Assignment

Sanabila Khoirunnisa 130120409

Import necessary library

In [457…
```
import numpy as np  # useful for many scientific computing in Python
import pandas as pd # primary data structure library
```

Read the data

In [458…
```
df_can = pd.read_excel('https://s3-api.us-geo.objectstorage.softlayer.net/cf-course
                       sheet_name='Canada by Citizenship',
                       skiprows=range(20),
                       skipfooter=2)
```

Drop unnecessary columns

In [459…
```
# in pandas axis=0 represents rows (default) and axis=1 represents columns.
df_can.drop(['AREA','REG','DEV','Type','Coverage'], axis=1, inplace=True)
df_can.head(2)
```

Out[459]:

	OdName	AreaName	RegName	DevName	1980	1981	1982	1983	1984	1985	...	2004
0	Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	2978
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	1450

2 rows × 38 columns

Rename columns title

In [460…
```
df_can.rename(columns={'OdName':'Country', 'AreaName':'Continent', 'RegName':'Regi
df_can.columns
```

Out[460]:
```
Index([  'Country', 'Continent',    'Region',   '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')
```

Add a 'Total' column

In [461…
```
df_can['Total'] = df_can.sum(axis=1)
df_can
```

```
C:\Users\sanabila\AppData\Local\Temp\ipykernel_15140\2515980790.py:1: FutureWarnin
g: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only valid c
olumns before calling the reduction.
  df_can['Total'] = df_can.sum(axis=1)
```

Out[461]:

	Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	...	200
0	Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	343
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	122
2	Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	...	362
3	American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	...	
4	Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
190	Viet Nam	Asia	South-Eastern Asia	Developing regions	1191	1829	2162	3404	7583	5907	...	185
191	Western Sahara	Africa	Northern Africa	Developing regions	0	0	0	0	0	0	...	
192	Yemen	Asia	Western Asia	Developing regions	1	2	1	6	0	18	...	16
193	Zambia	Africa	Eastern Africa	Developing regions	11	17	11	7	16	9	...	9
194	Zimbabwe	Africa	Eastern Africa	Developing regions	72	114	102	44	32	29	...	61

195 rows × 39 columns

Question 1: Let's compare the number of immigrants from India and China from 1980 to 2013.

Step 1: Get the data set for China and India, and display dataframe.

In [462…

```
#Set 'Country' column as index
df_can.set_index('Country', inplace=True)
```

In [463…

```
### type your answer here
df_ChIn = df_can.loc[['China', 'India']]
df_ChIn
```

Out[463]:

Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	1986	...	2005
China	Asia	Eastern Asia	Developing regions	5123	6682	3308	1863	1527	1816	1960	...	42584
India	Asia	Southern Asia	Developing regions	8880	8670	8147	7338	5704	4211	7150	...	36210

2 rows × 38 columns

Step 2: Plot graph. We will explicitly specify line plot by passing in `kind` parameter to `plot()`.

In [464…
```
#Import matpolib for visualization
import matplotlib.pyplot as plt
```

