Data Visualization

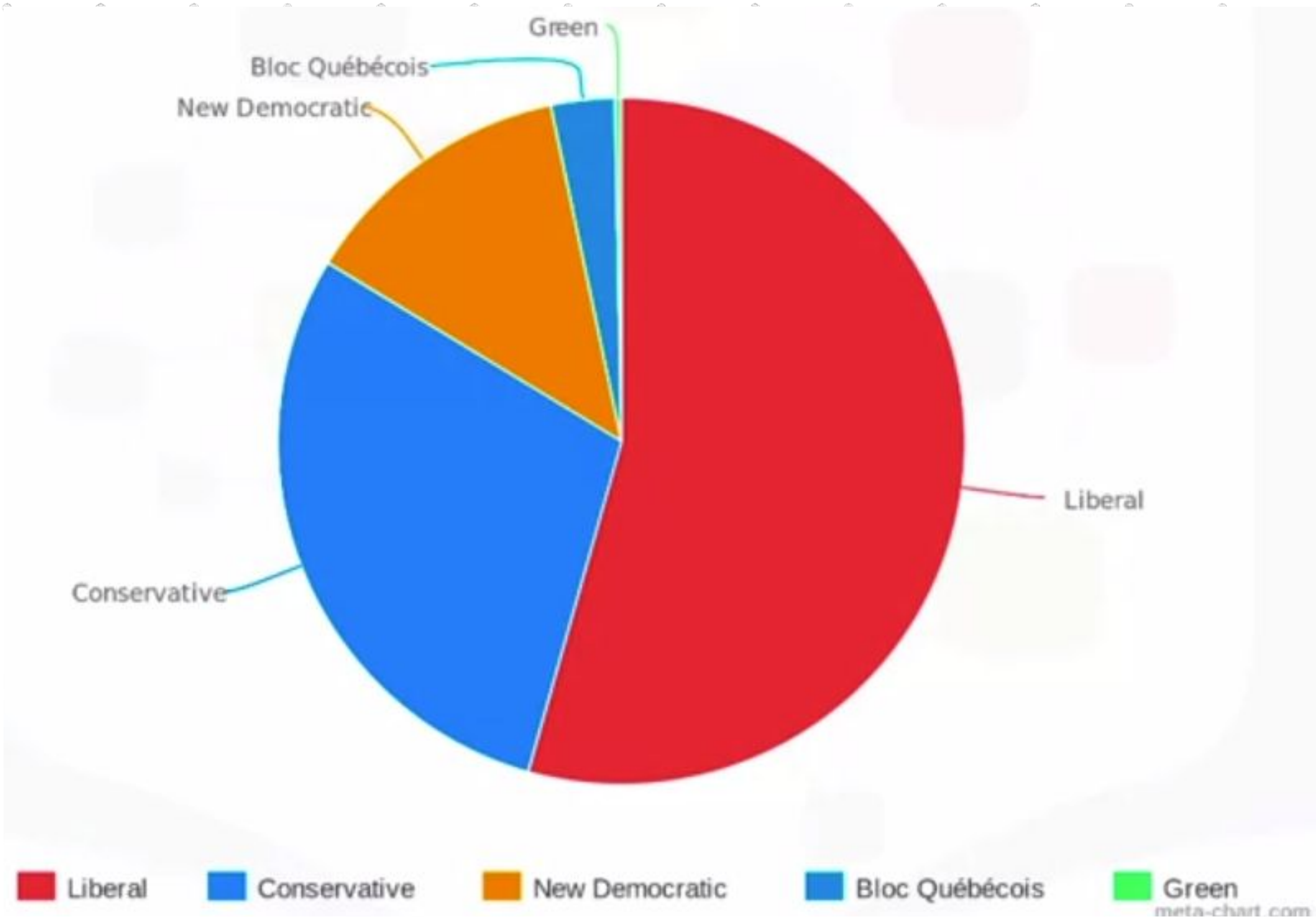
Agenda

1. Pie Chart
2. Boxplot
3. Scatter Plot
4. Area Plot - Complete Example

Pie Chart

Pie Chart

- is a circular statistical graphic
  - divided into slices
  - to illustrate numerical proportion



- **Example**
  - the Canadian federal election in 2015
  - were Liberals in red won more than 50% of the seats in the House of Commons

- There are some very vocal opponents to the use of pie charts
  - Most argue that pie charts fail to accurately display data with any consistency

Pie Chart Example - Cell 1

In [1]:

```
import pandas as pd

df = pd.read_csv('canada-mig-dataset.csv')

df.head()
```

Out[1]:

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2004	2005	2006	2007	2008	2009	2010	2011	2012
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	2978	3436	3009	2652	2111	1746	1758	2203	2635
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1450	1223	856	702	560	716	561	539	620
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3616	3626	4807	3623	4005	5393	4752	4325	3774
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	0	1	0	0	0	0	0	0
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	0	1	1	0	0	0	0	1

5 rows × 43 columns

Pie Chart Example - Cell 2

```
In [2]: df0['Total'] = df0.iloc[:, 9:43].sum(axis=1)
        df0.head()
```

Out[2]:

Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2005	2006	2007	2008	2009	2010	2011	2012	2013	Total
nigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	3436	3009	2652	2111	1746	1758	2203	2635	2004	58639
nigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1223	856	702	560	716	561	539	620	603	15699
nigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3626	4807	3623	4005	5393	4752	4325	3774	4331	69439
nigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	1	0	0	0	0	0	0	0	6
nigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	1	1	0	0	0	0	1	1	15

× 44 columns

Pie Chart Example - Cell 3

In [3]:

```
df1 = df0.groupby('AreaName', axis = 0).sum()
df1.head()
```

Out[3]:

	AREA	REG	DEV	1980	1981	1982	1983	1984	1985	1986	...	2005	2006	2007	2008	2009	2010	2011	2012
AreaName																			
Africa	48762	49242	48708	3951	4363	3819	2671	2639	2650	3782	...	27523	29188	28284	29890	34534	40892	35441	38083
Asia	45815	109147	44197	31025	34314	30214	24696	27274	23850	28739	...	159253	149054	133459	139894	141434	163845	146894	152218
Europe	39044	39754	38743	39760	44802	42720	24638	22287	20844	24370	...	35955	33053	33495	34692	35078	33425	26778	29177
Latin America and the Caribbean	29832	30395	29766	13081	15215	16769	15427	13678	15171	21179	...	24747	24676	26011	26547	26867	28818	27856	27173
Northern America	1810	1810	1802	9378	10030	9074	7100	6661	6543	7074	...	8394	9613	9463	10190	8995	8142	7677	7892

5 rows × 38 columns

Pie Chart Example - Cell 3 (showing Total)

In [3]:

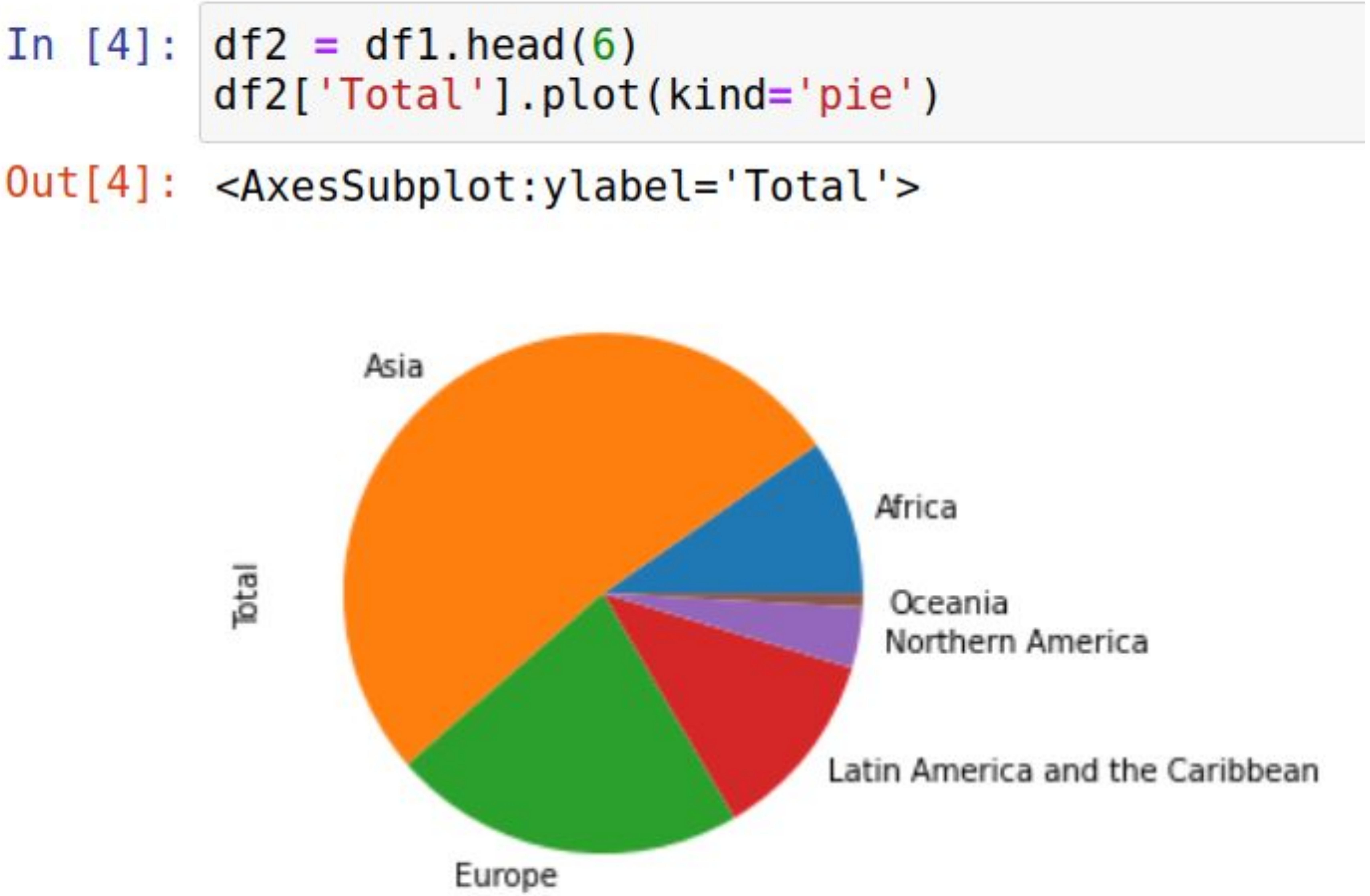
```
df1 = df0.groupby('AreaName', axis = 0).sum()
df1.head()
```

