第9回データの集計とグラフ作成2
第10回データの集計とグラフ作成3

目次

事例：簡単な回帰分析

- pandas、NumPy、SciPy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
  - Jupyterで新たなNotebookを[New]-[Python 3]として新規作成します。
  - 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 scipy.stats as st
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams["font.family"] = "IPAexGothic"

file = pd.ExcelFile("Downloads¥python.xlsx")
data = file.parse("宿泊客数")
print(data)
```

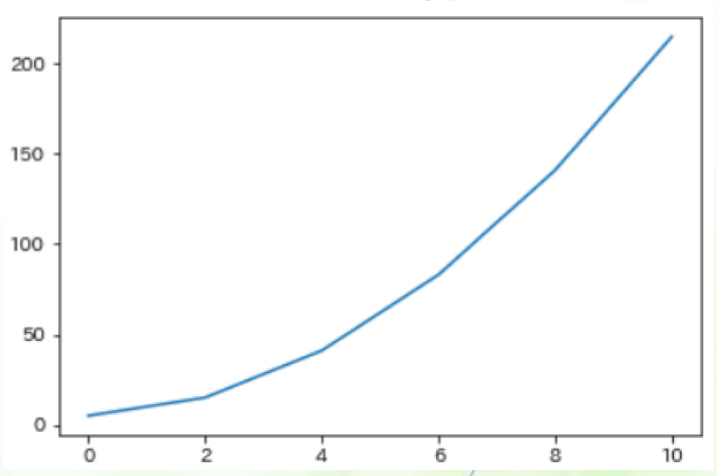
	都道府県	訪日外国人	日本人
0	東京都	18059960	39454990
1	大阪府	10008830	21001640
2	北海道	6554220	27000280
3	京都府	4602810	13046690

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。
  - 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。
- 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)
```

