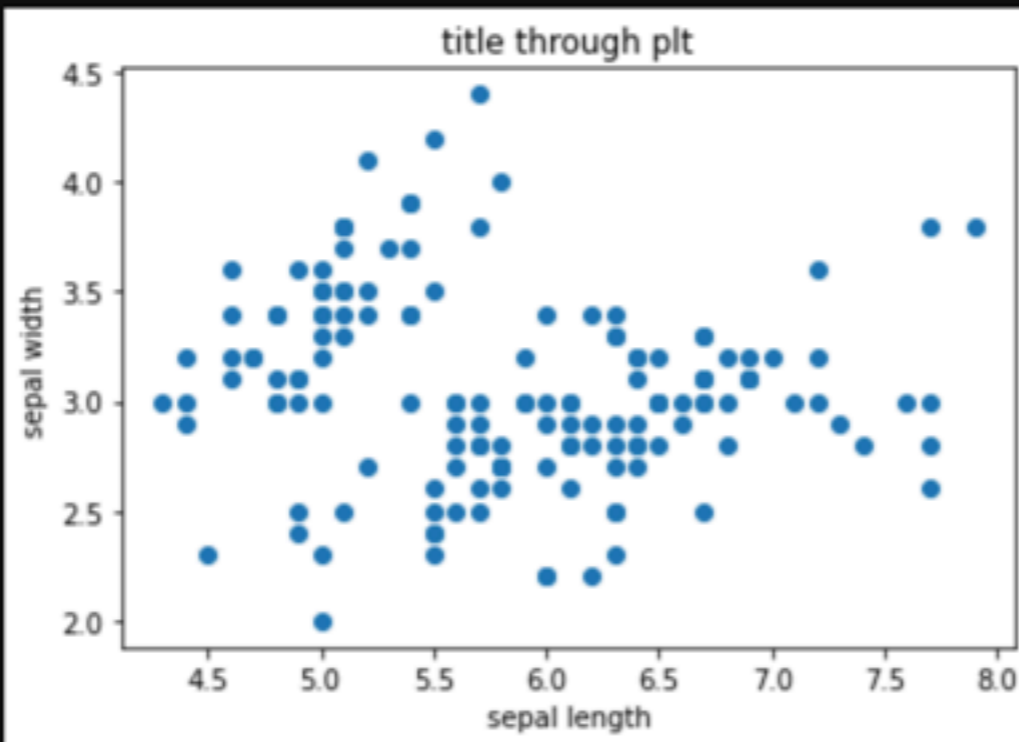
混ぜても危険ではない . . .

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

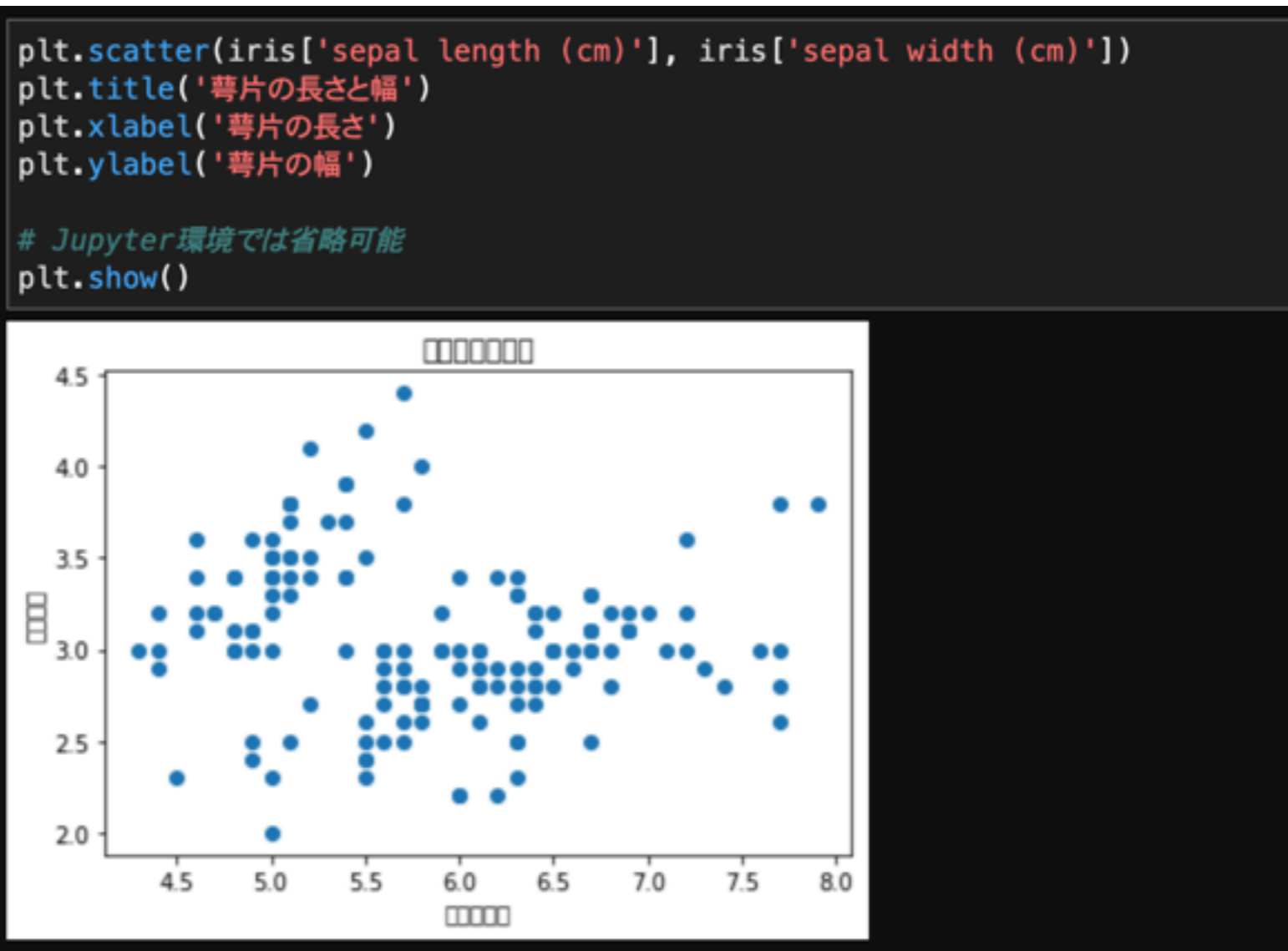
ax.scatter(iris['sepal length (cm)'], iris['sepal width (cm)'])
plt.title('title through plt')
ax.set_xlabel('sepal length')
ax.set_ylabel('sepal width')

# Jupyter環境では省略可能
plt.show()
```



タイトルだけMATLABスタイルにしてみた例

世界の人々を悩ませる豆腐問題



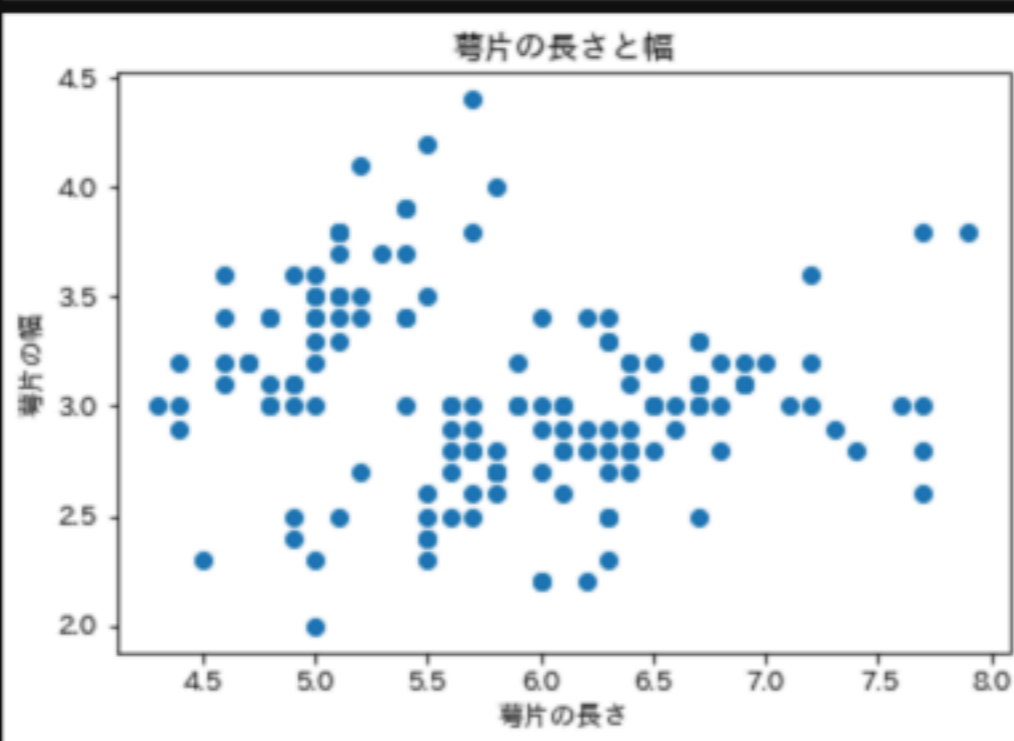
Matplotlibの設定で正しいフォントを指定するという解決作が一般的だった

japanize-matplotlib

```
import matplotlib.pyplot as plt
import japanize_matplotlib

plt.scatter(iris['sepal length (cm)'], iris['sepal width (cm)'])
plt.title('萼片の長さ と 幅')
plt.xlabel('萼片の長さ')
plt.ylabel('萼片の幅')

