

名前：はら ひろこ

趣味：すいみん

Python歴：1年くらい（Python以外に普段使うのは、Excel、R、Matlab）



## つまづくポイント

- ① 必要な基礎知識が多い、ような気がする。

演算 **+** は使うけれど、**%** や **//** はあまり使わない。  
全て覚える必要はない！

- ② テキストで学んでも、おもしろくない。

基礎を知らなくても、マネから入ればいい。

(例) python グラフ、、、で検索すれば、目当てに近いプログラムが出てくる。

- ③ 実務に即していない。

CSVファイルを読み込む時、`read(〇〇.csv)` とか書いているけど、ファイルの場所を意識してないでしょ！と言いたい。

# 0. Python初心者勉強会の予定

## 3回でやりたいこと

### Python の基礎

- Jupyter Notebookの使い方
- リストと辞書型/条件分岐とループ
- ライブラリの読み込み (Numpy/Scipy/Pandas/Matplotlib)

### 科学計算とデータ加工処理

- NumpyとScipy
- Pandas
- 欠損データと異常値、時系列データの取り扱い

### データの可視化

- グラフ
- シミュレーション

### 機械学習

Scikit-Learn/TensorFlow/...

### 画像処理

OpenCV/Scikit-Image/...

### Webスクレイピング

Requests/Beautiful Soup/...

### WEB開発

Django/Flask/Web2Py/...

### 音声録音・再生

PyAudio/scipy.signal/...

### ゲーム開発

PyGame/Arcade/PyGlet/...

# 0. Python初心者勉強会（前回のおさらい）

## 2-1. 配列の種類

List	Array	Series	Dataframe
Python 標準の配列	外部ライブラリの配列		
	numpy で定義された配列	Pandas で定義された	Pandas で定義された
['A大学', 'B大学', 'C大学'] [11, 12, 13] [123, 'abc', 456.789]	[[11 12 13] [21 22 23] [31 32 33]]		
—	import numpy		
list1 = ["A大学", "B大学", "C大学"]	ary2 = np.array([13], [21, 22, 23], 33])		
柔軟性：高い 処理速度：遅い			

## 3-1. 配列の相互変換

		To (変更後)			
		list	Array	Series	DataFrame
From (変更前)	list (ist1)	—			nd.DataFrame(li
	Array (ary1)	ary1.tolist()			
	Series (se1)	se1.values.tolist()			
	DataFrame (df1)	df1.values.tolist()			

## 4. DataFrame の操作

# 0. 参考テキスト



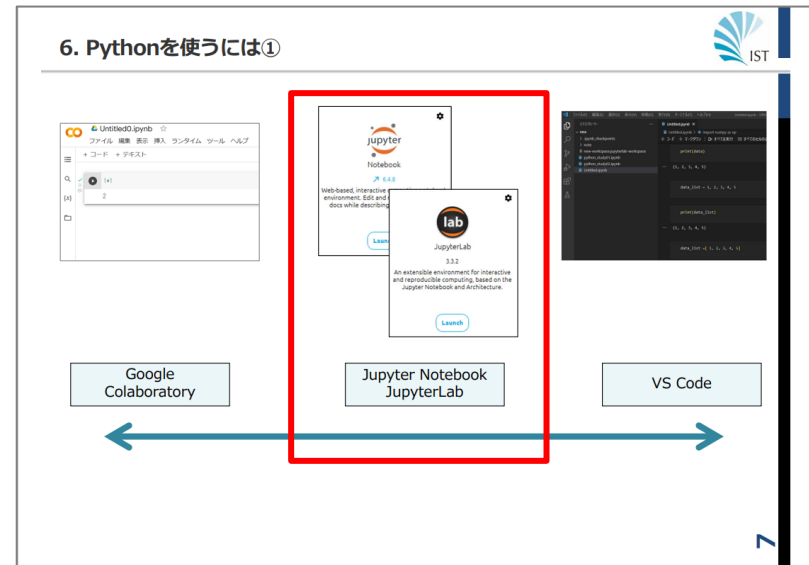
[https://mitani.cs.tsukuba.ac.jp/book\\_support/python/](https://mitani.cs.tsukuba.ac.jp/book_support/python/)

