Data Visualization

Agenda

1. Matplotlib Scripting Layer Examples
2. Histogram
3. Bar Chart
4. Questions

Matplotlib Scripting Layer Examples

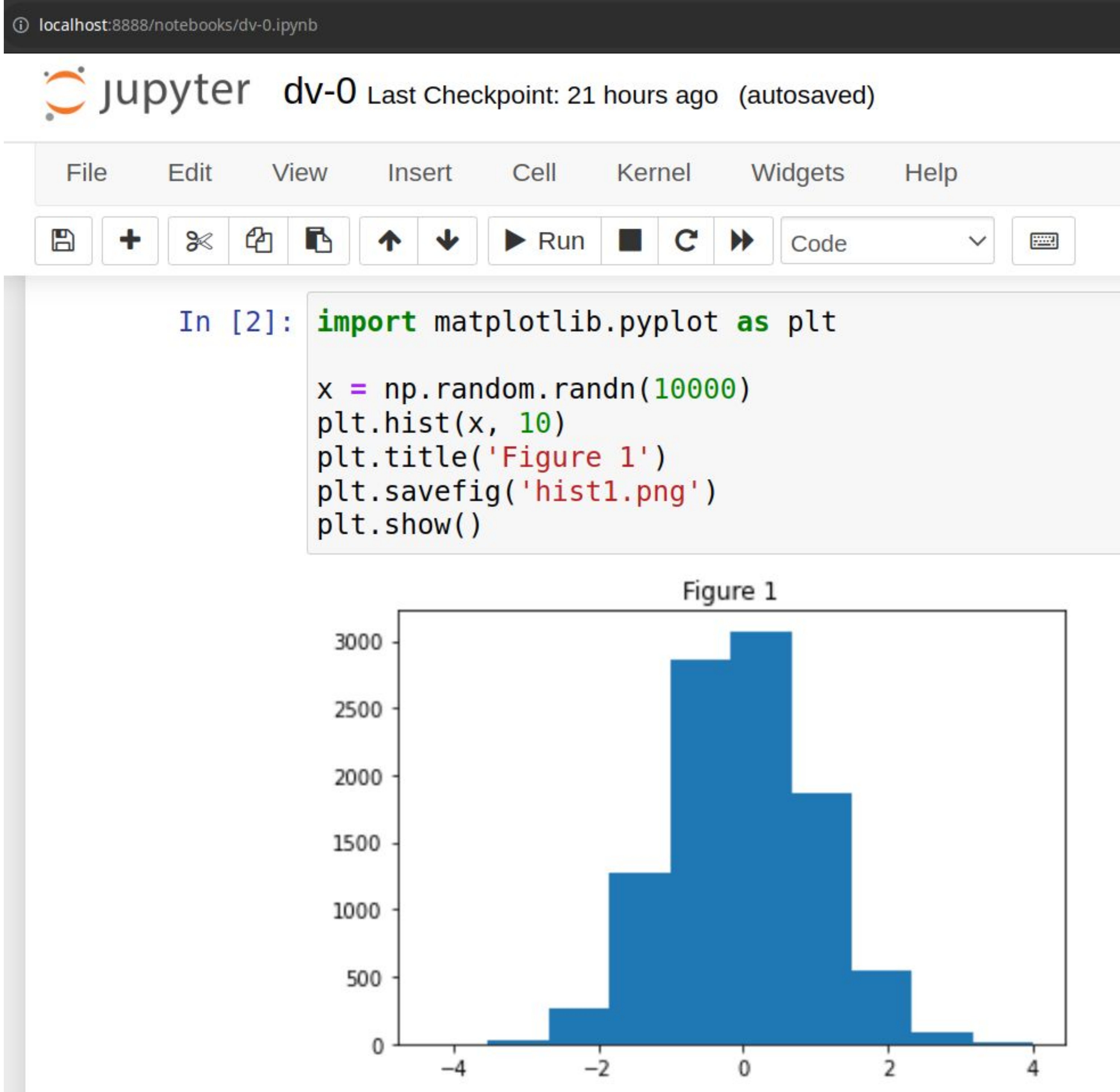
NumPy Example

localhost:8888/notebooks/dv-0.ipynb
jupyter dv-0 Last Checkpoint: 21 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run
Code

```
In [1]: import numpy as np

        np.random.randn(10) #Return 10 samples from the Standard Normal Distribution

Out[1]: array([-0.64393041,  0.0329367 , -0.16840147,  0.88846809,  0.76751103,
                0.18852699, -1.30213432,  0.58043701,  1.80149475, -0.18262329])
```