Out[3]:

A	REG	DEV	1980	1981	1982	1983	1984	1985	1986	...	2005	2006	2007	2008	2009	2010	2011	2012	2013	Total
2	49242	48708	3951	4363	3819	2671	2639	2650	3782	...	27523	29188	28284	29890	34534	40892	35441	38083	38543	618948
5	109147	44197	31025	34314	30214	24696	27274	23850	28739	...	159253	149054	133459	139894	141434	163845	146894	152218	155075	3317794
4	39754	38743	39760	44802	42720	24638	22287	20844	24370	...	35955	33053	33495	34692	35078	33425	26778	29177	28691	1410947
2	30395	29766	13081	15215	16769	15427	13678	15171	21179	...	24747	24676	26011	26547	26867	28818	27856	27173	24950	765148
0	1810	1802	9378	10030	9074	7100	6661	6543	7074	...	8394	9613	9463	10190	8995	8142	7677	7892	8503	241142

mns

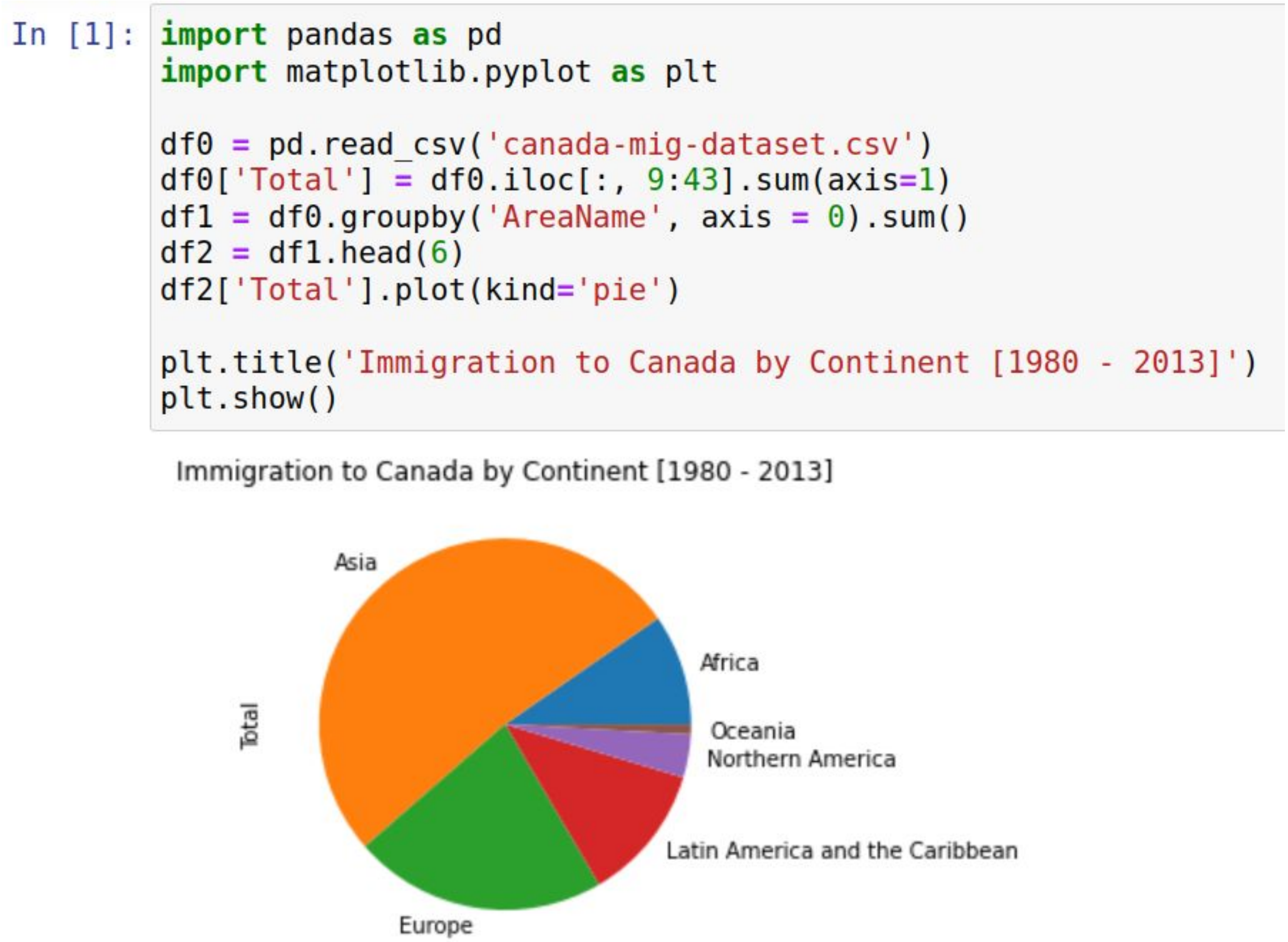
Pie Chart Example - Cell 4



```
In [4]: df2 = df1.head(6)
        df2['Total'].plot(kind='pie')

Out[4]: <AxesSubplot:ylabel='Total'>
```

Pie Chart - Complete Example



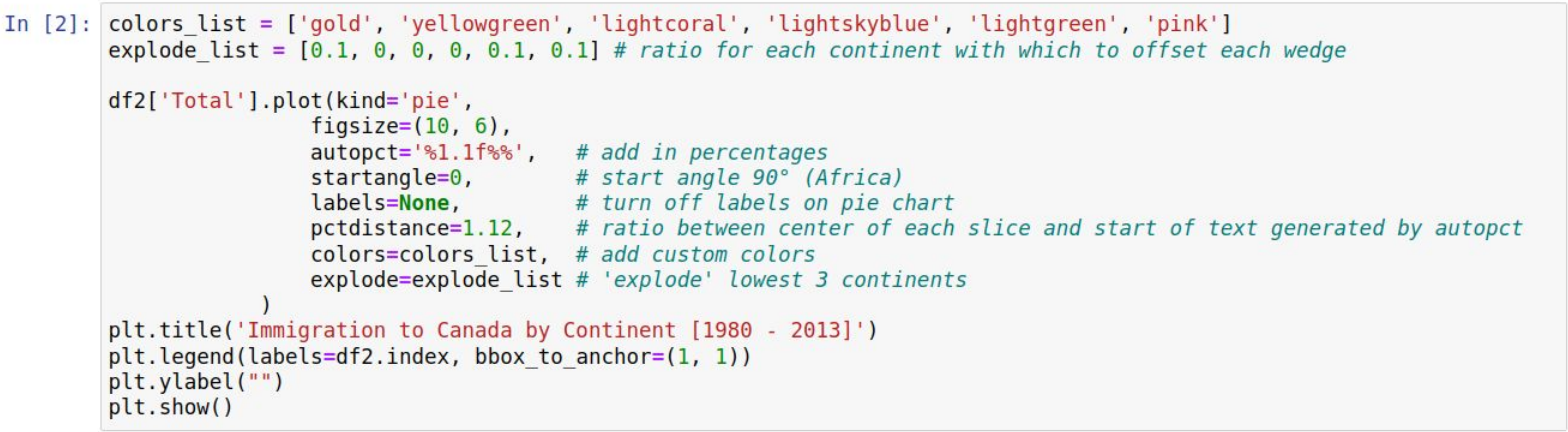
```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt

        df0 = pd.read_csv('canada-mig-dataset.csv')
        df0['Total'] = df0.iloc[:, 9:43].sum(axis=1)
        df1 = df0.groupby('AreaName', axis = 0).sum()
        df2 = df1.head(6)
        df2['Total'].plot(kind='pie')

        plt.title('Immigration to Canada by Continent [1980 - 2013]')
        plt.show()
```