WEB上にある無料のテキストを使ったり、Connpasのような勉強会に参加する方法もあります！

私はPyQ（月額3,040円）のPython独学プラットフォームで3か月間くらい写経しました。

## 全体の流れ

1. Pythonに触れる
2. Pythonの基本
3. 条件分岐と繰り返し
4. 組み込み型とオブジェクト
5. ユーザー定義関数
6. クラスの基本
7. 発展と応用



## 目次

### 1. csvファイルの読み込み

### 2. さまざまなグラフ

- 折れ線グラフ
- ヒストグラム
- 円グラフ

### 3. 参考図書

### 4. 質問

# 1. csvファイルの読み込み

# 1. csvファイルの読み込み

## test.csv

	A大学	B大学	C大学	D大学	E大学	F大学	G大学
2018年	1000	2000	3000	4000	5000	6000	7000
2019年	1001	2001	3001	4001	5001	6001	7001
2020年	1002	2002	3002	4002	5002	6002	7002
2021年	1003	2003	3003	4003	5003	6003	7003
2022年	1004	2004	3004	4004	5004	6004	7004

```
1 import pandas as pd
```

```
1 df = pd.read_csv("test.csv")
2 df
```

Unnamed: 0   A大学   B大学   C大学   D大学   E大学   F大学   G大学

0	2018年	1000	2000	3000	4000	5000	6000	7000
1	2019年	1001	2001	3001	4001	5001	6001	7001
2	2020年	1002	2002	3002	4002	5002	6002	7002
3	2021年	1003	2003	3003	4003	5003	6003	7003
4	2022年	1004	2004	3004	4004	5004	6004	7004

```
1 df = pd.read_csv("test.csv", index_col=0)
2 df
```

	A大学	B大学	C大学	D大学	E大学	F大学	G大学
2018年	1000	2000	3000	4000	5000	6000	7000
2019年	1001	2001	3001	4001	5001	6001	7001
2020年	1002	2002	3002	4002	5002	6002	7002
2021年	1003	2003	3003	4003	5003	6003	7003
2022年	1004	2004	3004	4004	5004	6004	7004

インデックスを指定



# 1. csvファイルの読み込み（参考：さまざまな引数を試してみる）

```
1 df1 = pd.read_csv("test.csv", header=None)
2 df1
```

	0	1	2	3	4	5	6	7
0	NaN	A大学	B大学	C大学	D大学	E大学	F大学	G大学
1	2018年	1000	2000	3000	4000	5000	6000	7000
2	2019年	1001	2001	3001	4001	5001	6001	7001
3	2020年	1002	2002	3002	4002	5002	6002	7002
4	2021年	1003	2003	3003	4003	5003	6003	7003
5	2022年	1004	2004	3004	4004	5004	6004	7004

```
1 df2 = pd.read_csv("test.csv", header=0)
2 df2
```

	Unnamed: 0	A大学	B大学	C大学	D大学	E大学	F大学	G大学
0	2018年	1000	2000	3000	4000	5000	6000	7000
1	2019年	1001	2001	3001	4001	5001	6001	7001
2	2020年	1002	2002	3002	4002	5002	6002	7002
3	2021年	1003	2003	3003	4003	5003	6003	7003
4	2022年	1004	2004	3004	4004	5004	6004	7004

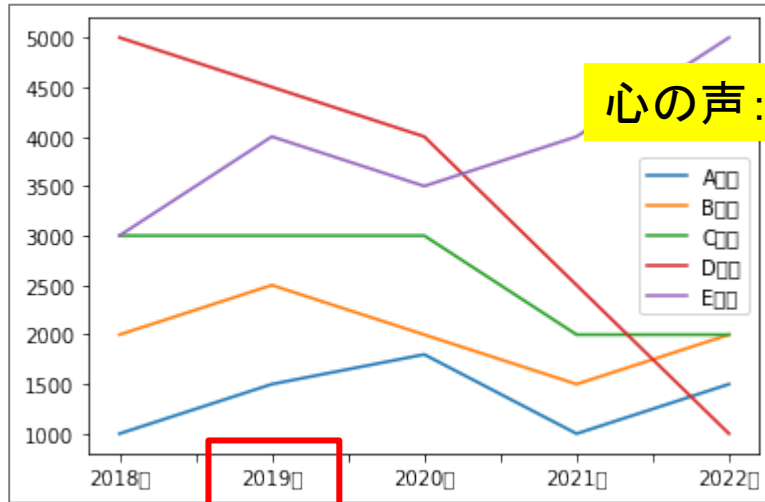
```
1 df3 = pd.read_csv("test.csv", header=1)
2 df3
```

	2018年	1000	2000	3000	4000	5000	6000	7000
0	2019年	1001	2001	3001	4001	5001	6001	7001
1	2020年	1002	2002	3002	4002	5002	6002	7002
2	2021年	1003	2003	3003	4003	5003	6003	7003
3	2022年	1004	2004	3004	4004	5004	6004	7004