# Jupyter環境では省略可能
plt.show()
```



pipでインストールして、matplotlibの後にimportするだけでOK

(パッケージの名前にハイフンとアンダースコアが混在することはよくあるので注意)

Matplotlibの生みの親

John Hunterは2012年8月に44歳の若さでこの世を去りました



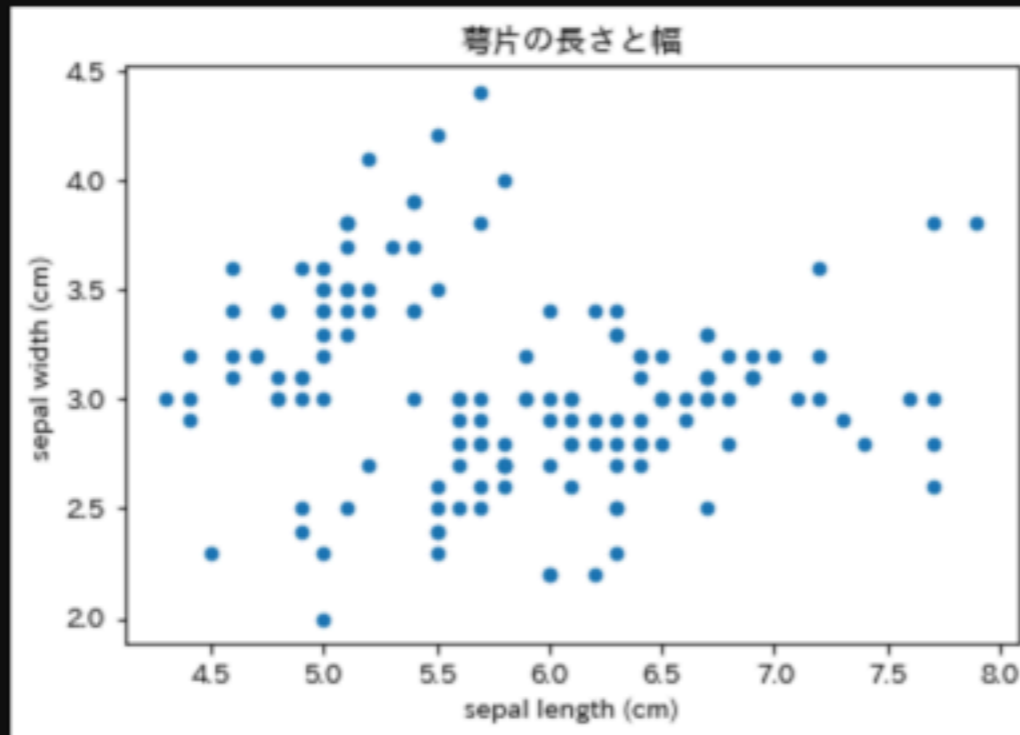
About John Hunter

John Hunter was the creator of **Matplotlib** and one of the founding board members of NumFOCUS. Husband to Miriam and father to three daughters—Clara, Ava, and Rahel—was diagnosed with cancer in late July 2012. He passed away on August 28, 2012 from complications arising from necessary cancer treatment.

John Hunter Matplotlib Summer Fellowshipでは夏の3ヶ月間、フルタイムでMatplotlibの開発に貢献できる学生さん（1～2人）をサポート
Matplotlibの辣腕コントリビュータからの指導あり

pandas.DataFrameになっているなら

```
# japanize_matplotlibを含めた日本語の設定が必要  
iris.plot.scatter('sepal length (cm)', 'sepal width (cm)', title='萼片の長さ&幅')  
  
<AxesSubplot:title={'center': '萼片の長さ&幅'}, xlabel='sepal length (cm)', ylabel='sepal width (cm)'
```

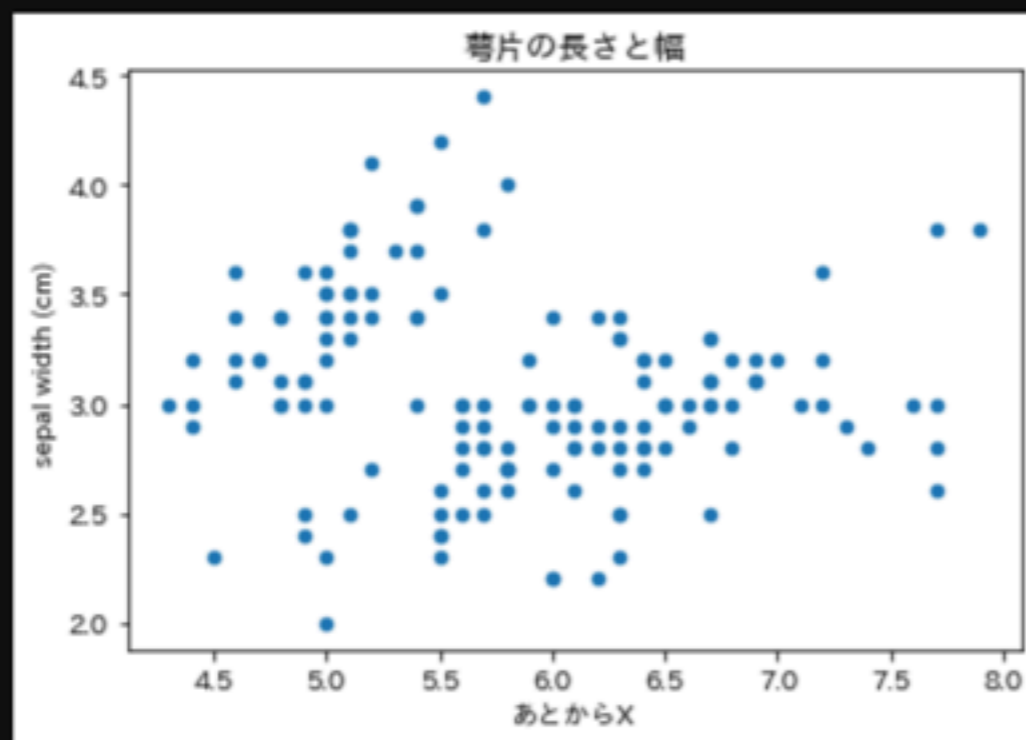


列名を指定するだけでよいので便利
タイトルなどはメソッドの引数で調整可能

基本はMatplotlibなので