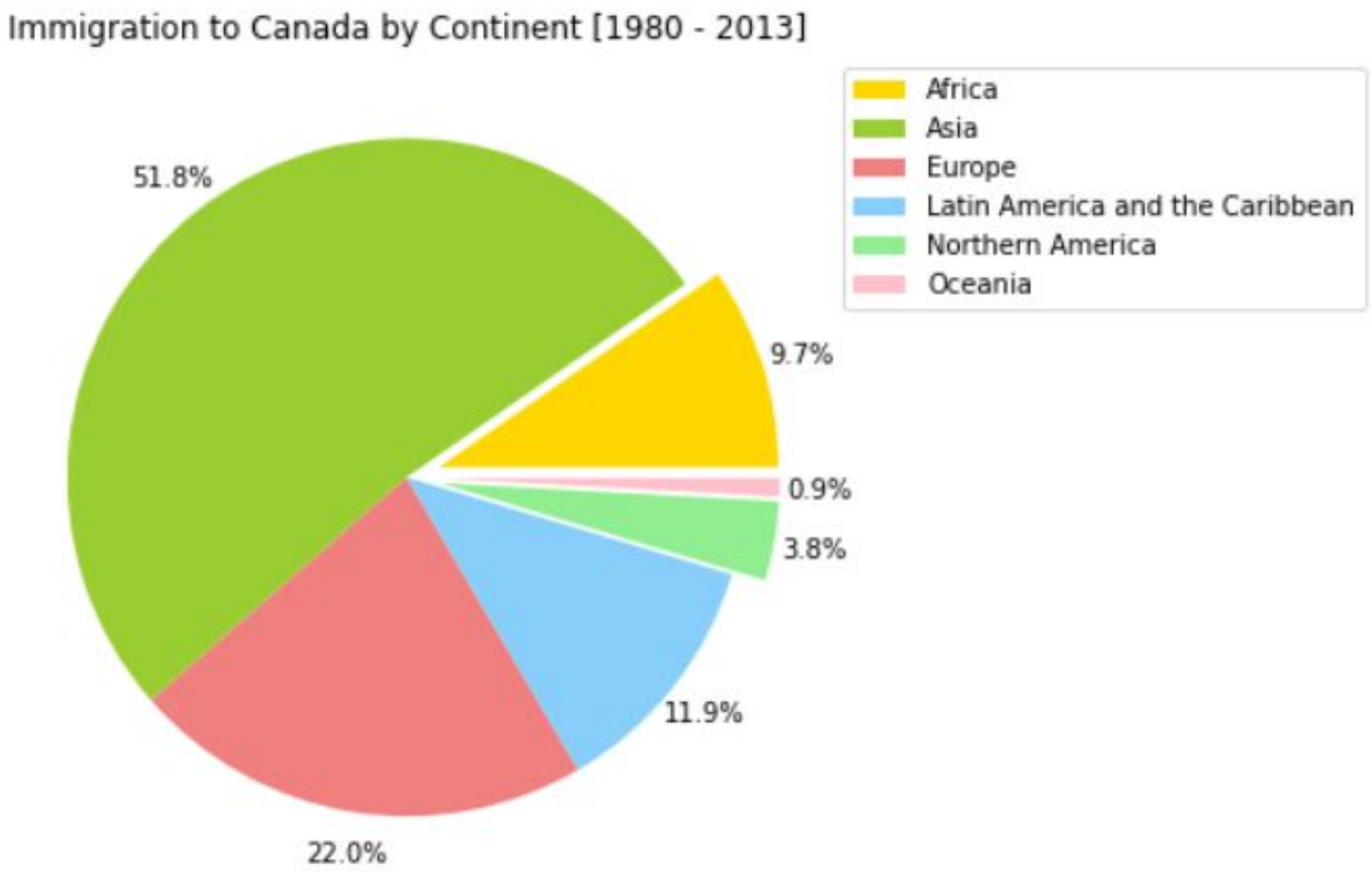
Immigration to Canada by Continent [1980 - 2013]

Pie Chart Example - Enhancement



```
In [2]: colors_list = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue', 'lightgreen', 'pink']
        explode_list = [0.1, 0, 0, 0, 0.1, 0.1] # ratio for each continent with which to offset each wedge

        df2['Total'].plot(kind='pie',
                          figsize=(10, 6),
                          autopct='%1.1f%%',   # add in percentages
                          startangle=0,        # start angle 90° (Africa)
                          labels=None,         # turn off labels on pie chart
                          pctdistance=1.12,    # ratio between center of each slice and start of text generated by autopct
                          colors=colors_list,  # add custom colors
                          explode=explode_list # 'explode' lowest 3 continents
                      )
        plt.title('Immigration to Canada by Continent [1980 - 2013]')
        plt.legend(labels=df2.index, bbox_to_anchor=(1, 1))
        plt.ylabel("")
        plt.show()
```



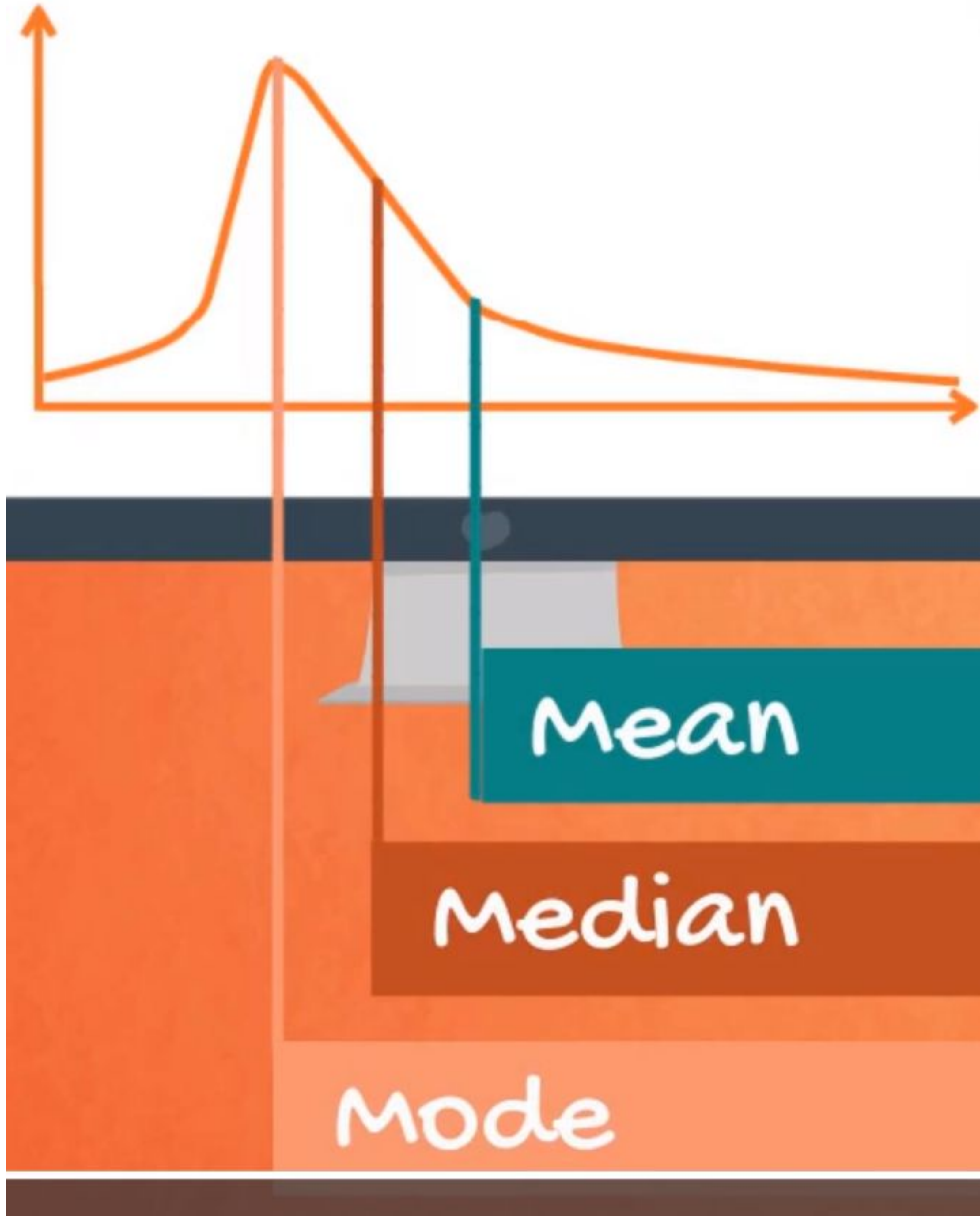
Boxplot

Mean vs Median vs Mode

- **Mean (average)**
  - is found by adding all numbers in the data set and then
  - dividing by the number of values in the data set