Scripting Layer Example 1

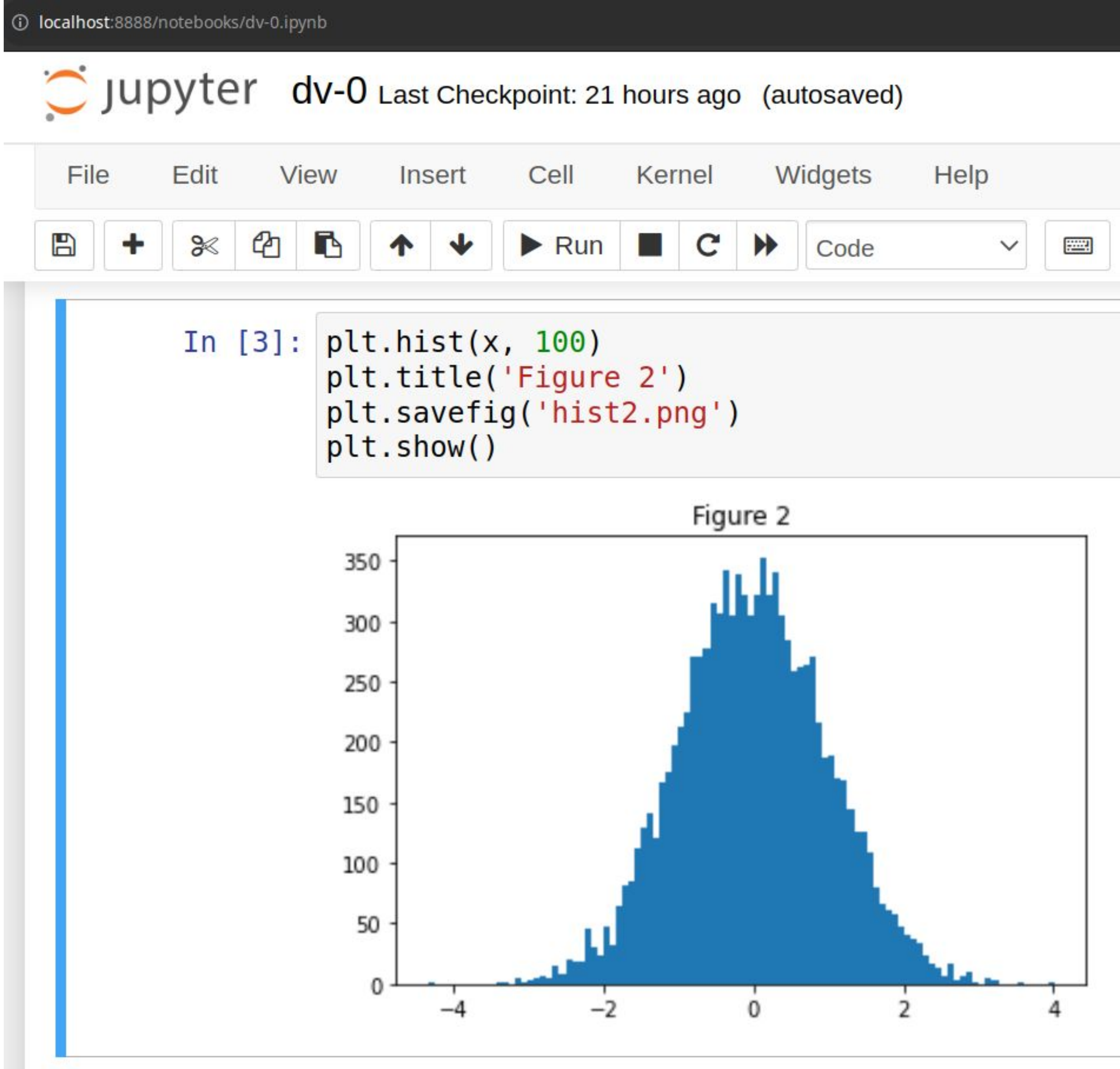
- `import` **`pyplot`**
  - ○ from the **matplotlib library**


localhost:8888/notebooks/dv-0.ipynb
jupyter
dv-0 Last Checkpoint: 21 hours ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code
In [2]:

```
import matplotlib.pyplot as plt

x = np.random.randn(10000)
plt.hist(x, 10)
plt.title('Figure 1')
plt.savefig('hist1.png')
plt.show()
```

