作図の基本

初めてのプログラミング

2020年度

只木進一（理工学部）

今日の目的

- 作図の基本：グラフ
- 注意：使用している環境では、図中に日本語を表示できない

- https://github.com/first-programming-saga/plotSample

簡単な図

- matplotlibを利用
  - title()：図の表題
  - scatter()：散布図（データを点で表示）
  - plot()：データを折れ線で表示
  - legend()：凡例表示
  - show()：図を表示
  - savefig()：ファイルへ保存

plotsample/simplePlot.ipynb

作図の基本

- データと目的に応じた適切は方法
  - 時系列→折れ線
  - 相関→散布図
- 図で何を示したいか
  - 表示方法、表示範囲

plot の使い方

```
plt.plot(x,y,label='theory',linewidth=2)
```

- plot(x,y)
  - xにx軸のデータリスト、yにy軸のデータリストを与え、折れ線で結ぶ
- plot(x,y,'bo')
  - 青'b'の点'o'でデータをプロット
- その他のキーワード
  - label : ラベル
  - linewidth : 線の太さ

https://matplotlib.org/api/index.html

scatterの使い方

```
plt.scatter(x,z,label='data',color='red',marker='s')
```

- scatter(x,y)
  - xにx軸のデータリスト、yにy軸のデータリストを与え、プロット
- その他のキーワード
  - color : 色
  - label : ラベル
  - linewidth : 線の太さ
  - marker : 点の形

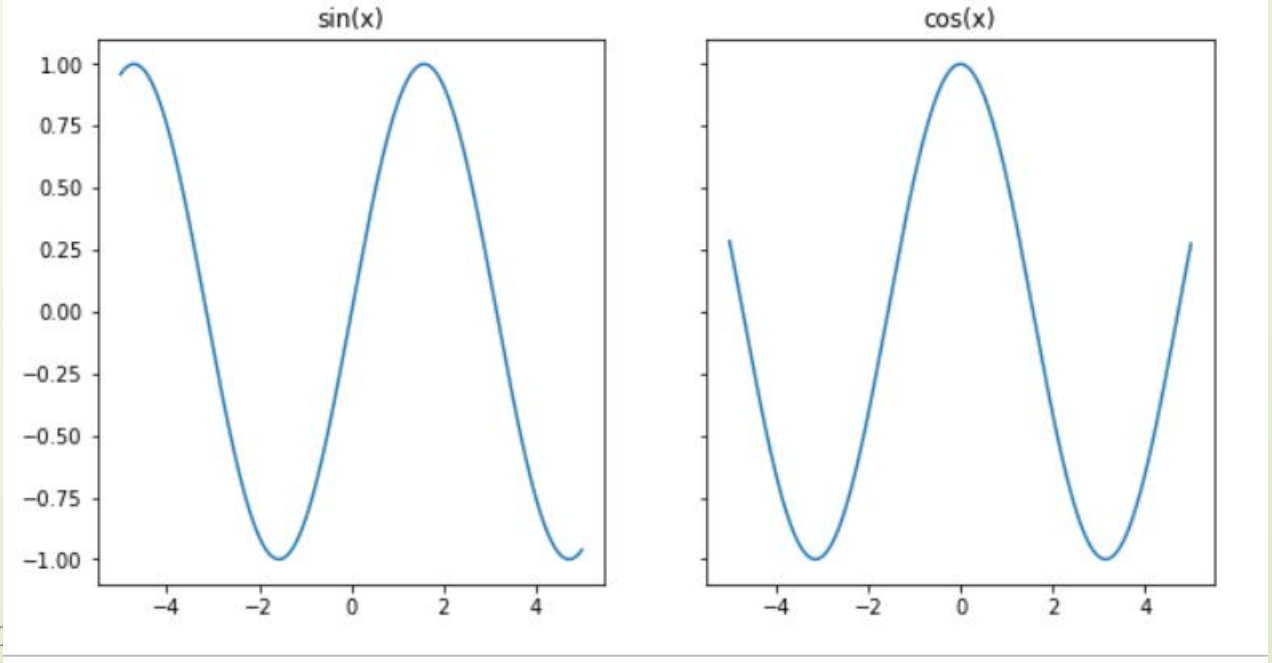
https://matplotlib.org/api/markers_api.html#module-matplotlib.markers

