■1.表データの読み書き

|  |
| --- |
| Pandasモジュール |
| Pandasとは、表データを読み込んで追加、削除、抽出、集計、書き出しなど行える外部モジュール |

◆表データ

行と列で出来ているデータのことを指します

**1列**

**1行**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 名前 | 国語 | 数学 | 英語 |
| 0 | のび太 | 3 | 1 | 1 |
| 1 | しずか | 80 | 90 | 88 |
| 2 | 出木杉 | 100 | 100 | 100 |

**インデックス**

**ヘッダー(項目名)**

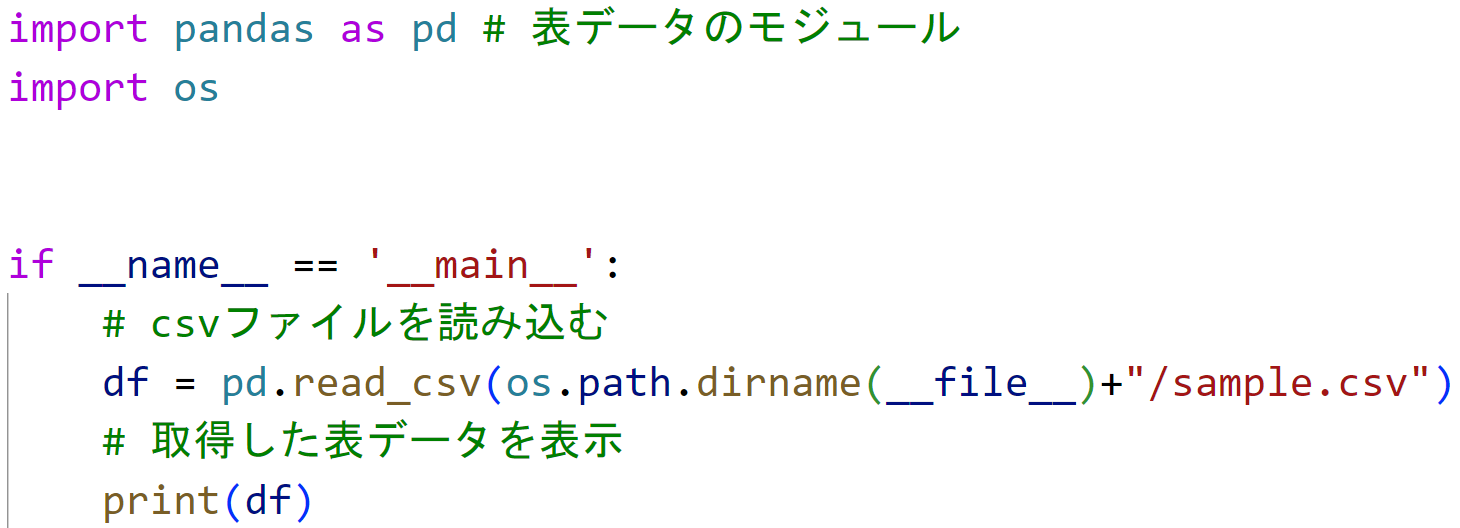
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 名前 | 国語 | 数学 | 英語 |
| 0 | のび太 | 3 | 1 | 1 |
| 1 | しずか | 80 | 90 | 88 |
| 2 | 出木杉 | 100 | 100 | 100 |

◆1.1 csvファイルを読み込む

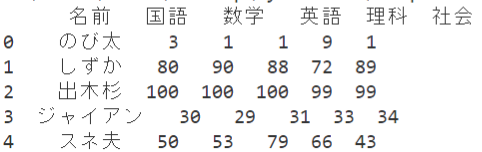
pandasを活用してsample.csvのデータを取り込んでみましょう！

----------------------------------------------------------------------------------

ファイル名：pandasSample.py



これで実行してみましょう!

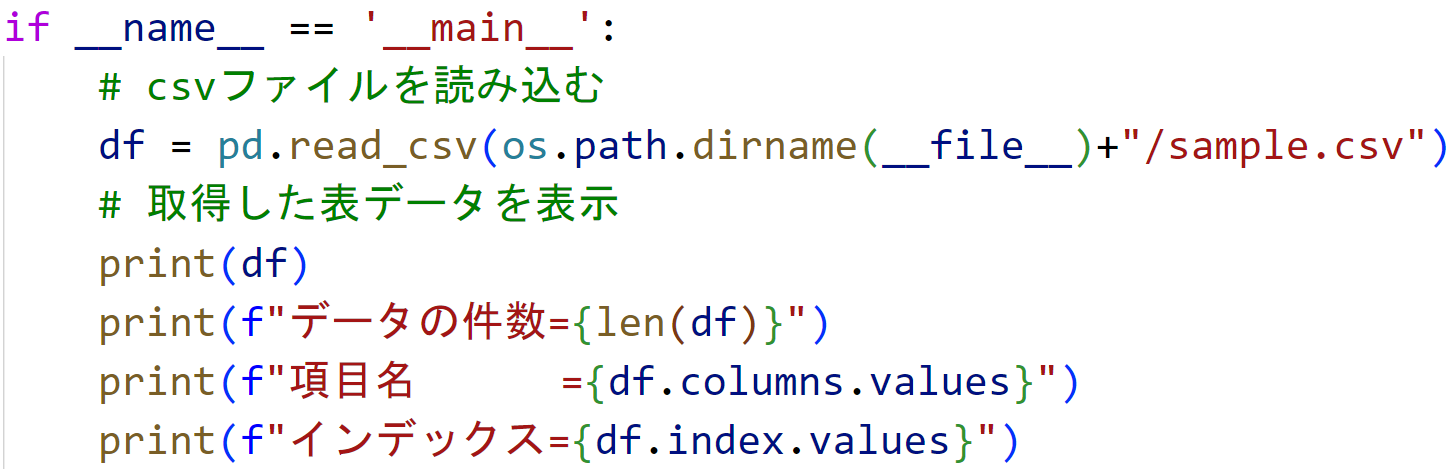
右図のように取得できましたね

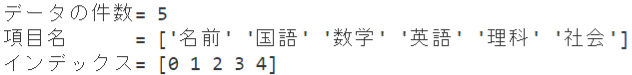
◆表データの解析

次は表データの内容を解析してみましょう

|  |  |  |
| --- | --- | --- |
| データ件数の取得 | 項目名の取得 | インデックスの取得 |
| len(df) | df.columns.values | df.index.values |