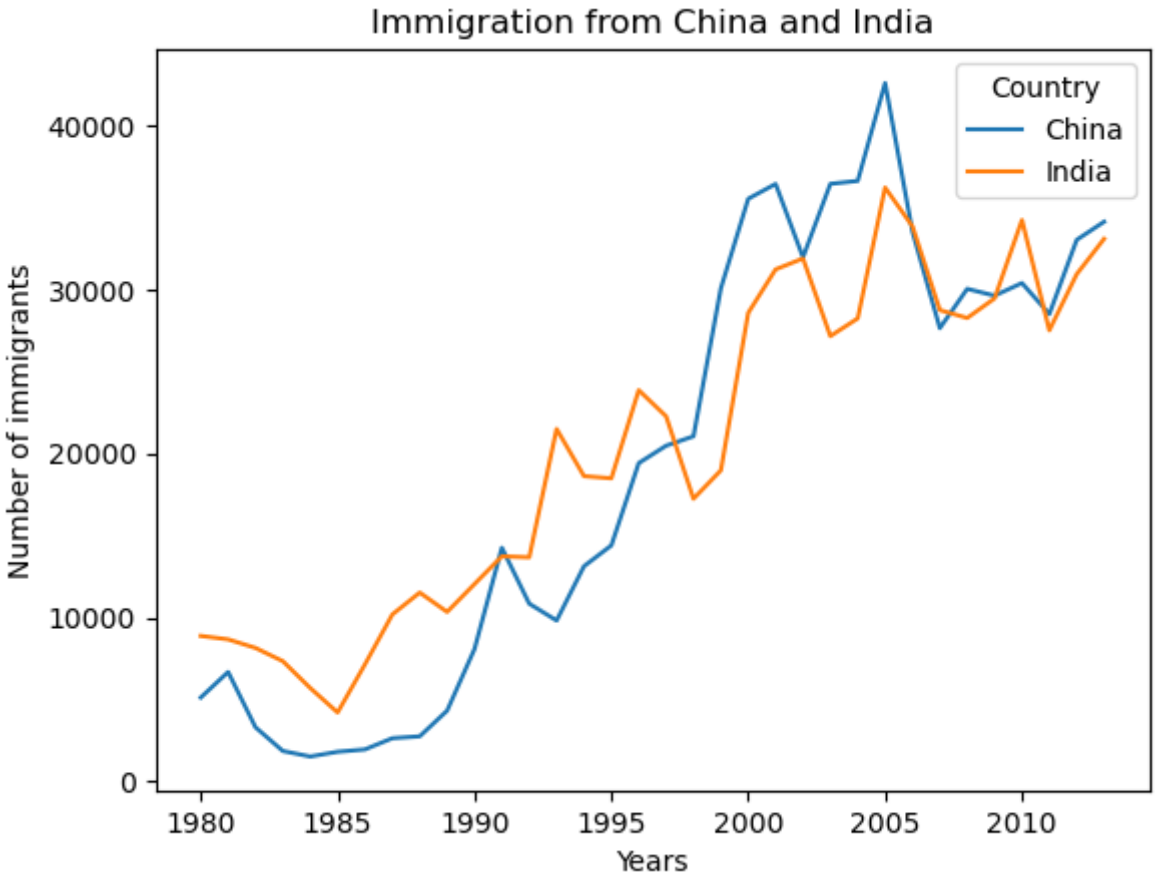
In [465…
```
#Change column data type from integer to string
df_can.columns = list(map(str, df_can.columns))
```

In [466…
```
# useful for plotting later on
years = list(map(str, range(1980, 2014)))
years
```

Out[466]:
```
['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']
```

In [467…

```
### type your answer here
# Retrieving immigration data from China and India
df_ChIn = df_can.loc[['China', 'India'], years]

df_ChIn = df_ChIn.transpose()
df_ChIn

# Change index data type to integer
df_ChIn.index = df_ChIn.index.map(int)

# Plot data in line plot form
df_ChIn.plot(kind='line')

plt.title('Immigration from China and India')
plt.ylabel('Number of immigrants')
plt.xlabel('Years')

# need this line to show the updates made to the figure
plt.show()
```



Question 2: Compare the trend of top 5 countries that contributed the most to immigration to Canada.

Step 1: Get the data set for top 5 countries

In [468…

```
### type your answer here
df_can.sort_values(['Total'], ascending=False, axis=0, inplace=True)

# get the top 5 entries
df_top5 = df_can.head()

# transpose the dataframe
df_top5 = df_top5[years].transpose()
```

```
df_top5.head()
```