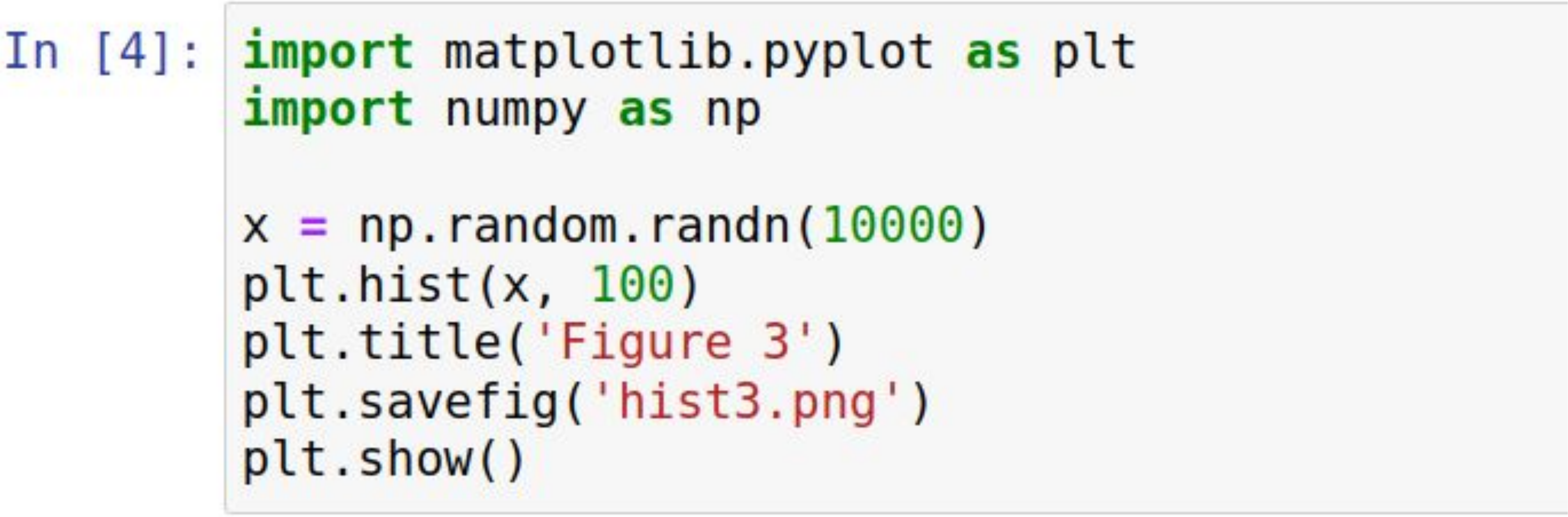
Scripting Layer Example 2

- all methods
  - creating/manipulating
    - histogram
    - other Artist objects
  - are part of **pyplot**



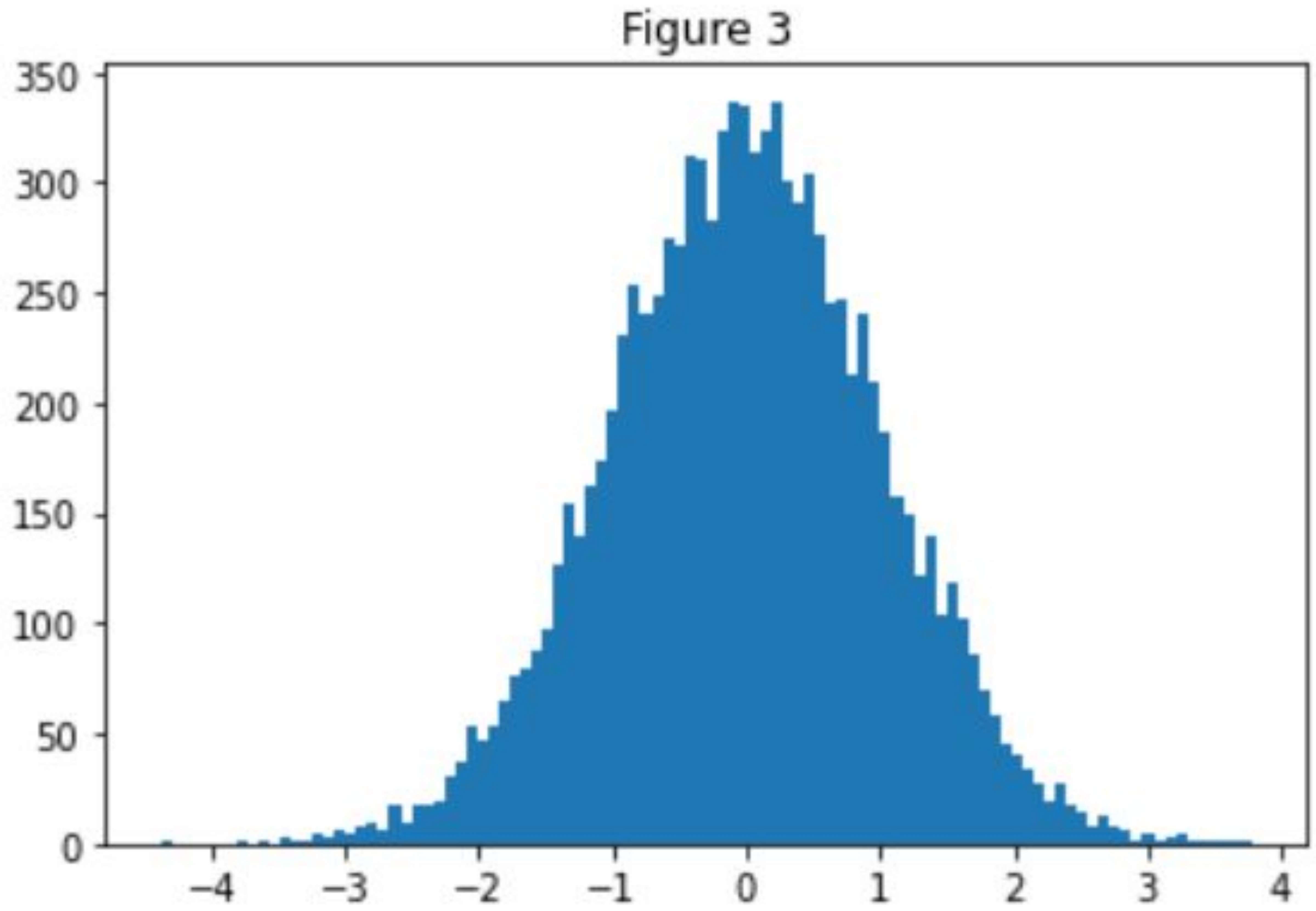
```
plt.hist(x, 100)
plt.title('Figure 2')
plt.savefig('hist2.png')
plt.show()
```