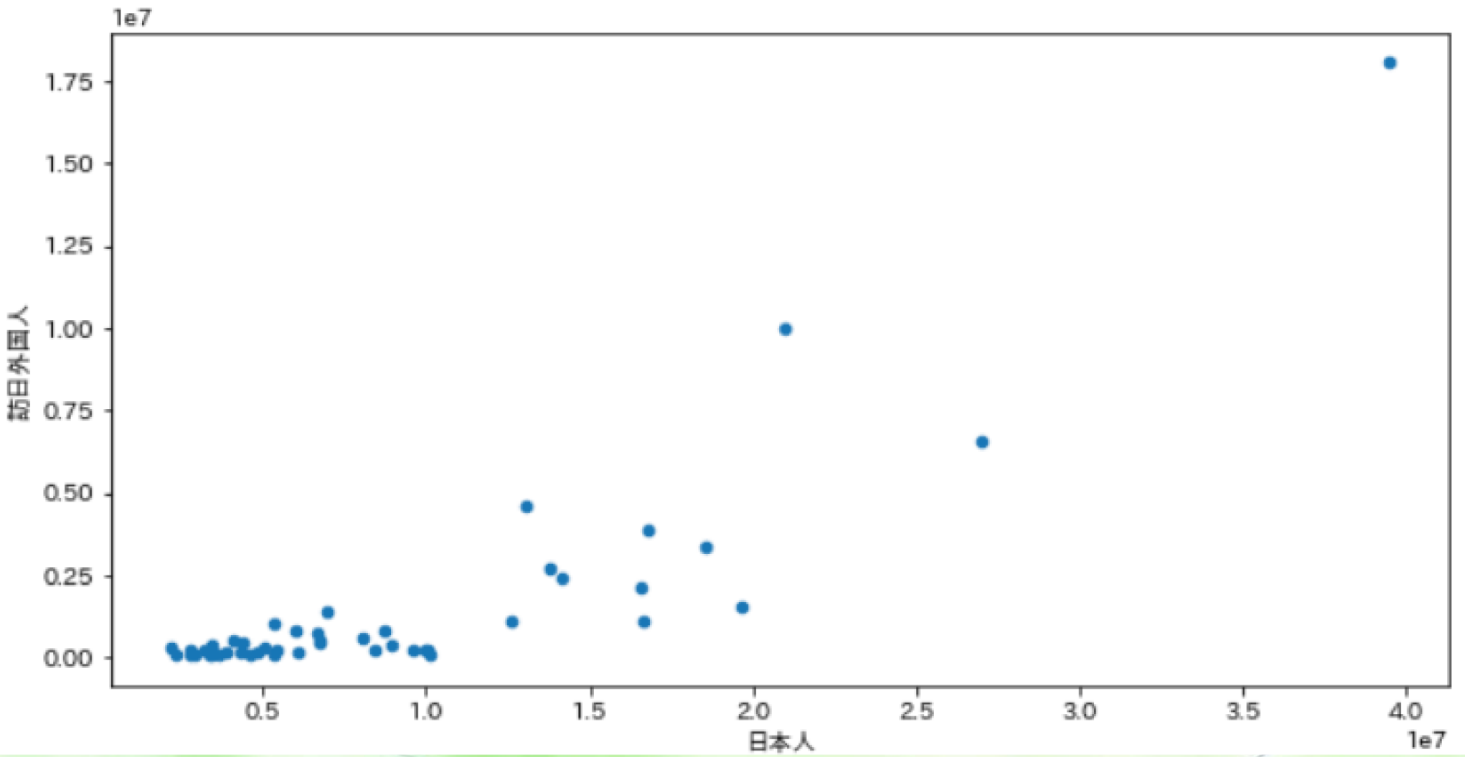
より直感的な記述が可能

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。

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

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

- 軸のannotateメソッドで座標を指定してラベルを描画できます。

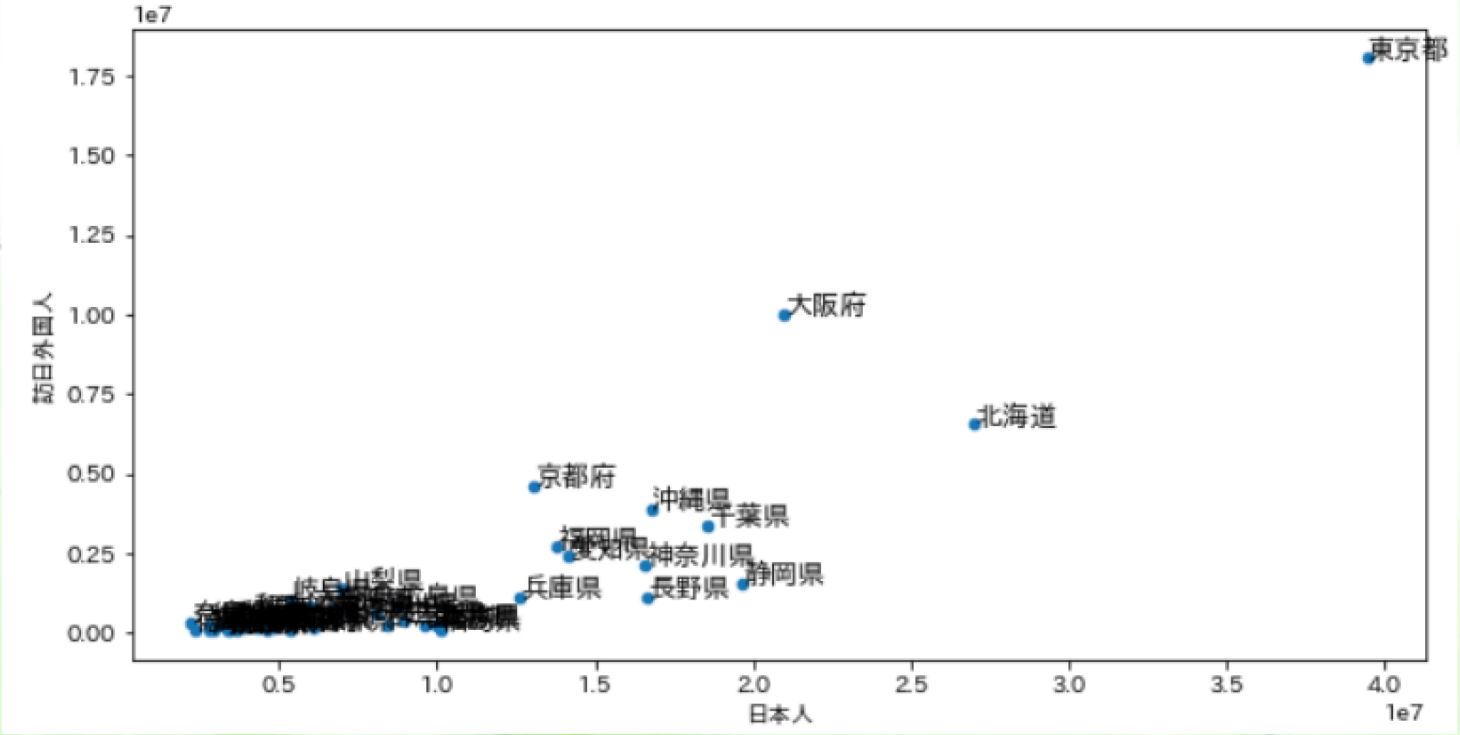
```
fig, axis = plt.subplots(figsize=(10, 5))
data.plot.scatter(ax = axis, x="日本人", y="訪日外国人")
for k, v in data.iterrows():
    axis.annotate(v[0], xy=(v[2],v[1]), size=12)
```

