

## クレジット:

UTokyo Online Education Education コンピュータシステム概論 2018 小林克志

## ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



## Cartopy を利用した地理情報の可視化

Cartopy (<http://scitools.org.uk/cartopy/>) は地図生成や地理空間可視化を目的とした地理空間データ処理パッケージである。

以下の例は、Cartopy を利用して、[Natural Earth](http://www.naturalearthdata.com) (<http://www.naturalearthdata.com>) から日本の都道府県の境界データを取得し、matplotlib の patch を作成、描画するプログラムである:

```
In [4]: %matplotlib notebook
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import cartopy.io.shapereader as shapereader

# 10m 解像度のデータ読み込み
shpfilename = shapereader.natural_earth(resolution='10m',
                                       category='cultural',
                                       name='admin_1_states_provinces')

# 国内の都道府県のみ取得
reader = shapereader.Reader(shpfilename)
provinces = []
for province in reader.records():
    if(province.attributes["admin"] == "Japan"):
        provinces.append(province)

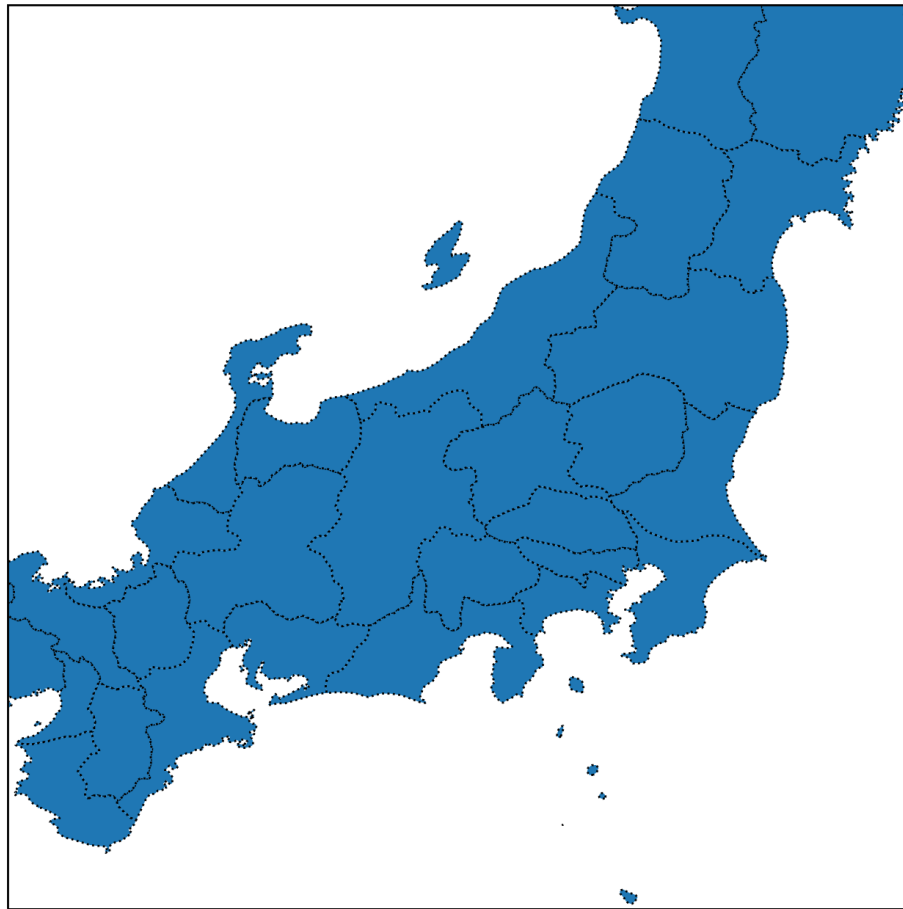
# Figure, Subplot 生成
plt.figure(figsize=[8,8])
ax = plt.axes(projection=ccrs.PlateCarree())

# 都道府県のパッチで埋めていく
for province in provinces:
    geometry = province.geometry
    ax.add_geometries(geometry, ccrs.PlateCarree(), edgecolor="black",
                      linestyle=":")

# 日本以外を含む海岸線を描画
# ax.coastlines(resolution='10m')

# 描画範囲を緯度、経度で指定
ax.set_extent([135, 142, 33, 40])
plt.show()
```

<IPython.core.display.Javascript object>



都道府県データから、英語名 "name"、日本語名 "name\_local" を確認してみる。残念ながら "Shizuoka" に対応する日本語名はないようだ:

```
In [5]: for province in provinces:  
        print(province.attributes["name"], province.attributes["name_  
        local"])
```

Hiroshima 広島県  
Okayama 岡山県  
Shimane 島根県  
Tottori 鳥取県  
Yamaguchi 山口県  
Saga 佐賀県  
Fukuoka 福岡県  
Kumamoto 熊本県  
Miyazaki 宮崎県  
Ehime 愛媛県  
Kagawa 香川県  
Kochi 高知県  
Oita 大分県  
Tokushima 徳島県  
Aichi 愛知県  
Gifu 岐阜県  
Ishikawa 石川県  
Mie 三重県  
Nagano 長野県  
Shizuoka  
Toyama 富山県  
Hokkaido 北海道  
Fukui 福井県  
Hyōgo 兵庫県  
Kyoto 京都府  
Nara 奈良県  
Osaka 大阪府  
Shiga 滋賀県  
Wakayama 和歌山県  
Chiba 千葉県  
Ibaraki 茨城県  
Kanagawa 神奈川県  
Saitama 埼玉県  
Tochigi 栃木県  
Tokyo 東京都  
Yamanashi 山梨県  
Akita 秋田県  
Aomori 青森県  
Fukushima 福島県  
Iwate 岩手県  
Miyagi 宮城県  
Niigata 新潟県  
Yamagata 山形県  
Nagasaki 長崎県  
Kagoshima 鹿児島県  
Okinawa 沖縄県  
Gunma 群馬県