Scripting Layer Complete Example



In [4]:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.randn(10000)
plt.hist(x, 100)
plt.title('Figure 3')
plt.savefig('hist3.png')
plt.show()
```



Histogram

Histogram

- represents the **frequency distribution** of a **numeric dataset**
- is a way to represent the **frequency distribution** of a variable

- The way it works is:
  - partitions the spread of the numeric data into bins
  - assigns each datapoint in the dataset to a bin
  - counts the number of datapoints that have been assigned to each bin

- **Vertical axis**
  - is actually the frequency or the number of datapoints in each bin

Histogram Example - Cell 1

In [1]:

```python
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

Histogram Example - Cell 2

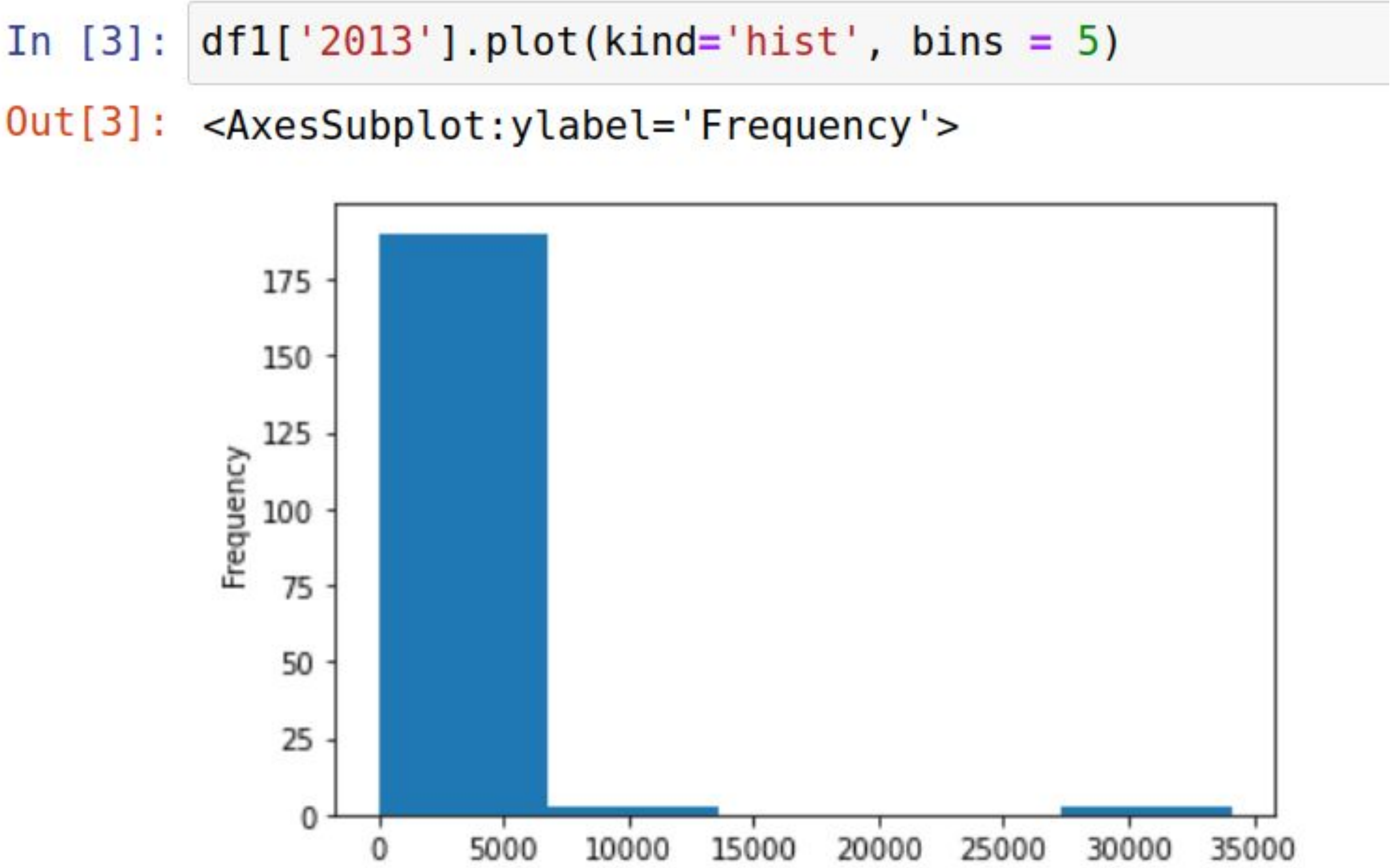
In [2]:
```
df1 = df.set_index('OdName')
df1.head()
```

Out[2]:

	Type	Coverage	AREA	AreaName	REG	RegName	DEV	DevName	1980	1981	...	2004	2005	2006	2007	2008	2009	2010	2011
OdName																			
Afghanistan	Immigrants	Foreigners	935	Asia	5501	Southern Asia	902	Developing regions	16	39	...	2978	3436	3009	2652	2111	1746	1758	2203
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
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

