11. エクセルのデータを作図する

プログラミング・データサイエンス I

2022/6/30

1　今日の目的

今日の目的

- Excel ファイルのデータを作図する
- DataFrame や Series の作図メソッド
- 九州と佐賀の人口を例に

前回は、プログラムの中に、作図するデータがありました。今日は、エクセルにあるデータを読み込んで、作図しましょう。サンプルプログラムを取得してください。

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

2　Excel データの作図: Drawing with data in Excel

Excel データの作図: Drawing with data in Excel

- DataFrame から Series を取り出す
- Series のメソッドで作図する

2.1　九州各県の人口推移

以前に、エクセルファイルを扱うライブラリ Pandas を用いました。エクセルファイルを読みこむと DataFrame という表形式のデータになることを説明しました。この Pandas は、前回利用した matplotlib と連携した作図のための機能を持っています。つまり、matplotlib がデータ列として利用していたリストを変形することなく、Pandas のデータのままで作図することができます。

プログラムと一緒に、KyushuPopulation.xlsx という Excel ファイルがあります。

表 1　九州各県の人口推移

県	1990	1995	2000	2005	2010	2015
福岡	4,758	4,873	4,968	5,014	5,039	5,120
佐賀	882	885	884	874	859	847
長崎	1,573	1,554	1,532	1,502	1,450	1,413
熊本	1,848	1,862	1,870	1,858	1,834	1,818
大分	1,246	1,243	1,236	1,225	1,207	1,191
宮崎	1,182	1,187	1,187	1,173	1,153	1,136
鹿児島	1,805	1,794	1,787	1,763	1,722	1,691

表 1 のように、国勢調査に基づく九州各県の人口推移が 5 年毎に入っています。一行取り出すと、一つの県の人口の時系列になります。はじめに、これをプロットしましょう。

ソースコード 2.1　Series をプロットする

```
1  def drawTimeSequence(df :pandas.DataFrame): #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, color = colors[pref])
```

KyushuPopulation.ipynb を開けてください。データを描画している部分が、ソースコード 2.1 です。DataFrame である df を引数として受け取り、各行、つまり各県のデータを Series である ser として取り出します。最後の行で、各県のデータをプロットしています。

Series 形式で、各県のデータを取り出したため、年と組になったデータとなります。そのため、年を横軸として折れ線を引くことができます。

課題 1　Series 形式になったデータを print() を使って、その内容を確認しなさい。

課題 2　ソースコード 2.1 の 5 行目はコメントアウトしています。ここで呼び出しているのは、ソースコード 2.3 の部分で、九州全体の人口の総和を求め、df に追加する部分です。コメントアウトを外して実行し、動作を確かめなさい。

ソースコード 2.2　Series をプロットする

```
filename = 'KyushuPopulation.xlsx'
with pandas.ExcelFile(filename) as f:
    df = pandas.read_excel(f)
df.set_index('県', inplace = True) #左端のカラムにインデクスを設定
#totalPopulation(df)
#図の準備
plt.figure(figsize = (15, 15), facecolor = 'white')
plt.rcParams['font.size'] = 20
plt.title('九州の人口')
plt.xlabel('年')
plt.ylabel('人口 ($\\times1,000$)')
#各県のデータをプロット
drawTimeSequence(df)

plt.legend()
plt.show()
```

ソースコード 2.3　九州全体の人口を計算する

```
def totalPopulation(df: pandas.DataFrame):
    total = df.sum(axis = 'index')
    df.loc['九州全体'] = total
```

2.2　九州各県の人口割合

次は、九州各県の人口の割合を図示しましょう。このような場合には、円グラフを使うことが多いでしょう。円グラフを表示する関数も用意されています。KyushuPopulationPie.ipynb を開けてください。

ソースコード 2.4 を見てください。1995 年と 2015 年とで、九州全体の人口に対する各県の割合の変化を円グラフにするコードです。一行二列の図を ax に描くことにします。7 行目は、全体に対するタイトルを指定しています。9 行目からの for ループで、各年の図を作成しています。対象となる年を指定することで、DataFrame から列を Series として取り出し、作図しています (図 1)。13 行目の plot.pie() の最初の引数に、作図する ax を指定しています。実行して、確認してください。実行結果を見ると、自動で百分率を計算し、表示していることがわかります。