## 2. さまざまなグラフ

## 2-1. 折れ線グラフ

```
1 df.plot()
```

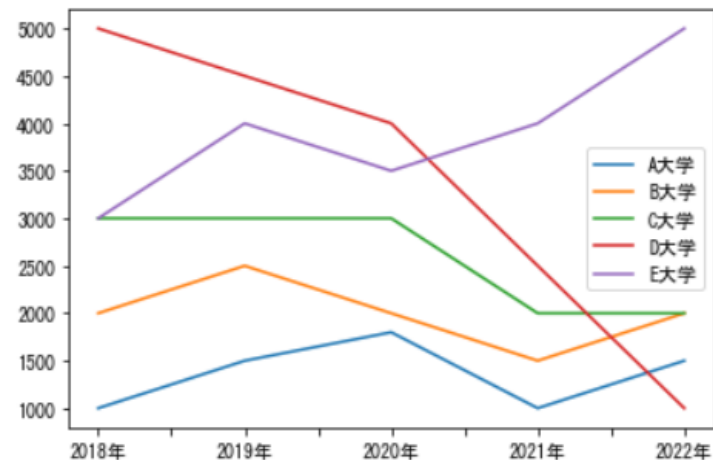


心の声: きっと他の人も同じように困っているに違いない。

日本語の文字化け発生！

```
1 # Matplotlibでの日本語表示
2 # (参考)https://www.yutaka-note.com/entry/matplotlib\_japanese
3 plt.rcParams['font.family'] = "MS Gothic"
4
5 df.plot()
```