追記して動作を確認します



実行してみましょう！

データの情報が表示出来ていますね

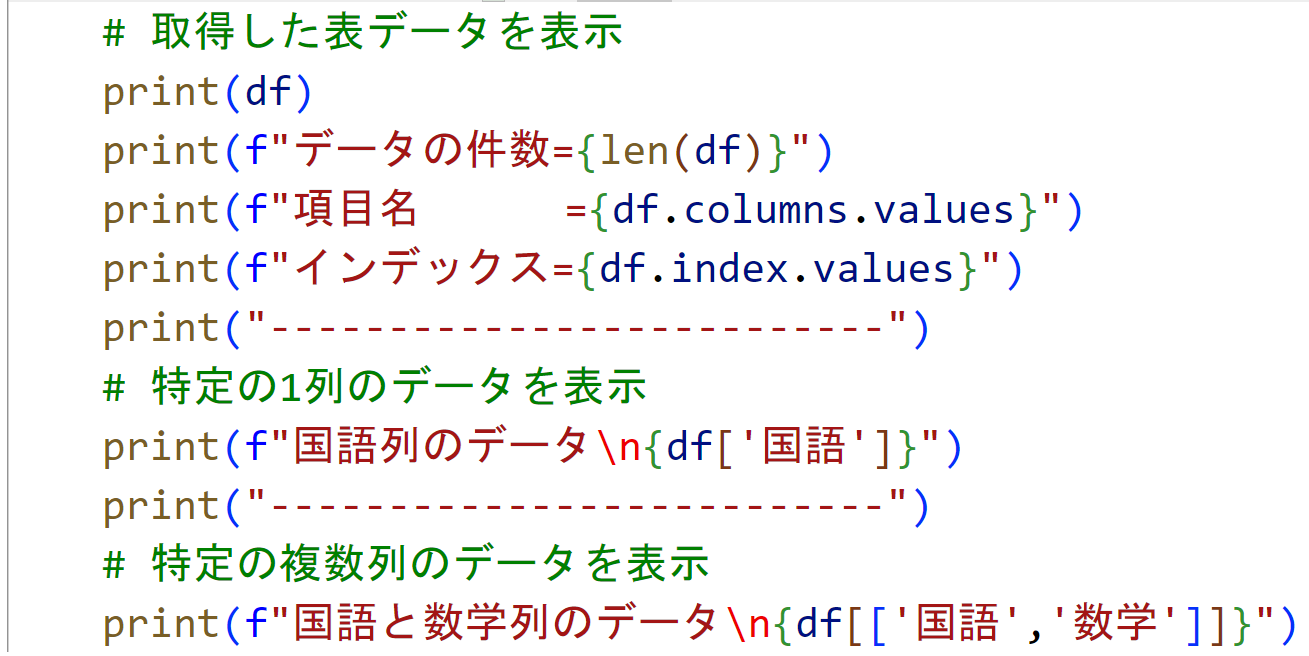
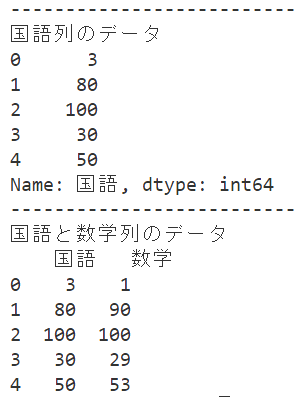
◆1.2 列データ、行データの表示

次は特定の列、特定の行を表示させてみましょう

|  |  |
| --- | --- |
| 1列のデータを取得 | 複数列のデータを取得 |
| df[“要素”] | df [ [ “要素1”,”要素2”] ] |

dfには多次元配列で格納されていますのでdf[“要素”]で指定することが可能です。

複数を指定する場合はdf [ [ 要素1,要素2] ]と指定することで可能です



これで実行してみましょう

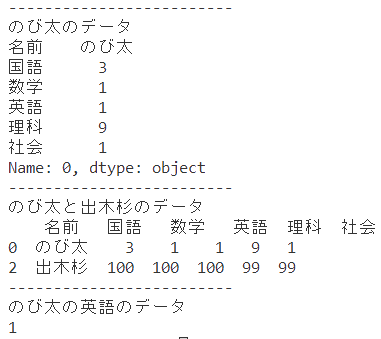
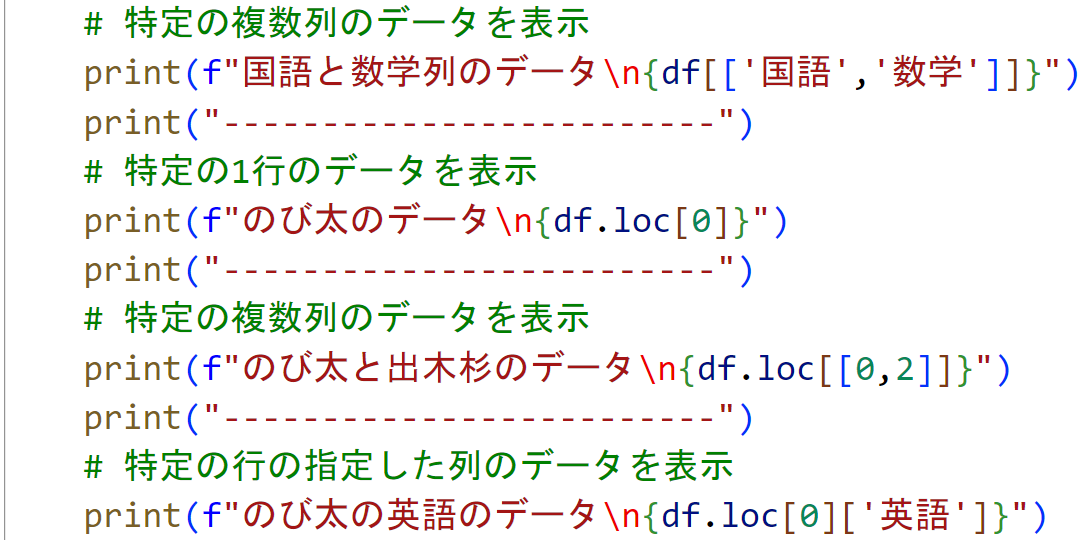
“ ”(ダブルコーテーション)内なので

重複しないよう‘ ’(ｼﾝｸﾞﾙｺｰﾃｰｼｮﾝ)で記述

◆1.3 行の表示

続いて特定の行を表示させてみましょう

|  |  |  |
| --- | --- | --- |
| 1行のデータを取得 | 複数行のデータを取得 | 1つの要素データを取得 |
| df.loc[行番号] | df.loc[[行番号1、行番号2]] | df.loc[行番号1][“列名”] |