dataの行を取り出して、インデックスをkに、各列をvに入れる

都道府県名

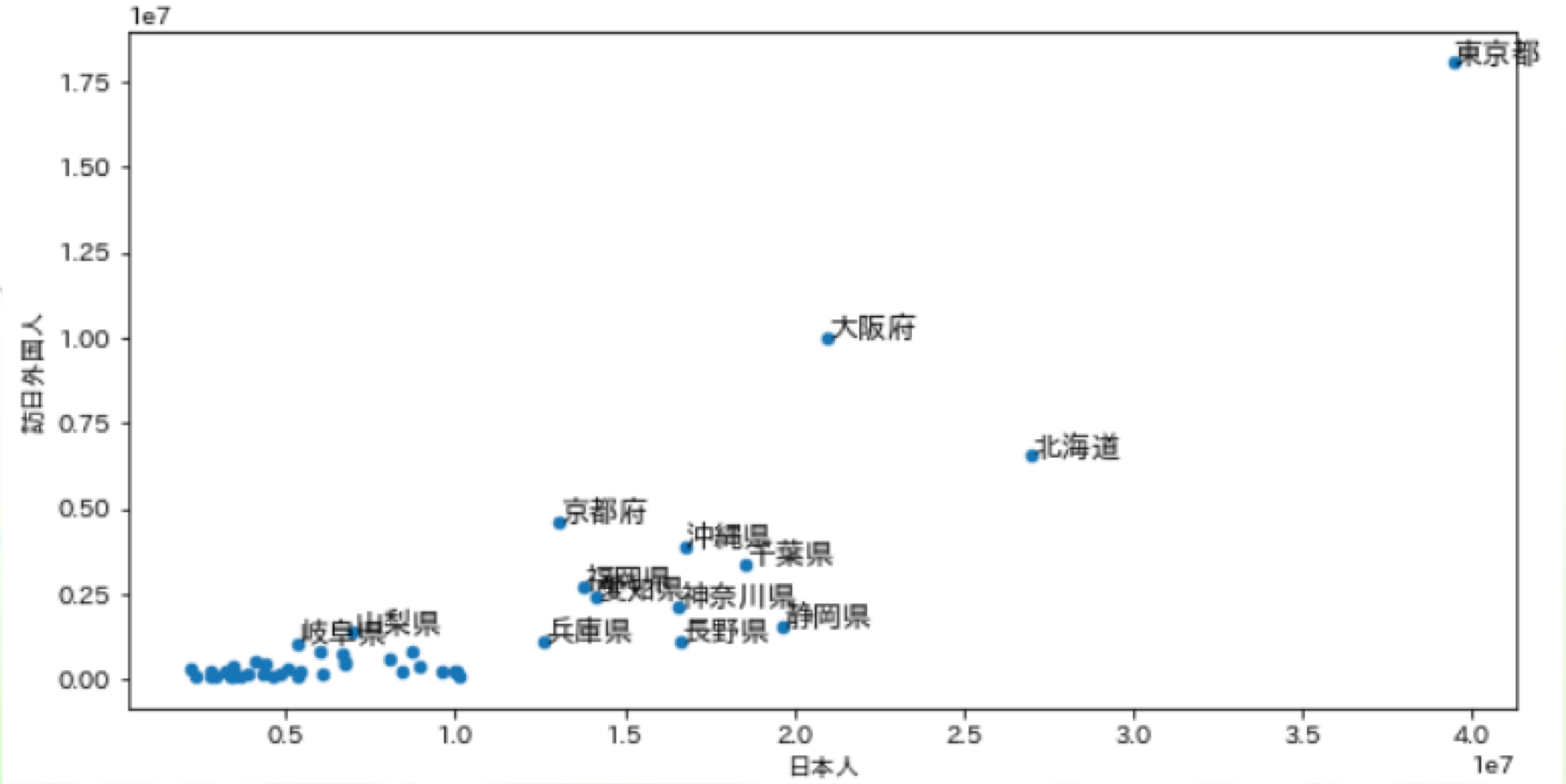
座標をタプルで指定

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

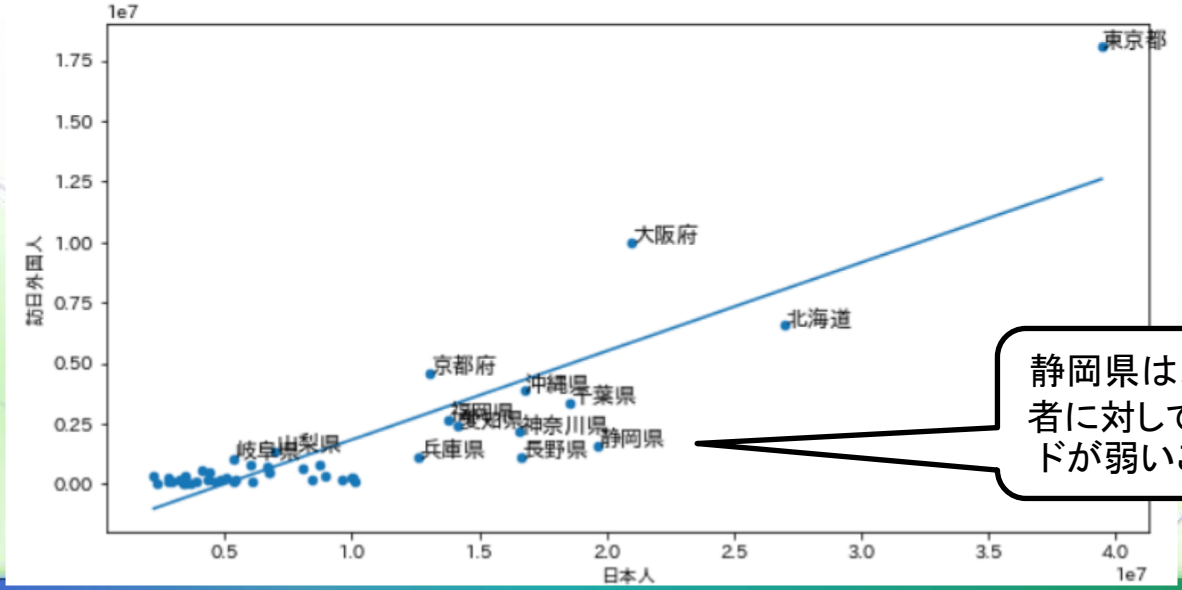
- 2組のデータの中心的な分布傾向を表す回帰直線を求めてグラフに描いてみましょう。SciPyのstatsモジュールには、回帰直線の傾きと切片を求めるlinregress関数が用意されています。
- 回帰直線の式：y = ax + b
  傾きaと切片bを求めると、グラフに直線が引けます。

aは傾き、
bは切片

```
a, b, r, p, err = st.linregress(data.日本人, data.訪日外国人)
sx = np.array([data.日本人.min(), data.日本人.max()])
sy = a * sx + b
plt.plot(sx, sy)
```