ソースコード 2.4　1995 年と 2015 年の人口比率

```
#図の準備
year = [1995, 2015]
size = 10
xsize = size * len(year)
fig, ax = plt.subplots(1, len(year), figsize=(xsize, size),
        facecolor = 'w')
plt.rcParams['font.size'] = 20
plt.suptitle('九州の人口')
#各県のデータをプロット
for i in range(len(year)):
    ser = df[year[i]]
    ser.name = ''  #Series の名前を消去：消さないと左に小さく表示されてしまう
    ser.plot.pie(ax = ax[i], startangle = 90, counterclock = False,
        autopct = '%1.2f%%', title = str(year[i]))

plt.savefig('KyushuPopulationPie.pdf')
plt.show()
```

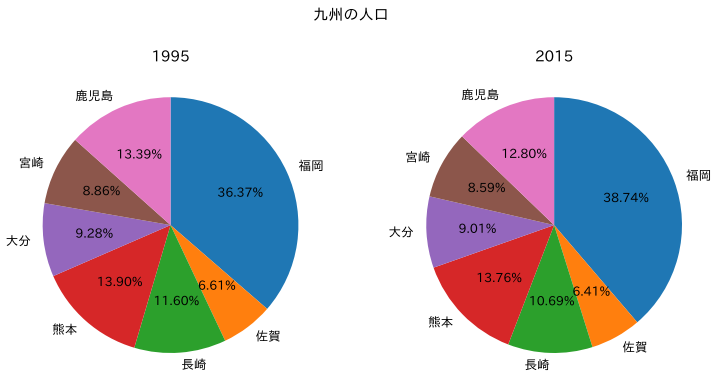

図 1　九州各県の人口

課題 3　1995 年、2005 年、2015 年の三つを作図するように、ソースコード 2.4 を変更し、動作を確かめなさい。

3　実データを扱う: 佐賀県の人口

実データを扱う: 佐賀県の人口

- 「ビックデータ&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/
- 人口の推移を図示

インターネット上には、様々なデータが公開されています。佐賀県が公開している人口推移のデータを使って作図をしましょう。配布したファイル中にある jinkou.xlsx を開けてください。このファイルは、以下の URL にあります。

http://data.bodik.jp/dataset/77e0cc66-c15d-4473-b3df-2664fe8e2e63/resource/8dc71515-526a-4168-866c-05d2cc8dad7b/download/jinkou.xlsx

以前に見たものと同じファイルです。列のラベルの西暦化、V 列を行のラベルに使う、空白行を除くなどの処理を、以前と同じように行っておきます。

ソースコード 3.1　指定した行の作図

```
1 plotList=['佐　賀　県', '市部','郡部']
2 for label in plotList:
3     ser = data.loc[label]/1000
4     ser.plot(label = label, linewidth = 3)
```

ソースコード 3.1 が作図の中核部分です。plotList に行のラベルを指定しています。各行について、そのデータを 1/1000 しているのが、2 行目です。このように、Series の各要素に対して、一括して処理を行うことができます。3 行目で、データを折れ線で作図します。実行結果を図 2 に示します。

課題 4　シートには、年代層別の人口がありました。年少 (0〜14)、生産年齢 (15〜64)、老年 (65 以上) の人口の推移を作図しなさい。データ中の行の名前を確認する必要があります。

これが今日の確認テストです。後で実施してください。

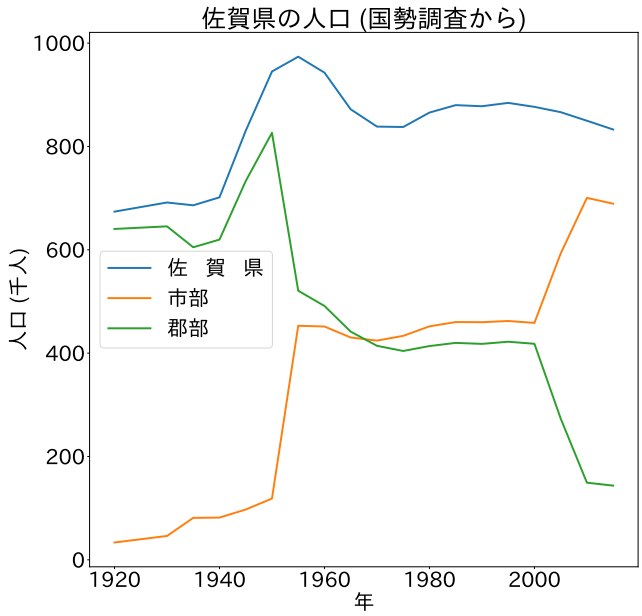

図 2　佐賀県の人口推移: 市部と郡部

4　積み上げグラフ:Stacked Bar Charts

各年齢層の占める割合の変化を示す方法として、前節の折れ線グラフのほかに、積み上げグラフがあります (図 3)。

`matplotlib` では、棒グラフを描く際に、その基点、つまり棒の縦方向の原点を指定するパラメタ `bottom` を持っています。これを使って作図しましょう。サンプルプログラムは `populationStackedBar.ipynb` です。