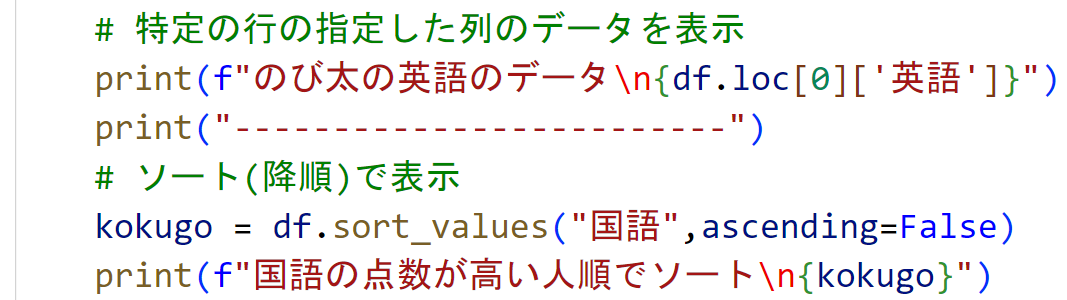
実行して動作を確認しましょう。

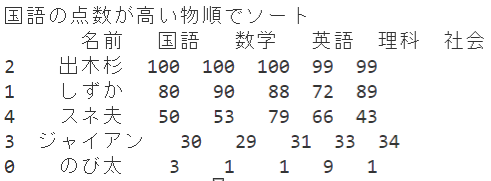
◆1.4 データの並び替え

項目を指定してソート(並び替え)をすることが出来ます。

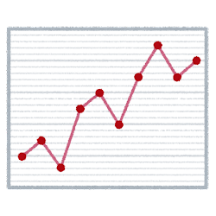
|  |  |
| --- | --- |
| 並び替え(昇順) | 並び替え(降順) |
| df.sort\_values(“列名”) | df.sort\_values(“列名”, ascending=False) |

実際に国語の点数を降順にするプログラムをやってみましょう！





実行して動作を確認しましょう！

■2.グラフで表示

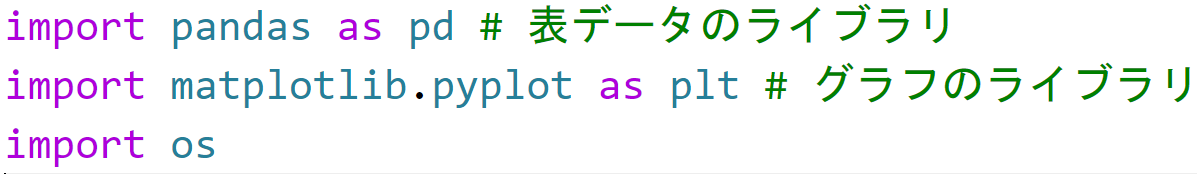
読み込んだ表データを活用し、グラフで表示してみましょう

グラフを使うのでmatplotlibモジュールを活用します

更に処理の中でplt.rc('font', family='BIZ UDGothic')も使用すると日本語にも対応できます。

-------------------------------------------------------------------------

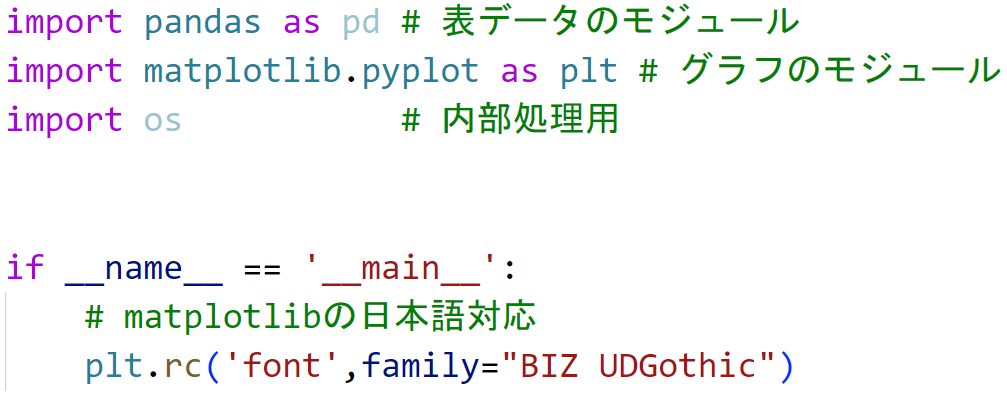
ファイル名：graphSample.py



◆グラフの日本語対応

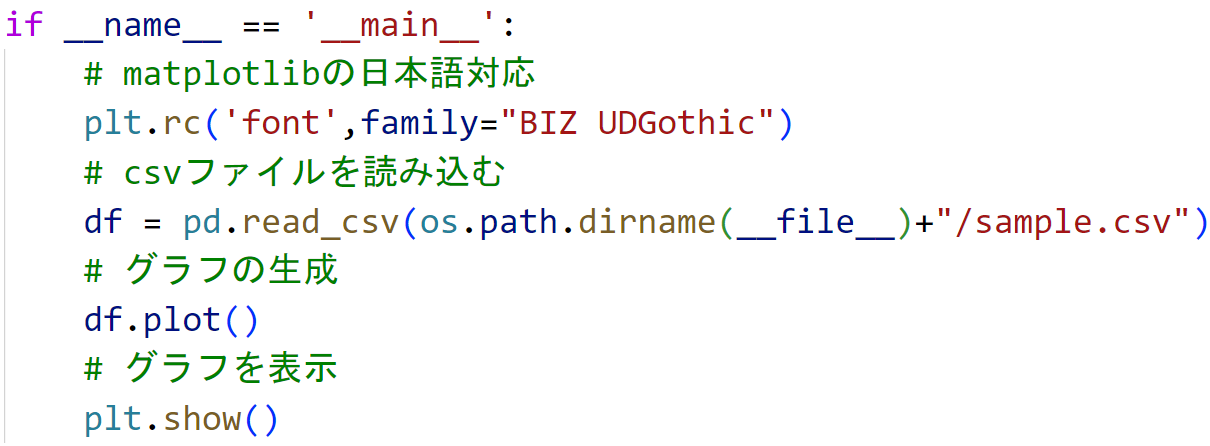
デフォルトのままだとグラフ表示時に日本語が文字化けしてしまうので

日本語に対応させましょう

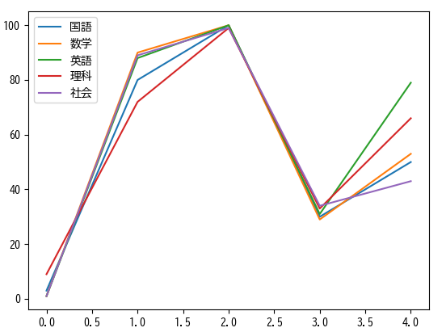


◆グラフの生成と表示

あとはデータを読み込み、グラフで表示させましょう



これで実行して動作を確認します。



右の図のようになりましたか？

ただし、このままでは

①グラフが折れ線

②下の目盛りの数字の意味

③左上の枠は一体なに？

という部分が気になりますね

①：グラフにも目的に合ったものを選ぶ必要があります。

|  |  |  |
| --- | --- | --- |
| 目的 | グラフの種類 | 命令 |
| 変化を見たい時 | 折れ線グラフ |  |
| 値の大小を比較したい時 | 棒グラフ | DataFrame.plot.bar() |
| 変化の要因を知りたい時 | 積み上げ棒グラフ | DataFrame.plot.barh() |
| 全体の構成の割合を見たい時 | 円グラフ | DataFrame.plot.bar(stacked=True) |
| データの散らばりを見たい時 | 箱ひげグラフ | DataFrame.plot.box() |
| 変化の大きさを強調してみたい時 | 面グラフ | DataFrame.plot.area() |

