

## ▼ Chapter 8 - Exercise 2: Canada

Dữ liệu Canada.xlsx chứa thông tin nhập cư vào Canada từ năm 1980 đến năm 2013. Bộ dữ liệu chứa dữ liệu hàng năm về dòng người di cư đến Canada được ghi nhận, trình bày thông tin inflows and outflows theo nơi sinh, quốc tịch hoặc nơi cư trú trước đó / tiếp theo cho cả người nước ngoài và quốc tịch. Chúng tôi sẽ tập trung vào dữ liệu nhập cư Canada.

```
import folium
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

## ▼ Map

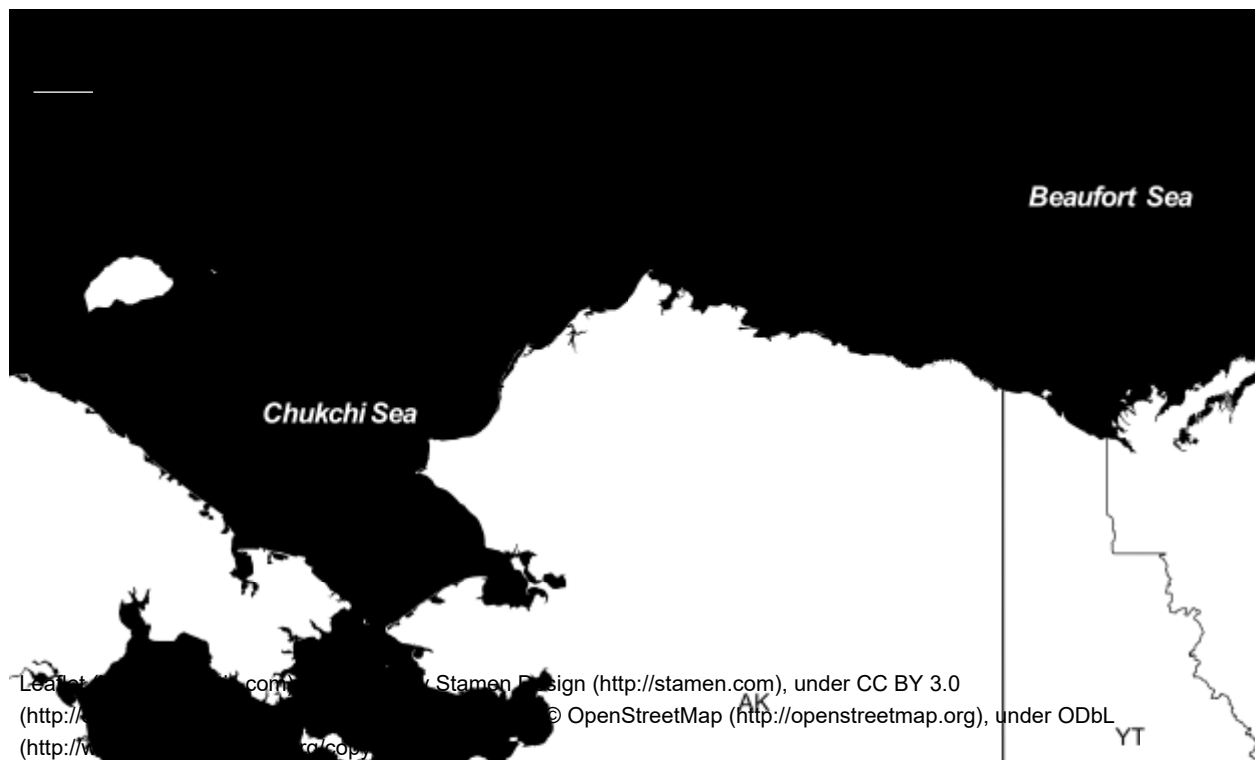
```
# Câu 1: Hiển thị bản đồ thế giới
world_map = folium.Map()
world_map
```



```
# Câu 2: Tạo bản đồ với center là Canada (location=[56.130, -106.35]) và zoom level (zoom_start=4)
canada_map = folium.Map(location=[56.130, -106.35], zoom_start=4)
canada_map
```



```
# Câu 3: Tạo Stamen Toner Map với center là Canada, và zoom level là 4
canada_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Toner')
canada_map
```



```
# Câu 4: Tạo Stamen Terrain Map với center là Canada, và zoom level là 4
canada_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Terrain')
canada_map
```



Leaflet (<https://leafletjs.com>) | Map tiles by Stamen Design (<http://stamen.com>), under CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0>). Data by © OpenStreetMap (<http://openstreetmap.org>), under CC BY SA (<http://creativecommons.org/licenses/by-sa/3.0>).

```
# Câu 5: Tạo Stamen Watercolor Map với center là Canada, và zoom level 4
world_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Watercolor')
```

world\_map







## ▼ Choropleth Map

```
# Câu 1: Đọc dữ liệu Canada.xlsx và lưu vào df_can,  
# tìm hiểu về dữ liệu với: describe, head, shape, columns  
df_can = pd.read_excel('data\Canada.xlsx', sheet_name='Canada by Citizenship',  
                      skiprows=range(20), skipfooter=2)
```

(<http://openstreetmap.org>). under CC BY SA (<http://creativecommons.org/licenses/by-sa/3.0>).