定義域から直線の方程式により値域を求める

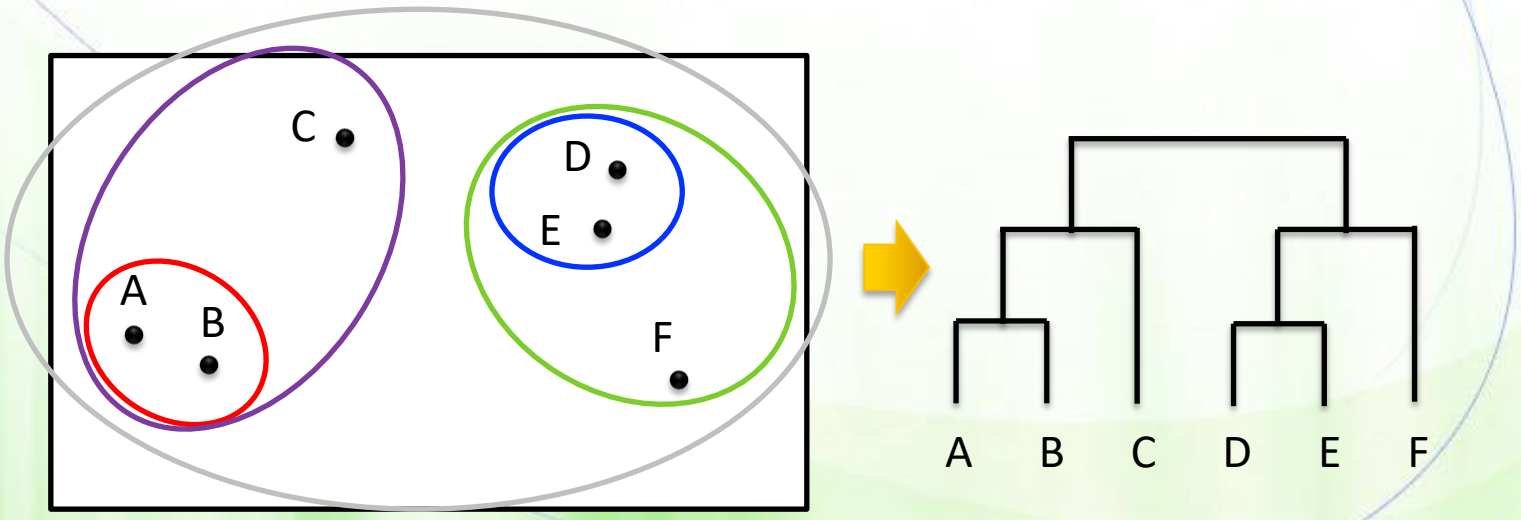
直線を引く定義域



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

事例：クラスタリング

- データの集合を、クラスタと呼ぶ部分集合に分ける処理をクラスタリングやクラスター分析と呼びます。
- クラスタは、データ同士がどれだけ似ているかを距離という指標をもとに評価して分割します。
- クラスタリングには、階層的手法と非階層的手法があります。
- 階層的手法
  - もっとも似ているデータから順番にクラスタにまとめていく方法で、途中経過が階層的に表すことができ、最終的に樹形図（デンドログラム）というトーナメント表のような図を得ることができます。



データセット

- 都道府県を宿泊施設の種類別にクラスタリングしてみましょう。
  - 宿泊施設数の多い20都道府県について
  - 種類
    - 旅館
    - リゾートホテル
    - ビジネスホテル
    - シティホテル
    - 簡易宿所
    - 会社、団体の宿泊所
- 準備
  - Jupyterで新たなNotebookを[New]-[Python 3]として新規作成します。
  - 今回のデータは、python.xlsxの「宿泊施設数」シートになります。

	A	B	C	D	E	F	G
1	都道府県	旅館	リゾート	ビジネス	シティホ	簡易宿所	会社・団体
2	東京都	1668200	367810	24030850	11909250	1713830	416380
3	北海道	3638330	2213410	9971510	4252600	420410	88770
4	大阪府	269970	1189350	11740500	6217540	683610	119020
5	愛知県	583160	279110	8524180	2467780	339580	116920
6	神奈川県	971920	1136460	6519380	2378860	721880	504520

データの取り込みと確認

階層的クラスタリングに必要な関数をインポート

```
import pandas as pd
import numpy as np
from scipy.spatial.distance  import pdist
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams["font.family"] = "IPAexGothic"

file = pd.ExcelFile("Downloads¥python.xlsx")
data = file.parse("宿泊施設数")
print(data)
```

	都道府県	旅館	リゾートホテル	ビジネスホテル	シティホテル	簡易宿所	会社・団体の宿泊所
0	東京都	1668200	367810	24030850	11909250	1713830	416380
1	北海道	3638330	2213410	9971510	4252600	420410	88770
2	大阪府	269970	1189350	11740500	6217540	683610	119020
3	愛知県	583160	279110	8524180	2467780	339580	116920

データ間の距離計算

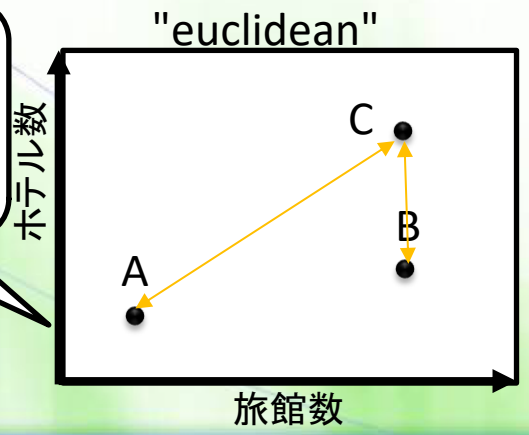
- まず、データ間の距離をpdist関数で計算します。
  - 都道府県名の列を除いた数値だけをpdist関数に渡します。
  - また、距離の計算方法を指定します。"euclidean"は、ユークリッド距離で計算するように指示しています。
    - "euclidean"：ユークリッド距離を指定。データ同士を結んだ直線の距離のこと。
    - "cosine"：コサイン類似度を指定。データに含まれる項目の割合がどれだけ似ているか。

```
dist = pdist(data.iloc[:,1:], "euclidean")
```