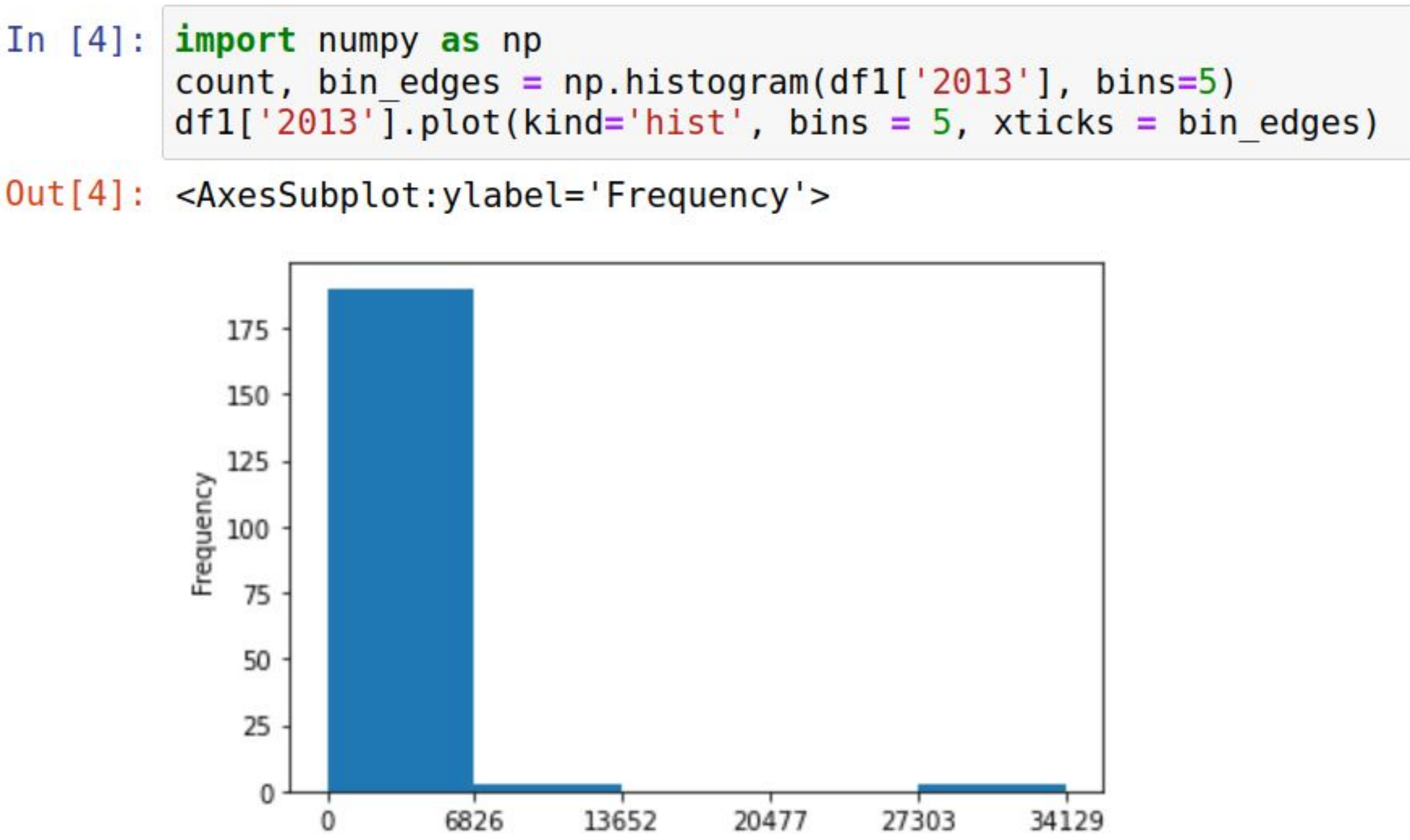
5 rows × 42 columns

Histogram Example - Cell 3



```
In [3]: df1['2013'].plot(kind='hist', bins = 5)

Out[3]: <AxesSubplot:ylabel='Frequency'>
```