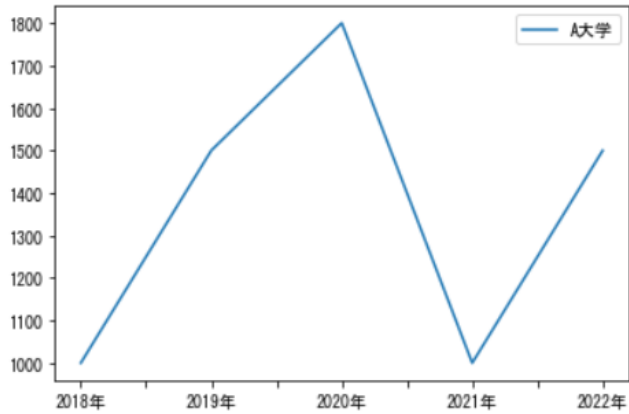
<AxesSubplot:>



## 2-1. 折れ線グラフ

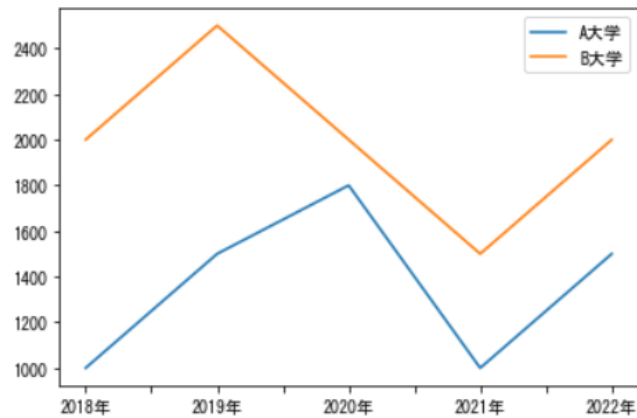
```
1 df.plot( y='A大学')
```

<AxesSubplot:>

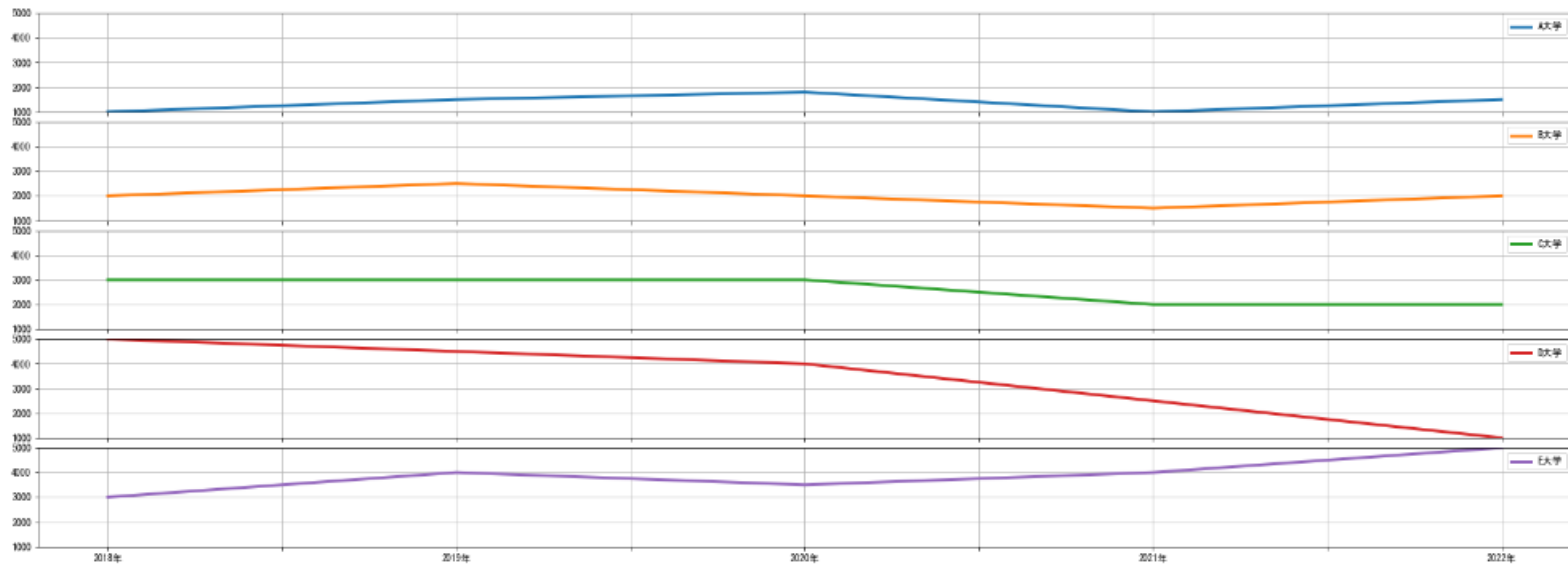


```
1 df.plot(y=['A大学', 'B大学'])
```

<AxesSubplot:>



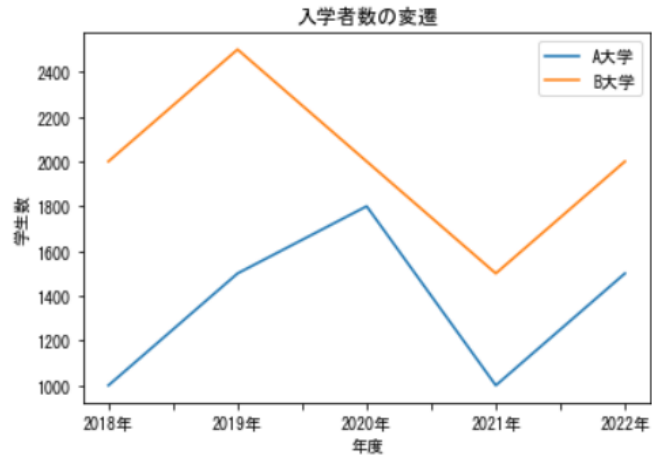
```
1 df.plot(subplots=True, figsize=(28.00, 10.00), grid=True, lw=3, ylim=(1000, 5000))
2 plt.subplots_adjust(hspace=0.1)
```



## 2-1. 折れ線グラフ

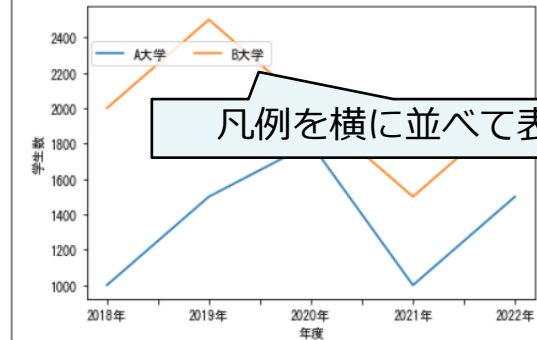
```
1 df.plot(y=["A大学", "B大学"], xlabel="年度", ylabel="学生数", title="入学者数の変遷")
```

<AxesSubplot:title=['center':'入学者数の変遷'], xlabel='年度', ylabel='学生数'>



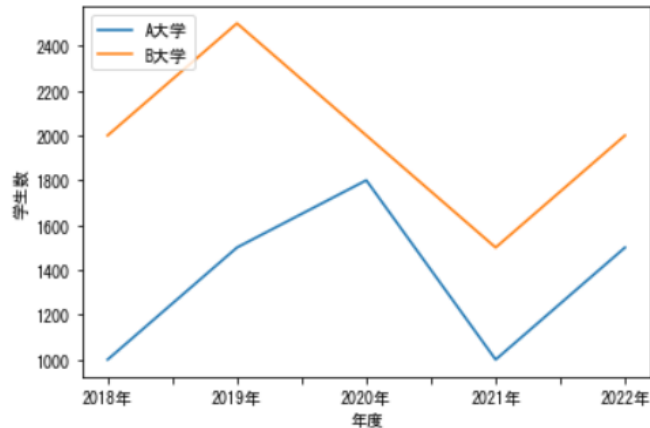
```
1 df.plot(y=["A大学", "B大学"], xlabel="年度", ylabel="学生数")
2 plt.legend(loc=(0.01, 0.8), ncol=2)
```