②：下の目盛りの数字の意味

現状左の数字は「点数」、下の目盛りは「インデックスの数字」となっており

これだと誰の点数か判断しにくいですね。

インデックスではなく、名前に変更して表示されるようが良さそうです。

例：plt.legend(loc=”lower right”)

③：左上の枠は一体なに？

これは凡例(はんれい)といって、それぞれの色が何を指しているか表示しています。

デフォルトで表示位置は自動で決まるのですが plt.legend(loc=”lower right”) と書くことで右下に出来ます

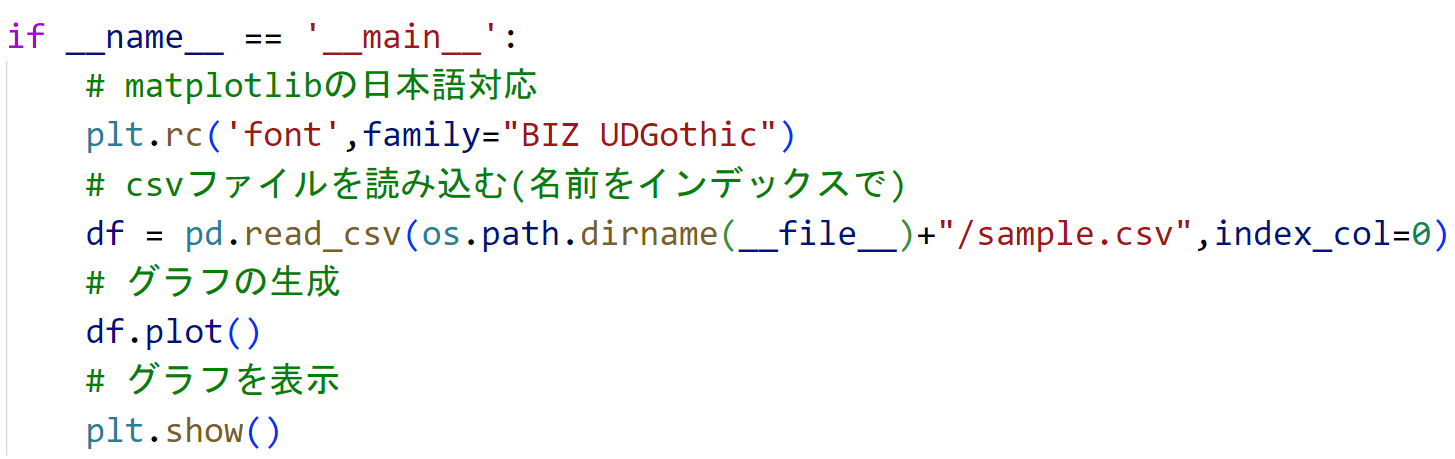
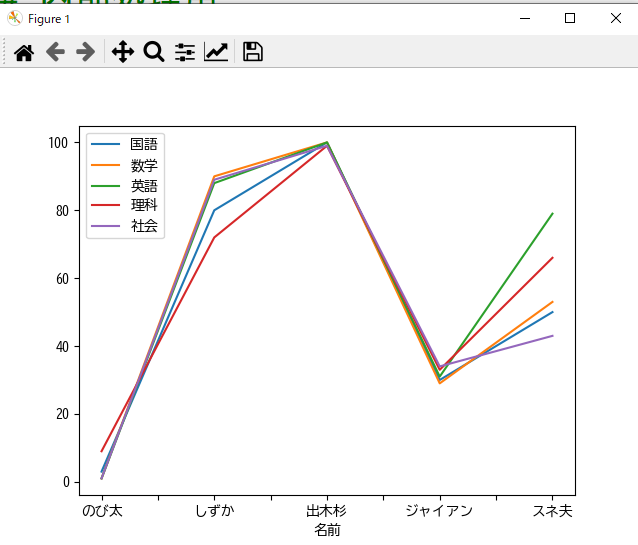
では実際に使ってみて動作を確認してみましょう！

----------------------------------------------------------------

◆インデックス名の指定

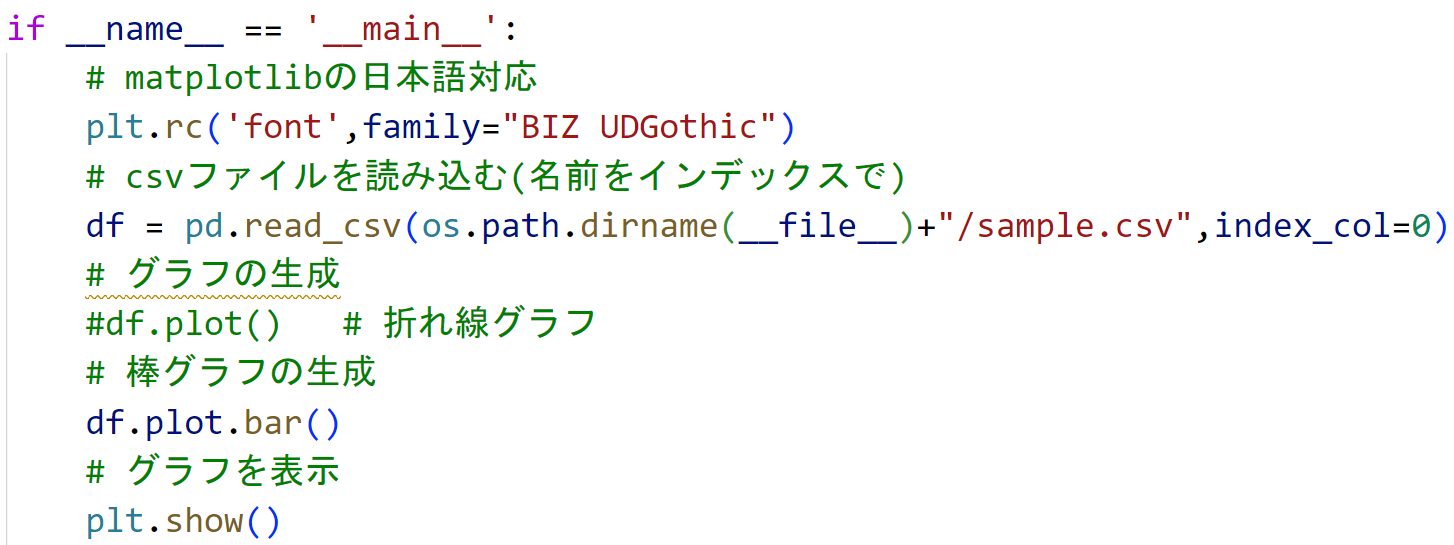
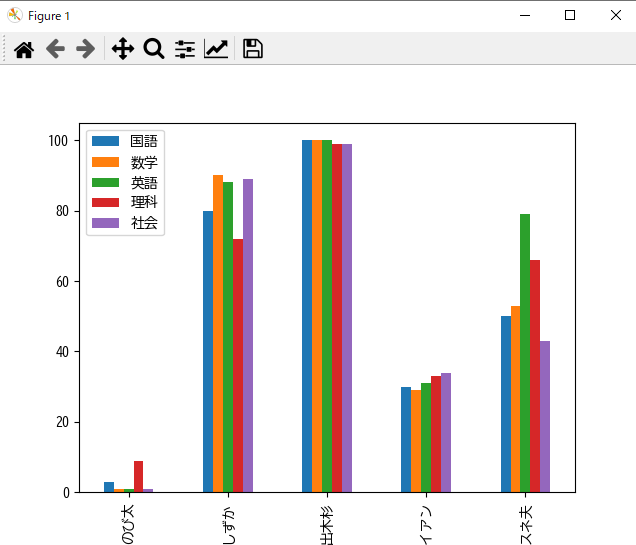
CSVの読み込みの際[ index\_col=0 ]と指定することで

