Immigration to Canada from 1980 to 2013

Will use Matplotlib and Seaborn library for Data Visualization

Add required libraries

In [1]:

```
import numpy as np
import pandas as pd

#!conda install -c anaconda xlrd --yes
```

In [3]:

```
df_can = pd.read_excel('https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DV0101EN/labs/Data_Files/Canada.xlsx',
                       sheet_name='Canada by Citizenship',
                       skiprows=range(20),
                       skipfooter=2)

print ('Data read into a pandas dataframe!')
```

```
Data read into a pandas dataframe!
```

In [4]:

```
df_can.head()
```

Out[4]:

	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
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions

5 rows × 43 columns

Clean the data set to remove a few unnecessary columns.

In [5]:

```
df_can.drop(['AREA','REG','DEV','Type','Coverage'], axis=1, inplace=True)
df_can.rename(columns={'OdName':'Country', 'AreaName':'Continent', 'RegName':'Region'},
inplace=True)
df_can.head(2)
```

Out[5]:

	Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	...	200
0	Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	297
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	145

2 rows × 38 columns

Adding a 'Total' column that sums up the total immigrants by country over the entire period 1980 - 2013, as follows:

In [6]:

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

In [7]:

```
print('data dimensions:', df_can.shape)
print(df_can.columns)
df_can.head(2)
```

```
data dimensions: (195, 39)
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,     'Total'],
      dtype='object')
```

Out[7]:

	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 rows × 39 columns

Visualizing Data using Matplotlib

In [8]:

```
# Import libraries
%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt
```

Line Plots (Series/Dataframe)

In [9]:

```
mpl.style.use(['ggplot'])
```

In [10]:

```
years = list(map(str, range(1980, 2014)))

df_can.columns = list(map(str, df_can.columns))
df_can.set_index('Country', inplace=True)
```

In [11]:

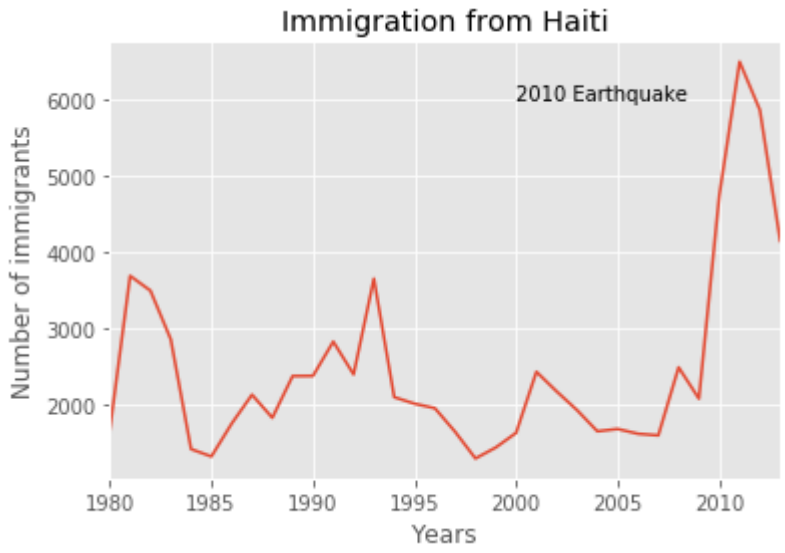
```
haiti = df_can.loc['Haiti', years] # passing in years 1980 - 2013 to exclude the 'tota
l' column
haiti.head()
```

Out[11]:

```
1980    1666
1981    3692
1982    3498
1983    2860
1984    1418
Name: Haiti, dtype: object
```

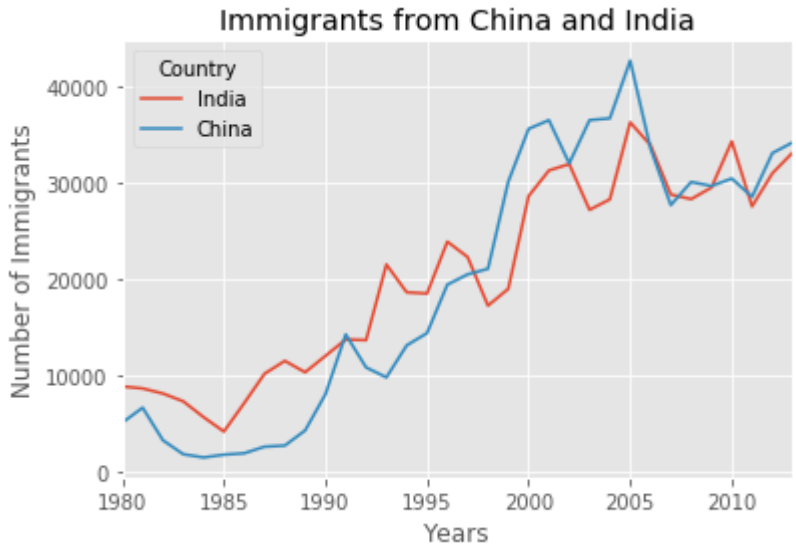
In [12]:

```
haiti.index = haiti.index.map(int) # let's change the index values of Haiti to type int
eger for plotting
haiti.plot(kind='line')

plt.title('Immigration from Haiti')
plt.ylabel('Number of immigrants')
plt.xlabel('Years')
plt.text(2000, 6000, '2010 Earthquake')

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



In [13]:

```
df_CI = df_can.loc[['India', 'China'], years]
df_CI = df_CI.transpose()

df_CI.index = df_CI.index.map(int)
df_CI.plot(kind='line')

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

plt.show()
```



In [14]:

```
df_top5 = df_can.sort_values(by='Total', ascending=False, axis=0, inplace=False)
df_top5.head(2)
```

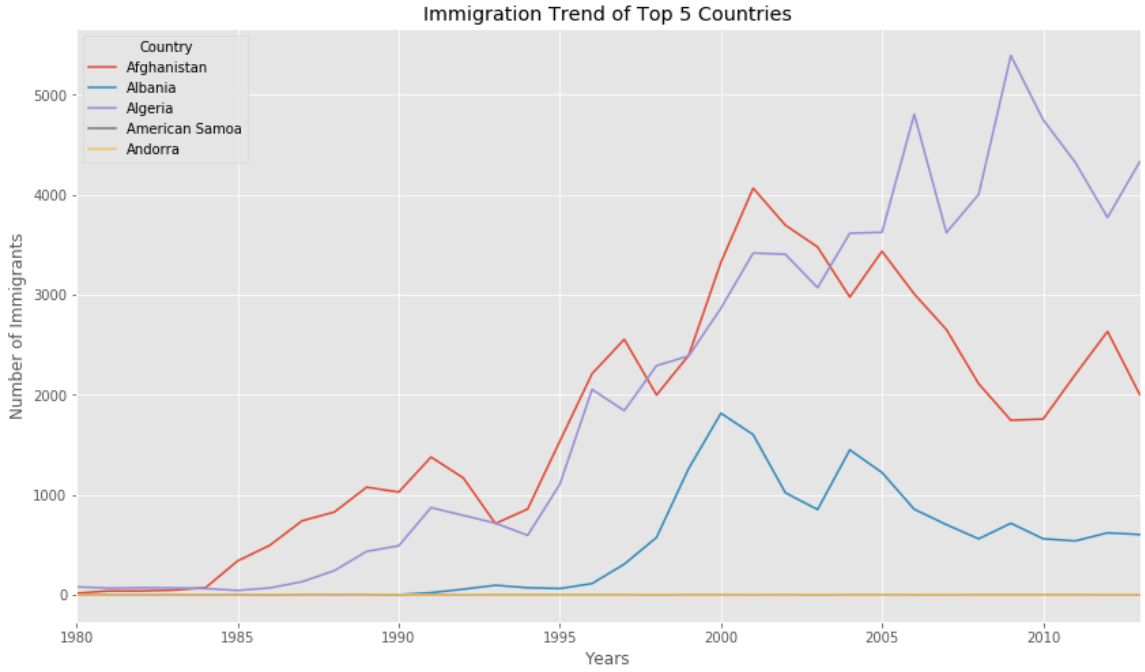
Out[14]:

Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	1986	...	20
India	Asia	Southern Asia	Developing regions	8880	8670	8147	7338	5704	4211	7150	...	362
China	Asia	Eastern Asia	Developing regions	5123	6682	3308	1863	1527	1816	1960	...	425

2 rows × 38 columns

In [15]:

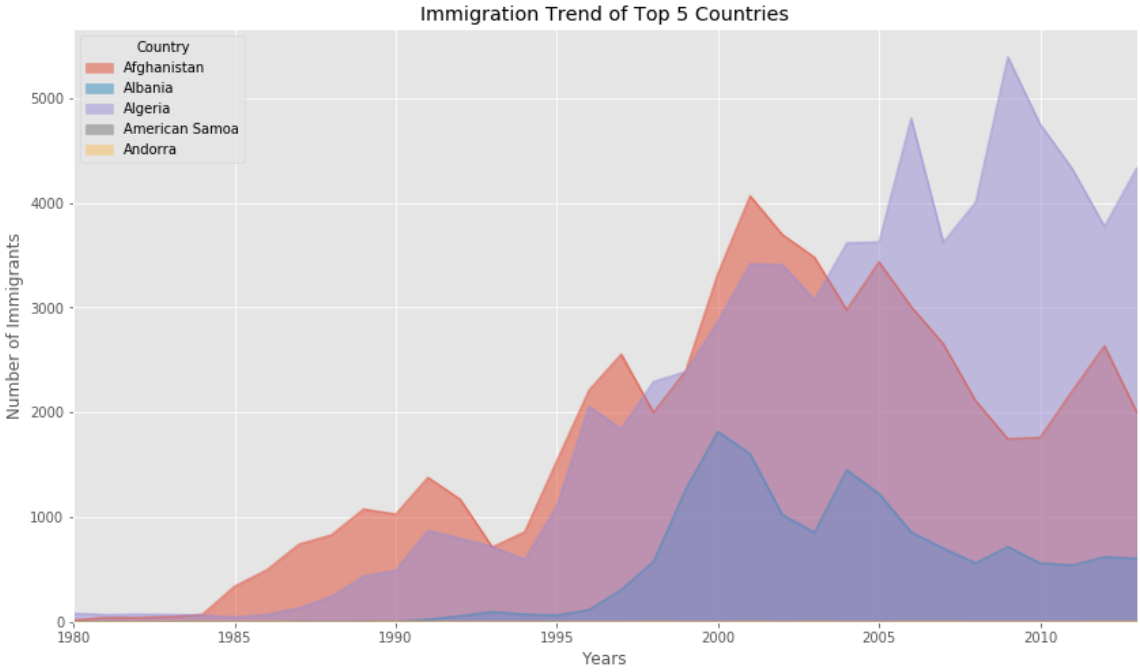
```
df_top5 = df_can.head(5)