複数の図を描く（省略）

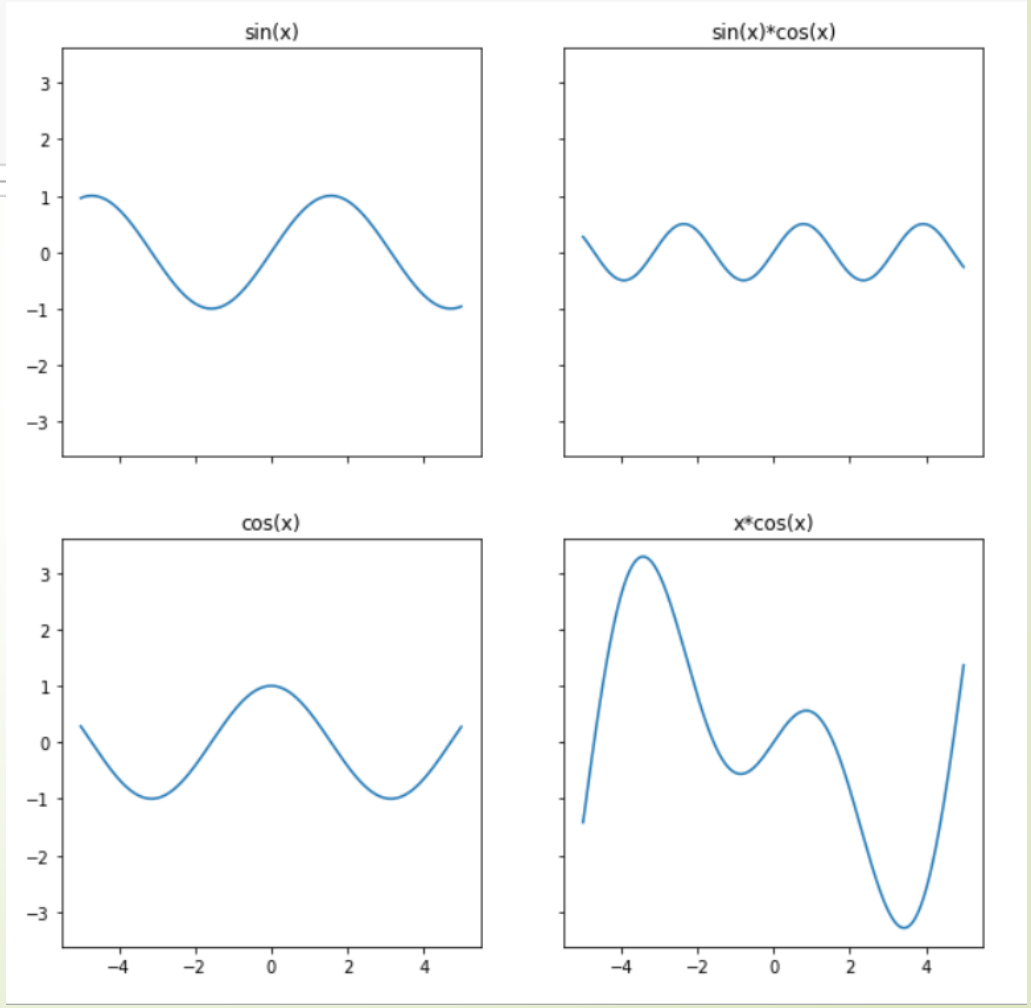
- fig,ax=subplots()
  - 行と列を指定
  - 戻り値は、全体(fig)と各作図領域(ax)

