第9回データの集計と可視化2

pandas、NumPyを用いた回帰分析

目次

事例：簡単な回帰分析

- pandas、NumPyモジュールを使った、簡単な回帰分析ついて試します。
  - 回帰分析とは、ある2つの変数にどのような関係があるのかを推定する手法です。
  - 例えば、あるクラスの生徒について、身長tと体重wの一覧表があったとします。身長tが高い人ほど体重wが大きいといった関係がある場合に、tからw（またはその逆）を推定する式を求めるような分析が回帰分析です。
- 準備
  - Spyderを起動し、[新規]-[新規ファイル]を作成します。
  - 今回のデータは、python.xlsxの「宿泊客数」シートになります。下記コードで取り込み表示し内容を確かめます。

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["font.family"] = "IPAexGothic"
plt.rcParams['font.size'] = 15

file = pd.ExcelFile("~/Desktop/python.xlsx")
data = file.parse("宿泊客数")
print(data)
```

	都道府県	訪日外国人	日本人
0	東京都	18059960	39454990
1	大阪府	10008830	21001640
2	北海道	6554220	27000280
3	京都府	4602810	13046690

このデータは、年間の都道府県別の宿泊を伴う観光客の数です。

NumPyについて

- NumPyはPythonを用いた数値計算を効率的に記述し、高速に実行するためのモジュールです。
  - C言語で記述されているためベクトルや行列をメモリの効率性を確保しながら、高速に処理することができます。
  - 数値計算をより直感的に記述できます。PandasもNumPyを用いて構築されています。
- 例：$0 \leq x \leq 10$ において　$y = 2x^2 + x + 5$　のグラフを描く例
  - NumPyを使わない例

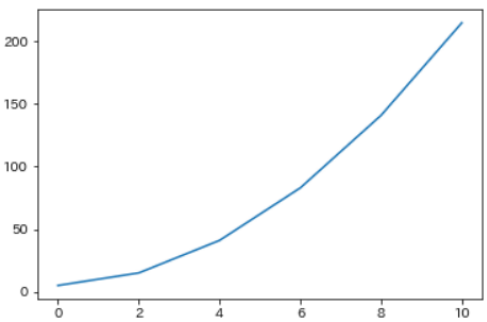
```
x = [0, 2, 4, 6, 8, 10]
y = [2 * i**2 + i + 5 for i in x]
plt.plot(x, y)
```

  - NumPyを使った例

pythonの配列をNumPyのベクトルに変換

```
x = np.array([0, 2, 4, 6, 8, 10])
y = 2 * x**2 + x + 5
plt.plot(x, y)
```

より直感的な記述が可能

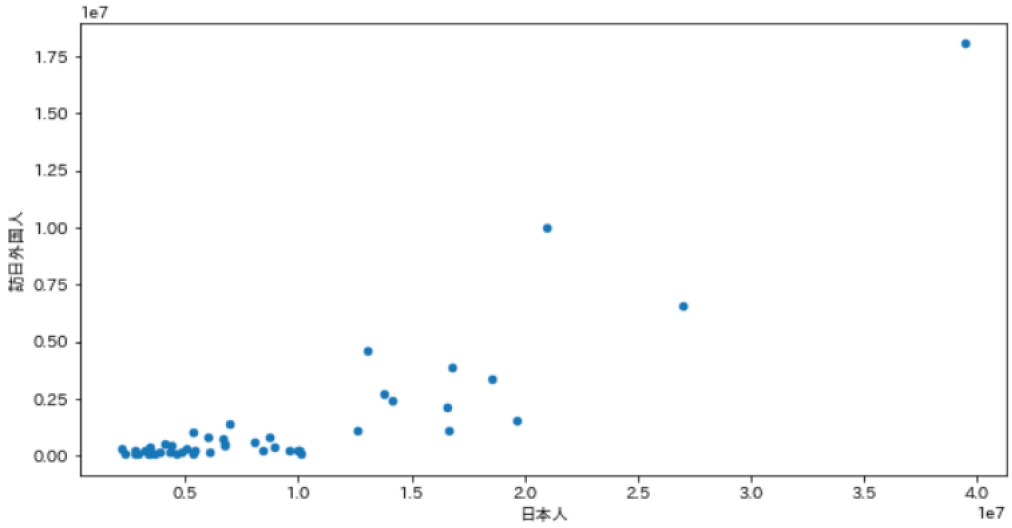


2つの量の関係をグラフ化する

- xとyの2つの軸について、データの散らばり具合を描くのが散布図です。散布図を描くことによって、2つの量の関係性を図示することができます。