df_top5 = df_top5[years].transpose()
df_top5.index = df_top5.index.map(int)


df_top5.plot(kind='line',figsize=(14, 8))

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

plt.show()
```



Area Plots

In [16]:

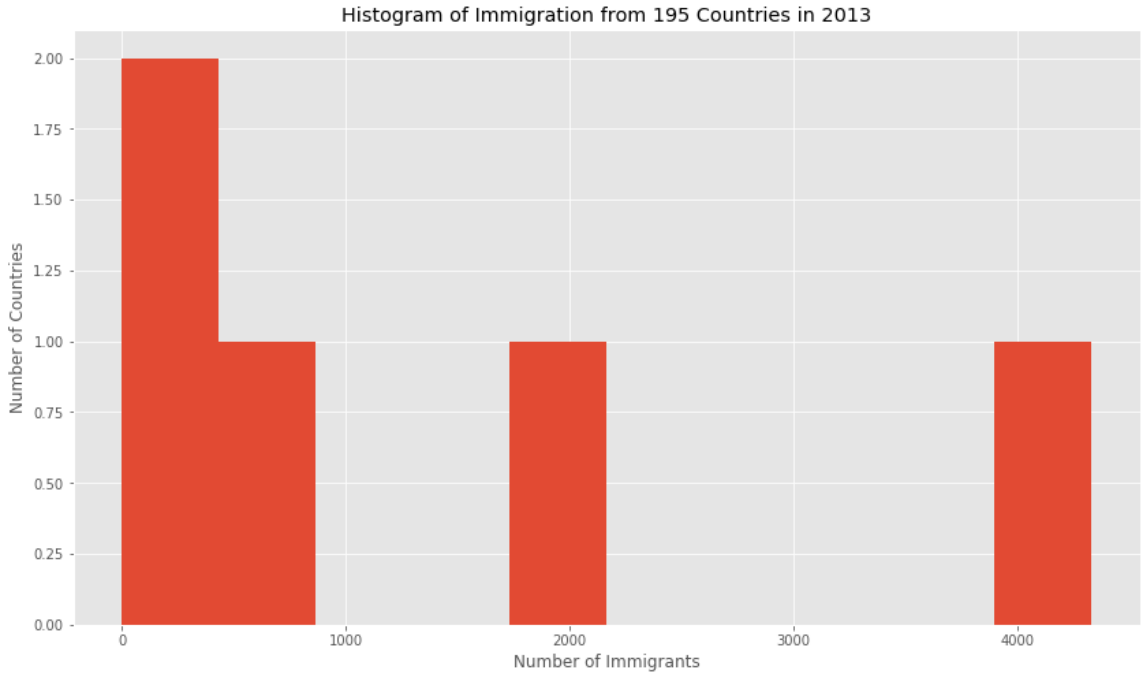
```
df_top5.plot(kind='area',
             alpha=0.5,
             stacked=False,
             figsize=(14, 8), # pass a tuple (x, y) size
             )