一番左の列(今回は名前の列)をインデックスとして読み込みます



◆棒グラフの設定

グラフを生成する際に、どの形式かを指定することで別のグラフに変更できます。

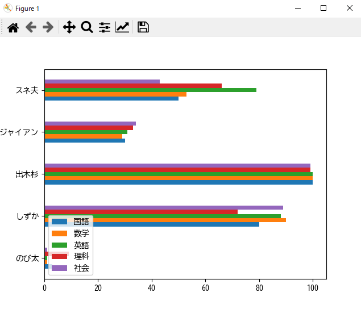


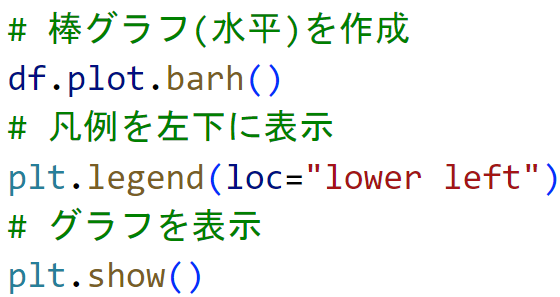
この状態で実行してみましょう！

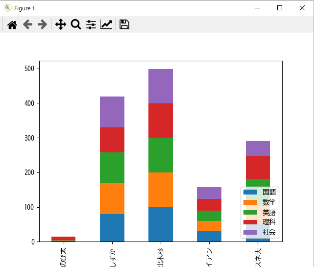
先程よりはわかりやすくなりましたね

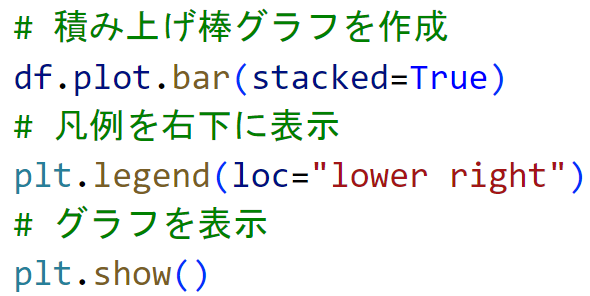
他のグラフも紹介だけしておきます。

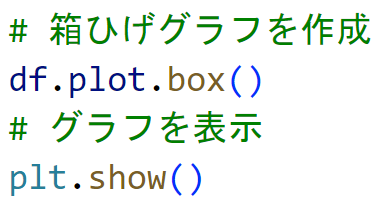
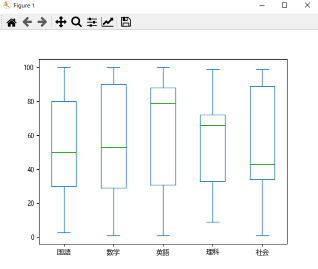
(興味があれば動かしてみて下さい)

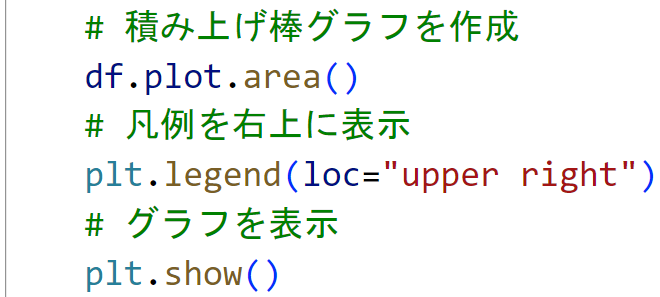
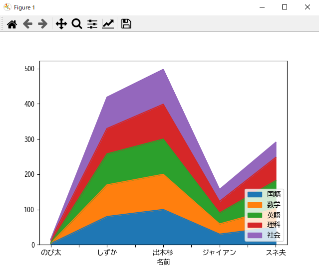












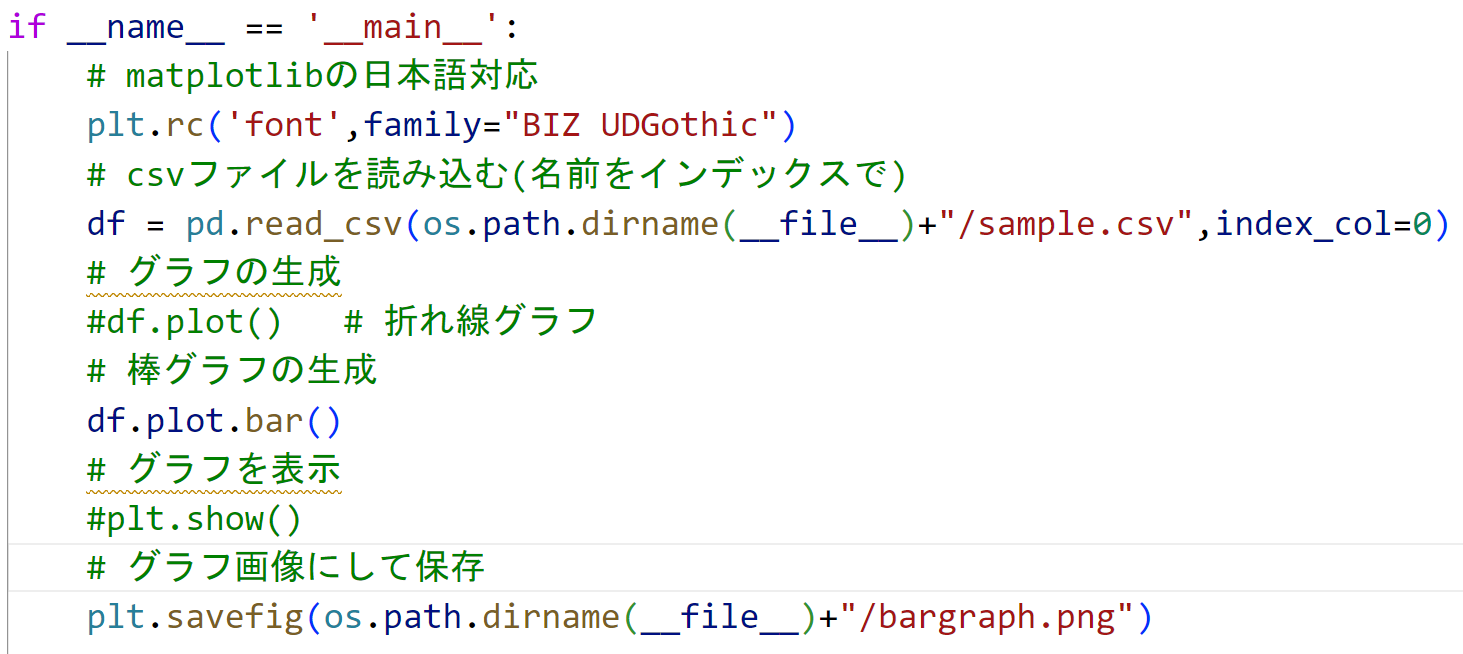
◆グラフの画像ファイル出力

棒グラフは表示だけでなく、画像ファイルとして保存することも出来ます

|  |  |
| --- | --- |
| グラフの表示 | グラフ画像の保存 |
| plt.show() | plt.savefig() |

書式：plt.savefig(“ファイル名.png”)