## ヒートマップによる可視化

住民基本台帳に基づく人口動態データ ([http://www.soumu.go.jp/menu\\_news/s-news/01gyosei02\\_02000148.html](http://www.soumu.go.jp/menu_news/s-news/01gyosei02_02000148.html))の人口増減率から都道府県単位のヒートマップを描いてみる:

```
In [6]: import matplotlib.pyplot as plt
import cartopy.crs as ccrs
```

```

import cartopy.feature as cfeature
import cartopy.io.shapereader as shapereader
import pandas as pd
import numpy as np

# Excel データ読み込み
df = pd.read_excel("000494956.xls", sheet_name=0, header=[1,2,3],
skiprows=[4])
# Omit "Unnamed" indices and adjust
for i, col in enumerate(df.columns.levels):
    columns = np.where(col.str.contains("Unnamed"), "", col)
    df.columns.set_levels(columns, level=i, inplace=True)
df.set_index("都道府県名", inplace=True)
df.index.name="都道府県名"

shpfilename = shapereader.natural_earth(resolution='10m',
category='cultural',
name='admin_1_states_provinces')

reader = shapereader.Reader(shpfilename)

provinces = []
for province in reader.records():
    if(province.attributes["admin"] == "Japan"):
        provinces.append(province)

plt.figure(figsize=[8,8])
ax = plt.axes(projection=ccrs.PlateCarree())

# カラーマップのためのスケールデータ
cmap = plt.cm.magma
cnorm = max(df["平成28年","増減率"]) - min(df["平成28年","増減率"])
coff = min(df["平成28年","増減率"])

for province in provinces:
    geometry = province.geometry
    if province.attributes["name_local"] == "":
        continue
    color = cmap((df["平成28年","増減率"][province.attributes["name_local"]] - coff) / cnorm)
    ax.add_geometries(geometry, ccrs.PlateCarree(), edgecolor="black", linestyle=":",
facecolor=color)

ax.set_title("Population Growth Rate H28 - H29")

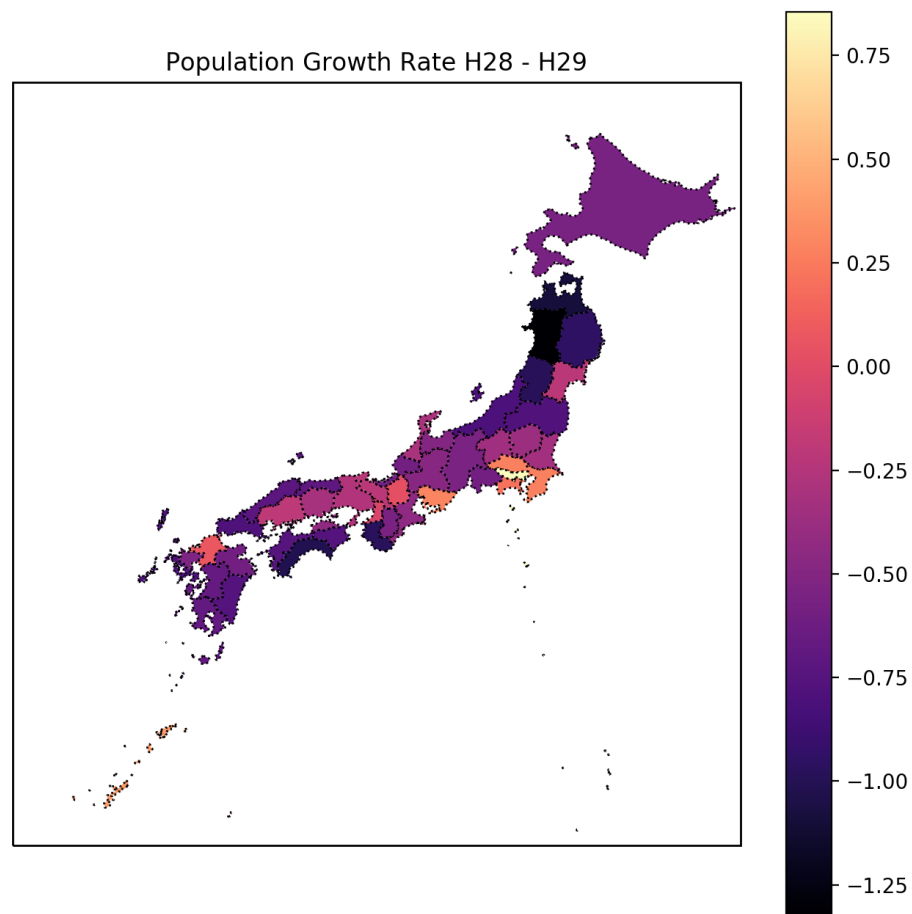
#ax.coastlines(resolution='10m')
ax.set_extent([125, 146, 25, 47])

sm = plt.cm.ScalarMappable(cmap=cmap,norm=plt.Normalize(cnorm + coff,coff))
sm._A = []
plt.colorbar(sm,ax=ax)

plt.savefig("sample.pdf", dps = 900, bbox_inches="tight")
plt.show()

```

<IPython.core.display.Javascript object>



```
/Users/ikob/anaconda3/lib/python3.6/site-packages/ipykernel/kerne  
lbase.py:399: PerformanceWarning: indexing past lexsort depth may  
impact performance.  
    user_expressions, allow_stdin)
```