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

plt.show()
```



Histograms

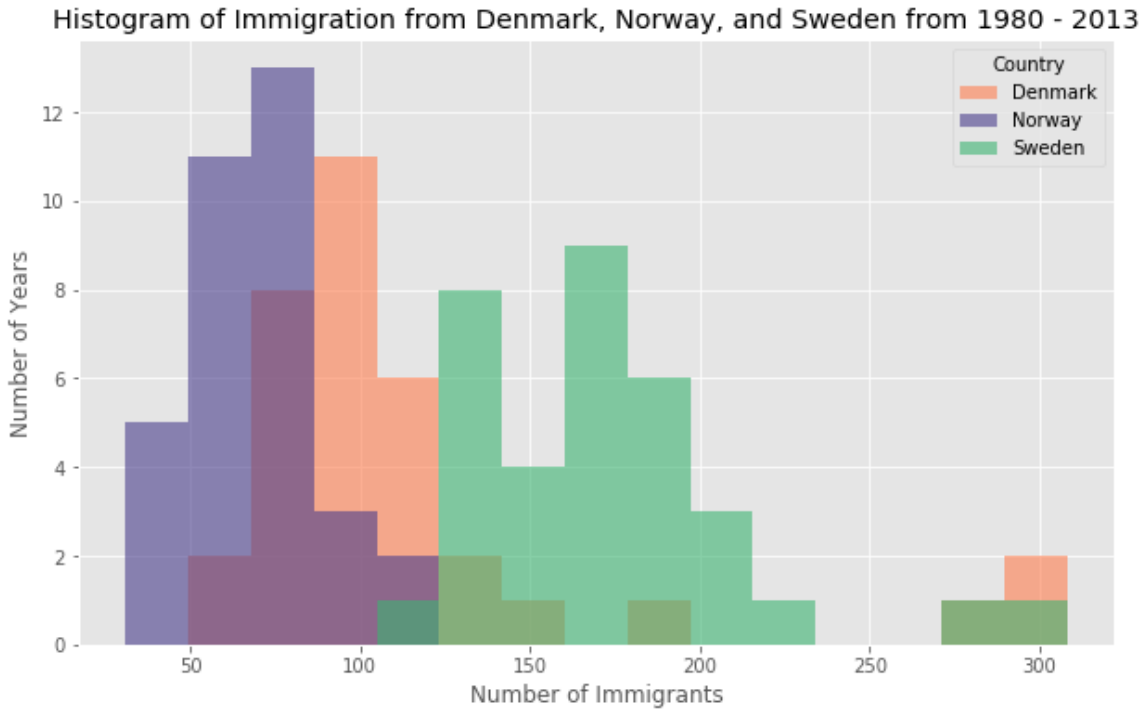
In [17]:

```
count, bin_edges = np.histogram(df_can['2013'])

df_can['2013'].head().plot(kind='hist',figsize=(14, 8),bins=10)

plt.title('Histogram of Immigration from 195 Countries in 2013')
plt.ylabel('Number of Countries')
plt.xlabel('Number of Immigrants')

plt.show()
```



In [18]:

```
df_t = df_can.loc[['Denmark', 'Norway', 'Sweden'], years].transpose()


# un-stacked histogram
df_t.plot(kind ='hist',
          figsize=(10, 6),
          bins=15,
          alpha=0.6,
          color=['coral', 'darkslateblue', 'mediumseagreen']
         )

plt.title('Histogram of Immigration from Denmark, Norway, and Sweden from 1980 - 2013')
plt.ylabel('Number of Years')
plt.xlabel('Number of Immigrants')

plt.show()
```



Bar Charts

In [19]:

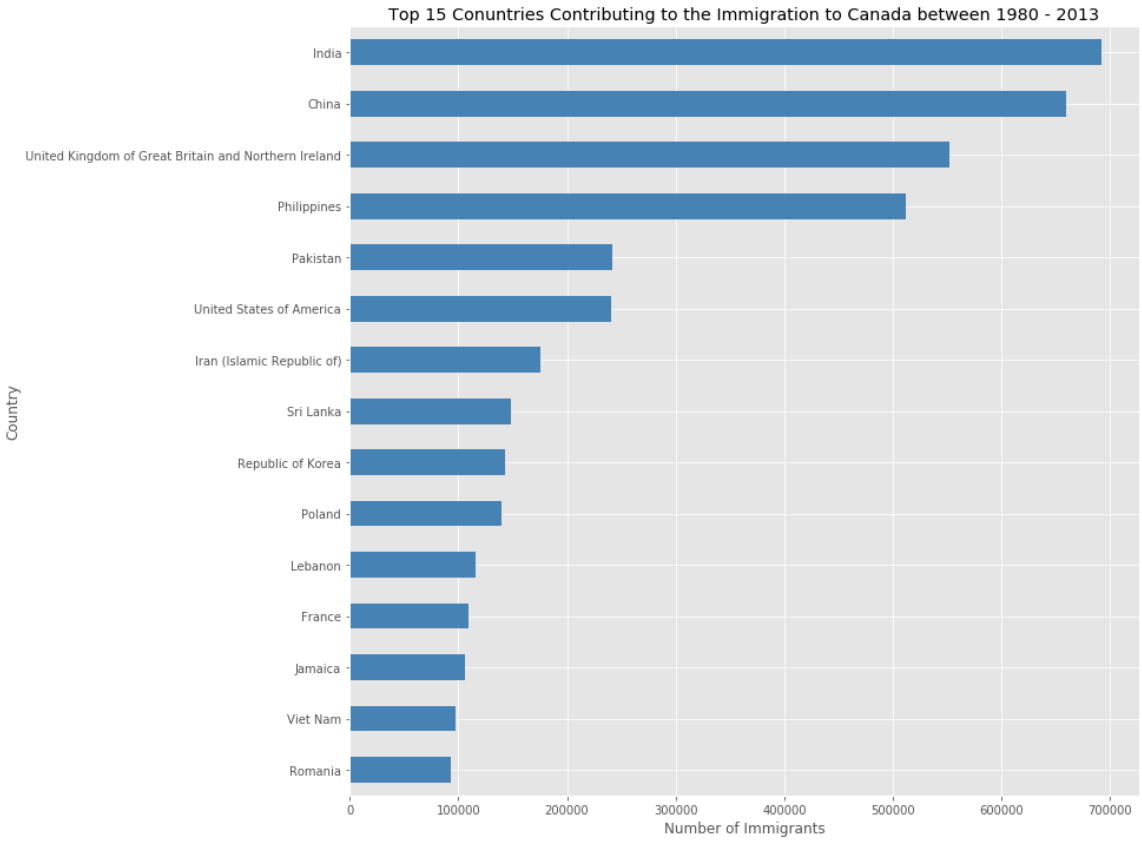
```
df_iceland = df_can.loc['Iceland', years]
df_iceland.plot(kind='bar', figsize=(10, 6))

