

コロナ感染者数グラフ作成

pandas + matplotlib でグラフ作成

```
In [1]: import datetime
import pandas as pd
import matplotlib.pyplot as plt
import japanize_matplotlib
from japanmap import picture
```

NHKサイトからダウンロードしたcsvを読み込み

<https://www3.nhk.or.jp/news/special/coronavirus/data-widget/#mokuji1>

```
In [2]: df = pd.read_csv('nhk_news_covid19_prefectures_daily_data.csv', encoding = ('utf8'))

# 日付のフォーマット修正
df['日付'] = df['日付'].replace('(\d+)/(\d+)/(\d+)', '\\1/0\\2/0\\3', regex=True)
df['日付'] = df['日付'].replace('(\d+)/0*(\d{2,})/0*(\d{2,})', '\\1/\\2/\\3', regex=True)
df
```

```
Out[2]:
```

	日付	都道府県コード	都道府県名	各地の感染者数_1日ごとの発表数	各地の感染者数_累計	各地の死者数_1日ごとの発表数	各地の死者数_累計
0	2020/01/16	1	北海道	0	0	0	0
1	2020/01/17	1	北海道	0	0	0	0
2	2020/01/18	1	北海道	0	0	0	0
3	2020/01/19	1	北海道	0	0	0	0
4	2020/01/20	1	北海道	0	0	0	0
...
24529	2021/06/16	47	沖縄県	115	19784	0	163
24530	2021/06/17	47	沖縄県	97	19881	4	167
24531	2021/06/18	47	沖縄県	86	19967	0	167
24532	2021/06/19	47	沖縄県	97	20064	0	167
24533	2021/06/20	47	沖縄県	61	20125	0	167

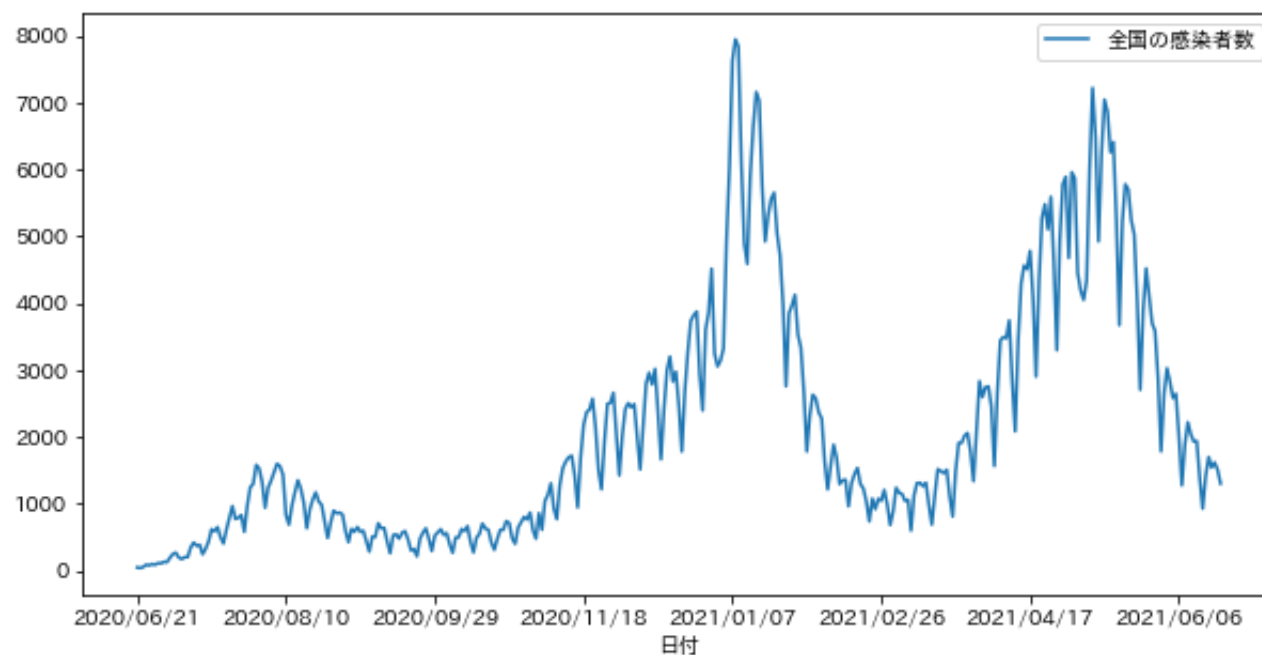
24534 rows × 7 columns

データを整形してグラフ表示

```
In [3]: # 折線グラフ化する直近のレコード数(7 = 直近7日分)
get_count = 365
```

```
In [4]: # 全国
df_all = df.copy()
df_all = df_all.groupby('日付').sum()
df_all = df_all.loc[:, ['各地の感染者数_1日ごとの発表数']]
df_all = df_all.rename(columns={'各地の感染者数_1日ごとの発表数': '全国の感染者数'})
df_all = df_all.sort_index()
df_all.tail(get_count).plot(figsize=(10, 5))
```