では、先ほどの棒グラフの表示のＰＧをグラフ画像の保存に変更してみましょう



実行後に画像ファイルが生成されていますね

■２．オープンデータを分析

オープンデータとは、企業や公的機関が公開しているデータのこと

オープンデータを活用し、分析や解析を行うことが出来ます。

|  |  |
| --- | --- |
| サイト名 | ＵＲＬ |
| e-Stat:政府統計の相互窓口 | https://www.e-stat.go.jp/ |
| キッズすたっと~探そう統計データ~ | https://dashboard.e-stat.go.jp/kids/ |
| データシティ鯖江 | https://data.city.sabae.lg.jp/ |
| 気象庁：過去の気象データ・ダウンロード | https://www.data.jma.go.jp/gmd/risk/obsdl/ |
| 郵便局：郵便番号データ | https://www.post.japanpost.jp/zipcode/download.html |
| DATA GO JP | https://www.data.go.jp/ |
| Link:Data | http://ja.linkdata.org/ |

今回は e-Stat のオープンデータを活用してみましょう

e-Stat:政府統計の相互窓口　【<https://www.e-stat.go.jp/>】

「人口推計　都道府県別」を入力し検索



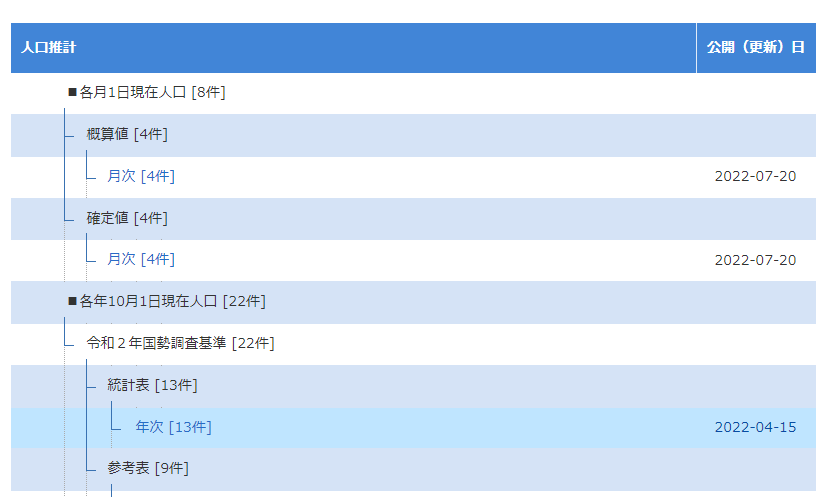
人口推計をクリック



データベースをクリック



各年10月1日現在人口の年次をクリック



「005 都道府県、男女別人口―総人口、日本人人口」のＤＢをクリック



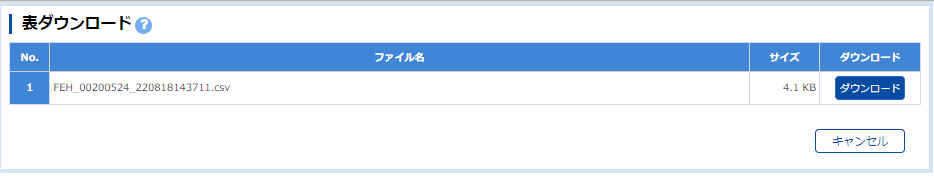
ダウンロードをクリック



以下と同じように選択しダウンロード



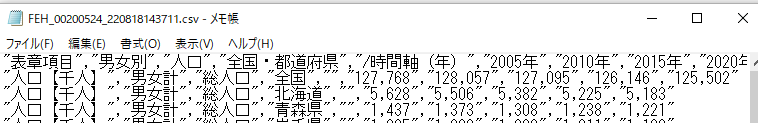
ダウンロードをクリック(※ファイル名の番号は毎回変わります)



ダウンロード出来ていますね



テキストファイルでファイルを開きで中身を確認してみましょう



ファイル名をわかりやすい物に変更しておきましょう。



1行目が「ヘッダー」になっているおり

各行のデータを指すには4列目の「全国・都道府県」が使えそうです。

Pandasを用いてデータの読み込み及び表示をさせましょう。

※先にファイルを扱いやすい場所(pythonの授業用フォルダ)に移しておきましょう

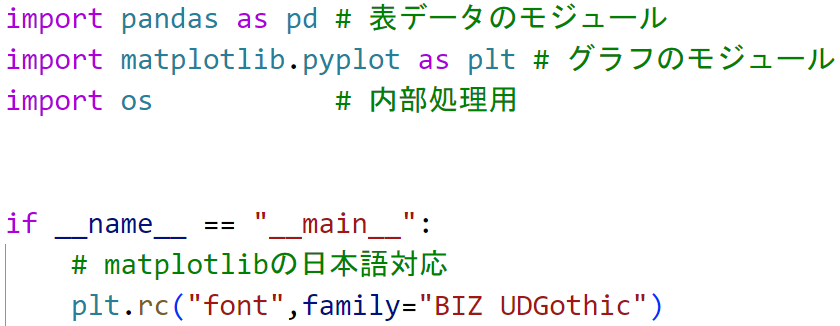
◆オープンデータファイルの読み込み

では、先ほどダウンロードした全国の人口のデータ(.csvファイル)を活用してみましょう！

--------------------------------------------------------------------------

ファイル名：dataAnalysis.py

今回は人口を棒グラフで表示させたいと思います

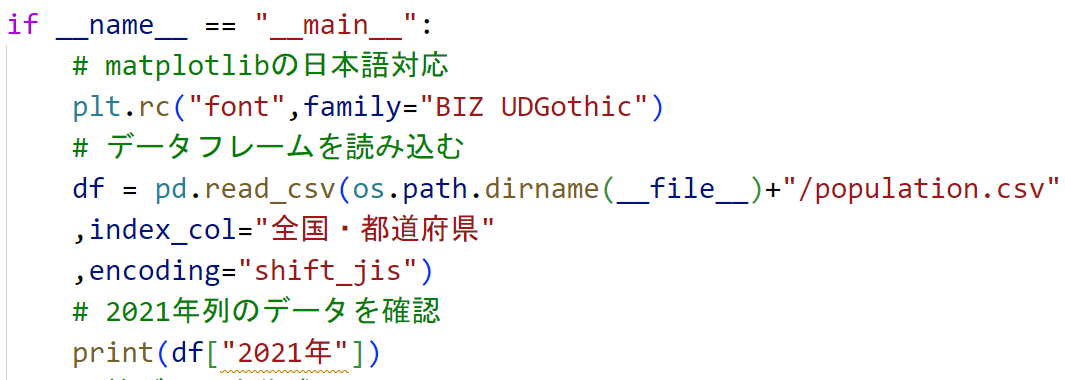
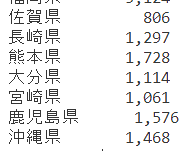