plt.xlabel('Year')
plt.ylabel('Number of immigrants')
plt.title('Icelandic immigrants to Canada from 1980 to 2013')

plt.show()
```

In [20]:

```
df_top = df_can.sort_values(by='Total', ascending=True, inplace=False)

df_top15 = df_top['Total'].tail(15)

df_top15.plot(kind='barh', figsize=(12, 12), color='steelblue')

plt.xlabel('Number of Immigrants')
plt.title('Top 15 Conuntries Contributing to the Immigration to Canada between 1980 - 2
013')

for index, value in enumerate(df_top15):
    label = format(int(value), ',')

df_top15.head()
```

Out[20]:

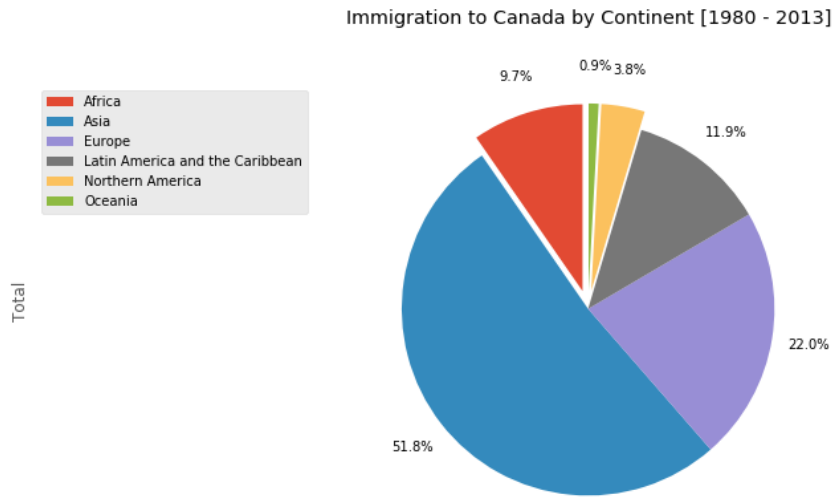
```
Country
Romania      93585
Viet Nam     97146
Jamaica     106431
France      109091
Lebanon     115359
Name: Total, dtype: int64
```



Pie Chart

In [27]:

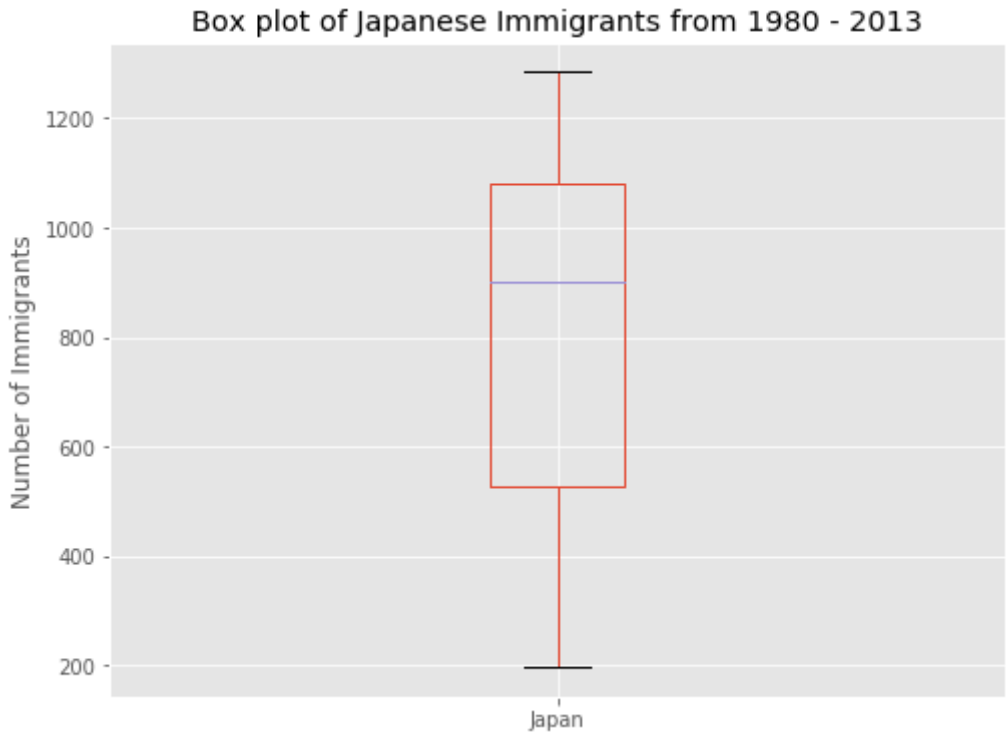
```
df_continents = df_can.groupby('Continent', axis=0).sum()