```
ax = iris.plot.scatter('sepal length (cm)', 'sepal width (cm)', title='萼片の長さ と幅')  
ax.set_xlabel('あとからX')
```

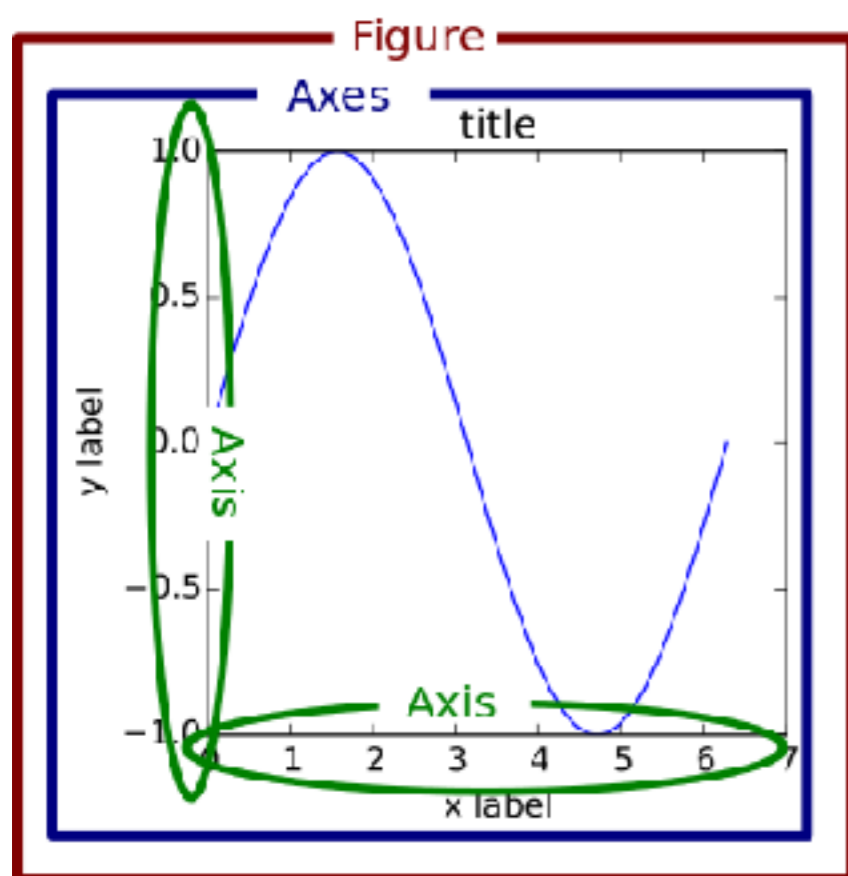
```
Text(0.5, 0, 'あとからX')
```



X軸のラベルを「あとからX」へ変更

pandasのメソッドの戻り値はAxesSubplotオブジェクト

Matplotlibを完全に理解したい方へ



https://matplotlib.org/1.5.1/faq/usage_faq.html

▲ この記事は最終更新日から3年以上が経過しています。

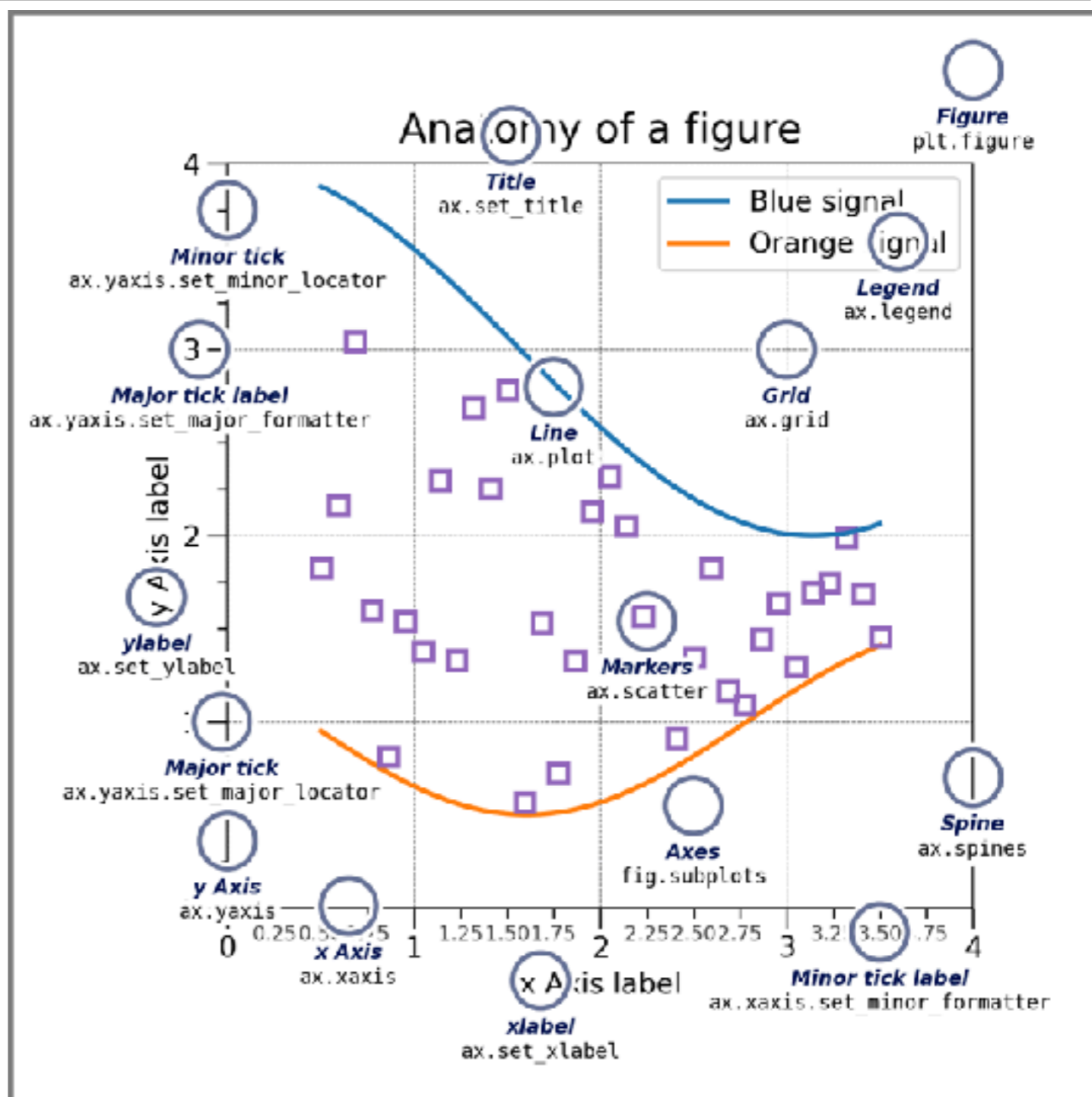
@skotaro

投稿日 2018年01月12日 更新日 2018年07月16日

早く知っておきたかったmatplotlibの基礎知識、 あるいは見た目の調整が捗るArtistの話

Python, matplotlib, pandas, seaborn, 実験屋のためのPython

<https://qiita.com/skotaro/items/08dc0b8c5704c94eafb9>



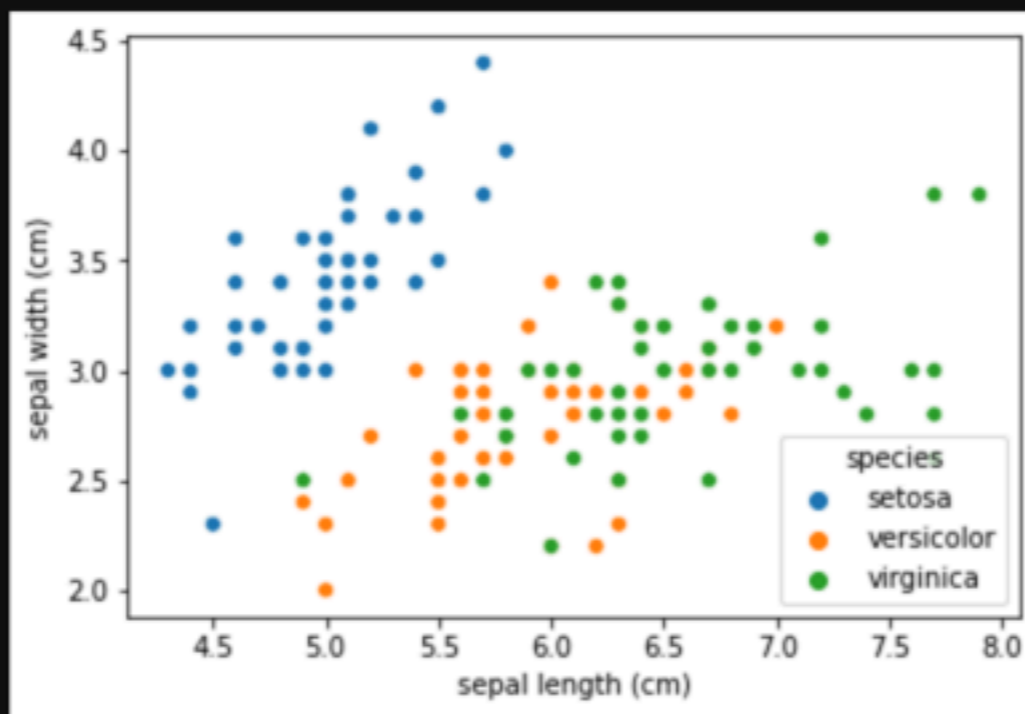
https://matplotlib.org/stable/tutorials/introductory/quick_start.html#parts-of-a-figure

seabornはいろいろ簡単にできる

```
import seaborn as sns

sns.scatterplot(data=iris,
                x="sepal length (cm)",
                y="sepal width (cm)",
                hue="species")

<AxesSubplot:xlabel='sepal length (cm)', ylabel='sepal width (cm)'
```



Data引数にDataFrameを渡すことが前提になった設計

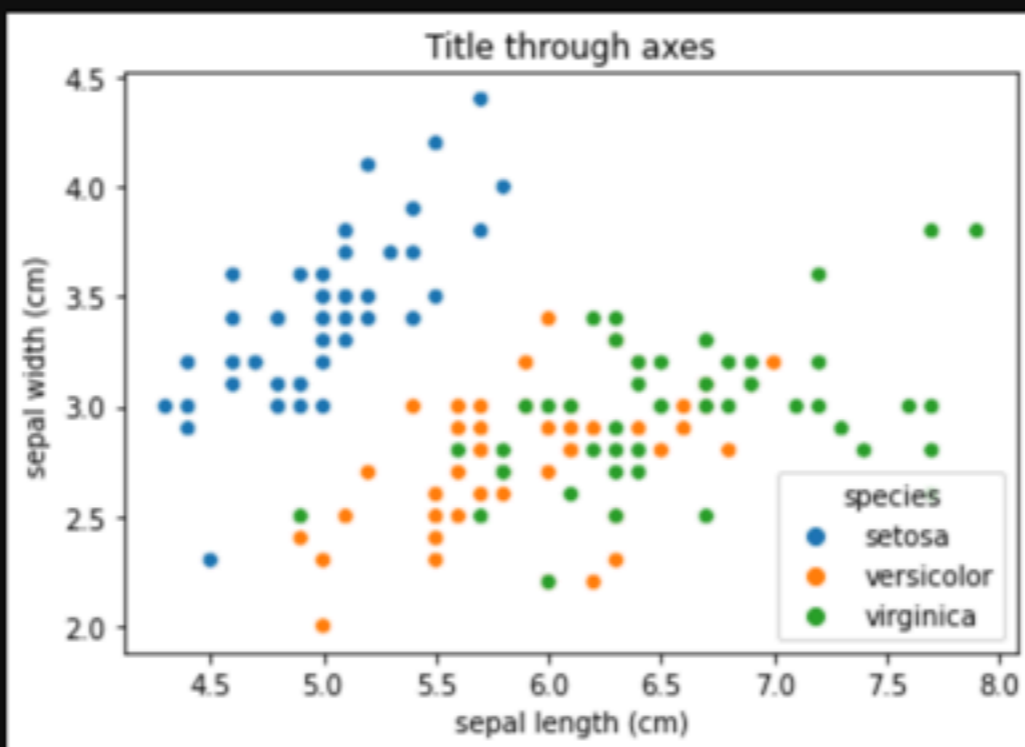
hue引数で色分けも簡単

Matplotlibを知っているとさらに良い