plotsample/multiPlot.ipynb
plotsample/multiPlot2.ipynb

```
1  def draw(x,y,z):
2      fig,ax=plt.subplots(1,2,figsize=(10,5),sharey=True)
3      ax[0].set_title('sin(x)')
4      ax[0].plot(x,y)
5      ax[1].set_title('cos(x)')
6      ax[1].plot(x,z)
7      plt.show()
```



```python
def draw(x,y1,y2,y3,y4):
    fig,ax=plt.subplots(2,2,figsize=(10,10),sharex=True,sharey=True)#xy軸のスケールを揃える
    ax[0,0].set_title('sin(x)')
    ax[0,0].plot(x,y1)
    ax[1,0].set_title('cos(x)')
    ax[1,0].plot(x,y2)
    ax[0,1].set_title('sin(x)*cos(x)')
    ax[0,1].plot(x,y3)
    ax[1,1].set_title('x*cos(x)')
    ax[1,1].plot(x,y4)
    plt.show()
```



図形を描く

- matplotlib.patches
  - 基本図形が用意されている
- 任意の多角形
  - 座標とMOVE/LINETOの組を対応付ける

plotsample/drawShapes.ipynb

円グラフ

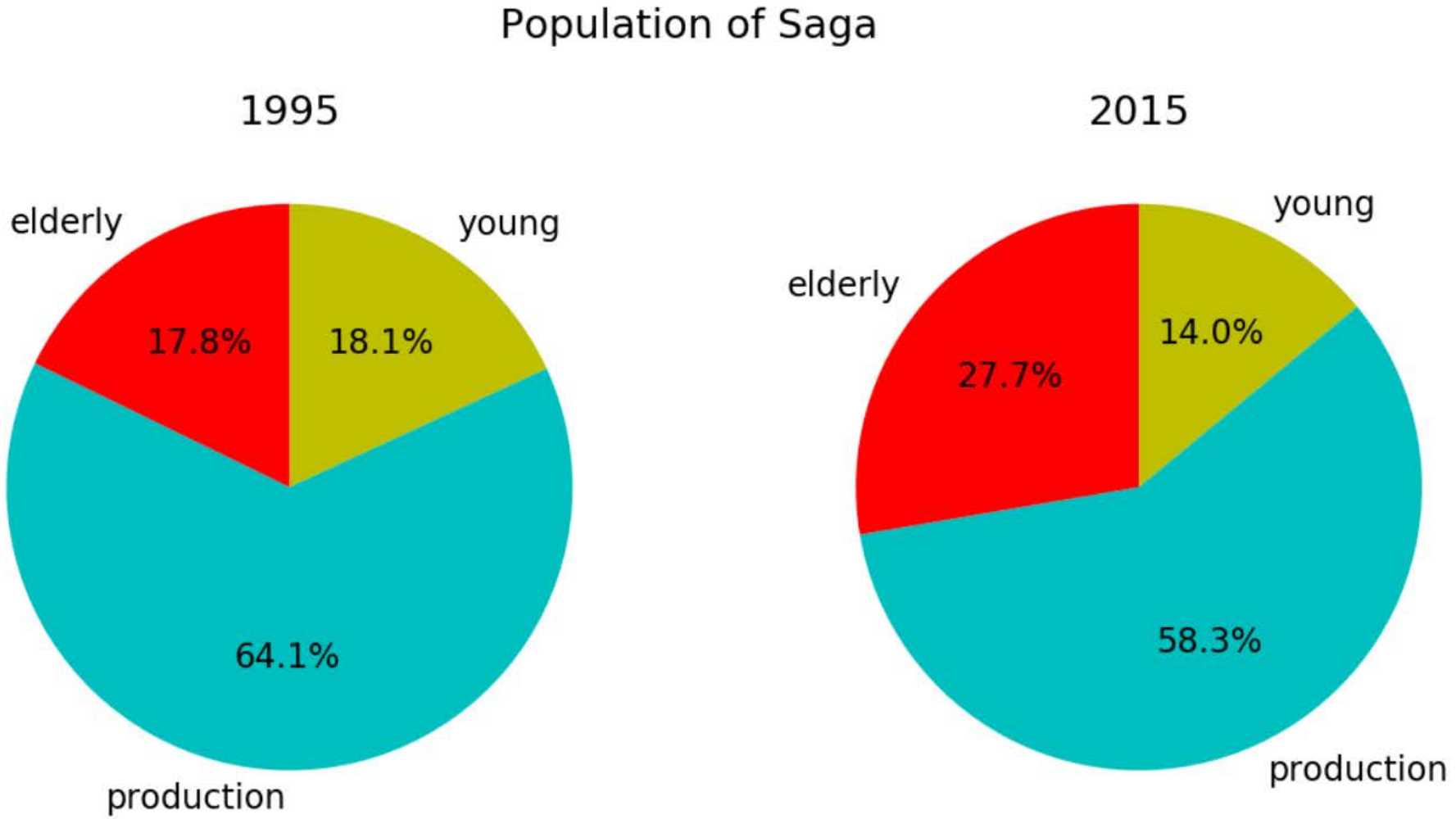
- pi()を使用

```
data2015=[116122,483019,229335]
data1995=[160307,566671,157329]
label=['young','production','elderly']
color=['y','c','r']

fig,ax=plt.subplots(1,2,figsize=(20,10))
plt.rcParams['font.size']=24 #文字の大きさ
fig.suptitle('Population of Saga') #全体のタイトル
#1995年の人口比率
ax[0].pie(data1995,labels=label,colors=color,
        startangle=90,counterclock=False,autopct='%1.1f%%')
ax[0].set_title('1995')
#2015年の人口比率
ax[1].pie(data2015,labels=label,colors=color,
        startangle=90,counterclock=False,autopct='%1.1f%%')
ax[1].set_title('2015')

plt.show()
```