explode_list = [0.1, 0, 0, 0, 0.1, 0.1] # ratio for each continent with which to offset
each wedge.

df_continents['Total'].plot(kind='pie',
                            figsize=(15, 6),
                            autopct='%1.1f%%',
                            startangle=90,
                            labels=None,         # turn off labels on pie chart
                            pctdistance=1.2,
                            explode=explode_list # 'explode' lowest 3 continents
                            )

# scale the title up by 12% to match pctdistance
plt.title('Immigration to Canada by Continent [1980 - 2013]', y=1.12)

plt.axis('equal')

# add legend
plt.legend(labels=df_continents.index, loc='upper left')

plt.show()
```



Box Plots

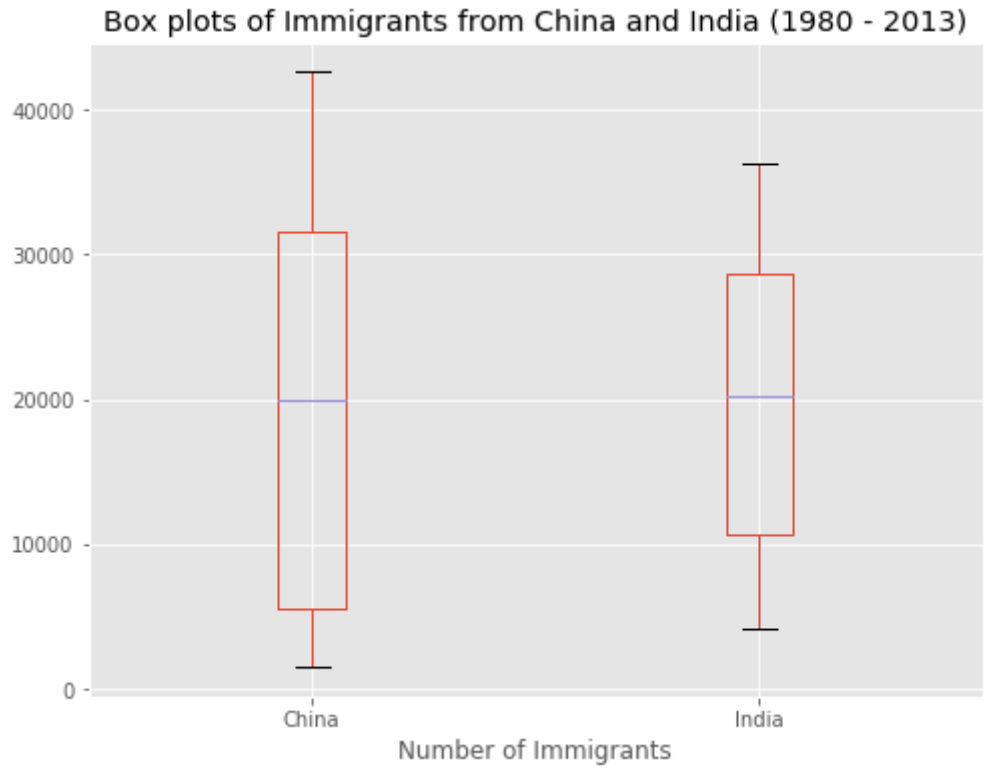
In [23]:

```
df_japan = df_can.loc[['Japan'], years].transpose()
df_japan.plot(kind='box', figsize=(8, 6))

plt.title('Box plot of Japanese Immigrants from 1980 - 2013')
plt.ylabel('Number of Immigrants')

plt.show()
```



In [25]:

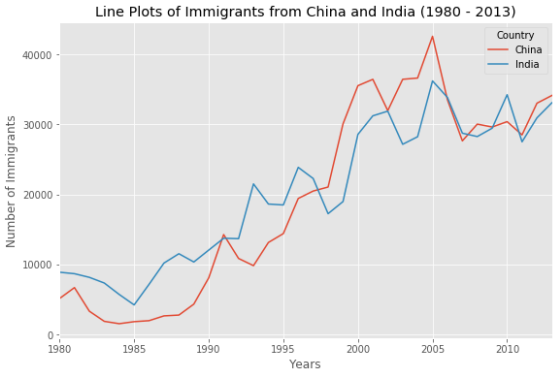
```
df_CI= df_can.loc[['China', 'India'], years].transpose()
df_CI.plot(kind='box', figsize=(8, 6))

plt.title('Box plots of Immigrants from China and India (1980 - 2013)')
plt.xlabel('Number of Immigrants')

plt.show()
```



Subplots

In [33]:

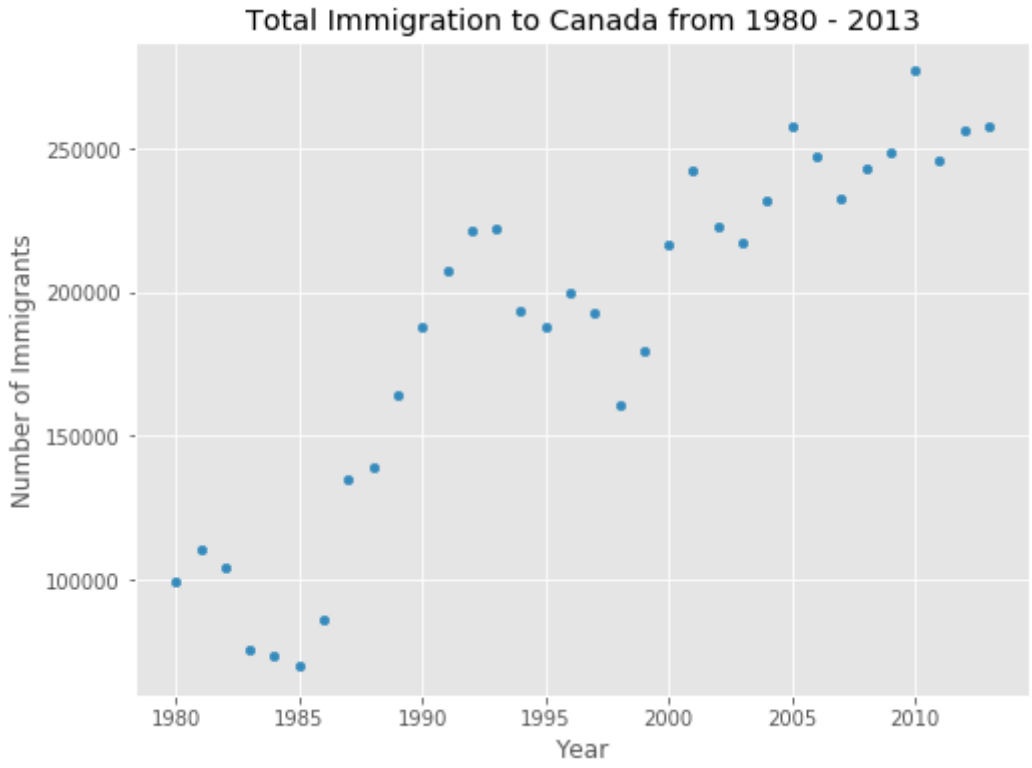
```
fig = plt.figure()

ax0 = fig.add_subplot(1, 2, 1)
ax1 = fig.add_subplot(1, 2, 2)

# Subplot 1: Box plot
df_CI.plot(kind='box', figsize=(20, 6), ax=ax0)
ax0.set_title('Box Plots of Immigrants from China and India (1980 - 2013)')
ax0.set_xlabel('Number of Immigrants')
ax0.set_ylabel('Countries')

# Subplot 2: Line plot
df_CI.plot(kind='line', figsize=(20, 6), ax=ax1)
ax1.set_title ('Line Plots of Immigrants from China and India (1980 - 2013)')
ax1.set_ylabel('Number of Immigrants')
ax1.set_xlabel('Years')

plt.show()
```



Scatter Plots

In [35]:

```
df_total = pd.DataFrame(df_can[years].sum(axis=0))

df_total.index = map(int, df_total.index)
df_total.reset_index(inplace = True)

df_total.columns = ['Year', 'Total']

df_total.plot(kind='scatter', x='Year', y='Total', figsize=(8, 6))

plt.title('Total Immigration to Canada from 1980 - 2013')
plt.xlabel('Year')
plt.ylabel('Number of Immigrants')

plt.show()
```



Will plot a linear line of best fit, and use it to predict the number of immigrants in 2015

In [37]:

```
x = df_total['Year']      # year on x-axis
y = df_total['Total']     # total on y-axis

fit = np.polyfit(x, y, deg=1)
fit
```

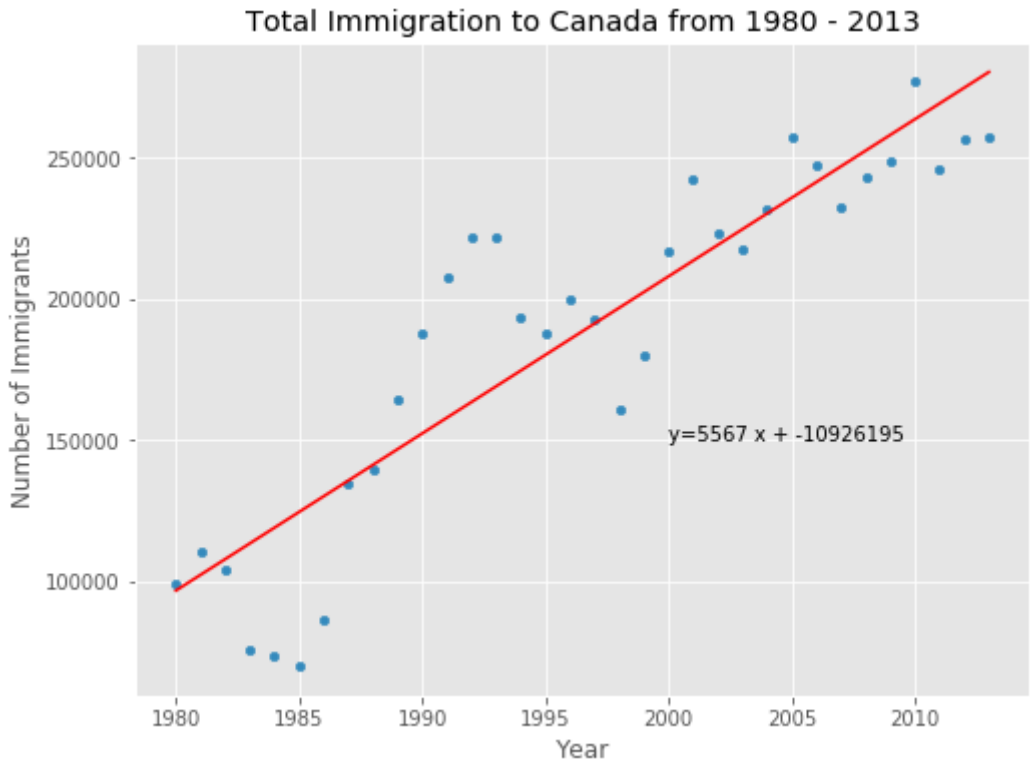
Out[37]:

```
array([ 5.56709228e+03, -1.09261952e+07])
```

Since we are plotting a linear regression `y= a*x + b` , our output has 2 elements `[5.56709228e+03, -1.09261952e+07]` with the the slope in position 0 and intercept in position 1.

In [39]:

```
df_total.plot(kind='scatter', x='Year', y='Total', figsize=(8, 6))

plt.title('Total Immigration to Canada from 1980 - 2013')
plt.xlabel('Year')
plt.ylabel('Number of Immigrants')

plt.plot(x, fit[0] * x + fit[1], color='red')
plt.annotate('y={0:.0f} x + {1:.0f}'.format(fit[0], fit[1]), xy=(2000, 150000))

plt.show()

'No. Immigrants = {0:.0f} * Year + {1:.0f}'.format(fit[0], fit[1])
```



Out[39]:

```
'No. Immigrants = 5567 * Year + -10926195'
```

No. Immigrants = 5567 * 2015 - 10926195
No. Immigrants = 291,310

In [ ]:

```
```

In [43]:

```
df_countries = df_can.loc[['Denmark', 'Norway', 'Sweden'], years].transpose()
df_total = pd.DataFrame(df_countries.sum(axis=1))

df_total.reset_index(inplace=True)
df_total.columns = ['year', 'total']
df_total['year'] = df_total['year'].astype(int)

df_total.plot(kind='scatter', x='year', y='total', figsize=(10, 6))
plt.title('Immigration from Denmark, Norway, and Sweden to Canada from 1980 - 2013')
plt.xlabel('Year')
plt.ylabel('Number of Immigrants')

plt.show()
```