```
ax = data.plot.scatter(x="日本人", y="訪日外国人", figsize=(10, 5))
```

plot.scatterメソッドで散布図を描く



散布図のサンプルにラベルを描画

- グラフのannotateメソッドで座標を指定してラベルを描画できます。

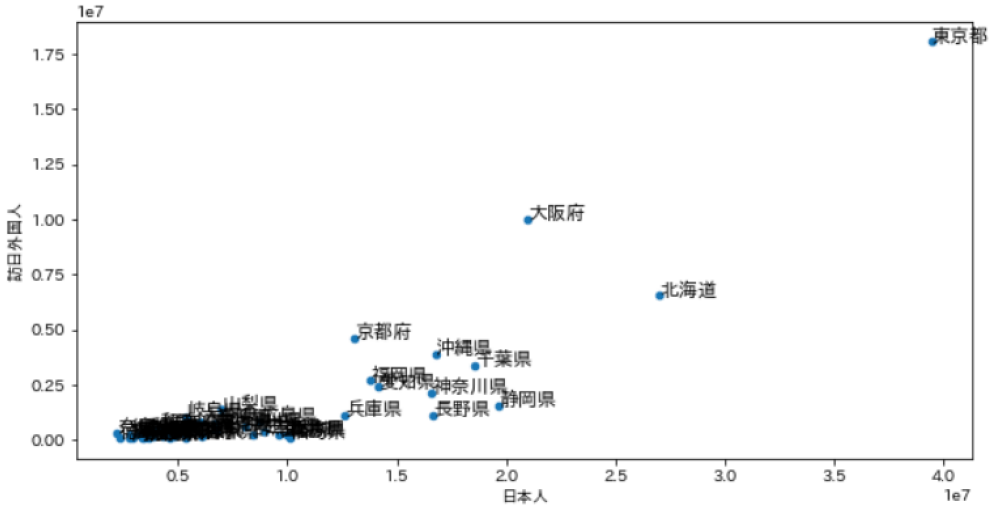
```
ax = data.plot.scatter(x="日本人", y="訪日外国人", figsize=(10, 5))
for k, v in data.iterrows():
    ax.annotate(v[0], xy=(v[2], v[1]), size=12)
```

dataの行を1つずつ取り出して、インデックスをkに、各列をvに入れる

都道府県名

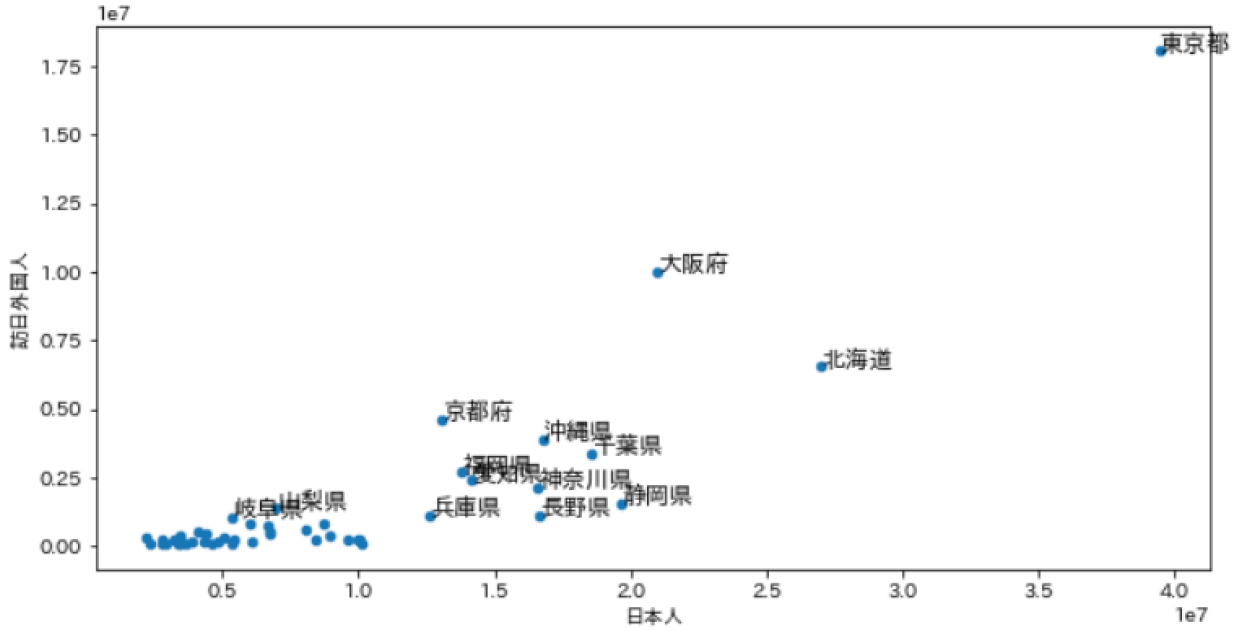
座標をタプルで指定

フォントサイズ



課題9

- 訪日外国人の少ない都道府県は、ラベルが重なってわかりづらいので、ラベルを表示しないようにしてください。



回帰直線を引く(1)

- 2組のデータの中心的な分布傾向を表す回帰直線を求めてグラフに描いてみましょう。
  - NumPyには、回帰曲線の係数を求めるpolyfitメソッドが用意されています。求める曲線の次数を1とすることで、1次方程式すなわち直線の傾きと切片を求めることができます。
  - 求めた回帰直線の傾きと切片から、poly1dメソッドで回帰直線の方程式を導くことができます。