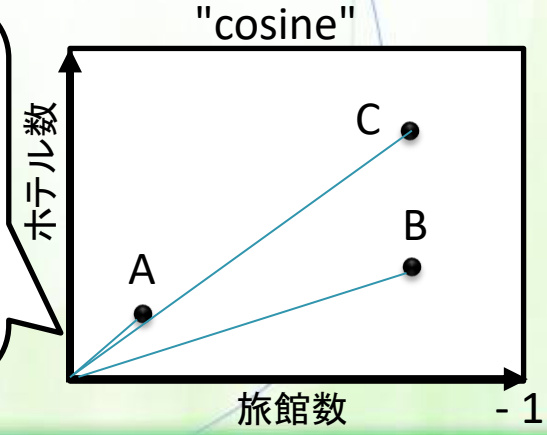
	都道府県	旅館	リゾートホテル	ビジネスホテル	シティホテル	簡易宿所
	会社・団体の宿泊所					
0	東京都	1668200	367810	24030850	11909250	1713830	416380
1	北海道	3638330	2213410	9971510	4252600	420410	88770
2	大阪府	269970	1189350	11740500	6217540	683610	119020
3	愛知県	583160	279110	8524180	2467780	339580	116920

data.iloc[:,1:]

ユークリッド距離では、AとCよりも、BとCが（直線距離が近いので）似ている



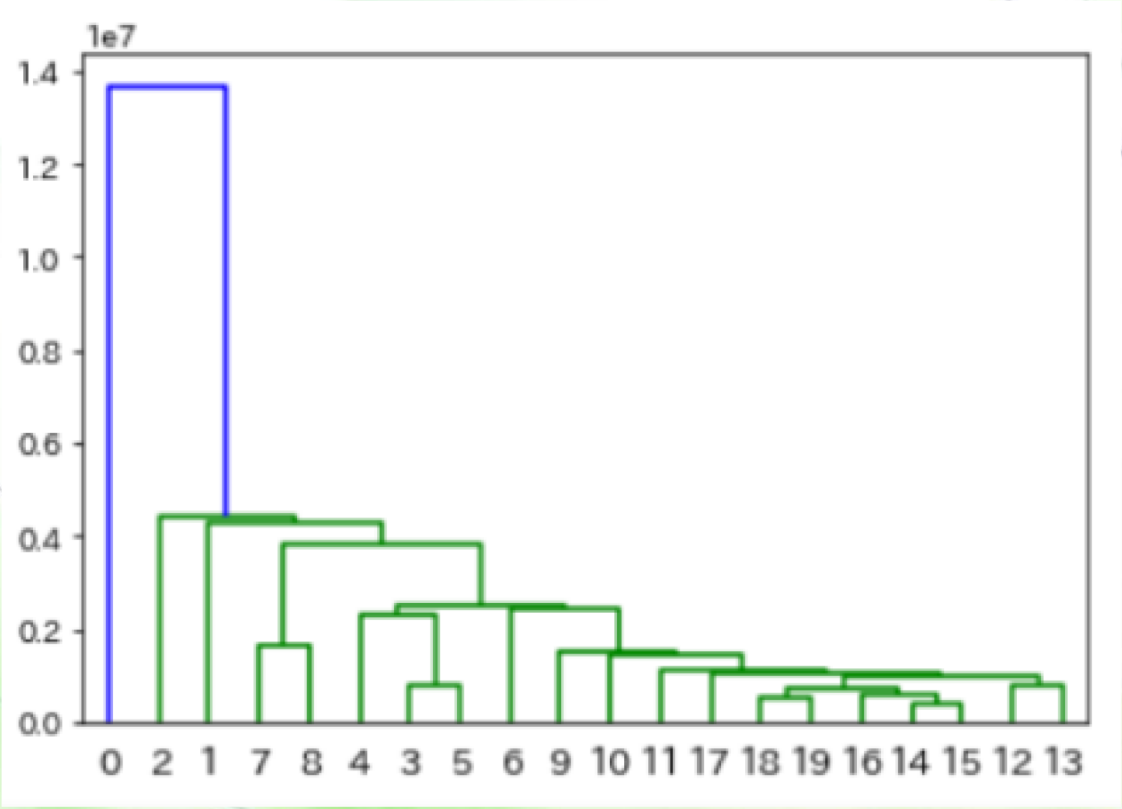
コサイン類似度では、AとCは原点から引いた直線の角度がより近いので似ている。つまり、旅館とホテルの割合が似ている。



階層的クラスタリングを可視化

- linkage関数で階層的クラスタリングを実行し、dendrogram関数で樹形図として可視化します。

```
result = linkage(dist)
dendrogram(result)
```