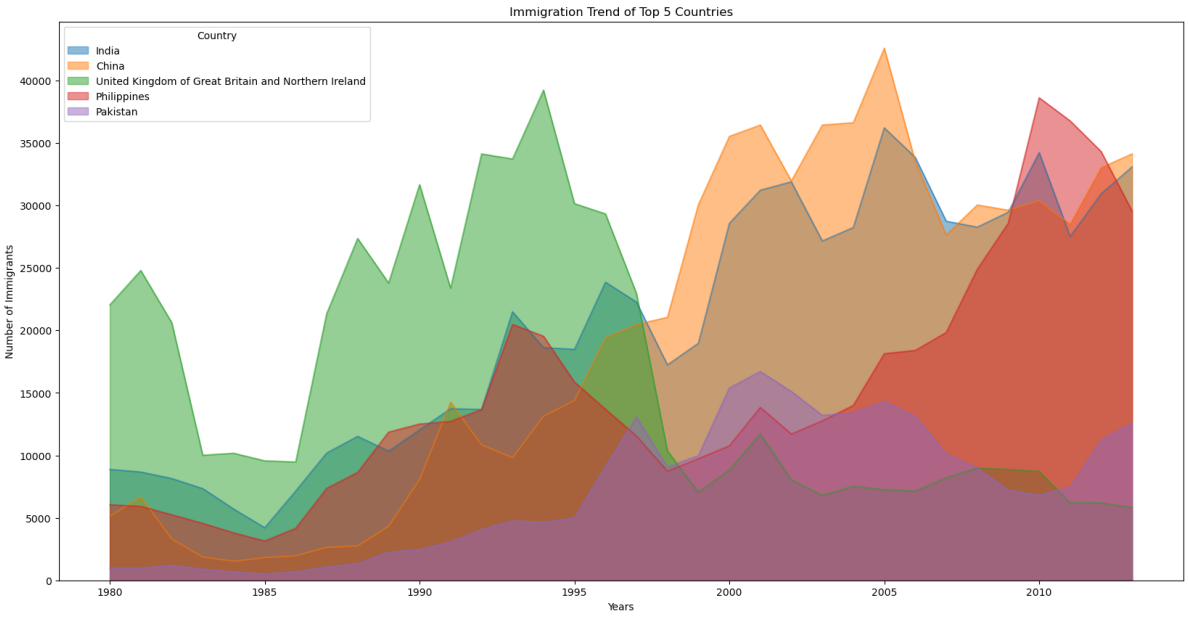
Out[468]:

Country	India	China	United Kingdom of Great Britain and Northern Ireland	Philippines	Pakistan
1980	8880	5123	22045	6051	978
1981	8670	6682	24796	5921	972
1982	8147	3308	20620	5249	1201
1983	7338	1863	10015	4562	900
1984	5704	1527	10170	3801	668

Step 2: Plot graph

In [469…

```
### type your answer here
df_top5.index = df_top5.index.map(int) # let's change the index values of df_top5 t
df_top5.plot(kind='area',
             stacked=False,
             figsize=(20, 10), # pass a tuple (x, y) size
             )

plt.title('Immigration Trend of Top 5 Countries')
plt.ylabel('Number of Immigrants')
plt.xlabel('Years')

plt.show()
```



Question 3: Create an unstacked area plot of the 5 countries that contributed the least to immigration to Canada **from** 1980 to 2013. Use a transparency value of 0.55.

In [470…

```
### type your answer here
df_can.sort_values(['Total'], ascending=True, axis=0, inplace=True)

# get the least 5 entries
df_Tail5 = df_can.head()

# transpose the dataframe
df_Tail5 = df_Tail5[years].transpose()
```