```
# Cho biết thông tin thống kê chung của df_can  
df_can.describe()
```

	AREA	REG	DEV	1980	1981	1982
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000
mean	912.764103	1249.015385	901.753846	508.394872	566.989744	534.723077
std	13.082835	1185.526885	0.431878	1949.588546	2152.643752	1866.997511
min	903.000000	905.000000	901.000000	0.000000	0.000000	0.000000
25%	903.000000	914.000000	902.000000	0.000000	0.000000	0.000000
50%	908.000000	922.000000	902.000000	13.000000	10.000000	11.000000

```
# In vài dòng dữ liệu đầu của df_can
df_can.head()
```

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions

```
# Cho biết kích thước của df_can
df_can.shape
```

```
(195, 43)
```

```
# Xem danh sách các cột của df_can
df_can.columns
```

```
Index(['Type', 'Coverage', 'OdName', 'AREA', 'AreaName', 'REG',
       'RegName', 'DEV', 'DevName', '1980', '1981', '1982',
```

```

1983,      1984,      1985,      1986,      1987,      1988,
1989,      1990,      1991,      1992,      1993,      1994,
1995,      1996,      1997,      1998,      1999,      2000,
2001,      2002,      2003,      2004,      2005,      2006,
2007,      2008,      2009,      2010,      2011,      2012,
2013],
dtype='object')

```

```

# Câu 2: Làm sạch dữ liệu:
# Bỏ đi những cột không cần thiết như 'AREA', 'REG', 'DEV', 'Type', 'Coverage'
df_can.drop(['AREA', 'REG', 'DEV', 'Type', 'Coverage'], axis=1, inplace=True)

# Đổi tên một số cột như sau:
#   'OdName' => 'Country', 'AreaName' => 'Continent', 'RegName' => 'Region'
df_can.rename(columns={'OdName': 'Country', 'AreaName': 'Continent', 'RegName': 'Region'},
               inplace=True)

# Đổi tất cả các cột sang kiểu string
df_can.columns = list(map(str, df_can.columns))

# Thêm cột Total chứa tổng lượng nhập cư qua các năm
df_can['Total'] = df_can.sum(axis=1)

```

```

# Câu 3: Xem thông tin dữ liệu lúc này:
# Hiển thị 5 dòng dữ liệu đầu của df_can sau khi làm sạch dữ liệu
df_can.head()

```

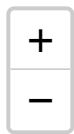
Country	Continent	Region	PopName	1980	1981	1982	1983	1984	1985
---------	-----------	--------	---------	------	------	------	------	------	------

```
# Cho biết kích thước của df_can sau khi làm sạch dữ liệu
df_can.shape
```

```
(195, 39)
```

```
        'Europe'      regions
```

```
# Câu 4: Tạo world map, với center [0, 0] là latitude và longitude, zoom level là 2, sử dụng tiles là OpenStreetMap
world_map = folium.Map(location=[0, 0], zoom_start=2, tiles='OpenStreetMap')
world_map
```



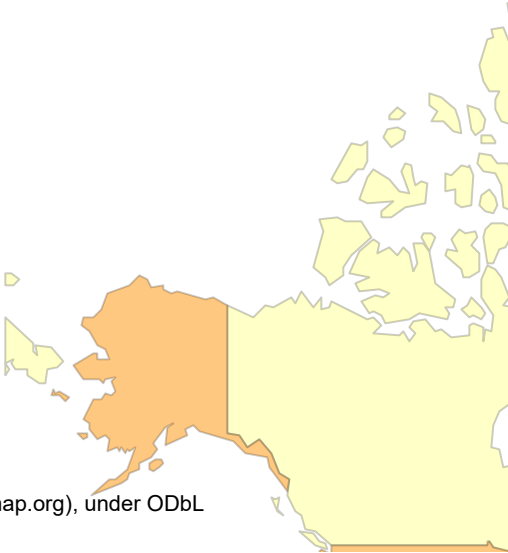
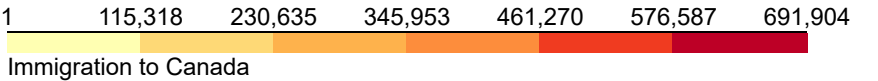
Leaflet (<https://leafletjs.com>) | Data by © OpenStreetMap (<http://openstreetmap.org>), under ODbL (<http://www.openstreetmap.org/copyright>).

```
# Câu 5: Tạo choropleth map sử dụng total nhập cư của từng quốc gia vào Canada từ năm 1980 đến năm 2013
```

```
# Lấy file GeoJSON có tên là world-countries.json  
world_geo = r'data\world-countries.json'
```

```
folium.Choropleth(  
    geo_data=world_geo,  
    data=df_can,
```

```
columns=['Country', 'Total'],  
key_on='feature.properties.name',  
fill_color='YlOrRd',  
fill_opacity=0.7,  
line_opacity=0.2,  
legend_name='Immigration to Canada'  
)  
.add_to(world_map)  
  
world_map
```



Leaflet (<https://leafletjs.com>) | Data by © OpenStreetMap (<http://openstreetmap.org>), under ODbL (<http://www.openstreetmap.org/copyright>).