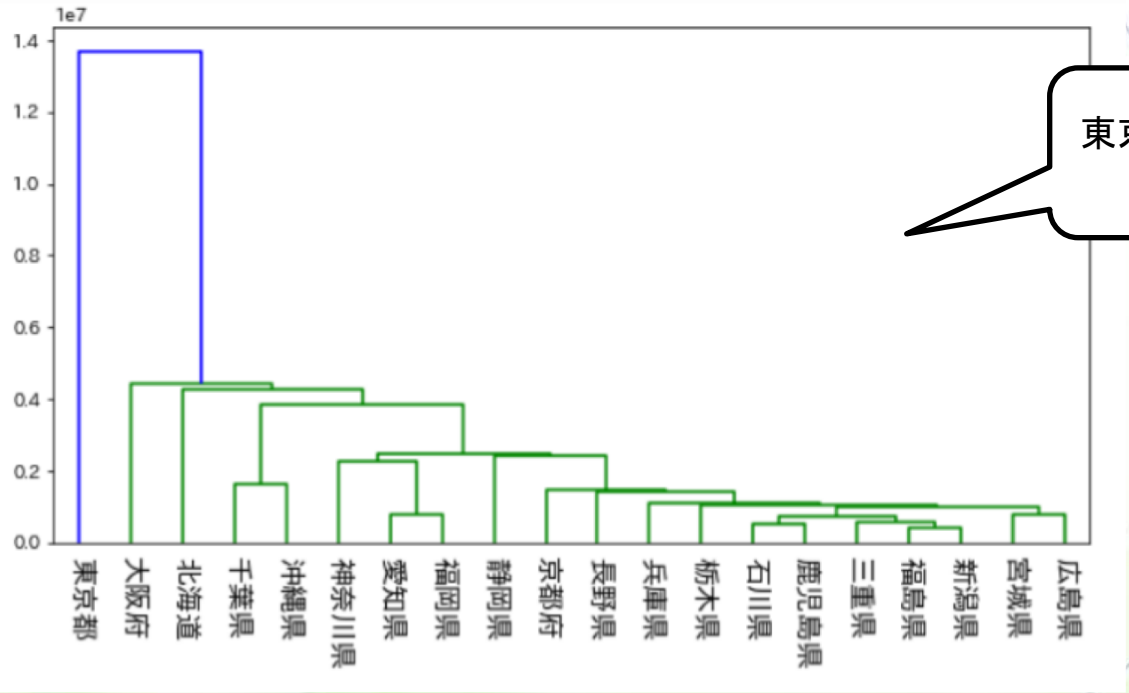
樹形図にラベルを表示

- 図のサイズをより大きくし、ラベルを表示しましょう。

```
result = linkage(dist)
plt.subplots(figsize=(10,5))
dendrogram(result, labels=data.iloc[:,0].tolist(),
           leaf_rotation=-90, leaf_font_size=15)
```

dataの１列目（都道府県名）をtolistメソッドでリストに変換してラベルに設定

ラベルを-90度回転させてフォントサイズ15を指定



東京都だけ仲間はずれ？

宿泊施設の割合を棒グラフで確認

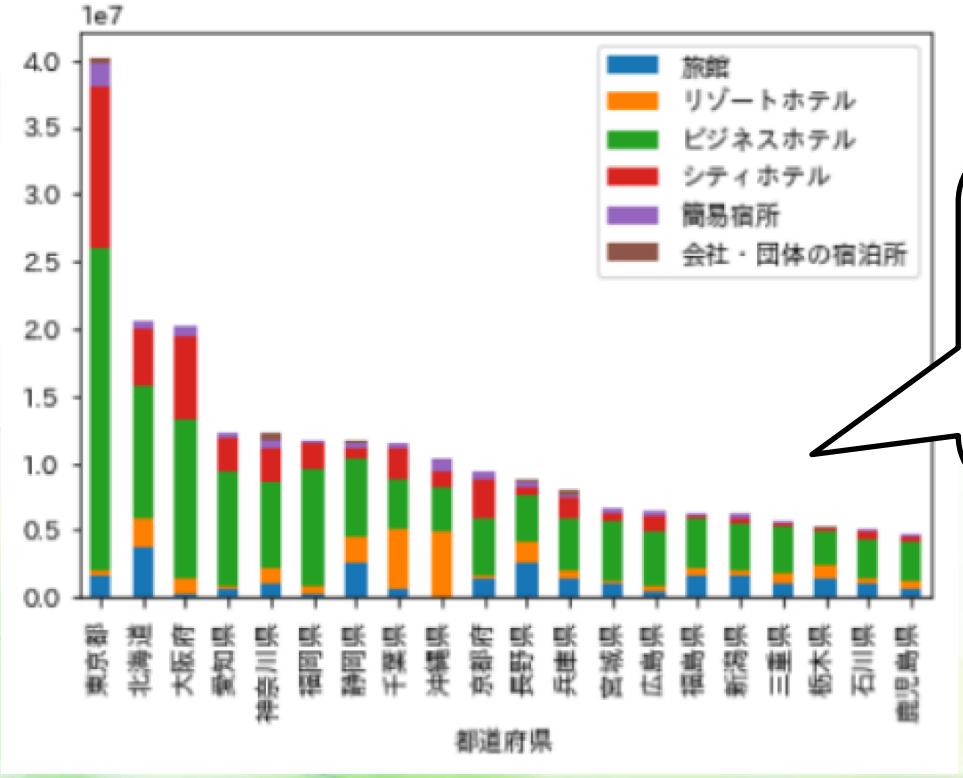
```
data.index = data.都道府県
data.loc[:,"旅館":].plot.bar(stacked=True)
```

dataの１列目（都道府県名）をindexに上書き（あとで特定の都道府県の行を取り出せるように）

上書き

	都道府県	旅館	リゾートホテル	ビジネスホテル	シティホテル	簡易宿所
0	東京都	1668200	367810	24030850	11909250	1713830
1	北海道	3638330	2213410	9971510	4252600	420410
2	大阪府	269970	1189350	11740500	6217540	683610

