

エクセルのデータを作 図する

初めてのプログラミング

2020年度

只木進一（理工学部）

今日のサンプルプログラム

- ▶ <https://github.com/first-programming-saga/SagaPopulation>

Excelのデータをプロット

- ▶ Pandasの作図機能を直接利用できる
- ▶ 例
 - ▶ 九州各県の人口推移
 - ▶ 佐賀県の人口推移：詳細

KyushuPopulation.ipynb

- ▶ KyushuPopulation.xlsxを読み込み、
県毎の人口推移をプロット

Prefecture	1990	1995	2000	2005	2010	2015
Fukuoka	4,758	4,873	4,968	5,014	5,039	5,120
Saga	882	885	884	874	859	847
Nakasaka	1,573	1,554	1,532	1,502	1,450	1,413
Kumamoto	1,848	1,862	1,870	1,858	1,834	1,818
Oita	1,246	1,243	1,236	1,225	1,207	1,191
Miyazaki	1,182	1,187	1,187	1,173	1,153	1,136
Kagoshima	1,805	1,794	1,787	1,763	1,722	1,691

pandas.Seriesを利用する利点

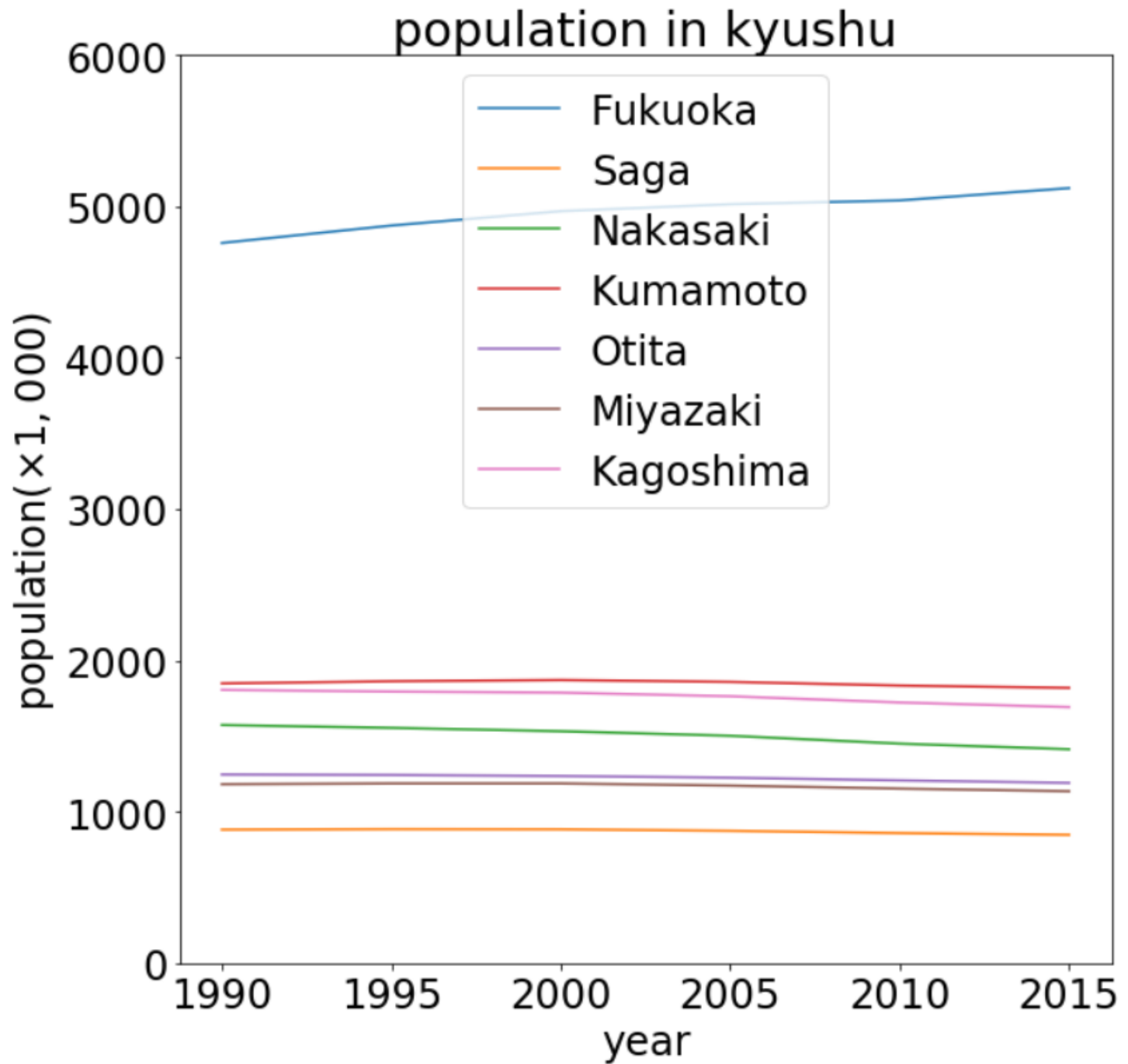
- ▶ 一行のデータをpandas.Seriesとして取り出し
- ▶ pandas.Series.plot()を使って折れ線グラフを作成
- ▶ indexと値の組になっている
- ▶ indexをx軸の値として利用できる