Excel を読み込んで成形した `DataFrame` 内には、年齢層を表すインデクスがあります。これを使うと、`Series` として取り出すことができるのは、前節で行いました。全部の列を作図すると、数が多すぎますから、一部を取り出すことにします。`Series.filter()` というメソッドは、引数に `Series` 中のインデクスのリストを渡すと、対応する部分だけからなる `Series` を返します。こうしてできる部分的データを作図すればよいでしょう。

次は、棒グラフの基点です。作図する `Series` と同じインデクスを持つ `Series` となる `bottom` を用意しましょう。データを作図する度に、`bottom` にデータを加算します。

ソースコード 4.1 を見てください。引数 `yearList` には、作図対象の年が入っていま

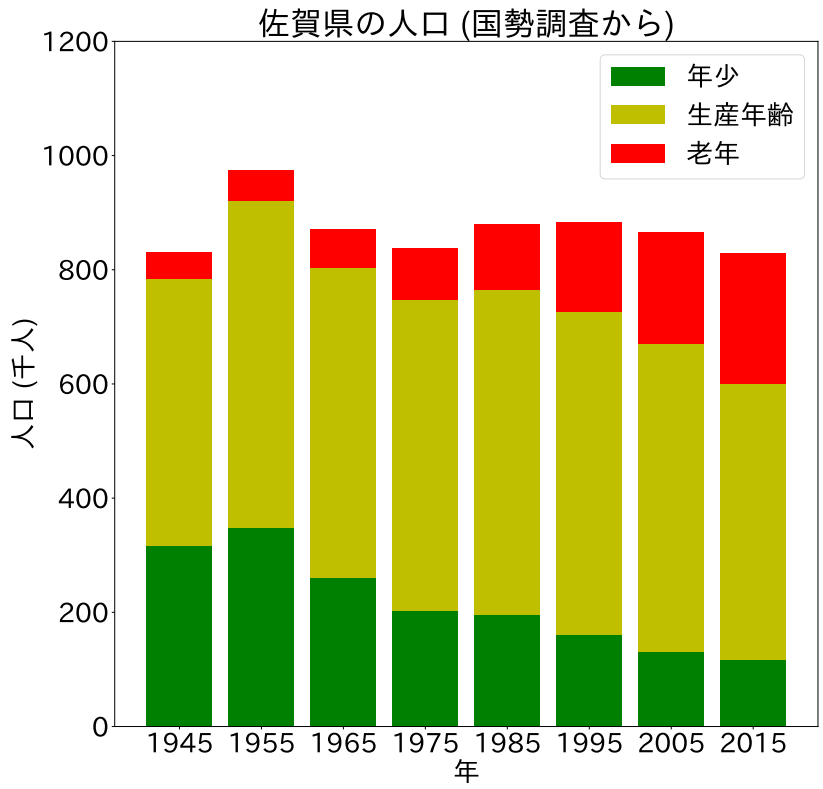

図 3　佐賀県の人口構成推移

す。6 行目から 10 行目で、各年齢層に対する行のデータから、対象年を抜き出し、さらに 1/1000 に行のラベルを指定しています。

12 行目は、`yearList` を使って、すべての値が 0 である `bottom` という `Series` を作っています。0 の数は `yearList` の長さです。またそのインデクスは、`yearList` になっていることに注意してください。

16 行目は、加工した `Series` の棒グラフとしての作図です。`bottom` が基点となっています。作図した `Series` を 18 行目で `bottom` に加算していきます。

ソースコード 4.1　積み上げグラフ

```
1  def plotSub(yearList:list[int], df:pandas.DataFrame):
2      """
3      yearList で指定された年の棒グラフを描画
4      """
5      #必要なデータに成形する：対象となる年のデータを取得し、1/1000 にする
6      population = [
7          df.loc['0〜14 歳'].filter(yearList)/1000,
8          df.loc['15〜64 歳'].filter(yearList)/1000,
9          df.loc['65 歳以上'].filter(yearList)/1000
10     ]
11     #積み上げの基点となる Series
12     bottom = pandas.Series([0]*len(yearList), index = yearList)
13     color = ['g', 'y', 'r']
14     label = ['年少', '生産年齢', '老年']
15     for k in range(len(population)):
16         population[k].plot.bar(bottom = bottom,
17             tick_label = yearList, color = color[k], label = label[k])
18         bottom += population[k]
```