Out[4]: <AxesSubplot: xlabel='日付'>

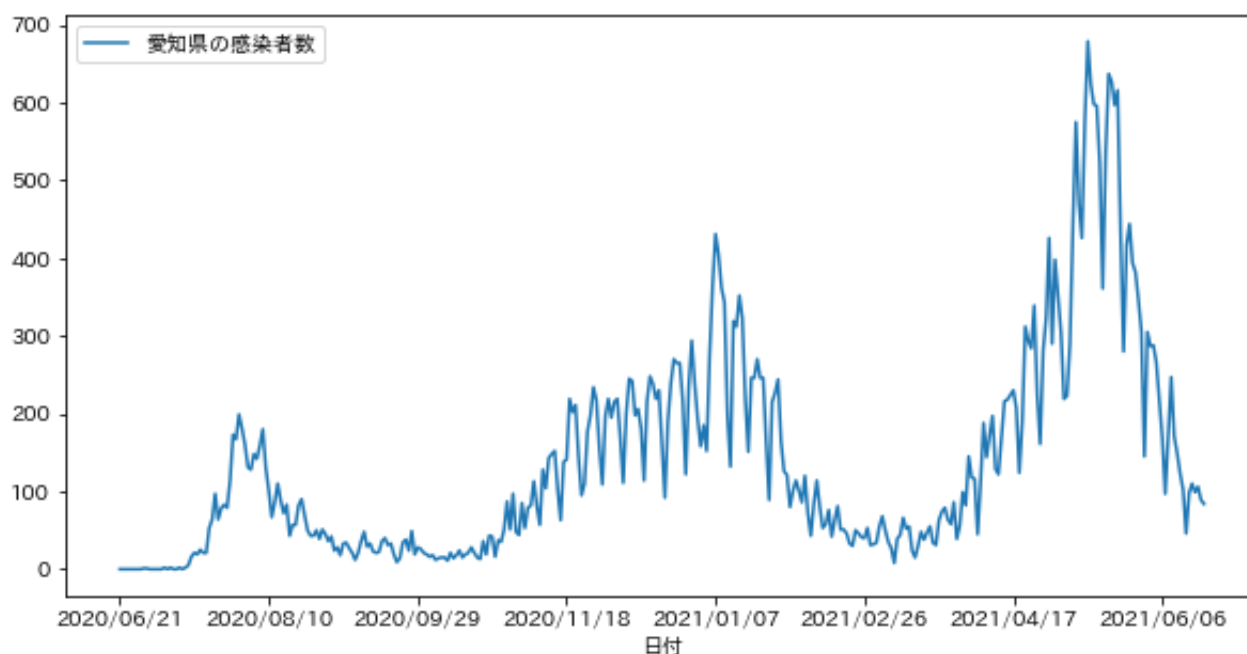


```

In [5]: # 愛知
df_aichi = df.copy()
df_aichi = df_aichi[df_aichi.都道府県名 == '愛知県']
df_aichi = df_aichi.set_index('日付')
df_aichi = df_aichi.loc[:, ['各地の感染者数_1日ごとの発表数']]
df_aichi = df_aichi.rename(columns={'各地の感染者数_1日ごとの発表数': '愛知県の感染者数'})
df_aichi = df_aichi.sort_index()
df_aichi.tail(get_count()).plot(figsize=(10, 5))

```

Out[5]: <AxesSubplot:xlabel='日付'>



昨日の県別感染者数で地図を色分け

```

In [6]: d_yest = '{0:%Y/%m/%d}'.format(datetime.date.today() - datetime.timedelta(days=1))

# csvからデータ抽出
df_yest = df.copy()
df_yest = df_yest[df_yest.日付 == d_yest]
df_yest = df_yest.set_index('都道府県コード')
df_yest = df_yest.loc[:, ['各地の感染者数_1日ごとの発表数']]

# japanmapで画像生成
cmap = plt.get_cmap('Reds')
norm = plt.Normalize(vmin=df_yest.各地の感染者数_1日ごとの発表数.min(),
                    vmax=df_yest.各地の感染者数_1日ごとの発表数.max())
fcol = lambda x: '#' + bytes(cmap(norm(x), bytes=True)[:3]).hex()
plt.subplots(figsize=(8,8))
plt.colorbar(plt.cm.ScalarMappable(norm, cmap))
plt.imshow(picture(df_yest.各地の感染者数_1日ごとの発表数.apply(fcol)));

```