先程、名前を変更したファイルを読み込みましょう。

その際、インデックスを”全国・都道府県”列に指定しましょう

今回使用するcsvはshift\_jisで作成されているので、

エンコーディングもshift\_jisとなるよう合わせておきます。

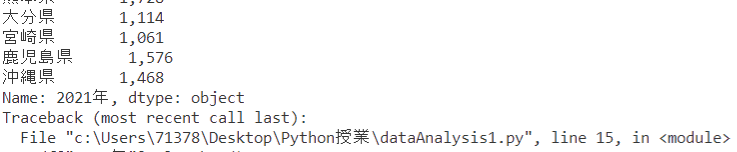


確認の為、実行し人口のデータを取得出来ている事を確認します。

◆人口を棒グラフで表示

そのうえで、2021年列の人口データを取得し、棒グラフを表示させましょう！





これで実行してみると...

**途中でエラー**になっていますね



これは「プロットする数値データがない」となっています

原因は、データの区切りは,(カンマ)という書式ですが

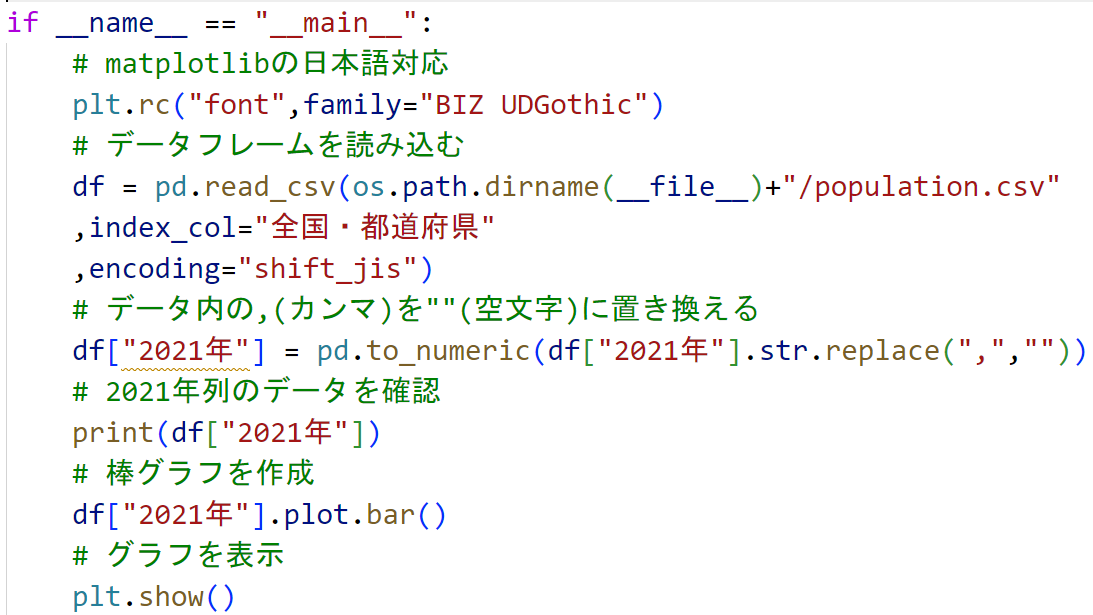
データ作成者によって、その書き方はバラバラです

今回は人口の数値が「127,768」と桁に,(カンマ)が使用されておりエラーになっています.

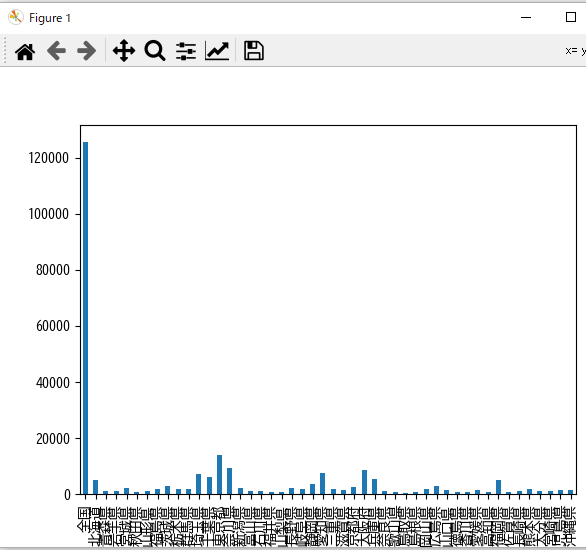
◆取得データの置き換え

数値内の,(カンマ)を“”(空文字)に置換するプログラムも追記しましょう

|  |
| --- |
| 文字列.replace(“検索文字”,”置換後文字”) |
| str.replace(“,” , ””) |



編集した上で再度実行してみましょう！



表示はされましたが、見にくいですね

①全国の数値も含まれている

②グラフの画面サイズが小さい

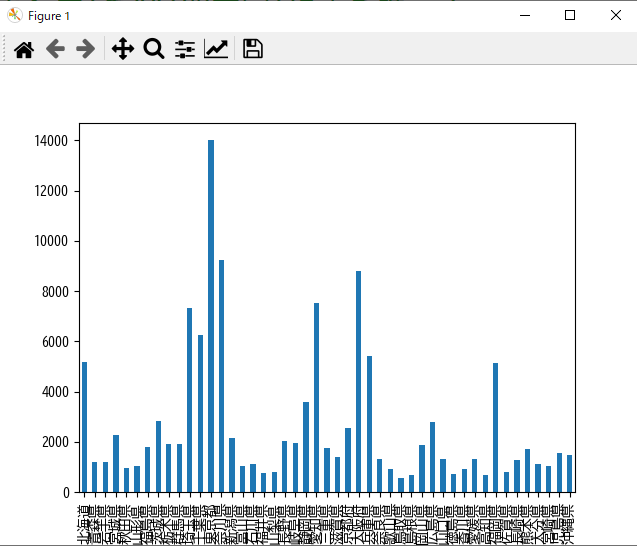
上記が原因かと思われます。

ＰＧを少し工夫して見やすくしていきましょう

◆データの削除

|  |  |
| --- | --- |
| 行のデータ削除 | 列のデータ削除 |
| df.drop(“行要素”,axis=0) | df.drop(“列要素”,axis=1) |