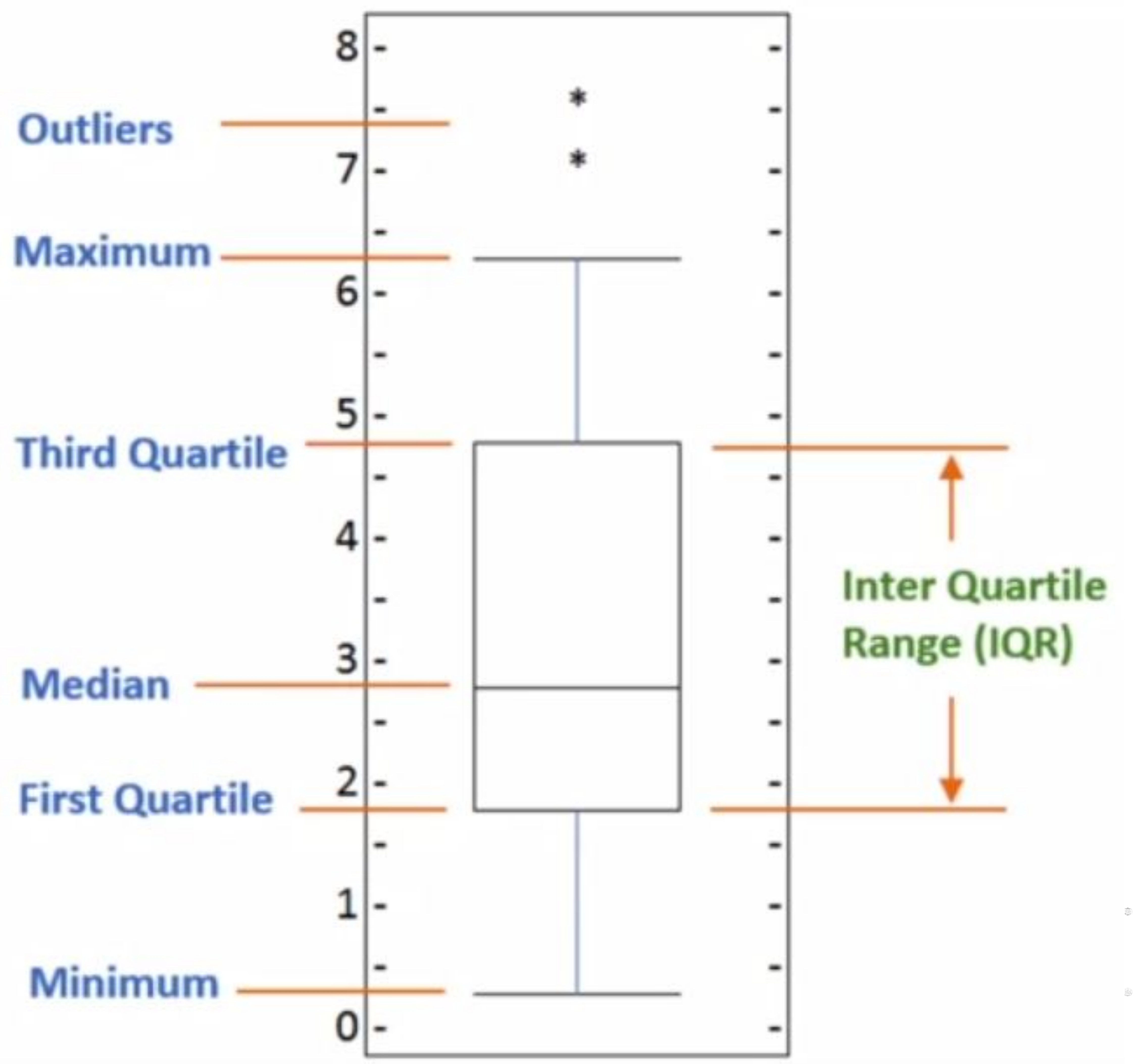
- **Median**
  - the middle value
  - when a data set is ordered from least to greatest

- **Mode**
  - the number that occurs most often in a data set



Boxplot

Statistically represents data distribution through **5 dimensions**



Boxplot

Statistically represents data distribution through **5 dimensions**

1. **minimum**: smallest number in the sorted data
2. **first quartile**: 1/4 of data points are less than this value
3. **median**: median of the sorted data
4. **third quartile**: 3/4 of data points are less than this value
5. **maximum**: highest number in the sorted data

** Display outliers as individual dots outside extremes

Boxplot Example - Cell 1

```
In [1]: import pandas as pd

        df = pd.read_csv('canada-mig-dataset.csv')

        df.head()
```

Out[1]:

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2004	2005	2006	2007	2008	2009	2010	2011	2012
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	2978	3436	3009	2652	2111	1746	1758	2203	2635
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1450	1223	856	702	560	716	561	539	620
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3616	3626	4807	3623	4005	5393	4752	4325	3774
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	0	1	0	0	0	0	0	0
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	0	1	1	0	0	0	0	1

5 rows × 43 columns

Boxplot Example - Cell 2

In [2]:

```
df1 = df.set_index('OdName')
df1.head()
```

Out[2]:

	Type	Coverage	AREA	AreaName	REG	RegName	DEV	DevName	1980	1981	...	2004	2005	2006	2007	2008	2009	2010	2011	2
OdName																				
Afghanistan	Immigrants	Foreigners	935	Asia	5501	Southern Asia	902	Developing regions	16	39	...	2978	3436	3009	2652	2111	1746	1758	2203	2
Albania	Immigrants	Foreigners	908	Europe	925	Southern Europe	901	Developed regions	1	0	...	1450	1223	856	702	560	716	561	539	
Algeria	Immigrants	Foreigners	903	Africa	912	Northern Africa	902	Developing regions	80	67	...	3616	3626	4807	3623	4005	5393	4752	4325	3
American Samoa	Immigrants	Foreigners	909	Oceania	957	Polynesia	902	Developing regions	0	1	...	0	0	1	0	0	0	0	0	
Andorra	Immigrants	Foreigners	908	Europe	925	Southern Europe	901	Developed regions	0	0	...	0	0	1	1	0	0	0	0	

5 rows × 42 columns

Boxplot Example - Cell 3

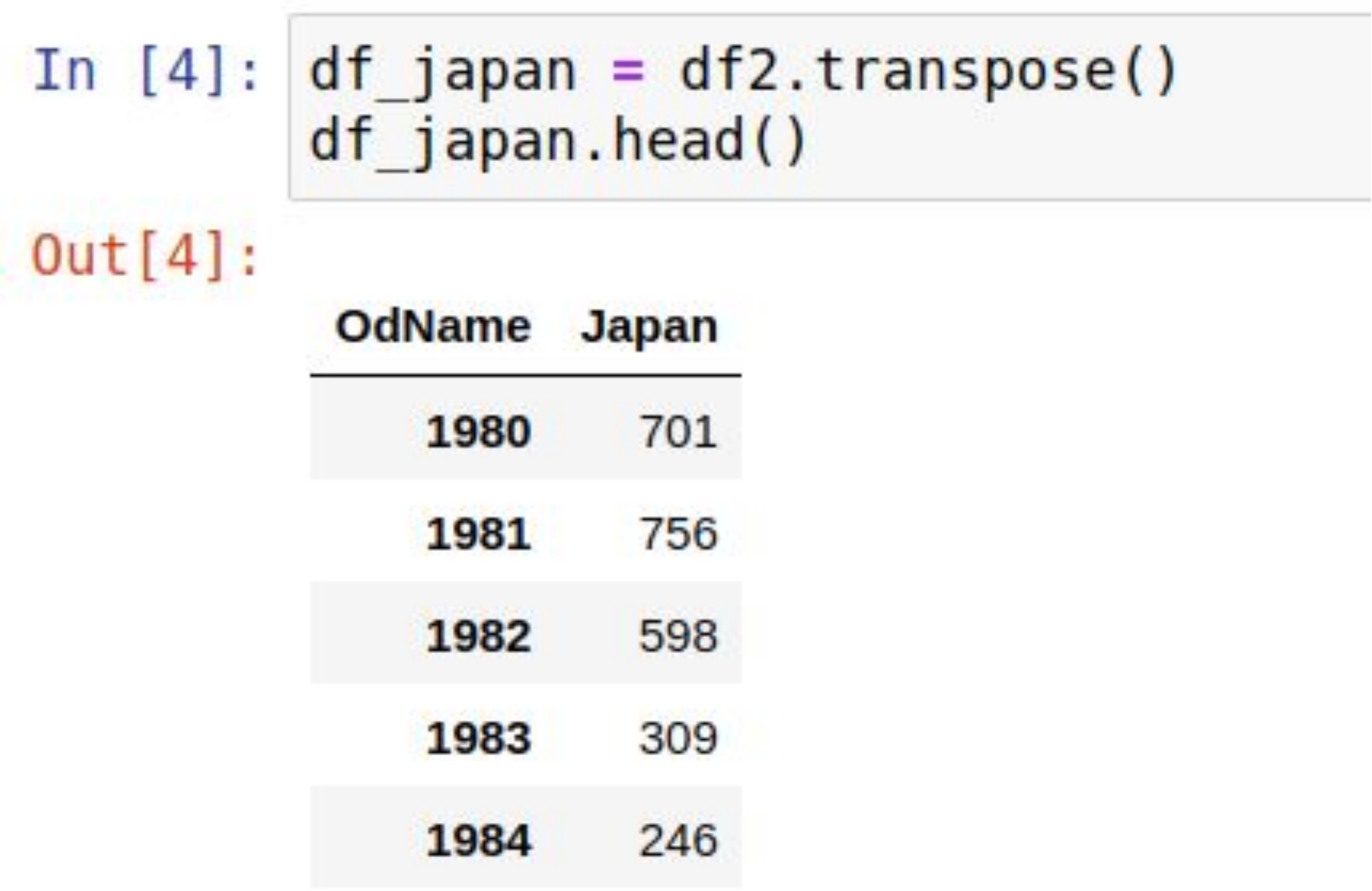
In [3]:
```
df2 = df1.loc[ ['Japan'], list(map(str, range(1980,2014))) ]
df2.head()
```

Out[3]:

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
OdName																					
Japan	701	756	598	309	246	198	248	422	324	494	...	973	1067	1212	1250	1284	1194	1168	1265	1214	982