```
import seaborn as sns
```

```
ax = sns.scatterplot(data=iris,  
                    x="sepal length (cm)",  
                    y="sepal width (cm)",  
                    hue="species")  
ax.set_title('Title through axes')
```

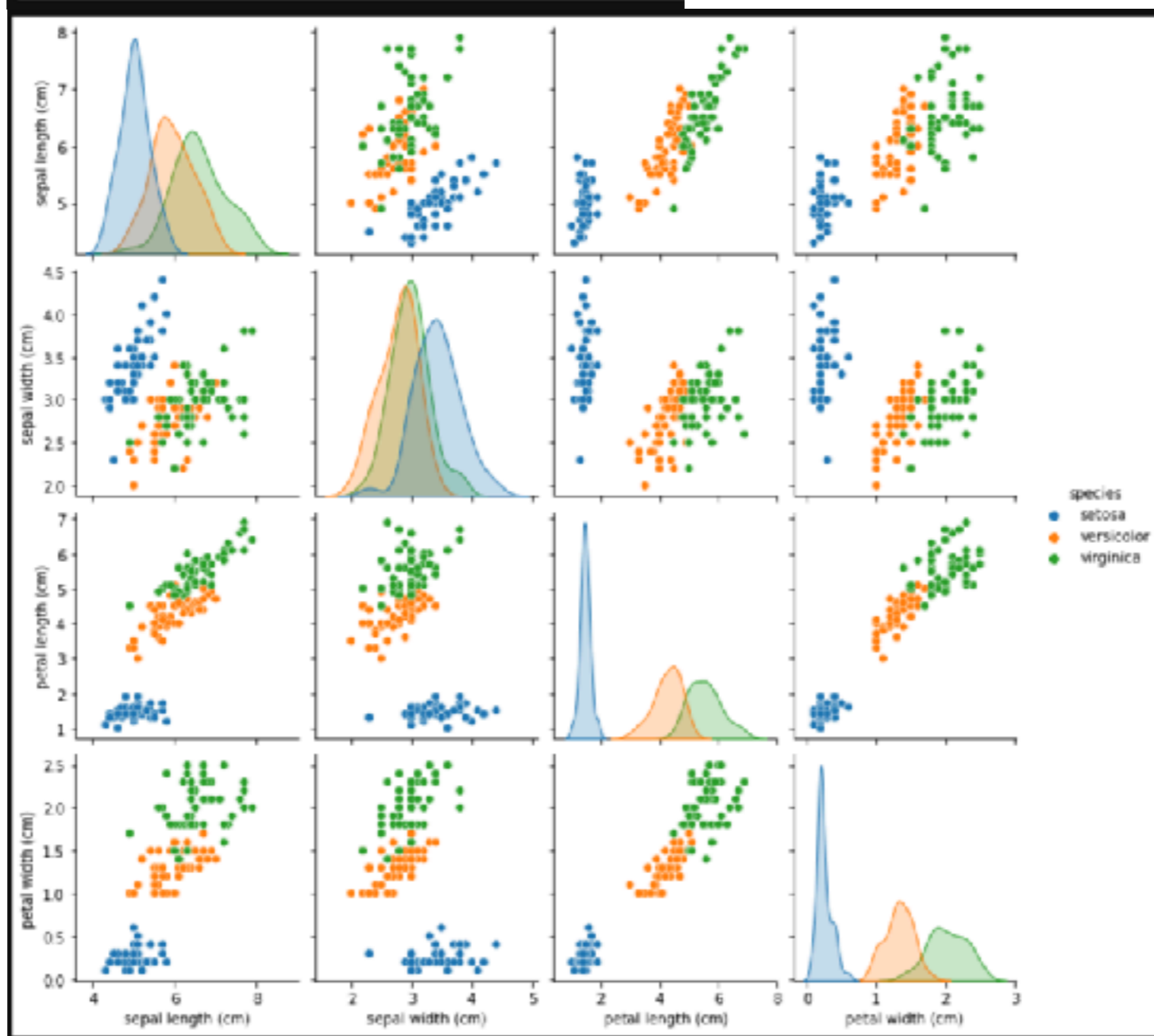
```
Text(0.5, 1.0, 'Title through axes')
```



裏側はMatplotlibなのでseabornの引数でできないこともMatplotlibが分かればできる

散布图行列

```
import seaborn as sns  
sns.pairplot(iris, hue="species")
```



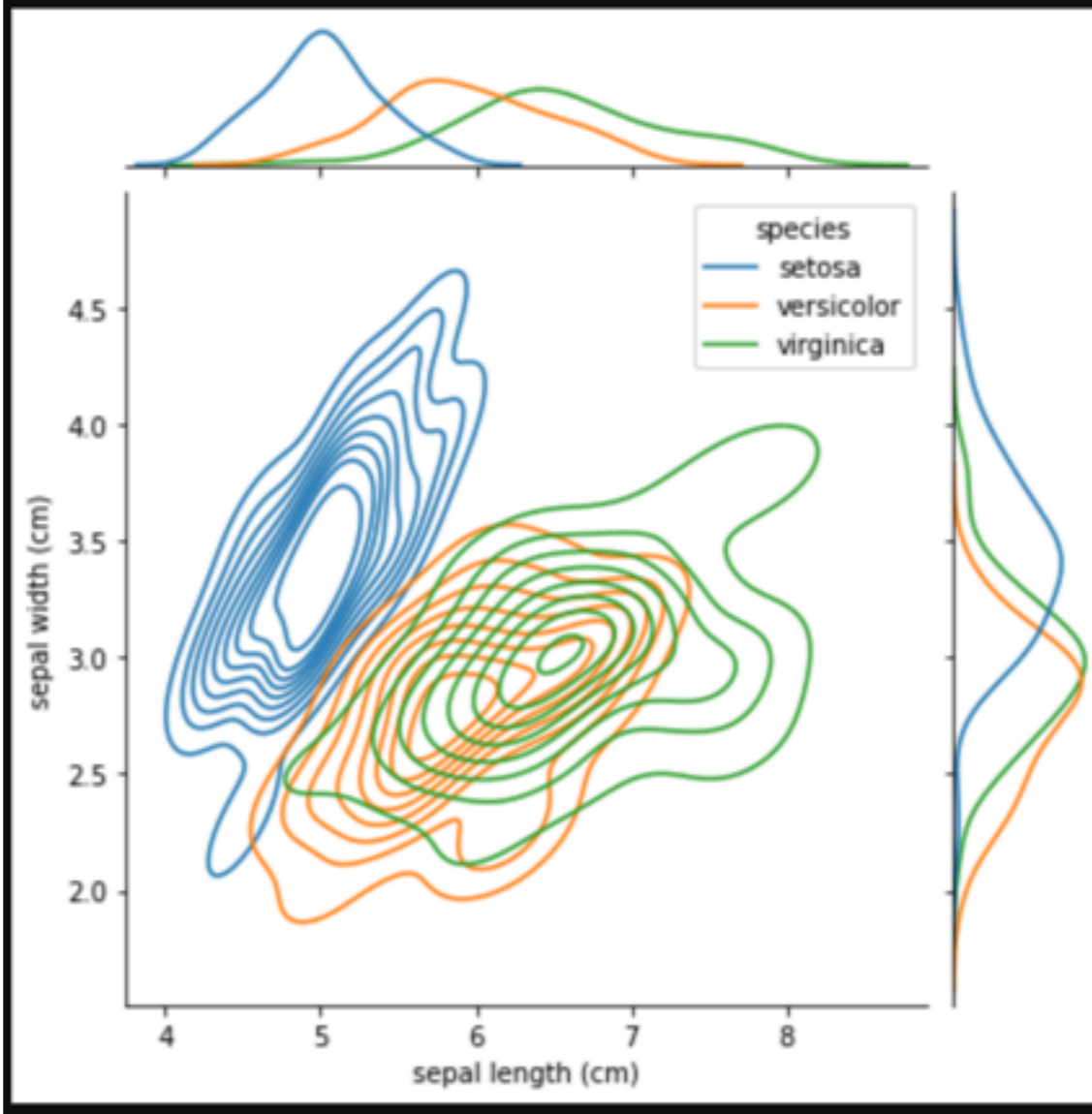
jointplot

Kernel Density Estimationを使った分布の推定

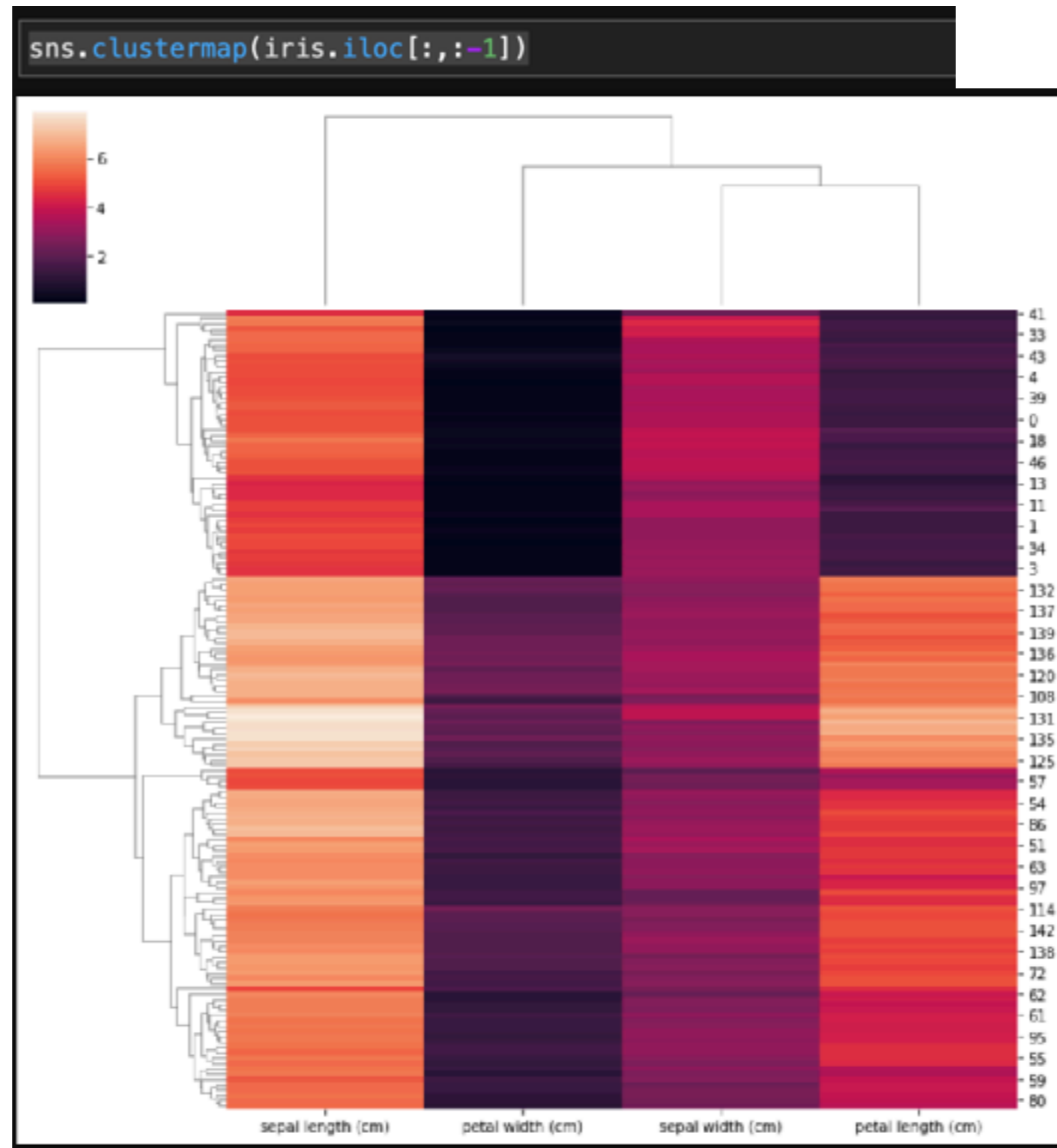
```
import seaborn as sns
```