<matplotlib.legend.Legend at 0x2709622b148>



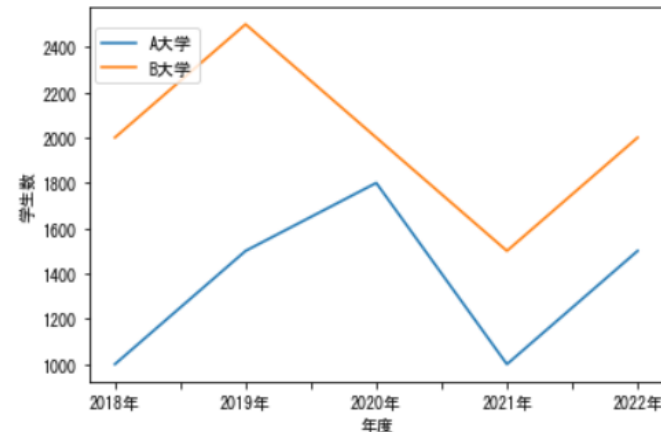
```
1 df.plot(y=["A大学", "B大学"], xlabel="年度", ylabel="学生数")
2 plt.legend(loc="upper left")
```

<matplotlib.legend.Legend at 0x27097788cc8>



```
1 df.plot(y=["A大学", "B大学"], xlabel="年度", ylabel="学生数")
2 plt.legend(loc=(0.01, 0.8))
```

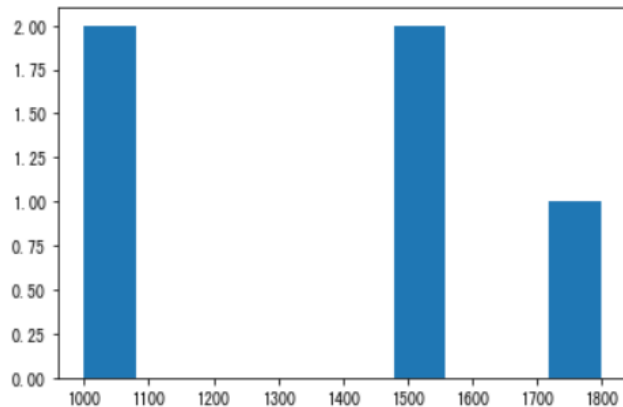
<matplotlib.legend.Legend at 0x2709793cb88>



## 2-2. ヒストグラム

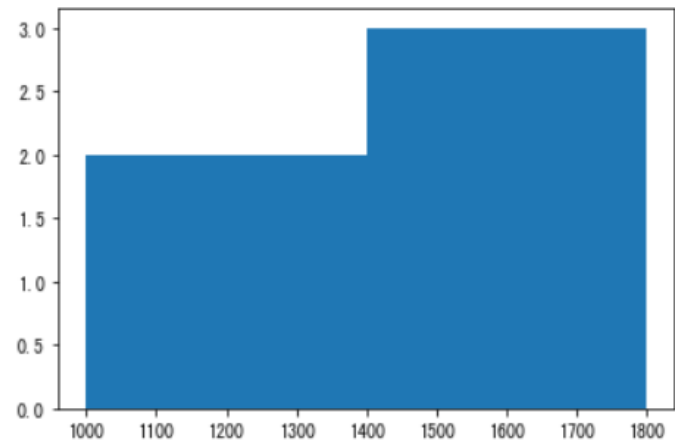
```
1 plt.hist(df["A大学"])
```

```
(array([2., 0., 0., 0., 0., 0., 2., 0., 0., 1.]),
 array([1000., 1080., 1160., 1240., 1320., 1400., 1480., 1560., 1640.,
        1720., 1800.]),
 <BarContainer object of 10 artists>)
```

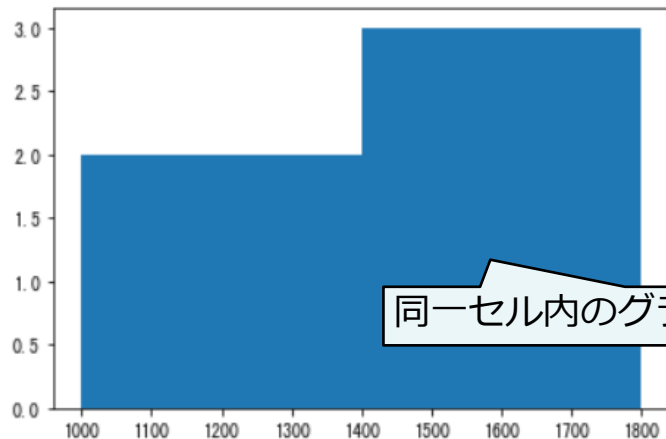


```
1 plt.hist(df["A大学"], bins=2)
```

```
(array([2., 3.]),
 array([1000., 1400., 1800.]),
 <BarContainer object of 2 artists>)
```



```
1 plt.hist(df["A大学"], bins=2)
2 plt.savefig("test.png")
```