1 rows × 34 columns

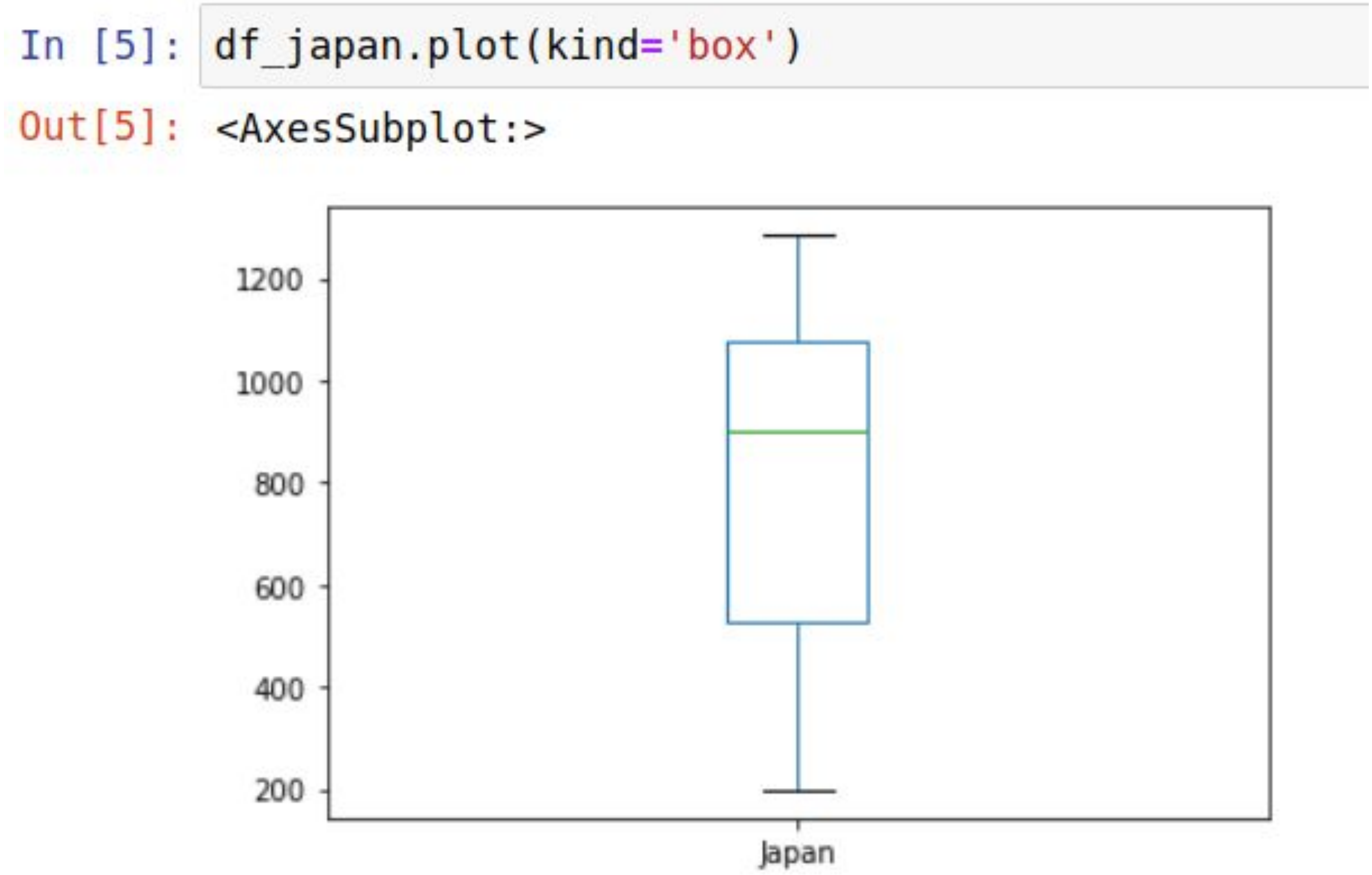
Boxplot Example - Cell 4



```
In [4]: df_japan = df2.transpose()
        df_japan.head()
```

```
Out[4]:
        OdName  Japan
          1980    701
          1981    756
          1982    598
          1983    309
          1984    246
```

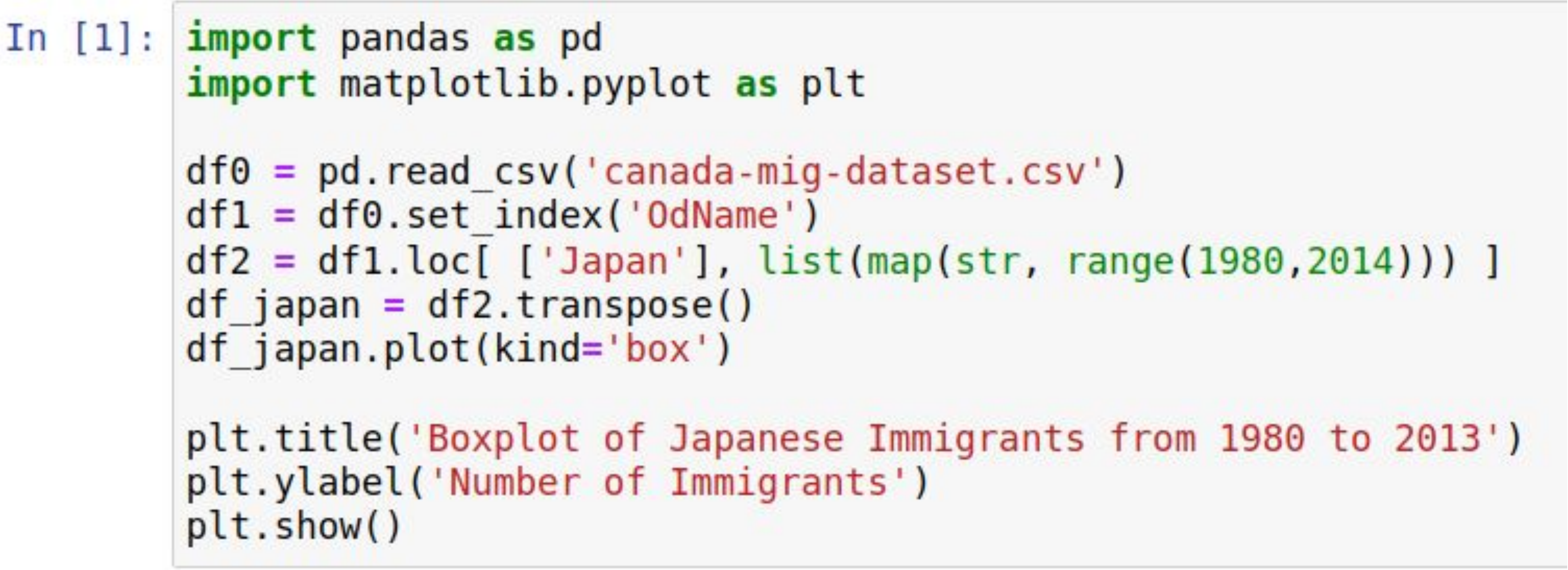
Boxplot Example - Cell 5



```
In [5]: df_japan.plot(kind='box')

Out[5]: <AxesSubplot:>
```

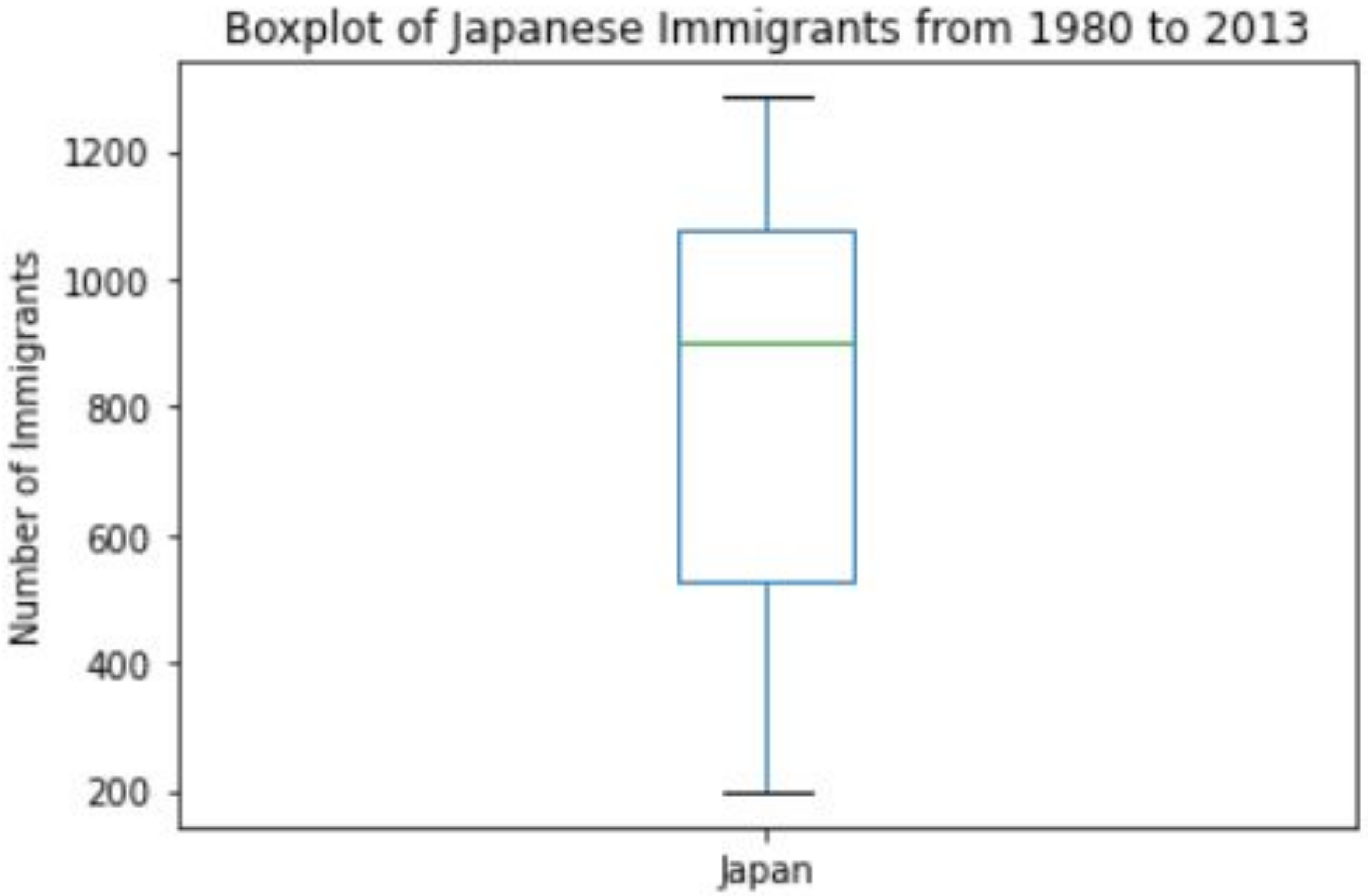
Boxplot - Complete Example



```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt

        df0 = pd.read_csv('canada-mig-dataset.csv')
        df1 = df0.set_index('OdName')
        df2 = df1.loc[ ['Japan'], list(map(str, range(1980,2014))) ]
        df_japan = df2.transpose()
        df_japan.plot(kind='box')

        plt.title('Boxplot of Japanese Immigrants from 1980 to 2013')
        plt.ylabel('Number of Immigrants')
        plt.show()
```