```
sns.jointplot(data=iris,  
             x="sepal length (cm)",  
             y="sepal width (cm)",  
             hue="species",  
             kind="kde")
```

```
<seaborn.axisgrid.JointGrid at 0x13cf65300>
```

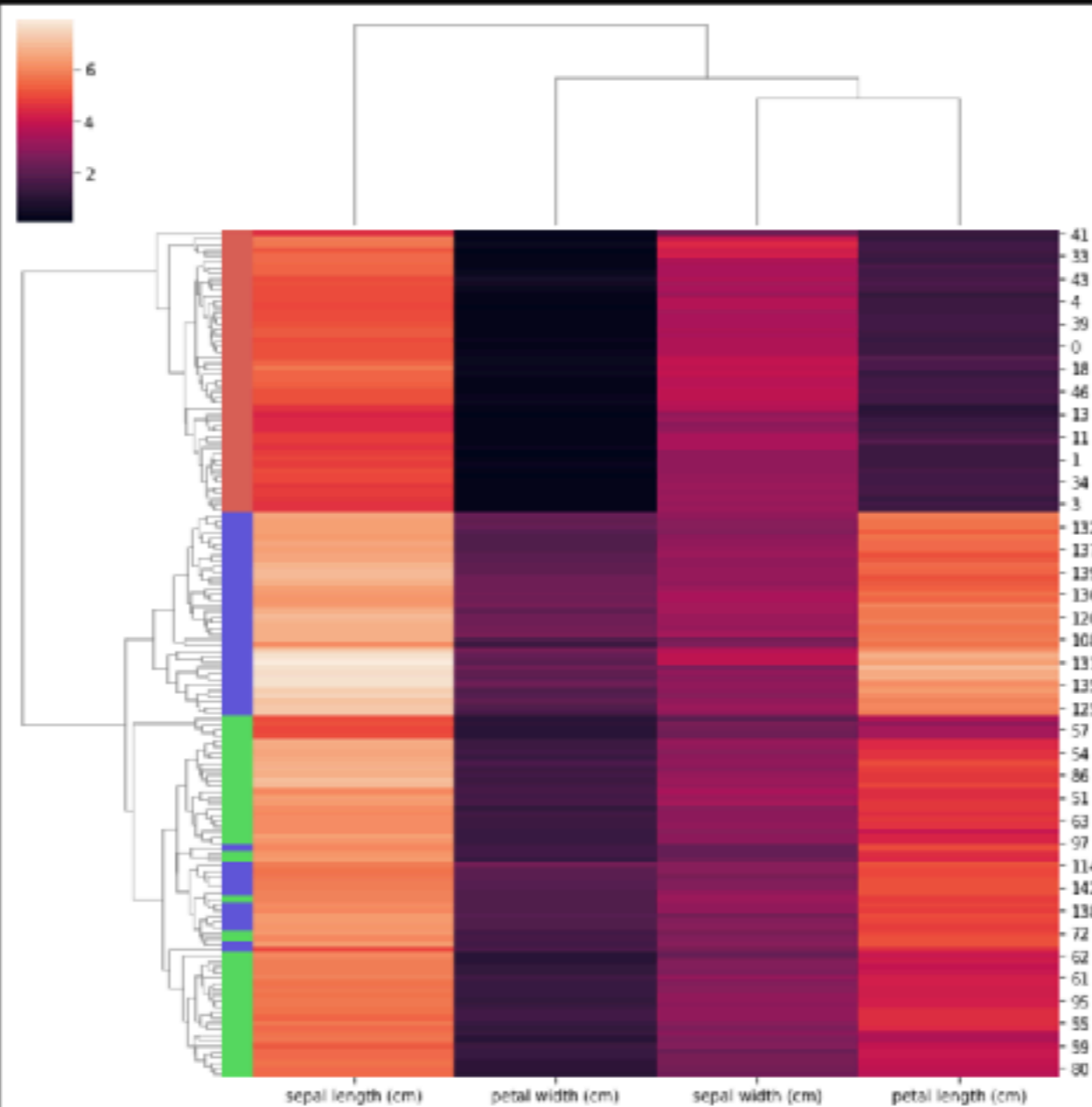


階層的クラスタリング



種類ごとに色を付ける