Pandas.Seriesのプロット機能

- ▶ Series.plot()
 - ▶ Seriesのデータをプロットする。
- ▶ 引数で指定できるもの
 - ▶ プロット方法：'line'、'bar'、'scatter'ほか
 - ▶ label：ラベル

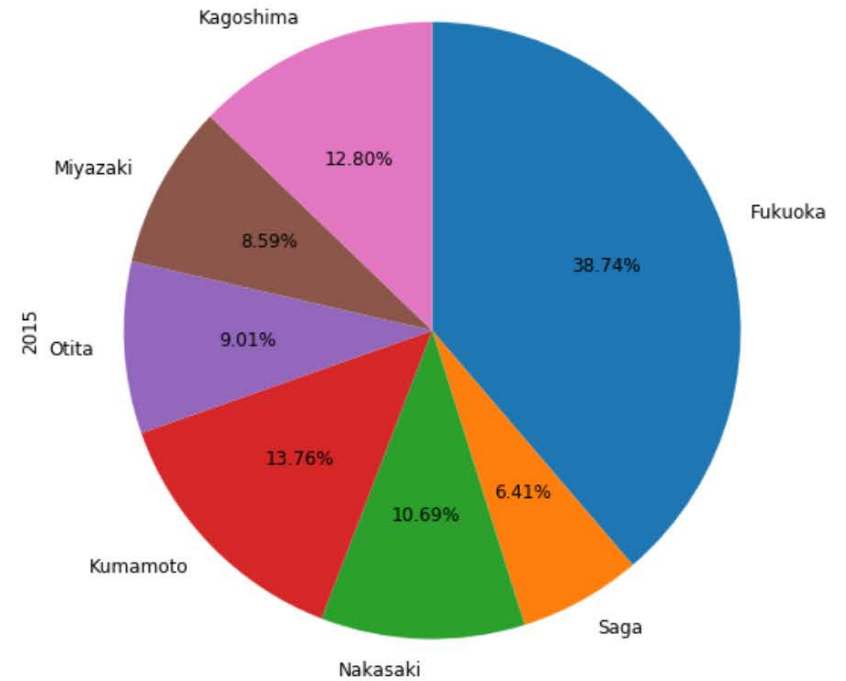
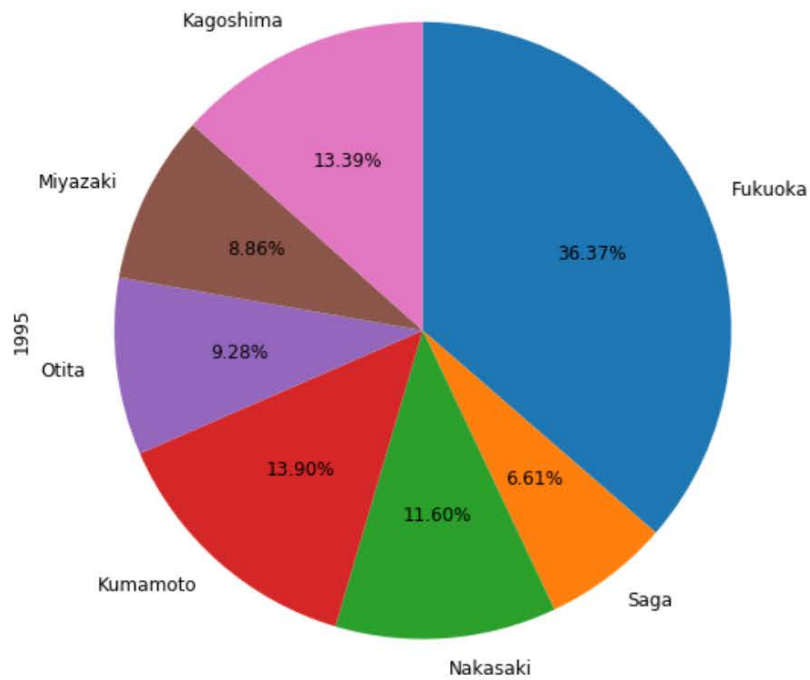
```
1 filename = 'KyushuPopulation.xlsx'
2 with pandas.ExcelFile(filename) as f:
3     df = pandas.read_excel(f)
4     df.set_index('Prefecture', inplace=True) #左端のカラムにインデクスを設定
5
6     #図の準備
7     plt.figure(figsize=(10,10))
8     plt.rcParams['font.size']=24
9     plt.title('population in kyushu')
10    plt.ylim(0,6000)
11    plt.xlabel('year')
12    plt.ylabel('population($¥¥times1,000$)')
13    #各県のデータをプロット
14    drawTimeSequence(df)
15
16    plt.legend()
17    plt.show()
```

```
1 def drawTimeSequence(df): #dataFrameを受け取る
2     for pref in df.index: #各行にある県名を取得
3         ser = df.loc[pref] #一行のデータ（時系列）を取得
4         #一行のデータをプロットし、ラベルとして県名を設定
5         ser.plot(label=pref)
```

```
1 filename = 'KyushuPopulation.xlsx'
2 with pandas.ExcelFile(filename) as f:
3     df = pandas.read_excel(f)
4 df.set_index('Prefecture', inplace=True) #左端のカラムにインデクスを設定
5
6 #図の準備
7 year = [1995, 2015]
8 xsize = 10*len(year)
9 fig, ax = plt.subplots(1, len(year), figsize=(xsize, 10))
10 plt.rcParams['font.size']=12
11 plt.suptitle('population in kyushu')
12 #各県のデータをプロット
13 for i in range(len(year)):
14     ser = df[year[i]]
15     ser.plot.pie(ax=ax[i], startangle=90, counterclock=False, autopct='%1.2f%%')
16 plt.show()
```