Scatter Plot

Scatter Plot

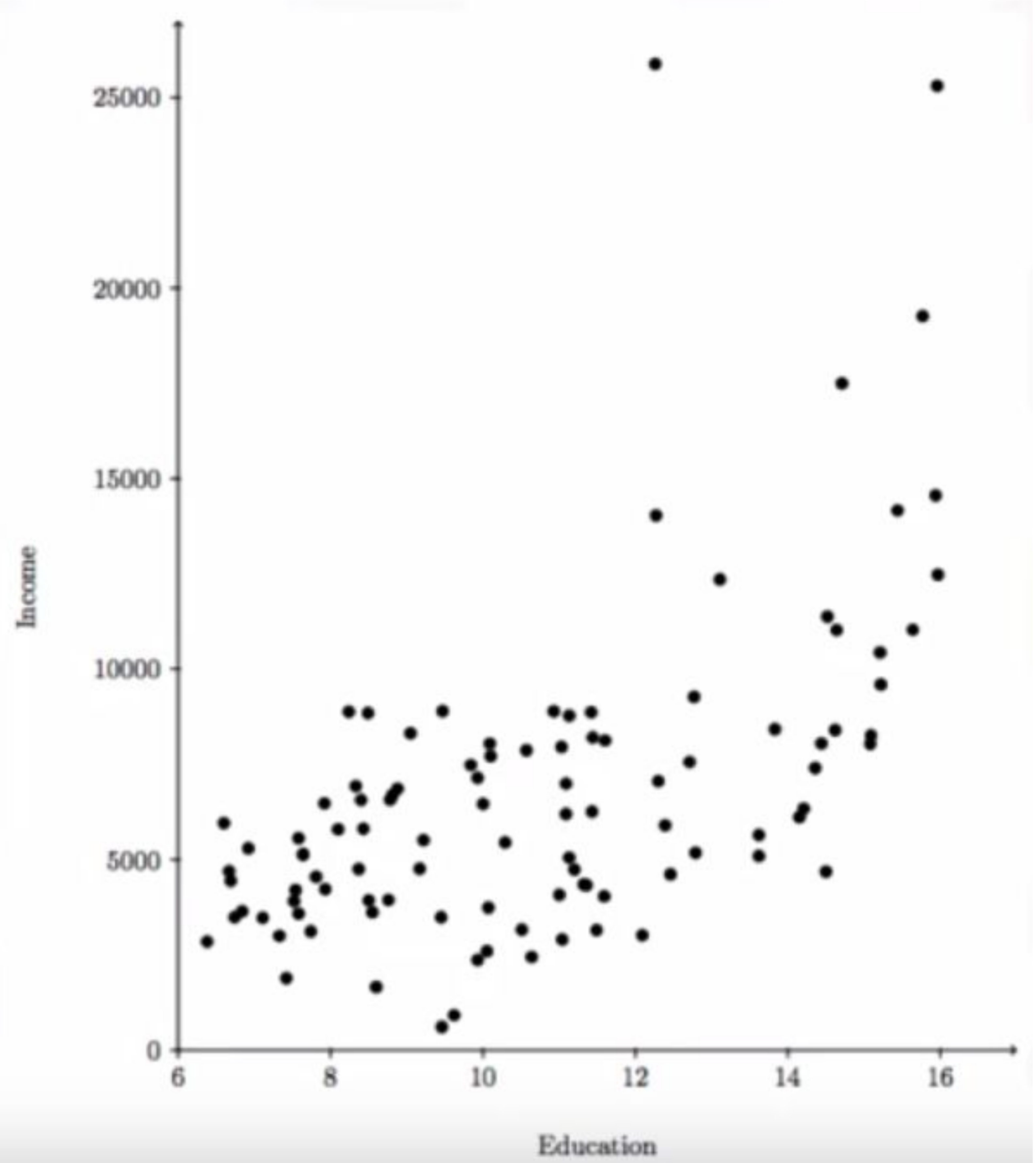
- Displays values pertaining to typically:
  - two variables against each other

- **Usually**
  - a **dependent variable** to be plotted against an **independent variable**
  - in order to determine if any **correlation** between the two variables exists

Scatter Plot Examples

- Scatter plot of income versus education
    - individual with more education years is likely to earn higher income

- Scatter plot of immigration
    - clearly depicts an overall rising trend
    - of immigration with time



Scatter Plot Example - Cell 1

In [1]:

```
import pandas as pd

df = pd.read_csv('canada-mig-dataset.csv')

df.head()
```

Out[1]:

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2004	2005	2006	2007	2008	2009	2010	2011	2012
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	2978	3436	3009	2652	2111	1746	1758	2203	2635
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1450	1223	856	702	560	716	561	539	620
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3616	3626	4807	3623	4005	5393	4752	4325	3774
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	0	1	0	0	0	0	0	0
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	0	1	1	0	0	0	0	1

5 rows × 43 columns

Scatter Plot Example - Cell 2

In [2]:

```
df1 = df0.iloc[:, 9:43]
df1.head()
```

Out[2]:

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	16	39	39	47	71	340	496	741	828	1076	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	2004
1	1	0	0	0	0	0	1	2	2	3	...	1450	1223	856	702	560	716	561	539	620	603
2	80	67	71	69	63	44	69	132	242	434	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	4331
3	0	1	0	0	0	0	0	1	0	1	...	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	2	0	0	0	...	0	0	1	1	0	0	0	0	1	1

5 rows × 34 columns

Scatter Plot Example - Cell 3

In [3]:

```
df1.loc["Total"] = df1.sum(axis=0)
df1.tail()
```

Out[3]:

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	...	2004	2005	2006	2007	2008	2009	2010	2011
192	1	2	1	6	0	18	7	12	7	18	...	124	161	140	122	133	128	211	160
193	11	17	11	7	16	9	15	23	44	68	...	56	91	77	71	64	60	102	69
194	72	114	102	44	32	29	43	68	99	187	...	1450	615	454	663	611	508	494	434
195	44000	18078	16904	13635	14855	14368	13303	17304	22279	27118	...	3739	4785	4583	4348	4197	3402	3731	2554
Total	143137	128641	121175	89185	88272	84346	99351	152075	161585	191550	...	235822	262242	251640	236753	247244	252170	280687	248748

5 rows × 34 columns

Scatter Plot Example - Cell 4

In [4]:

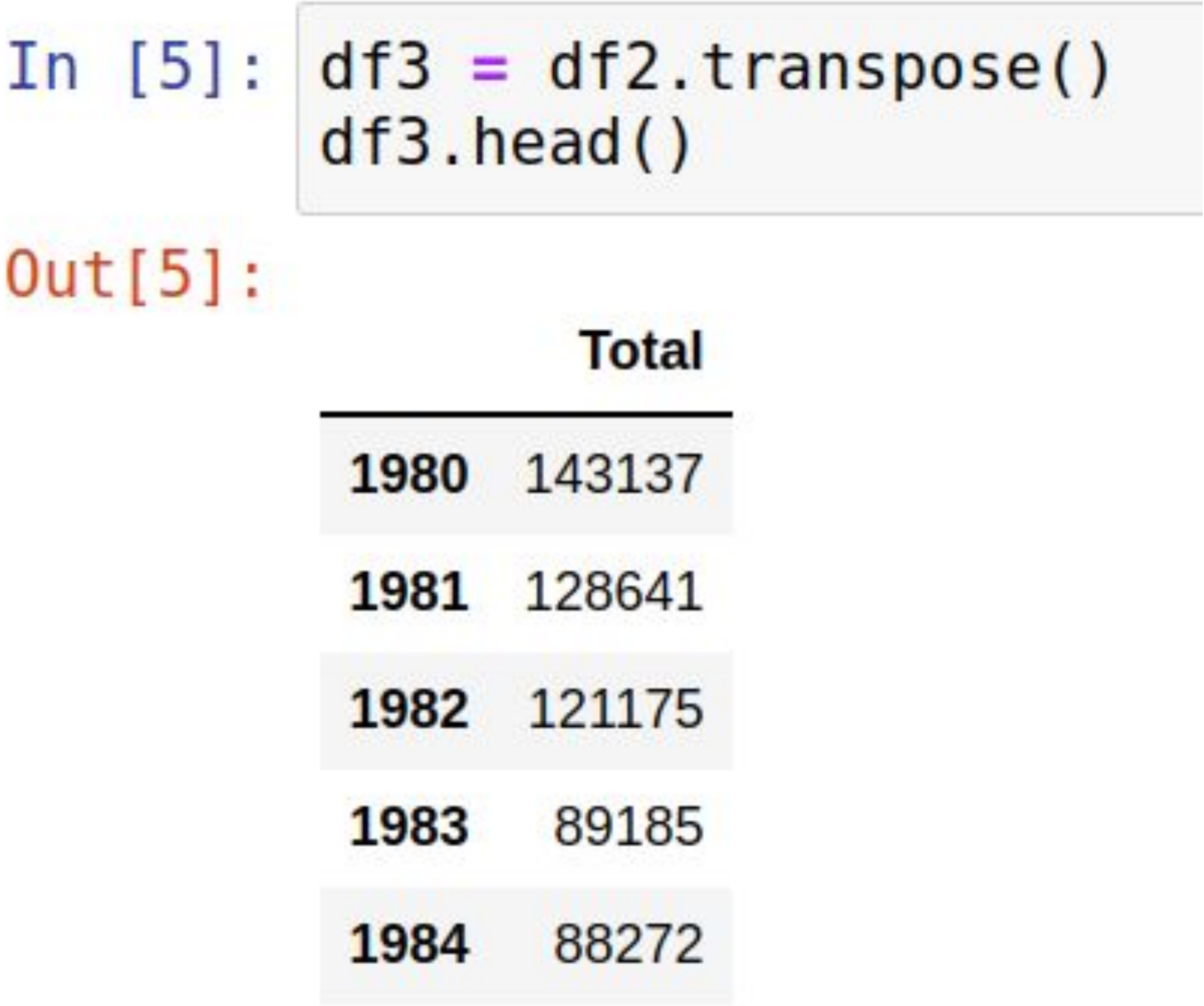
```
df2 = df1.tail(1)
df2.head()
```

Out[4]:

	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	...	2004	2005	2006	2007	2008	2009	2010	2011
Total	143137	128641	121175	89185	88272	84346	99351	152075	161585	191550	...	235822	262242	251640	236753	247244	252170	280687	248748

1 rows × 34 columns

Scatter Plot Example - Cell 5



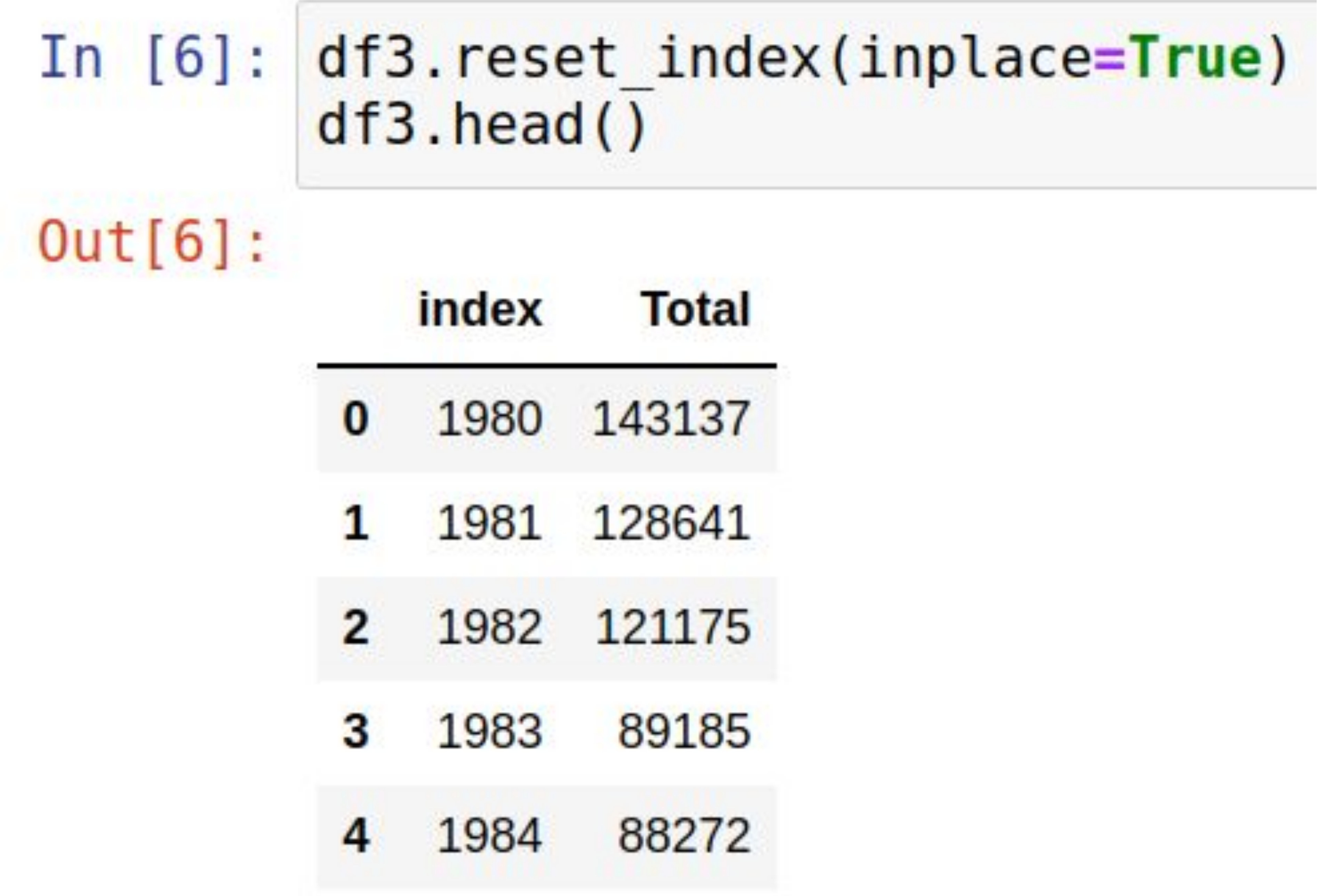
In [5]:

```
df3 = df2.transpose()
df3.head()
```

Out[5]:

	Total
1980	143137
1981	128641
1982	121175
1983	89185
1984	88272

Scatter Plot Example - Cell 6



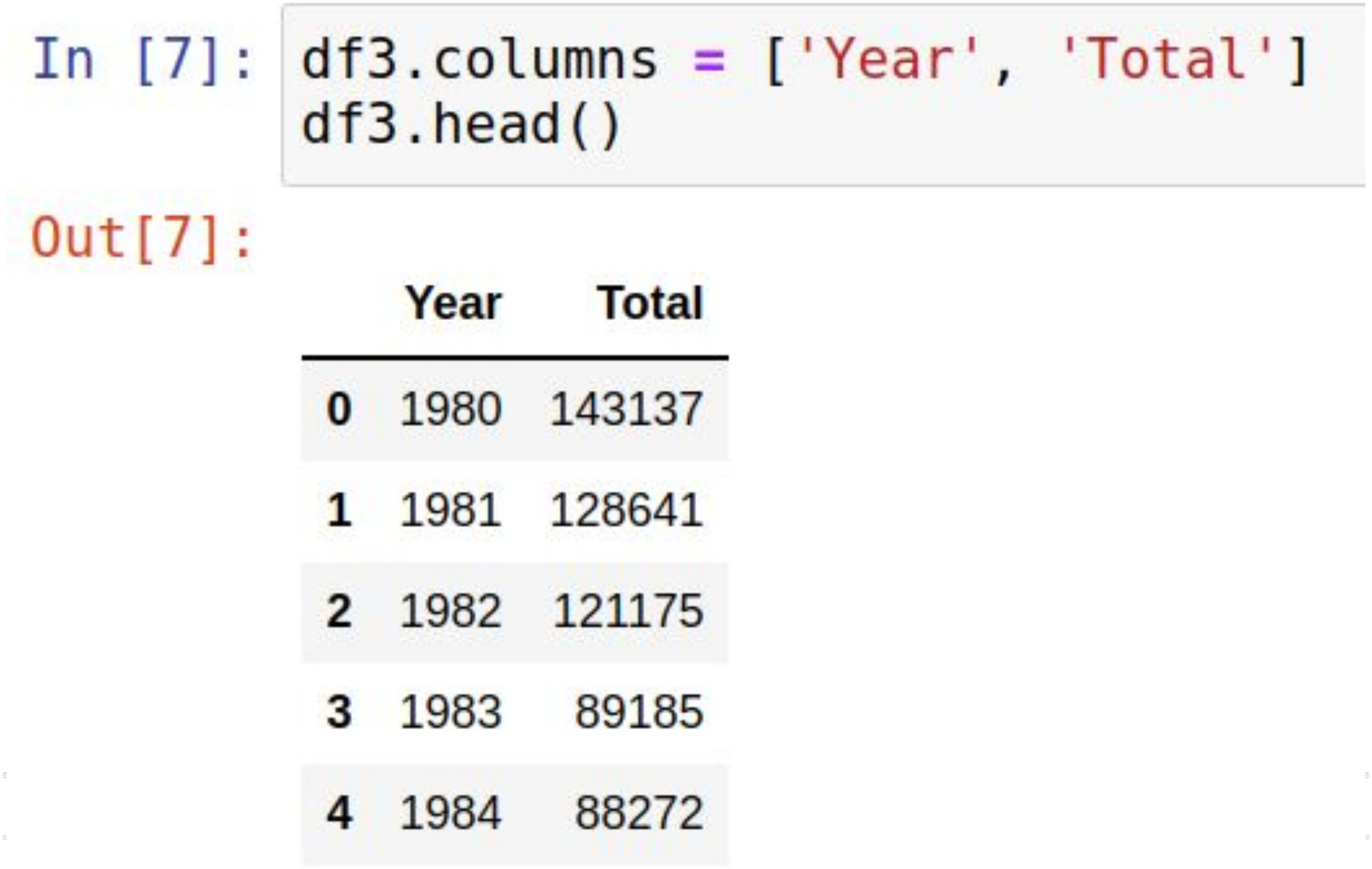
In [6]:

```
df3.reset_index(inplace=True)
df3.head()
```