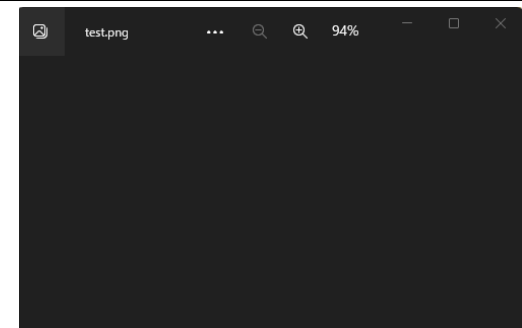


```
1 plt.savefig("test.png")
```

<Figure size 432x288>

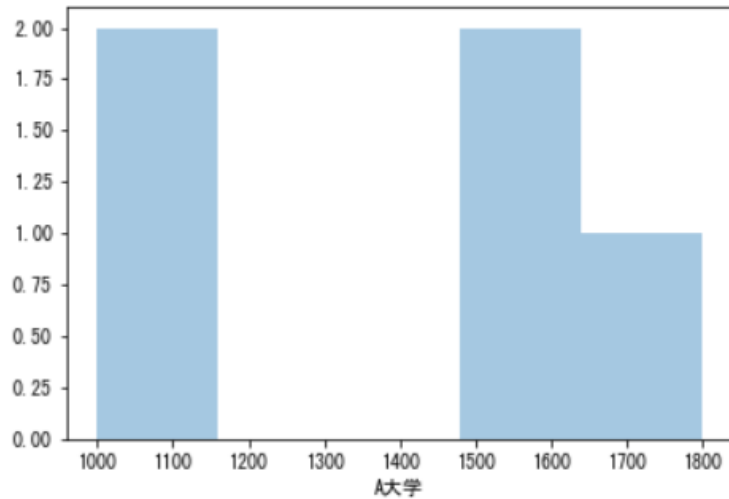
このセル内にはグラフデータの表記はない。  
そのため、空のデータが保存される。

同一セル内のグラフを保存する。

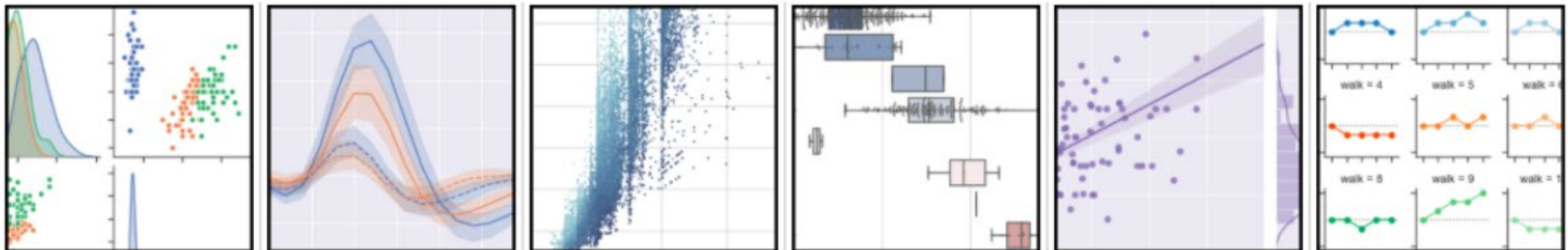


## 2-2. ヒストグラム（参考：seabornの紹介）

```
1 import seaborn as sns
2 sns.distplot(df["A大学"], kde=False, rug=False, bins=5)
3 plt.show()
```



### Seaborn データ可視化入門



(出展) <https://www.codexa.net/seaborn-python/>

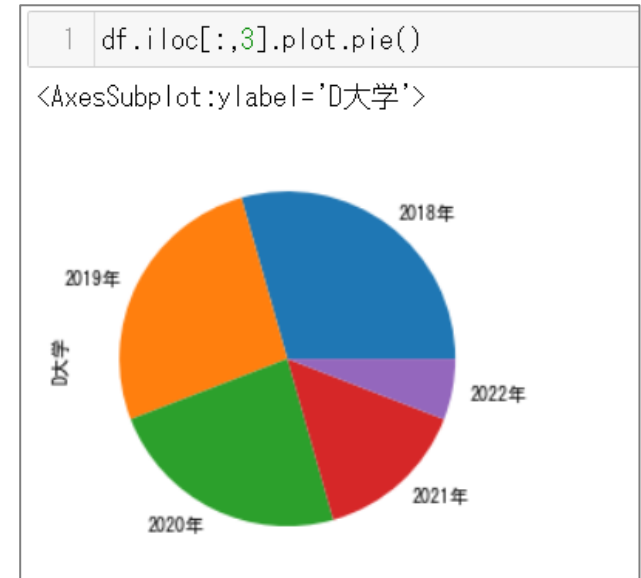
## 2-3. 円グラフ

```
1 df
```

	A大学	B大学	C大学	D大学	E大学
2018年	1000	2000	3000	5000	3000
2019年	1500	2500	3000	4500	4000
2020年	1800	2000	3000	4000	3500
2021年	1000	1500	2000	2500	4000
2022年	1500	2000	2000	1000	5000

```
1 df.iloc[:,3]
```