Histogram Example - Cell 4



```
In [4]: import numpy as np
        count, bin_edges = np.histogram(df1['2013'], bins=5)
        df1['2013'].plot(kind='hist', bins = 5, xticks = bin_edges)

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

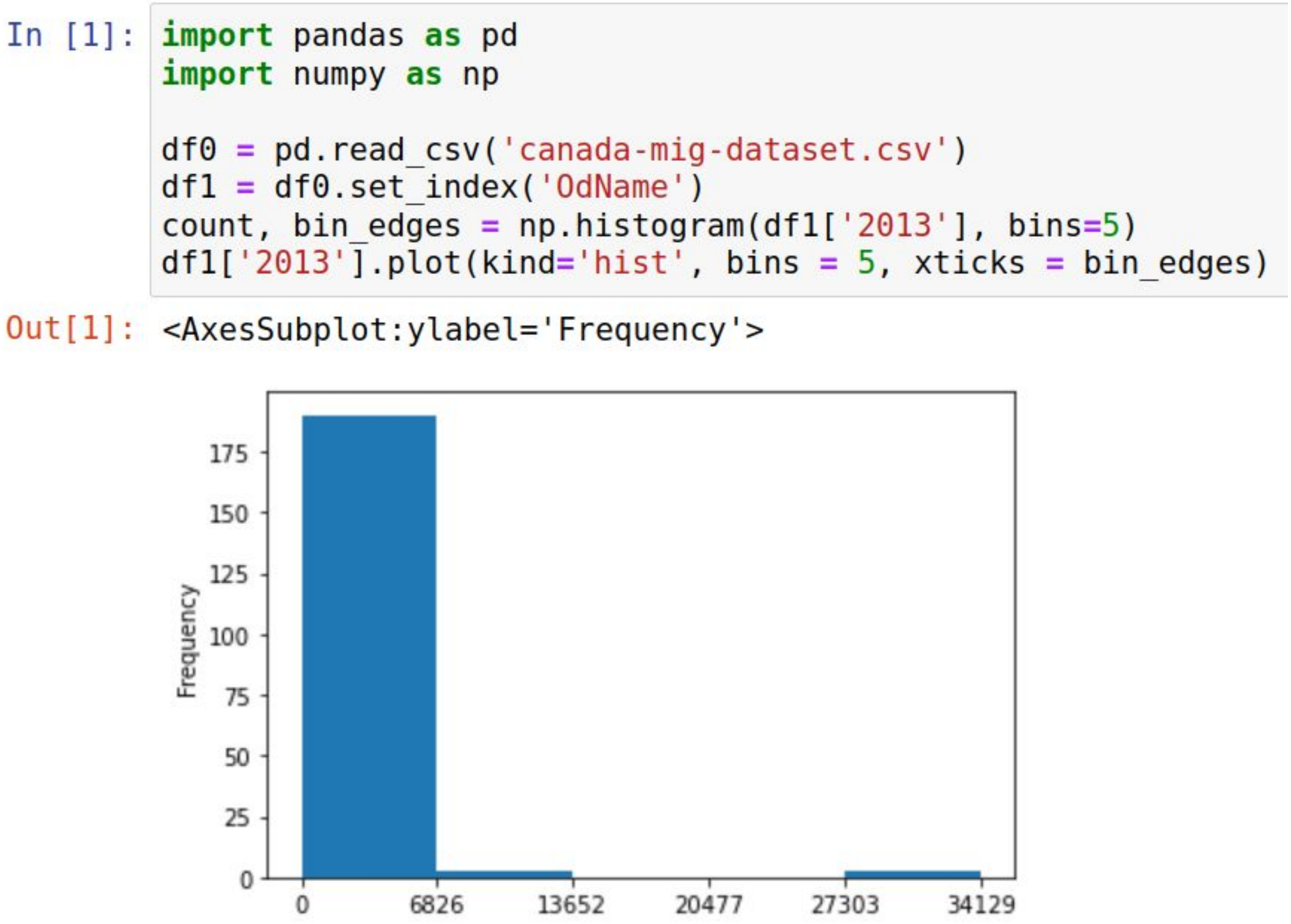
```
count = [190   3   0   0   3]
bin_edges = [    0.    6825.8 13651.6 20477.4 27303.2 34129. ]
```

Histogram

- **Numpy histogram function**
  - partitions the spread of the column data into bins of equal width
  - computes the number of data points that fall in each bin
  - returns the **frequency of each bin** (count) & **bin edges** (bin_edges)

- We then pass these **bin edges** as an additional parameter in our plot function to generate the histogram

Histogram - Complete Example



In [1]:

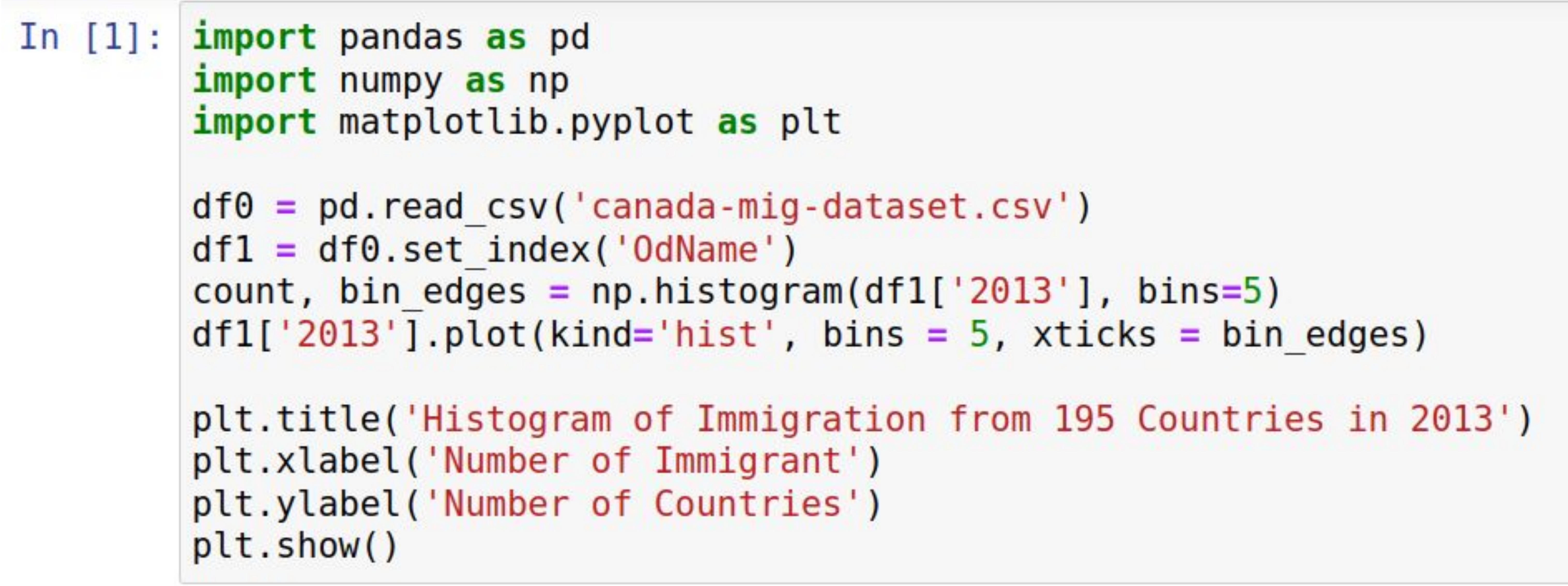
```
import pandas as pd
import numpy as np