```
# hlsパレットから3色もってくる
palette = sns.color_palette("hls", 3)
# インスタンスごとに色付け
row_colors = [palette[i] for i in iris_data.target]
# 種類ごとに色が付いたクラスタリング
sns.clustermap(iris.iloc[:, :-1], row_colors=row_colors)
```



Matplotlib～seabornを知りたい方へおすすめ



JupyterLabの使い方と
データの分析・可視化を徹底解説!

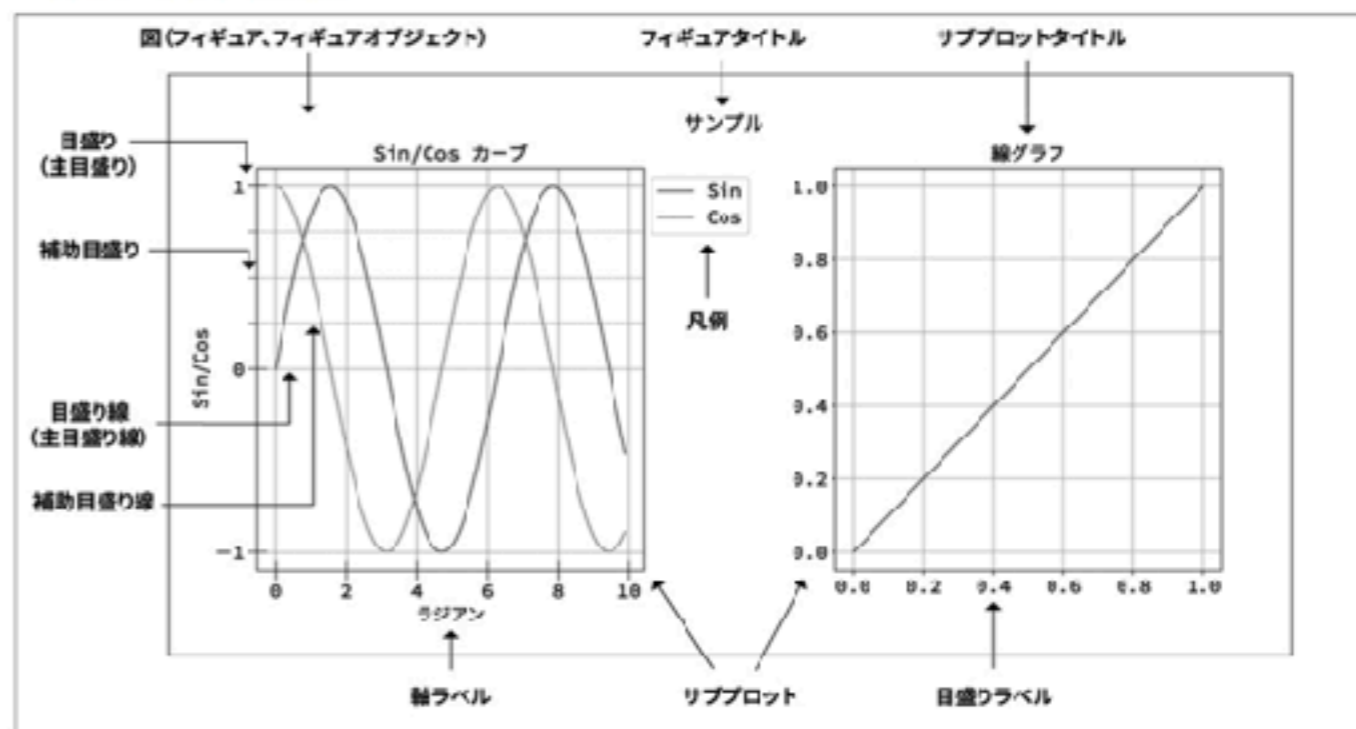
- データ分析の必須ツール JupyterLabの 基本から応用まで
- pandasによるデータの 集計・加工・可視化ノウハウ
- Matplotlibとseabornによる 効果的なグラフ作成

技術評論社

■ グラフの名称

本書では、数値データを可視化する「グラフ」を取り扱います。グラフの各部名称は図0.1の通り定義します。

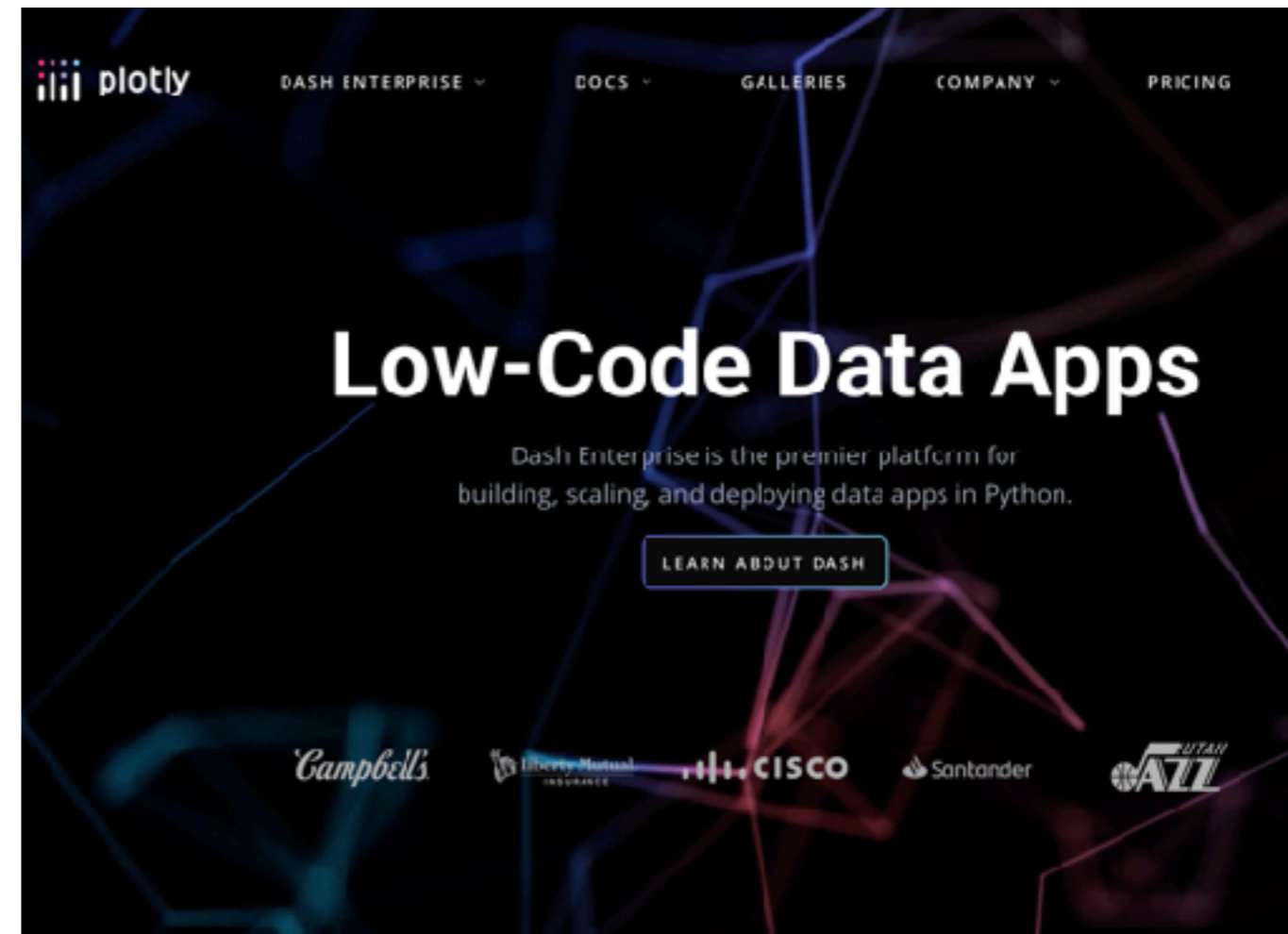
● 図 0.1 グラフ各部名称



Amazonの試し読みページより
詳しい解説があるので分かりやすい

Plotly

- JavaScript系の可視化ライブラリ Plotlyとデータサイエンス向けWebアプリケーションフレームワーク Dashを開発する会社
- Python以外の言語（R, Julia, F#など）に対応
- PlotlyやDashはオープンソースソフトウェア
- Dashアプリのホスティングなどで収入を得ている模様

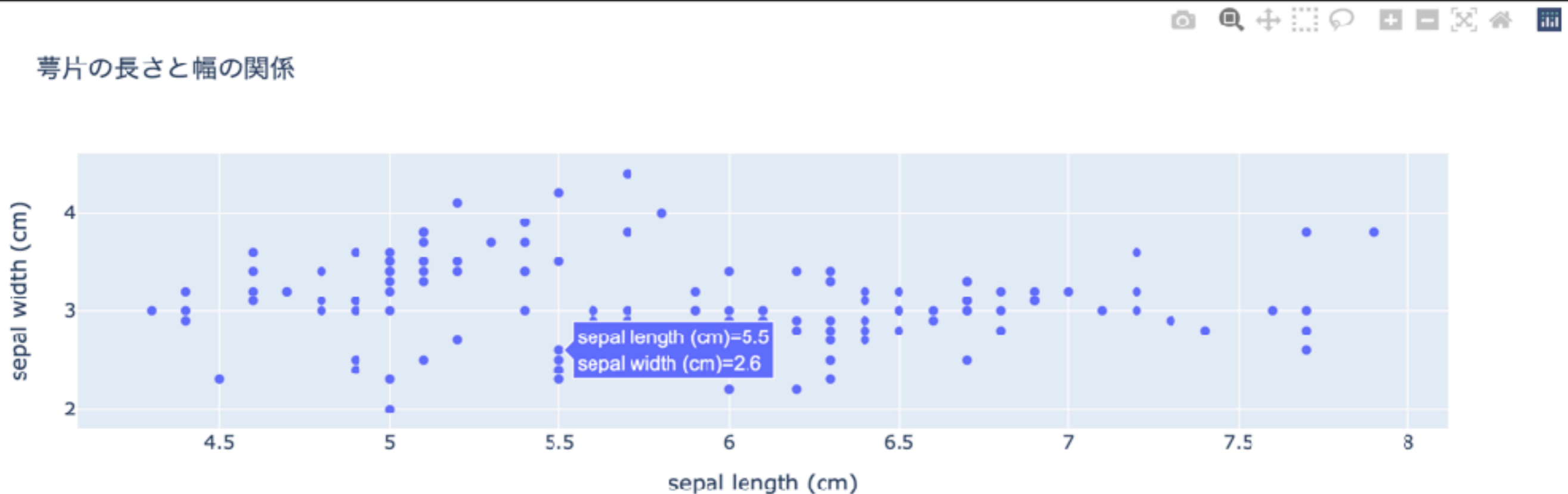


インタラクティブな可視化に便利

```
import plotly.express as px

fig = px.scatter(iris,
                 x="sepal length (cm)",
                 y="sepal width (cm)",
                 title="萼片の長さとの関係")

# Jupyter環境では省略可能
fig.show()
```



日本語の表示も問題なし

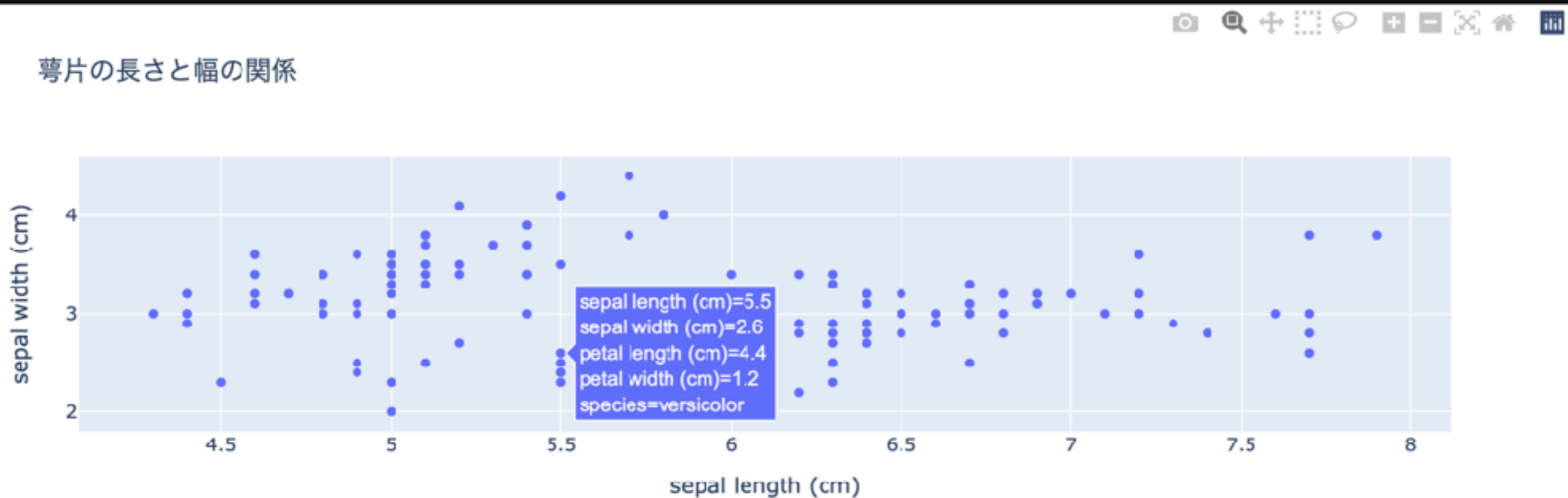
右上のツールで画像の保存、インタラクティブな拡大縮小が可能

hover_data引数

```
import plotly.express as px

fig = px.scatter(iris,
                 x="sepal length (cm)",
                 y="sepal width (cm)",
                 title="萼片の長さとの関係",
                 hover_data=["petal length (cm)", "petal width (cm)", "species"])