2018年 5000  
 2019年 4500  
 2020年 4000  
 2021年 2500  
 2022年 1000  
 Name: D大学, dtype: int64



```
1 df
```

	A大学	B大学	C大学	D大学	E大学
2018年	1000	2000	3000	5000	3000
2019年	1500	2500	3000	4500	4000
2020年	1800	2000	3000	4000	3500
2021年	1000	1500	2000	2500	4000
2022年	1500	2000	2000	1000	5000

```
1 df.loc[["2019年"]]
```

	A大学	B大学	C大学	D大学	E大学
2019年	1500	2500	3000	4500	4000



## 2-3. 円グラフ

```
1 df.loc[["2019年"]].plot.pie()
```

**ValueError**

Traceback (most recent call last)

C:\Users\Public\Documents\Wondershare\CreatorTemp\ipykernel\_19800\810301371.py in <module>

----> 1 df.loc[["2019年"]].plot.pie()

C:\ProgramData\Anaconda3\envs\Yura\lib\site-packages\pandas\plotting\\_core.py in pie(self, \*\*kwargs)

1548 and not kwargs.get("subplots", False)

1549 ):

-> 1550 raise ValueError("pie requires either y column or 'subplots=True'")

1551 return self(kind="pie", \*\*kwargs)

1552

**ValueError:** pie requires either y column or 'subplots=True'

```
1 df.loc[["2019年"]].plot.pie(subplots=True)
```

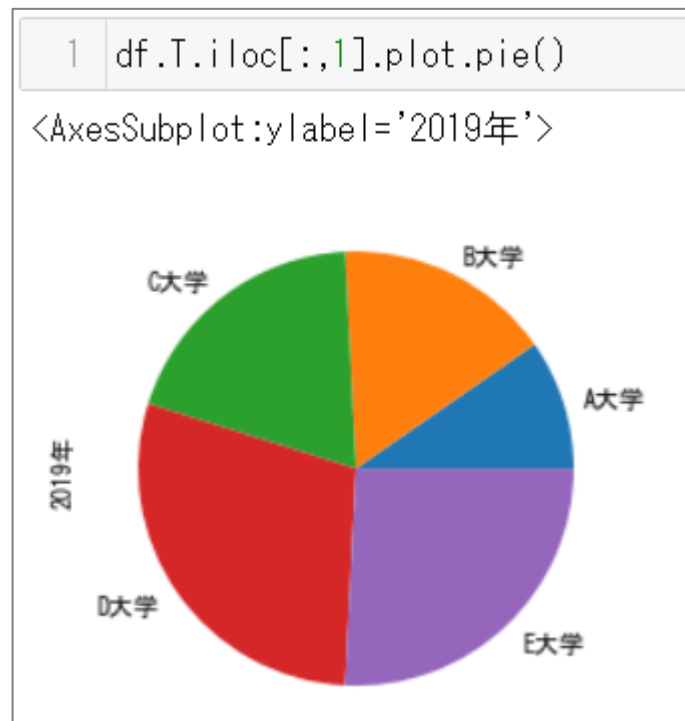
array([<AxesSubplot:ylabel='A大学'>, <AxesSubplot:ylabel='B大学'>, <AxesSubplot:ylabel='C大学'>, <AxesSubplot:ylabel='D大学'>, <AxesSubplot:ylabel='E大学'>], dtype=object)