Out[6]:

	index	Total
0	1980	143137
1	1981	128641
2	1982	121175
3	1983	89185
4	1984	88272

Scatter Plot Example - Cell 7



In [7]:

```python
df3.columns = ['Year', 'Total']
df3.head()
```

Out[7]:

	Year	Total
0	1980	143137
1	1981	128641
2	1982	121175
3	1983	89185
4	1984	88272

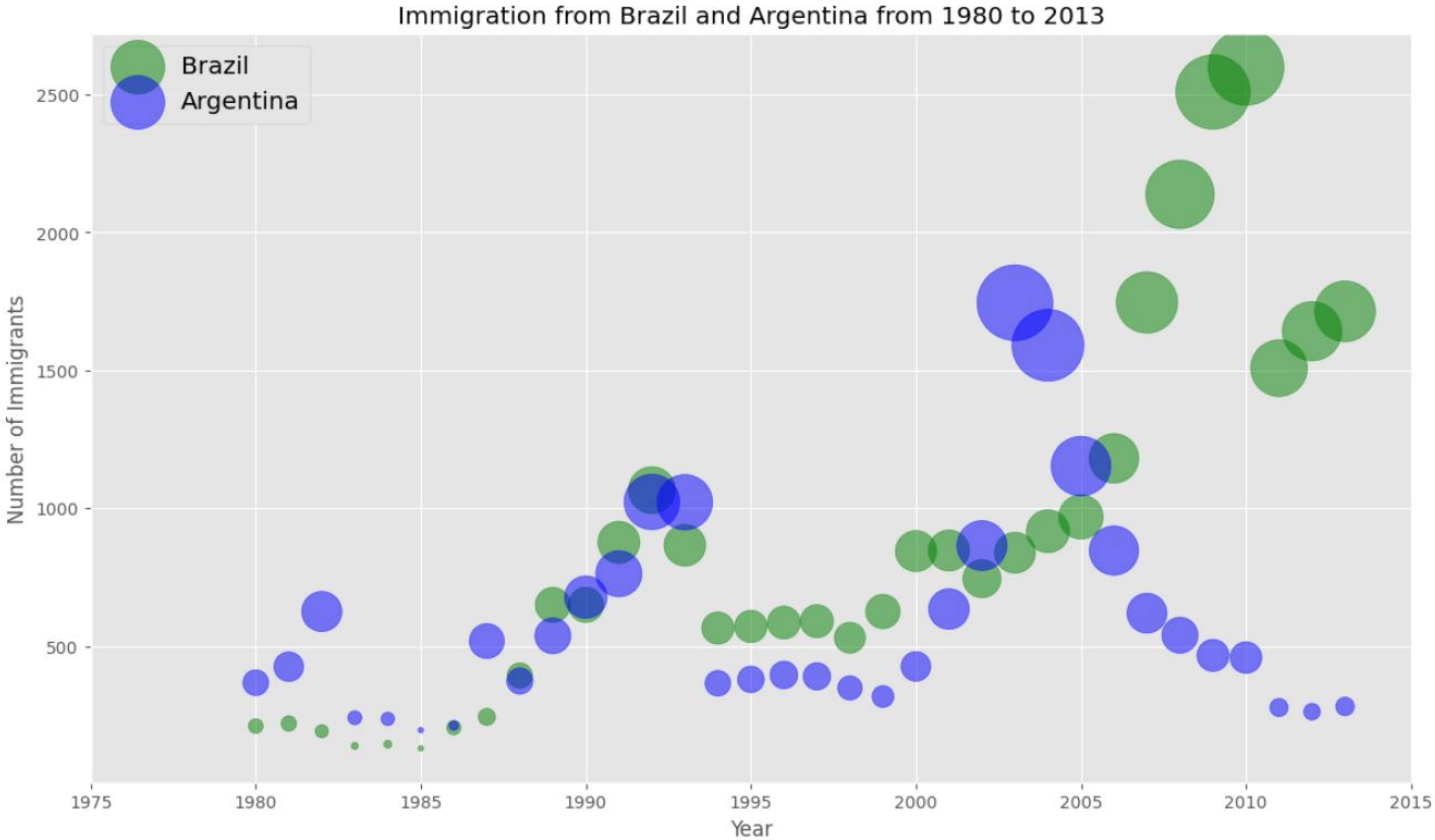
Scatter Plot Example - Cell 8

```
In [8]: df3.plot(kind='scatter', y='Total', x='Year', figsize=(16, 6))
Out[8]: <AxesSubplot:xlabel='Year', ylabel='Total'>
```

275000
250000
225000
200000
175000
150000
125000
100000
75000
Total
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
Year

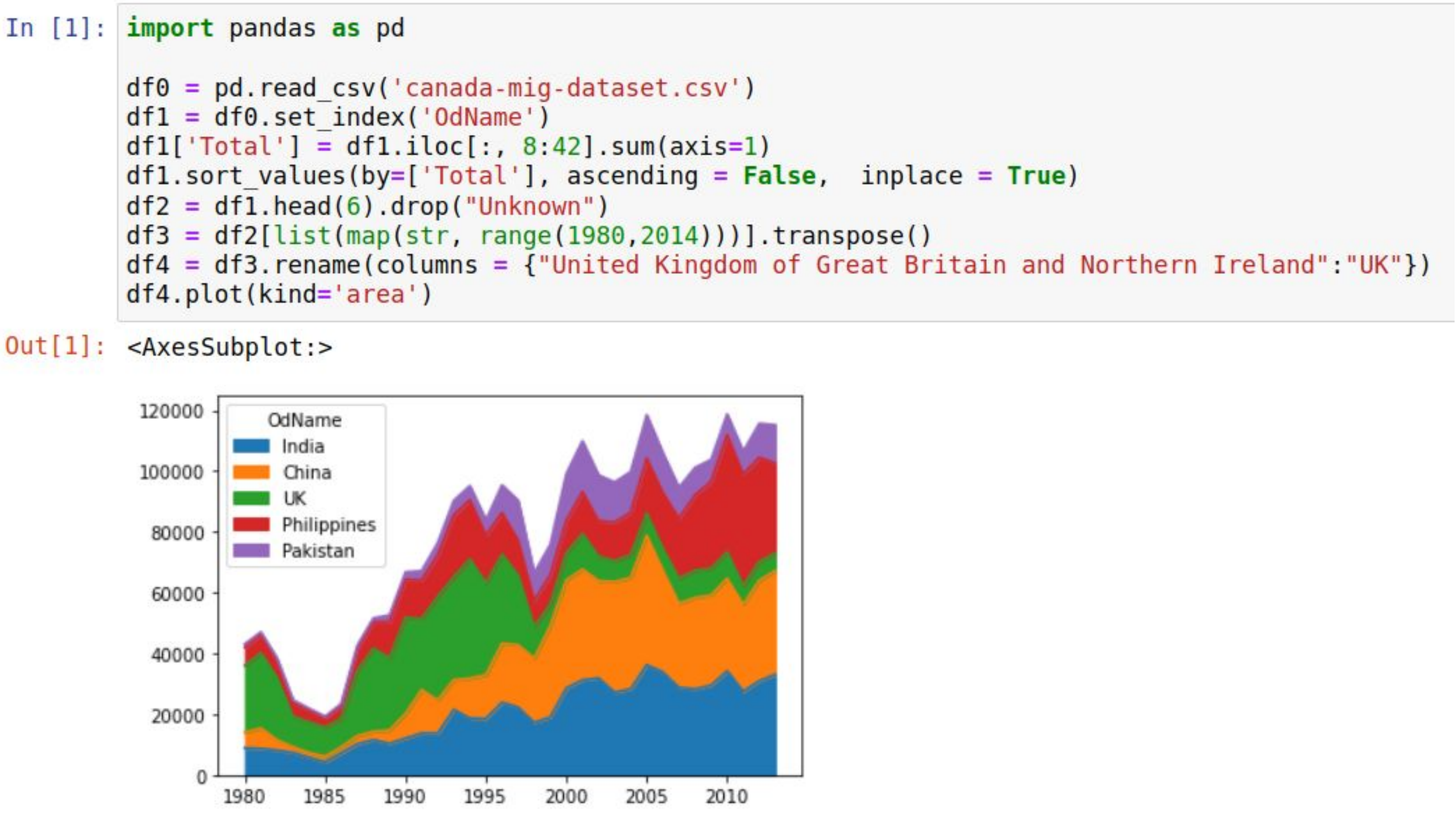
Bubble Plot

- A very interesting variation of the scatter plot



Area Plot - Complete Example

Area Plot - Complete Example - No Unknown



```
In [1]: import pandas as pd

        df0 = pd.read_csv('canada-mig-dataset.csv')
        df1 = df0.set_index('OdName')
        df1['Total'] = df1.iloc[:, 8:42].sum(axis=1)
        df1.sort_values(by=['Total'], ascending = False,  inplace = True)
        df2 = df1.head(6).drop("Unknown")
        df3 = df2[list(map(str, range(1980,2014)))].transpose()
        df4 = df3.rename(columns = {"United Kingdom of Great Britain and Northern Ireland":"UK"})
        df4.plot(kind='area')

Out[1]: <AxesSubplot:>
```

Questions

Links

https://github.com/fcai-b/dv

References

1. https://www.coursera.org/learn/python-for-data-visualization