```
z = np.polyfit(data.日本人, data.訪日外国人, 1)
print(z)
p = np.poly1d(z)
print(p)
```

1次方程式で回帰

```
[ 3.65448333e-01 -1.81342246e+06]
```

傾き　　切片

```
0.3654 x - 1.813e+06
```

poly1dにより、[a, b]という配列から、ax + bという方程式が生成できる

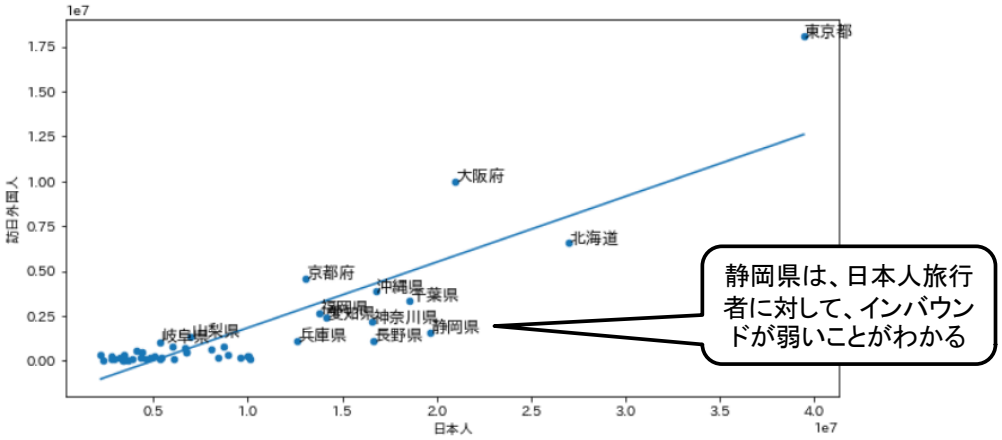
回帰直線を引く(2)

- np.linspaceメソッド：指定個の等間隔な数列を作ることができる
  - 例：`print(np.linspace(1, 10, 10))` ➡ `[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]`
- 回帰直線を引くには？
  - np.linspaceメソッドで、日本人の最小人数と最大人数までのN個の数列を作っておく
  - np.poly1dで求めた回帰直線の方程式に数列を当てはめて訪日外国人の推定値を求め、グラフを描く

```
xp = np.linspace(data.日本人.min(), data.日本人.max(), 100)
plt.plot(xp, p(xp))
plt.show()
```

回帰直線の方程式に数列xpを当てはめ

N=100



静岡県は、日本人旅行者に対して、インバウンドが弱いことがわかる

回帰曲線を描く

- np.polyfitメソッドの次数を2とすれば、2次方程式の回帰曲線が求められる

```
z = np.polyfit(data.日本人, data.訪日外国人, 2)
print(z)
p = np.poly1d(z)
print(p)
xp = np.linspace(data.日本人.min(), data.日本人.max(), 100)
plt.plot(xp, p(xp))
plt.show()
```