# Jupyter環境では省略可能
fig.show()
```



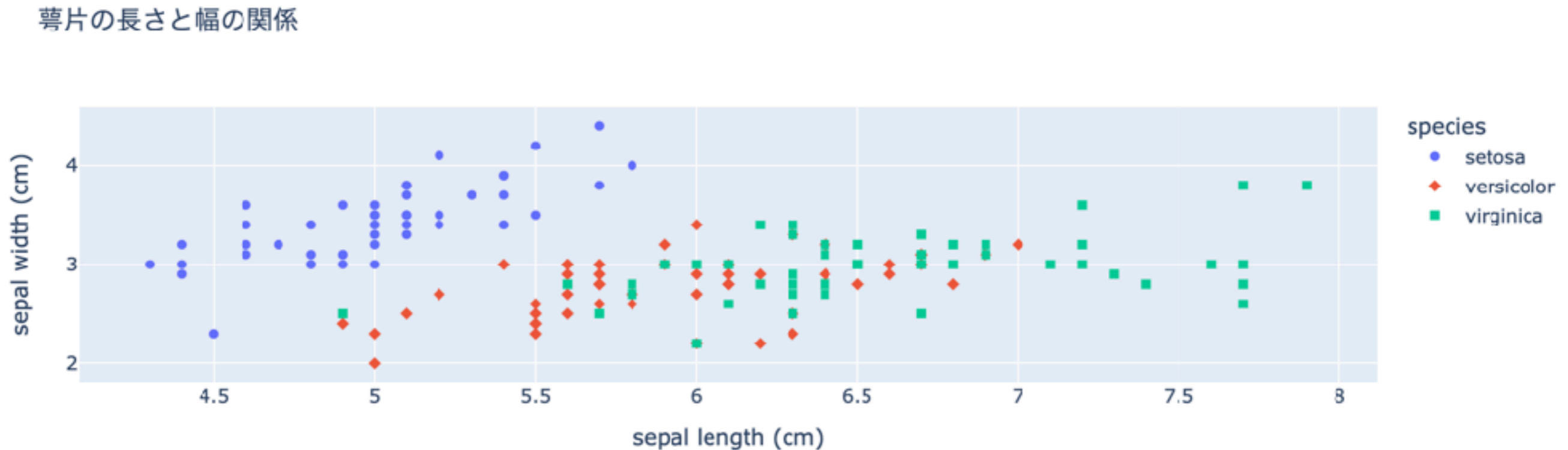
DataFrameの列名を指定すればそのデータが表示される

色づけも簡単

```
import plotly.express as px

fig = px.scatter(iris,
                 x="sepal length (cm)",
                 y="sepal width (cm)",
                 title="萼片の長さとの関係",
                 hover_data=["petal length (cm)", "petal width (cm)", "species"],
                 color="species",
                 symbol="species"
                )

# Jupyter環境では省略可能
fig.show()
```



symbol引数で点の形も変えています

こんな書き方もある

```
import plotly.graph_objects as go

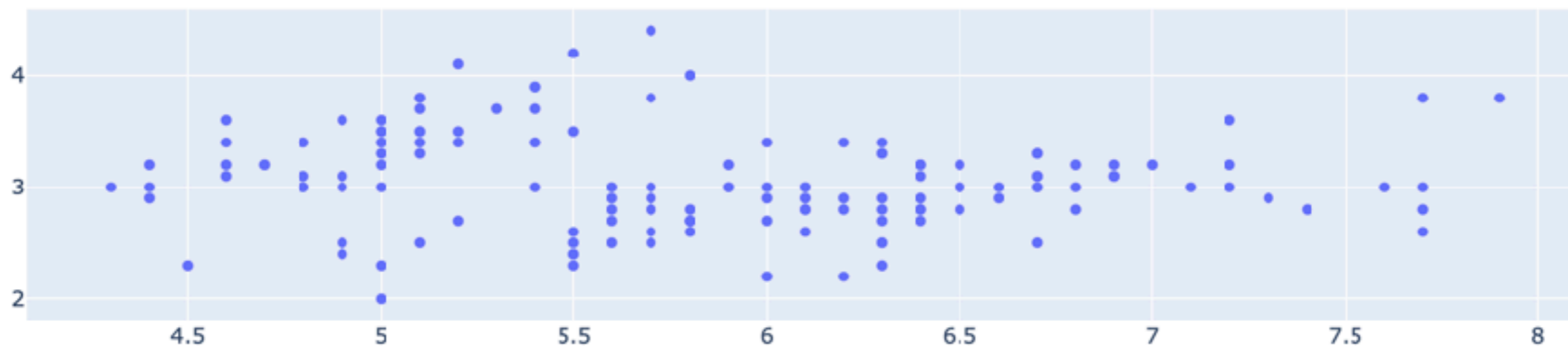
plot_data = go.Scatter(x=iris["sepal length (cm)"],
                       y=iris["sepal width (cm)"],
                       mode='markers')

fig = go.Figure(data=plot_data)

fig.update_layout(title="萼片の長さ と 幅")

# Jupyter環境では省略可能
fig.show()
```

萼片の長さ と 幅



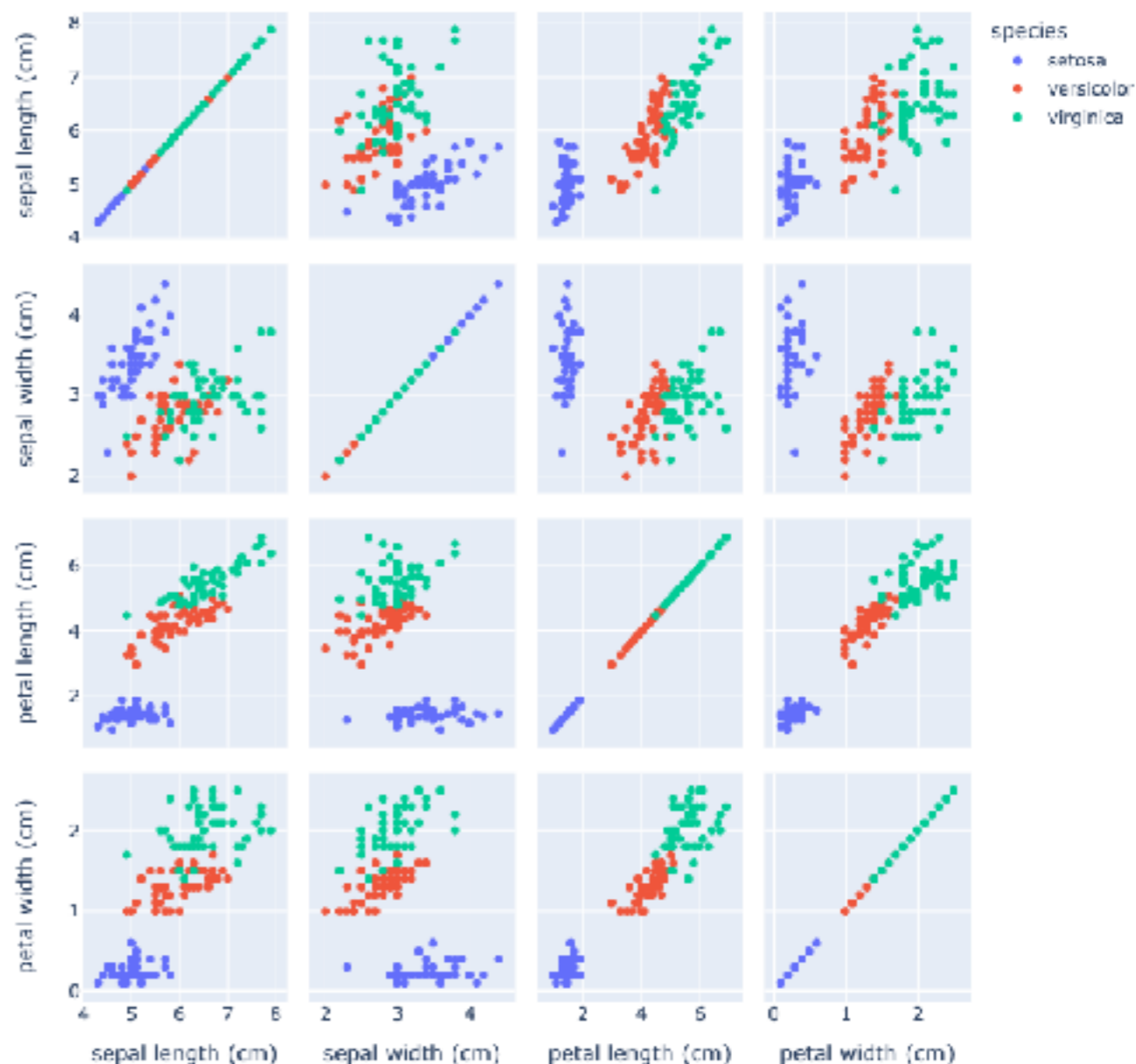
すいません。正直ほとんど使ったことありません・・・

散布図行列 (対角成分は同じ変数の散布図)