旅館の右側のデータでグラフを描く

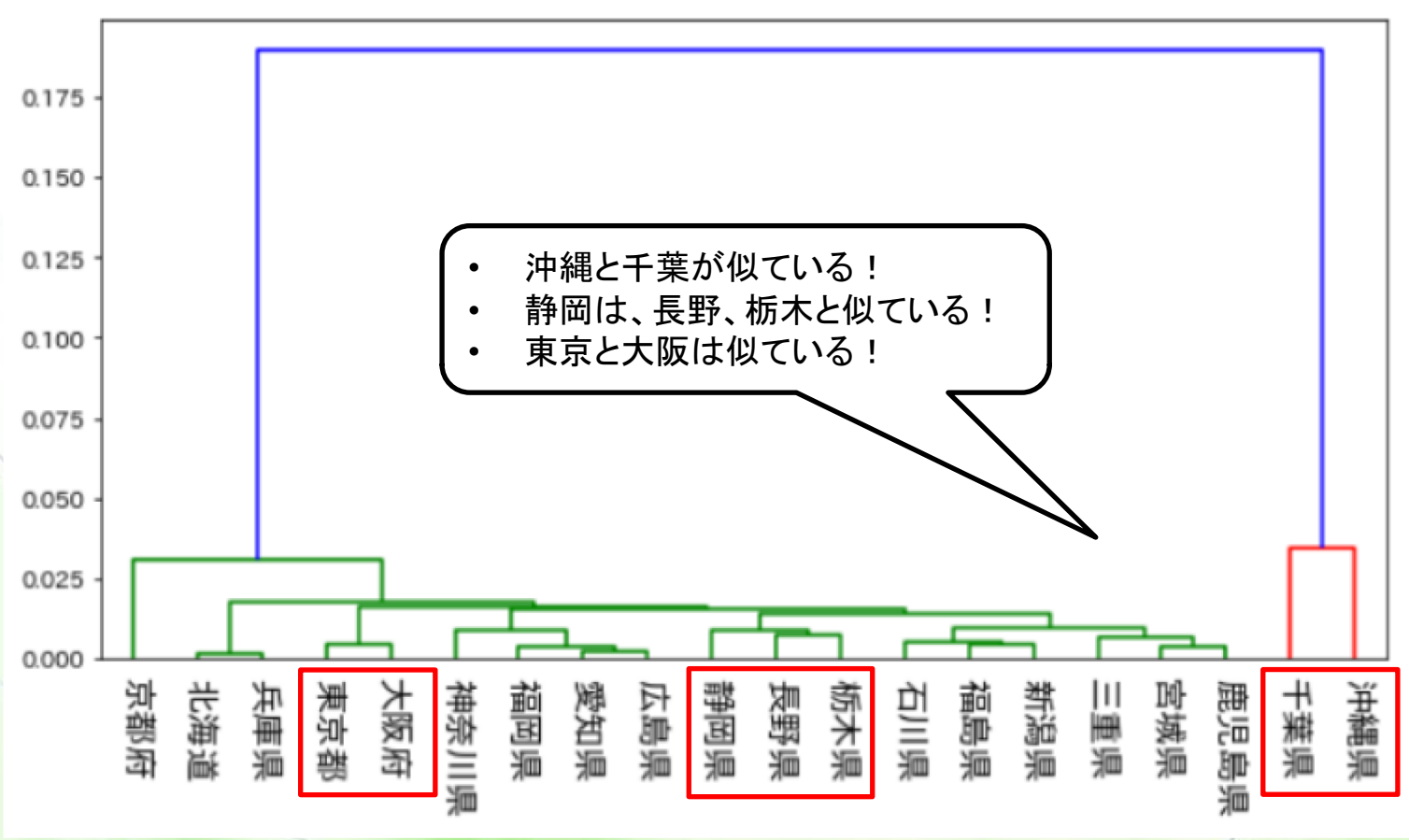


東京都だけ宿泊施設数が圧倒的に多い！ユークリッド距離では、宿泊施設の絶対数がクラスタ分類にかなり影響している。

距離計算をコサイン類似度に変更

- 宿泊施設の絶対数ではなく、構成比が似ているかでクラスタリングしたい！

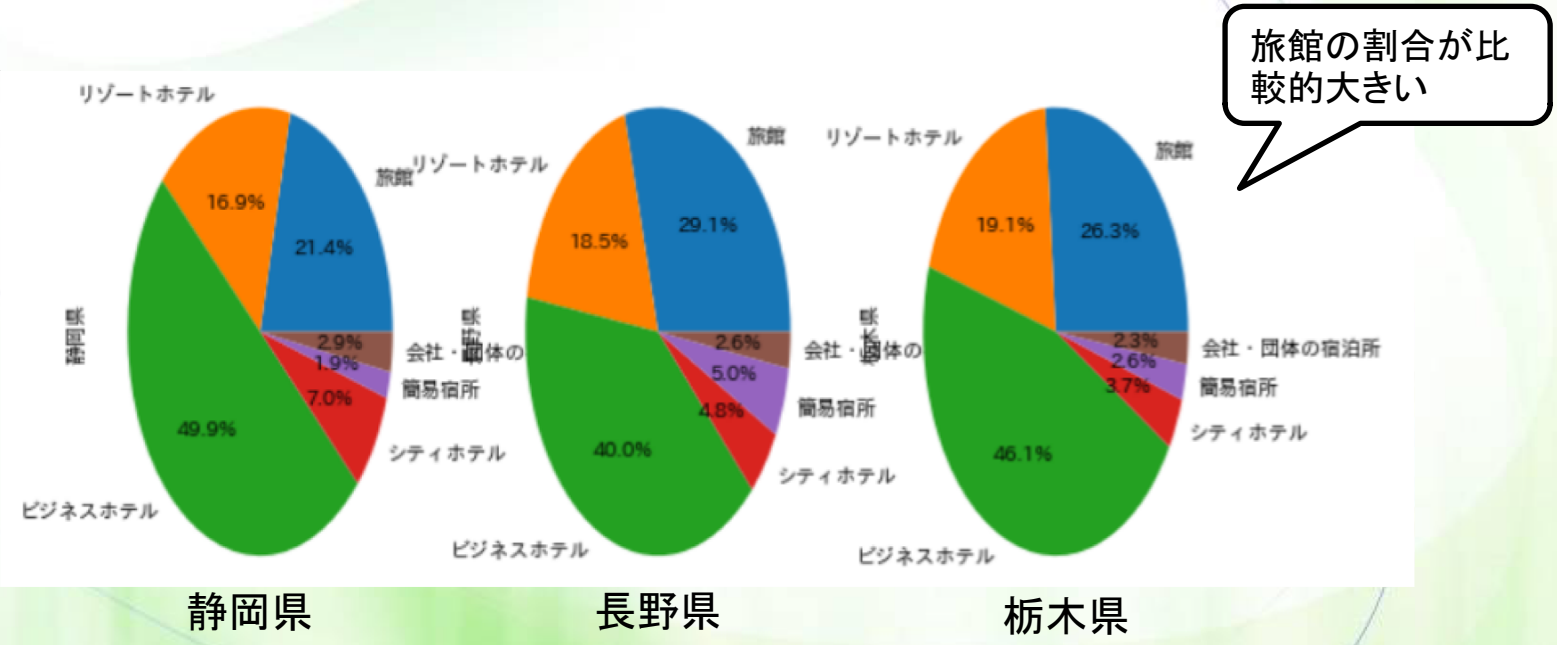
```
dist = pdist(data.iloc[:,1:], "cosine")
```