population in kyushu



population.ipynb

- ▶ 「ビッグデータ&オープンデータ・イニシアティブ九州」のデータを活用
 - ▶ <https://www.bodik.jp/>
- ▶ URL上のエクセルファイルを直接読み込む
 - ▶ URL上のエクセルをプログラムから読み込むのに工夫が必要なサイトがある

- ▶ 残念ながら、`index`を簡単には取り出せない
- ▶ `DataFrame.index`が使えない場合
 - ▶ 行数でデータを指定
 - ▶ `DataFrame.iloc(k)`
 - ▶ `k`は0から始まることに注意

- ▶ columnは取り出せるが、和暦が使われている
 - ▶ 西暦への変換写像を準備

```
1 url = 'http://data.bodik.jp/dataset/'¥
2     '77e0cc66-c15d-4473-b3df-2664fe8e2e63/resource/'¥
3     '8dc71515-526a-4168-866c-05d2cc8dad7b/download/jinkou.xlsx'
4
5 data = pandas.read_excel(url,header=3)#Excelの内容をDataFrameへ
6 prePlot()
7
8 pList=[(2,'Total'),(3,'Cities'),(4,'Counties')]
9 plotSub(pList,data)
10 plt.legend(loc='best')#凡例
11
12 plt.savefig('output.png')#画像保存
13 plt.show()#画像表示
```

```
1 def prePlot():
2     plt.figure(figsize=(10,10))
3     plt.title('Population in Saga (from National Census)')#図タイトル
4     plt.xlabel('year')
5     plt.ylabel('population (thousand)')
```

```
1 def plotSub(pList,df):
2     for loc,label in pList:
3         ser=evalPop(df,loc)
4         ser.plot(label=label)
```

```
1 def evalPop(data,pos):
2     vList,yList=[],[]
3     for k in wareki.keys():
4         p = data[k][pos]/1000.
5         y = wareki[k]#和暦から西暦へ
6         vList.append(p)
7         yList.append(y)
8     return pandas.Series(vList,index = yList)
```


1 #エクセル中の和暦ラベルと西暦の対応付け

```
2 wareki=[ '大正9年':1920, '昭和5年':1930, '10年':1935, '15年':1940,  
3         '20年':1945, '25年':1950, '30年':1955, '35年':1960, '40年':1965,  
4         '45年':1970, '50年':1975, '55年':1980, '60年':1985, '平成2年':1990,  
5         '7年':1995, '12年':2000, '17年':2005, '22年':2010, '27年':2015]
```