```
import plotly.express as px

fig = px.scatter_matrix(iris,
                       dimensions=["sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width (cm)"],
                       color="species",
                       width=800,
                       height=800)

fig.show()
```



後からの変更は

fig.update_layoutメソッドでも可能

データ可視化のWebアプリフレームワーク

 Cloud Gallery Components Community Docs Blog

Sign in

A faster way to build and share data apps

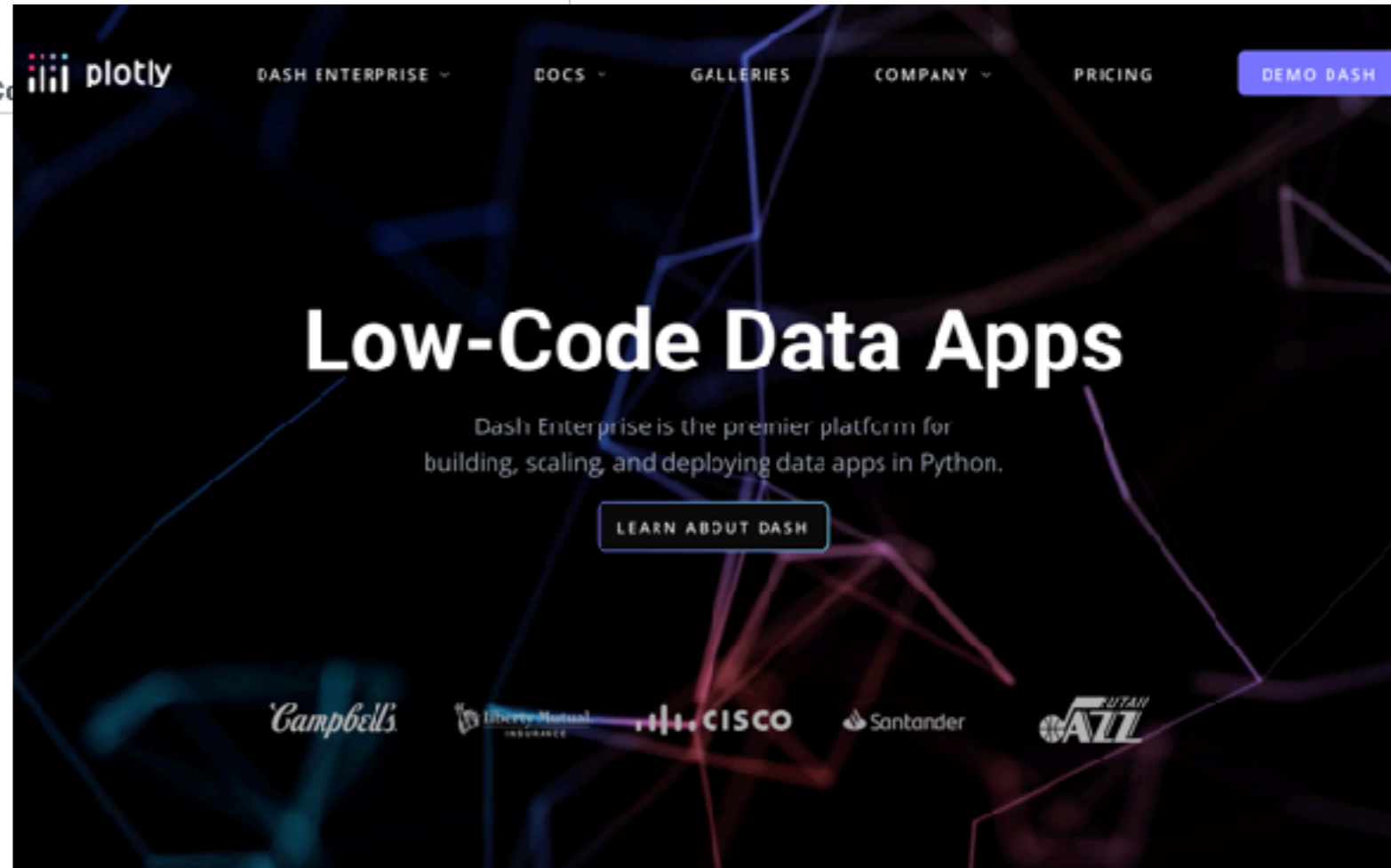
Streamlit turns data scripts into shareable web apps in minutes.
All in pure Python. No front-end experience required.

[Try Streamlit now](#)

[Sign up for Streamlit Cloud](#)

<https://plotly.com/>

<https://streamlit.io/>



The image shows a screenshot of the Plotly Dash Enterprise website. The background is dark with a network-like pattern of blue and purple lines. At the top, there is a navigation bar with the Plotly logo and links for DASH ENTERPRISE, DOCS, GALLERIES, COMPANY, and PRICING. A blue button labeled "DEMO DASH" is on the right. The main heading is "Low-Code Data Apps" in large white text. Below it, a sub-heading reads "Dash Enterprise is the premier platform for building, scaling, and deploying data apps in Python." A button labeled "LEARN ABOUT DASH" is centered below the text. At the bottom, there are logos for several partner companies: Campbell's, Liberty Mutual Insurance, CISCO, Santander, and JAZZ.

デモ コードは以下のGithub Gistにあります

<https://gist.github.com/tsjshg/97fc91a8bfc50e495a5296c06569629f>

Streamlit or Dash ?

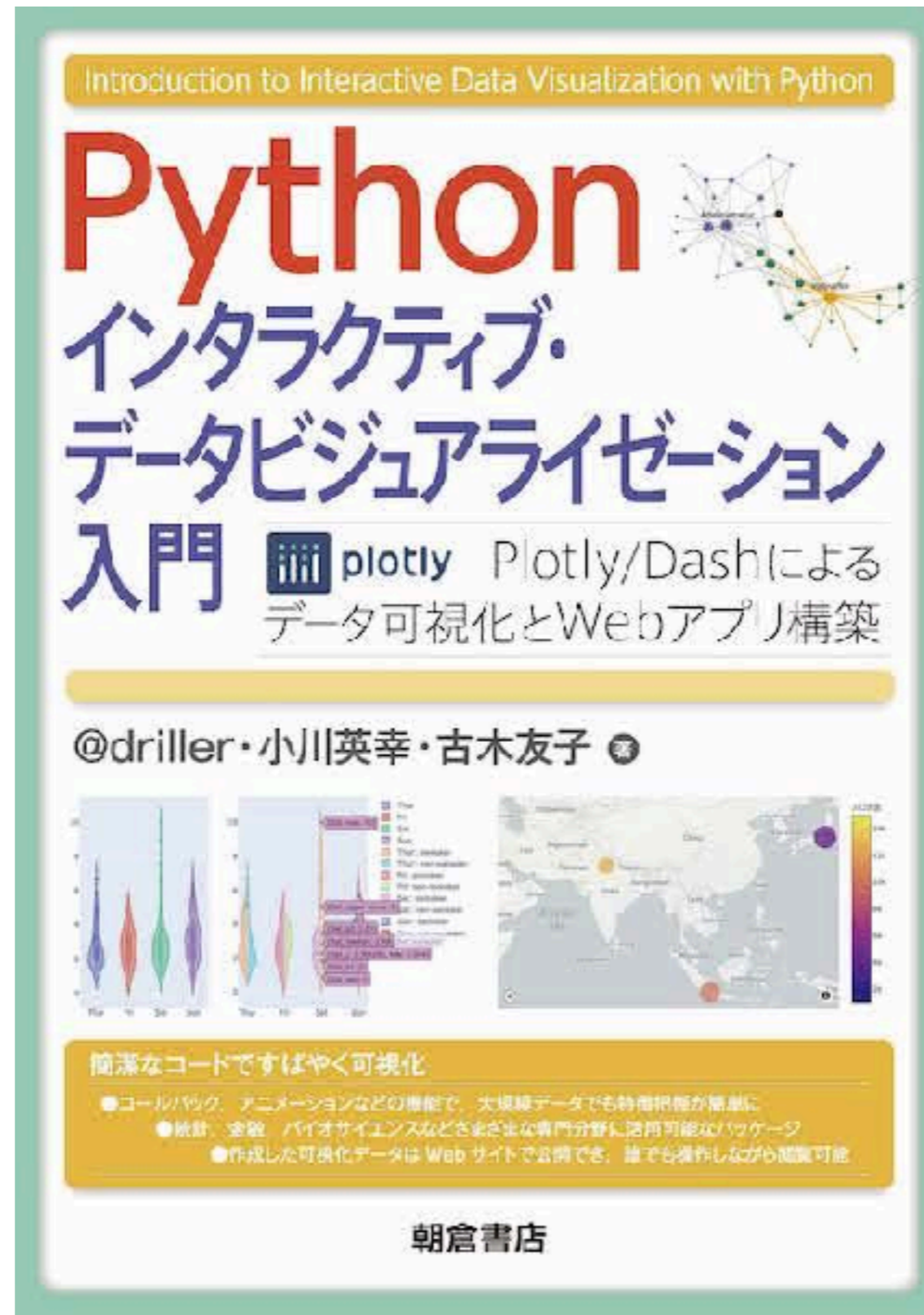
Streamlit

- Pythonだけでデータサイエンス用Webアプリが簡単に作れる
- △用意された基本デザインを踏襲する必要がありそう
- ○HTTPレスポンスに時間がかかる場合もよしなに対応
- △認証の仕組みが作りにくい？
- ○ニッキーの記事がある
 - <https://nikkie-ftnext.hatenablog.com/entry/streamlit-magical-cache>

Dash

- レイアウトのためにはHTMLとCSSの知識もあった方がよい
- ○自由に柔軟なデザインが可能
- △タイムアウトエラーが起きるような場合はコードを変更する必要
- ○BASIC認証ならすぐに実装できる
- ○日本語の解説本がある

Pythonインタラクティブ・データビジュアライゼーション入門 (朝倉書店)



まとめ

- MatplotlibはPythonで可視化の基本
 - ただ、そのままだとすこし不便
 - pandasやseabornの利用が便利
 - Matplotlibを知っていると細かいことができる
- JavaScriptを使ったインタラクティブな可視化
 - Plotlyが便利（Bokehの感想を誰かに教えてほしい）
- データサイエンス向けWebアプリケーションフレームワーク
 - Streamlitは少ないコードでそれなりのWebアプリがすぐできる
 - HTMLやCSSの知識があるならDashで凝ったアプリを作ることできるかも