円グラフで構成比を確認

- 静岡県、長野県、栃木県の宿泊施設の構成比を描いて確認してみましょう。

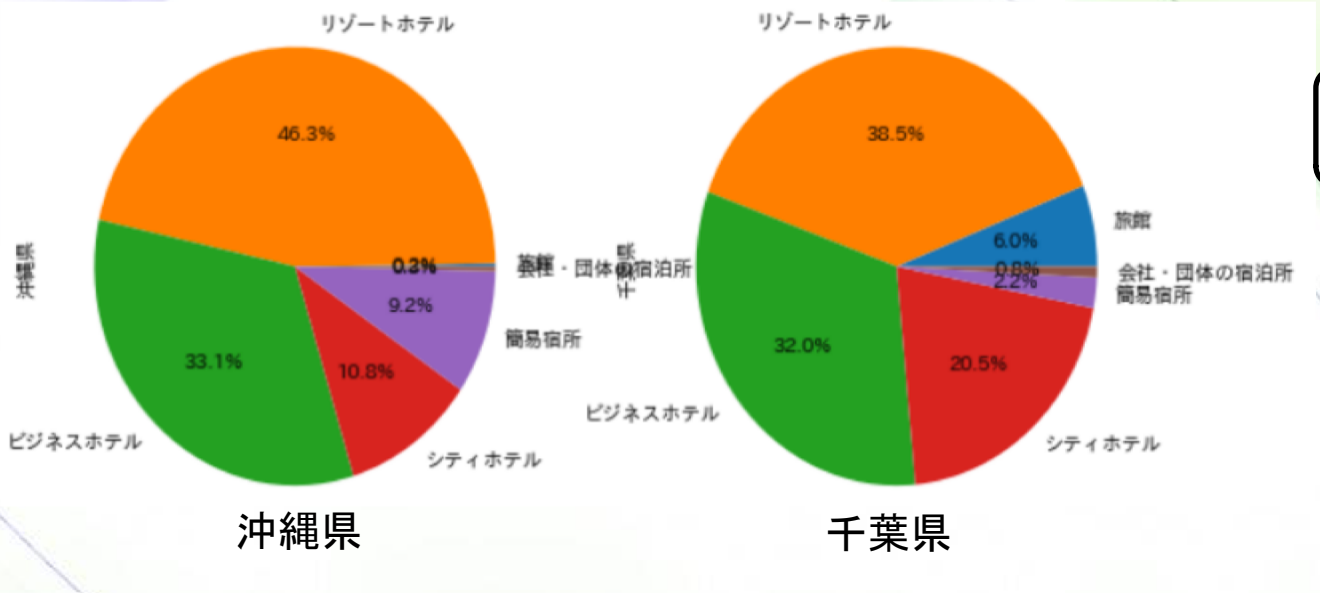
```
data.index = data.都道府県
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(10,5))
data.loc["静岡県","旅館":].plot.pie(ax=axes[0], autopct="%.1f%%")
data.loc["長野県","旅館":].plot.pie(ax=axes[1], autopct="%.1f%%")
data.loc["栃木県","旅館":].plot.pie(ax=axes[2], autopct="%.1f%%")
```



静岡県　長野県　栃木県

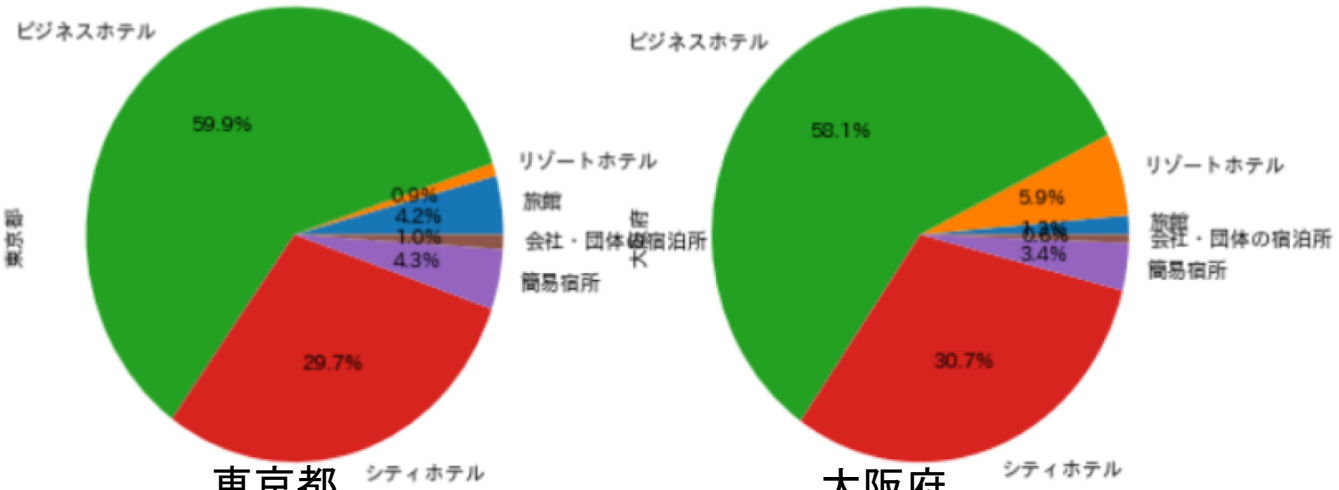
課題10

- 千葉県と沖縄県、東京都と大阪府の円グラフも描いてください。



沖縄県　　千葉県

リゾートホテルの割合が大きい



東京都　　大阪府

ビジネスホテルの割合が大きい