plotsample/piPlot.ipynb

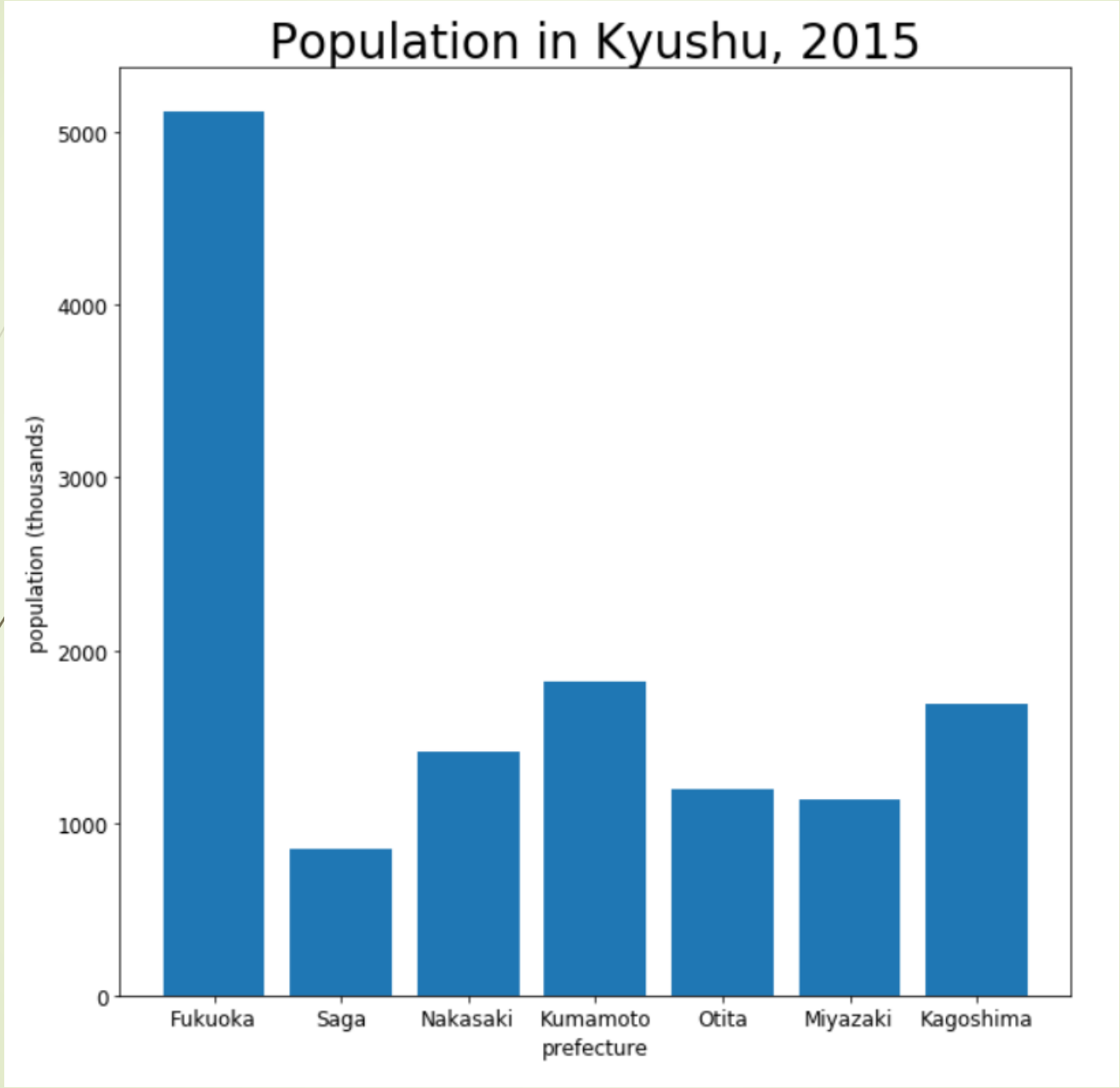

Population of Saga
1995
elderly
young
17.8%
18.1%
64.1%
production
2015
young
elderly
14.0%
27.7%
58.3%
production

棒グラフ

- bar()を使用
  - x軸、y軸のそれぞれのデータを準備
  - 各棒にラベルを指定できる

```
population=[5120,847,1413,1818,1191,1136,1691]
prefecture=['Fukuoka','Saga','Nakasaki','Kumamoto',
            'Otita','Miyazaki','Kagoshima']
x = [i for i in range(len(population))]
plt.figure(figsize=(10,10))
plt.rcParams['font.size']=12
plt.title('Population in Kyushu, 2015',fontsize=28)
plt.xlabel('prefecture')
plt.ylabel('population (thousands)')
plt.bar(x,population,tick_label=prefecture)
plt.show()
```


plotsample/plotBars.ipynb


Population in Kyushu, 2015
population (thousands)
0
1000
2000
3000
4000
5000
Fukuoka
Saga
Nakasaki
Kumamoto
Otita
Miyazaki
Kagoshima
prefecture

ヒストグラム

- hist()を使用
  - データとビン(bin)を指定
  - データから各ビンの頻度を自動計算してくれる

plotsample/histogram.ipynb

- bin
  - a container that you put waste in
  - a large container, usually with a lid, for storing things in

```
#元データ
data=[2.5,4.,5.5,9.2,0.7,8.8,6.1,7.1,4.3,1.6,
      3.4,1.4,7.7,4.4,5.5,9.6,6.8,8.6,2.1,
      8.9,2.9,1.1,2.1,3.7,2.0,1.4,0.5,5.7]
plt.figure(figsize=(10,10))
plt.title('histogram sample')
numBins = 10 #binの数
rWidth = 0.9 #描く幅は、等間隔の幅に対して0.9倍
plt.hist(data,range=(0,10),bins=numBins,rwidth=rWidth,color='cyan')
plt.show()
```