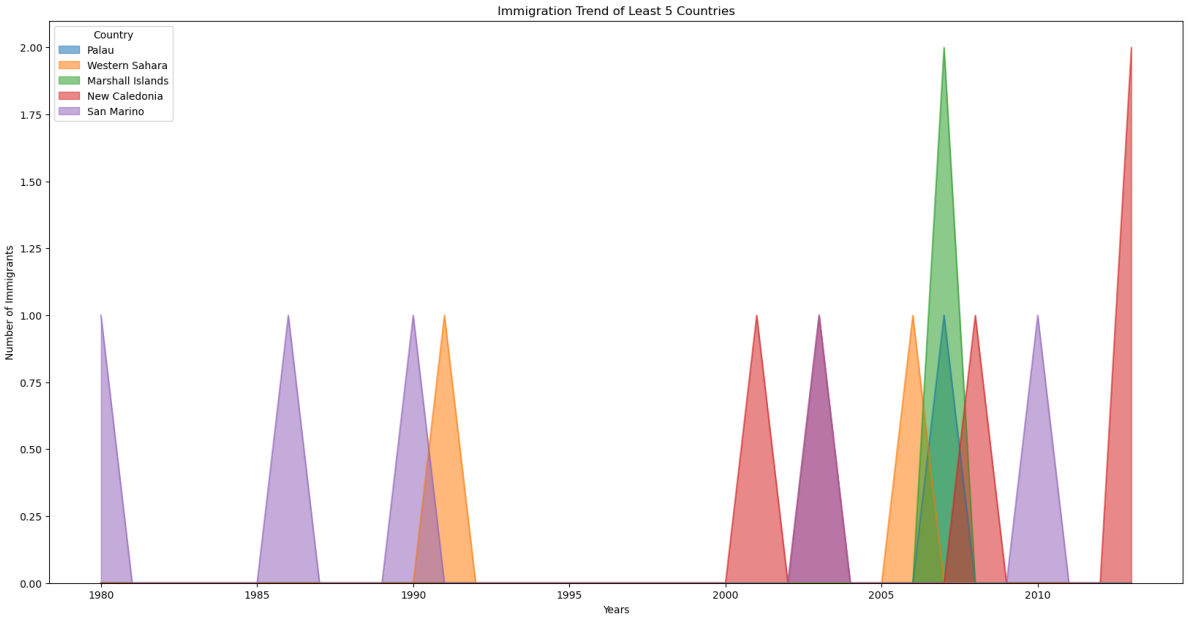
```
df_Tail5.head()
```

Out[470]:

Country	Palau	Western Sahara	Marshall Islands	New Caledonia	San Marino
1980	0	0	0	0	1
1981	0	0	0	0	0
1982	0	0	0	0	0
1983	0	0	0	0	0
1984	0	0	0	0	0

In [471…

```
df_Tail5.plot(kind='area',
              alpha=0.55, # 0-1, default value a= 0.5
              stacked=False,
              figsize=(20, 10),
             )

plt.title('Immigration Trend of Least 5 Countries')
plt.ylabel('Number of Immigrants')
plt.xlabel('Years')

plt.show()
```



Question 4: Display the immigration distribution for Greece, Albania, and Bulgaria for years 1980 - 2013? Use an overlapping plot with 15 bins and a transparency value of 0.35.

In [472…

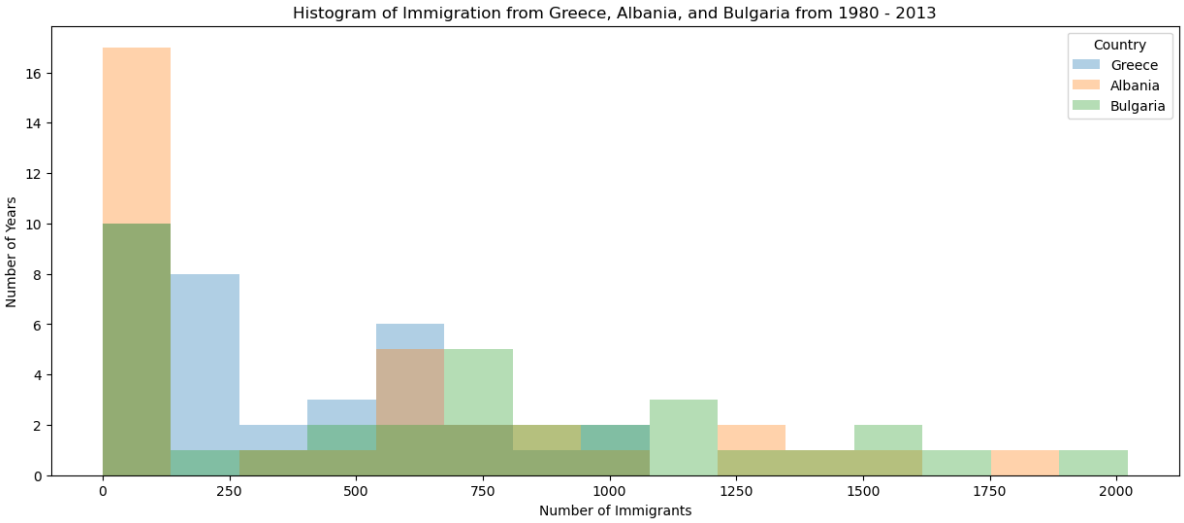
```
### type your answer here
df_t = df_can.loc[['Greece', 'Albania', 'Bulgaria'], years].transpose()
df_t.head()
```

Out[472]:

Country	Greece	Albania	Bulgaria
1980	1065	1	24
1981	953	0	20
1982	897	0	12
1983	633	0	33
1984	580	0	11

In [473…

```
# generate histogram
df_t.plot(kind='hist',alpha=0.35 ,figsize=(15, 6), bins=15)

plt.title('Histogram of Immigration from Greece, Albania, and Bulgaria from 1980 -
plt.ylabel('Number of Years')
plt.xlabel('Number of Immigrants')

plt.show()
```



Question 5: Create a *horizontal* bar plot showing the *total* number of immigrants to Canada from the top 15 countries, for the period 1980 - 2013. Label each country with the total immigrant count.

Step 1: Get the data pertaining to the top 15 countries.

In [474…

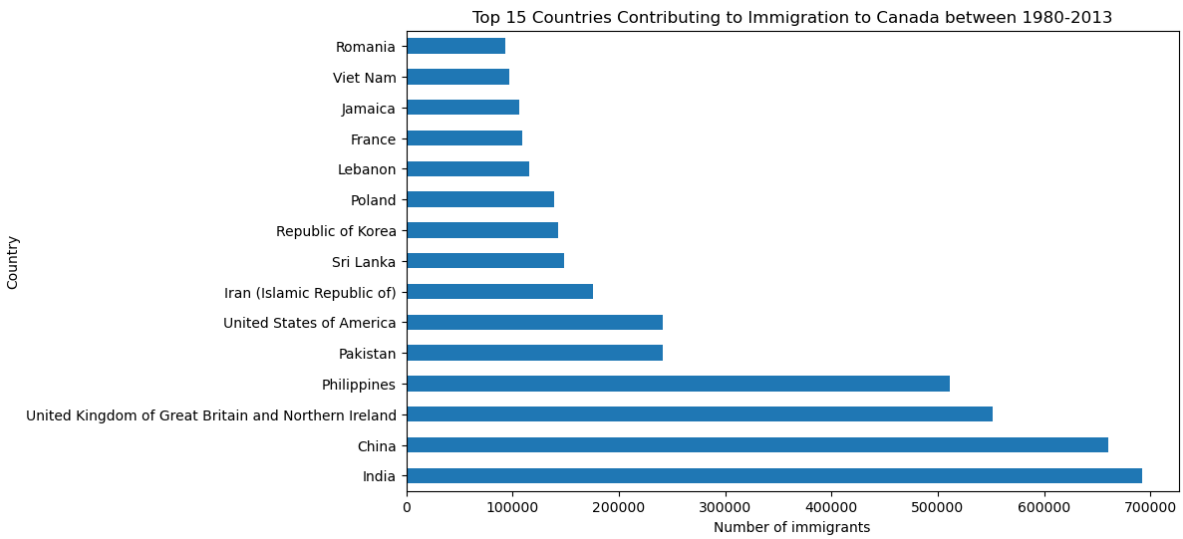
```
### type your answer here
df_can.sort_values(['Total'], ascending=False, axis=0, inplace=True)

# get the top 15 entries
df_top15 = df_can.head(15)

# transpose the dataframe
df_top15 = df_top15['Total'].transpose()

df_top15
```

Out[474]:
```
Country
India                                                  691904
China                                                  659962
United Kingdom of Great Britain and Northern Ireland   551500
Philippines                                            511391
Pakistan                                               241600
United States of America                               241122
Iran (Islamic Republic of)                             175923
Sri Lanka                                              148358
Republic of Korea                                      142581
Poland                                                 139241
Lebanon                                                115359
France                                                 109091
Jamaica                                                106431
Viet Nam                                                97146
Romania                                                 93585
Name: Total, dtype: int64
```

Step 2: Plot data:

1. Use `kind='barh'` to generate a bar chart with horizontal bars.
2. Make sure to choose a good size for the plot and to label your axes and to give the plot a title.

In [475…
```
### type your answer here
df_top15.plot(kind='barh', figsize=(10, 6))

plt.xlabel('Number of immigrants') # add to x-label to the plot
plt.ylabel('Country') # add y-label to the plot
plt.title('Top 15 Countries Contributing to Immigration to Canada between 1980-201

plt.show()
```



Thank you for completing this lab!