全国の行を削除するように.dropメソッドを活用してみましょう



実行してみましょう！

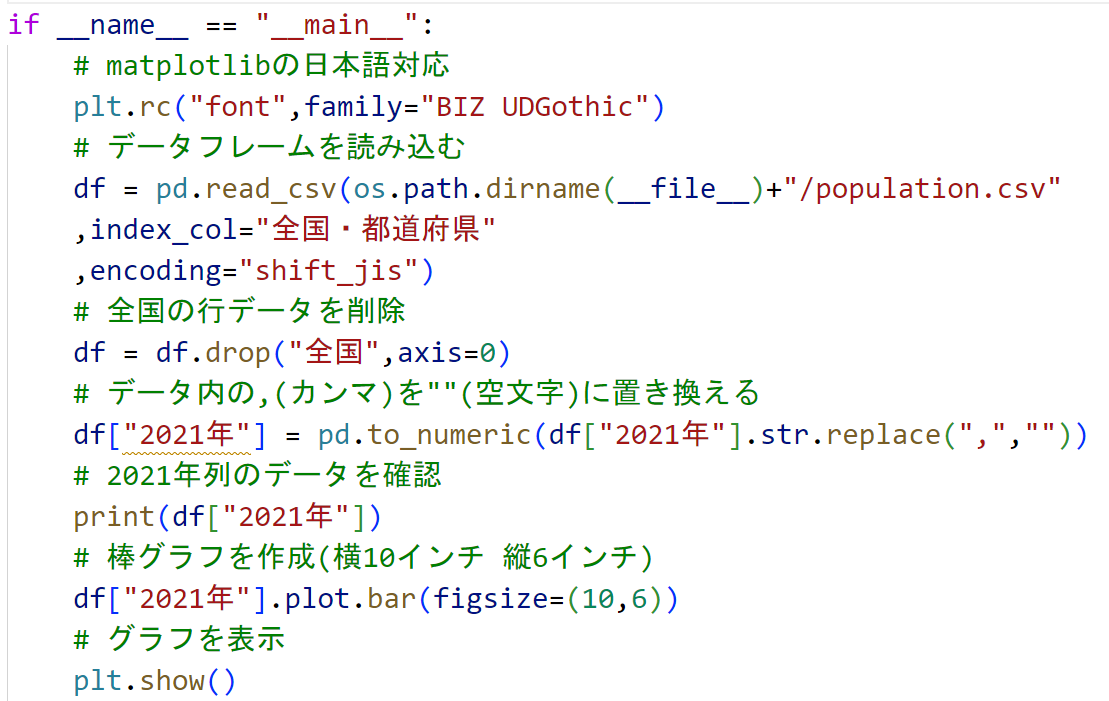
全国のデータが消えて比較しやすくなりました！

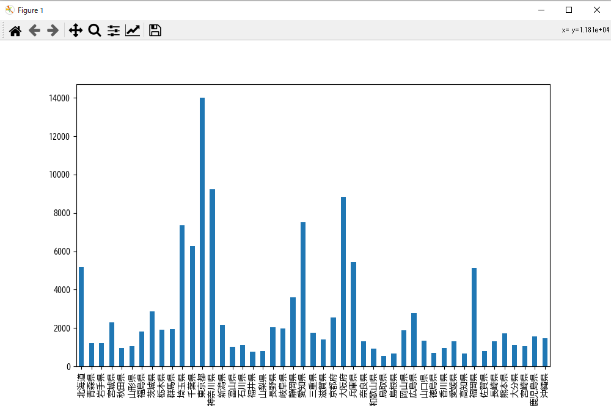
ただ、横が詰まっていてまだ見にくいですね。

◆グラフのサイズを変更

|  |
| --- |
| グラフの表示サイズ設定(インチ) |
| .bar(figsize=(横幅, 縦幅)) |

横幅を10インチ　縦幅を6インチに設定してみましょう！





この状態で実行してみましょう

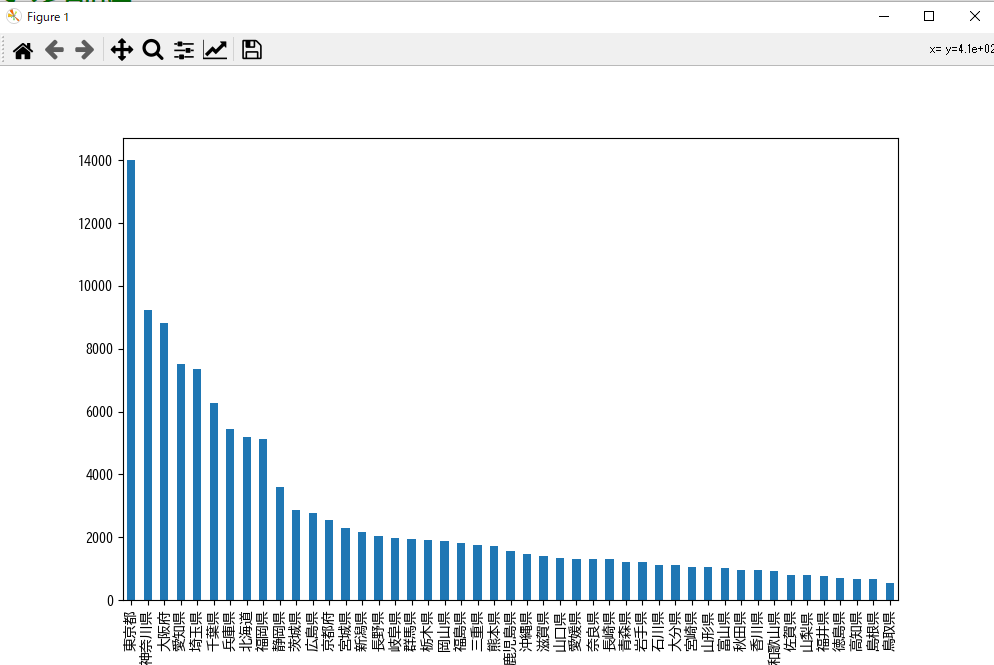
先程より横に余裕が出来て見やすくなりました！

◆データを並び替え

降順に並び替えるＰＧを追記し

より見やすいデータにしてPython入門編を終わりましょう





■最後に、

初回でもお伝えしましたが

**※Pythonは出来ることがとても多いです。**

**そして外部ライブラリは学習コストが高い傾向にあります。**

**その為、基本的に「実現したい目的を明確」にしてから、外部ライブラリを学びましょう。**

**やみくもに勉強しても、「掛けた時間を無駄にする」可能性が高いです。**

この選択授業の時間を使って

一番躓くであろう環境構築と、基礎知識のinputは出来たかと思います。

ここから先は、やりたい事や作りたいものをベースに調べたりテキストを読んだりして進んでみて下さい。

1号館1階にあるIT情報雑誌もPythonの回が多いです。

今の貴方なら書いてある内容が少しわかるかもしれません。

少しでも開発のきっかけや

何か得るものがあったなら幸いです。

4日間、お疲れ様でした。