## 2-3. 円グラフ

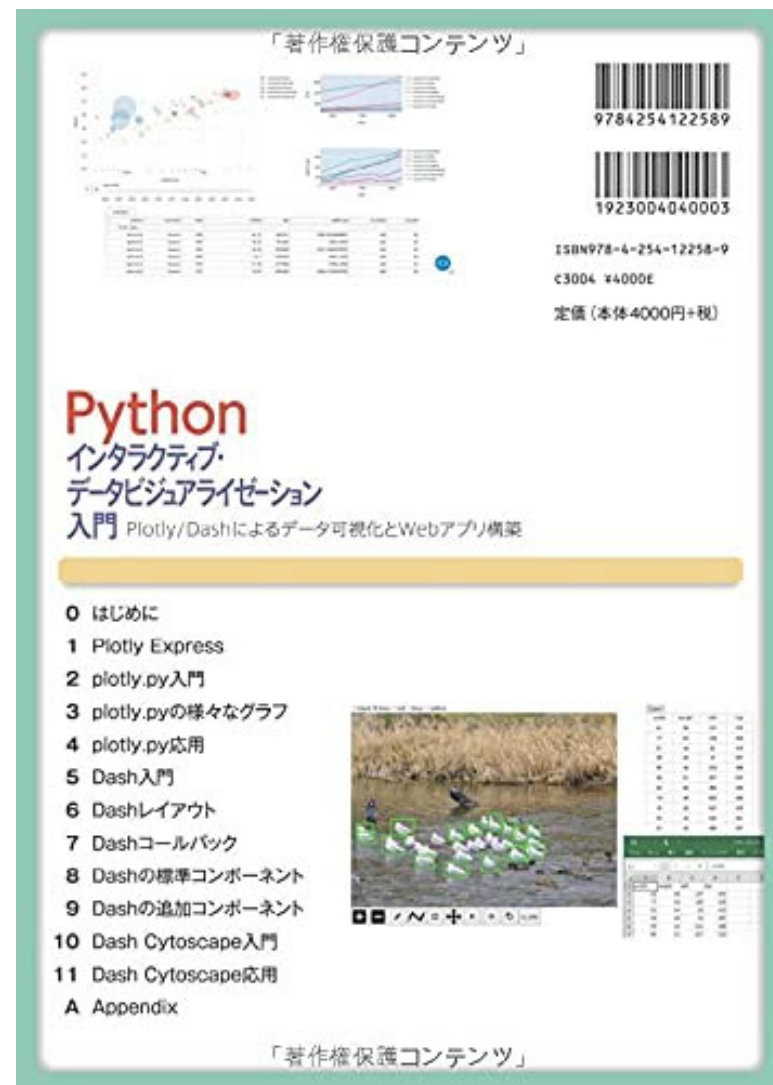
1	df.T				
	2018年	2019年	2020年	2021年	2022年
A大学	1000	1500	1800	1000	1500
B大学	2000	2500	2000	1500	2000
C大学	3000	3000	3000	2000	2000
D大学	5000	4500	4000	2500	1000
E大学	3000	4000	3500	4000	5000

Y軸を作る



### 3. 参考図書

### 3. グラフに関する参考図書



## 4. 質問

## 4. 質問

独自にプログラミングするのは難しいので、他の方がGitHub等インターネット上に公開しているプログラムを借用し、若干の改変を加えて使用している。

その解析結果を論文にするときの著作権などの問題への対応を教えてください。

利用するプログラムの利用条件による。GitHubで公開されているということは、閲覧は自由であることは間違いありませんが、改変・利用・再配布は、そのプログラムに定められた利用条件次第です。

OSSライセンスであるGPLv3ライセンスが付いたプログラムの場合には、改変・利用・再配布が認められているので、研究論文で何らかの言及をしなくても問題ないと考えられます（以下のQ&A参照）。

<https://www.gnu.org/licenses/gpl-faq.ja.html#RequireCitation>

さらに、解析結果となると、プログラムの元の著作者の著作権は及ばない（GPLかどうかに関わらず）です。

<https://www.gnu.org/licenses/gpl-faq.ja.html#GPLOutput>

なお、改変したプログラムにおいて、GPLライセンスの表示をすることは必要です。

<https://www.gnu.org/licenses/gpl-howto.html>

OSSライセンス付きのプログラムであったとしても、参考にした論文やGitHubアドレスは、記載する方がよさそう。

GPL以外にもOSSライセンスには色々な種類があり、OSSライセンス付きならば改変・利用・再配布はOKなので、研究目的で用いる場合はそれ程気を使わなくても良いと思います。ただ、OSSライセンスによりそれぞれ条件が異なるので、利用しようとしている元のプログラムの利用条件を確かめることが必要かなと思います。

<https://www.tohoho-web.com/ex/license.html>

ただ、以下の記事を見ると、GitHub上にはOSSライセンスを登録していないプログラムも多数公開されているということであり、その様な場合は**普通に著作権法が適用される**、つまり使用（実行・ソースコードの閲読・コンパイル）することはできますが、利用（複製・再配布・二次著作物の作成）することは（自由には）できないことになります。

<https://thinkit.co.jp/story/2014/02/24/4843>

その場合は、著作権者に許可を取って、プログラムの改変等を行うことが必要になると思います。

## 目次

1. csvファイルの読み込み
2. さまざまなグラフ
  - 折れ線グラフ
  - ヒストグラム
  - 円グラフ
3. 参考図書
4. 質問

ご清聴  
ありがとうございました