df0 = pd.read_csv('canada-mig-dataset.csv')
df1 = df0.set_index('OdName')
count, bin_edges = np.histogram(df1['2013'], bins=5)
df1['2013'].plot(kind='hist', bins = 5, xticks = bin_edges)
```

Out[1]:

```
<AxesSubplot:ylabel='Frequency'>
```

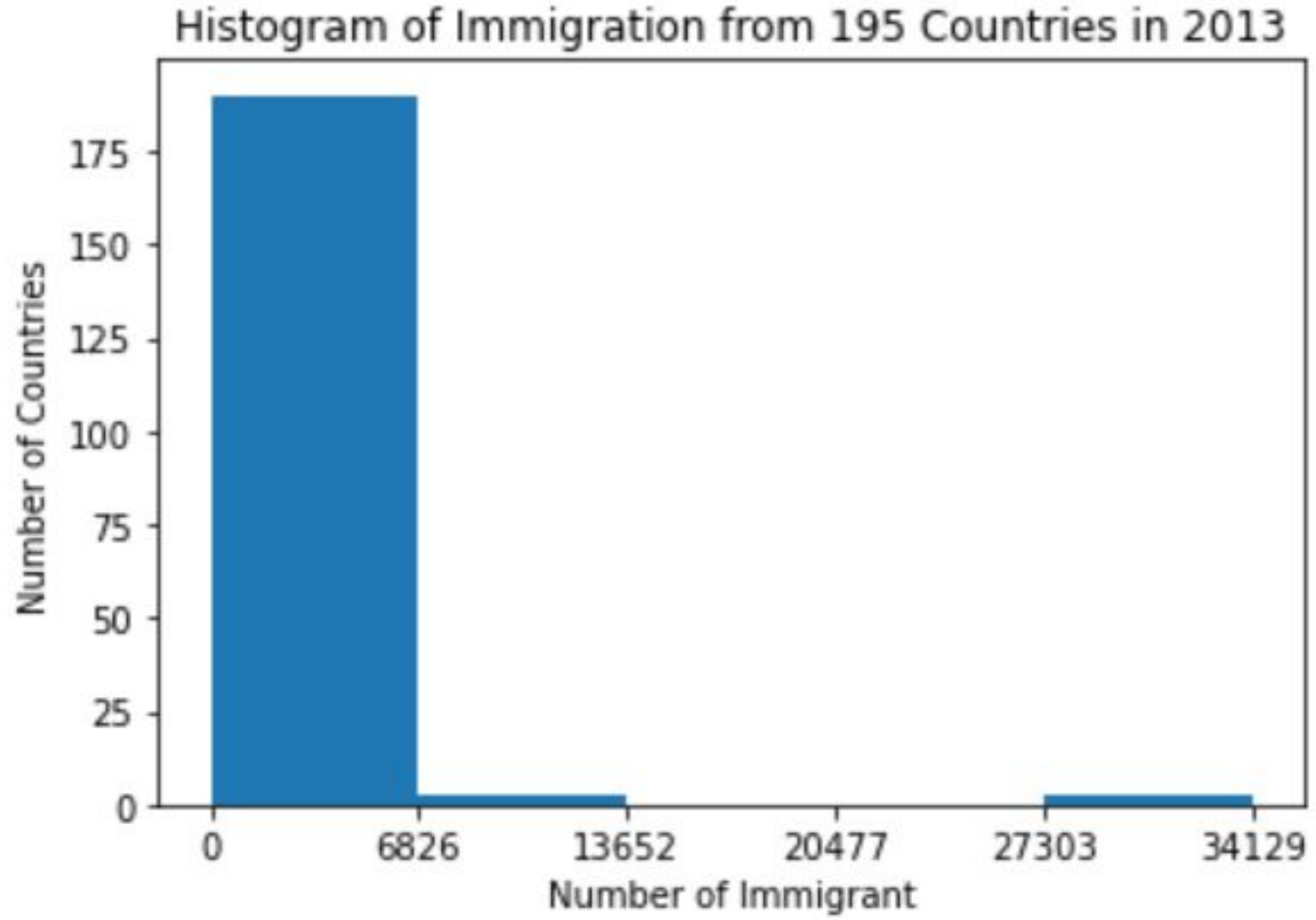
Histogram - Complete Example

- Add Title
- Add X Label
- Add Y Label



In [1]:

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

df0 = pd.read_csv('canada-mig-dataset.csv')
df1 = df0.set_index('OdName')
count, bin_edges = np.histogram(df1['2013'], bins=5)
df1['2013'].plot(kind='hist', bins = 5, xticks = bin_edges)

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



Bar Chart

Bar Chart (Bar Graph)

- is a very popular visualization tool
  - the length of each bar is proportional to the value of the item that it represents

- commonly used:
  - to compare the values of a variable at a given point in time

- Example:
  - visualizing in discrete fashion how immigration from Iceland to Canada evolved
  - bar height represents total immigration from Iceland to Canada in a specific year

Bar Chart Example - Cell 1

In [1]:

```python
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

Bar Chart Example - Cell 2

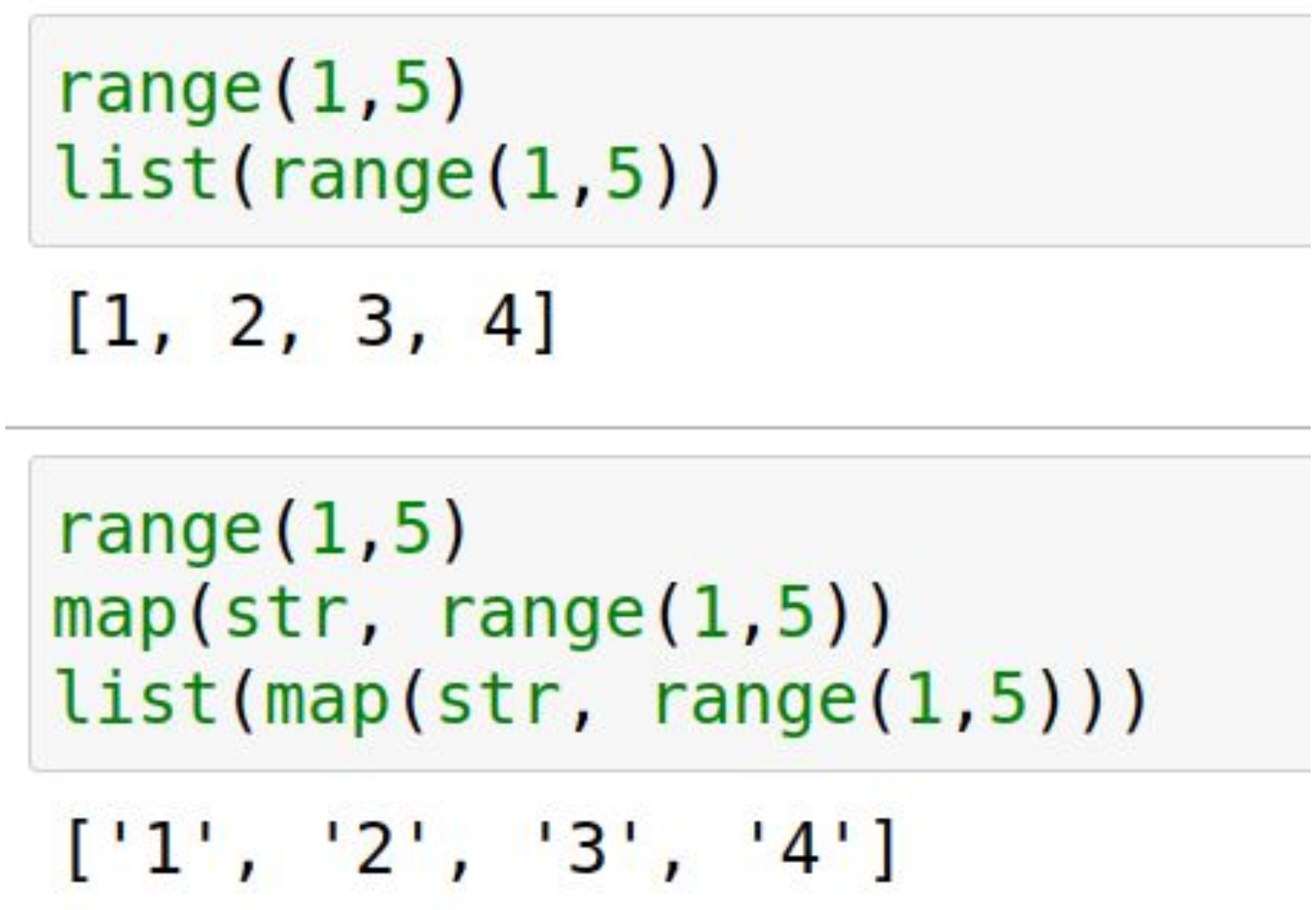
```
In [2]: df1 = df.set_index('OdName')
        df1.head()
```

Out[2]:

	Type	Coverage	AREA	AreaName	REG	RegName	DEV	DevName	1980	1981	...	2004	2005	2006	2007	2008	2009	2010	2011
OdName																			
Afghanistan	Immigrants	Foreigners	935	Asia	5501	Southern Asia	902	Developing regions	16	39	...	2978	3436	3009	2652	2111	1746	1758	2203
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
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

Explain: list(map(str, range(1,5)))



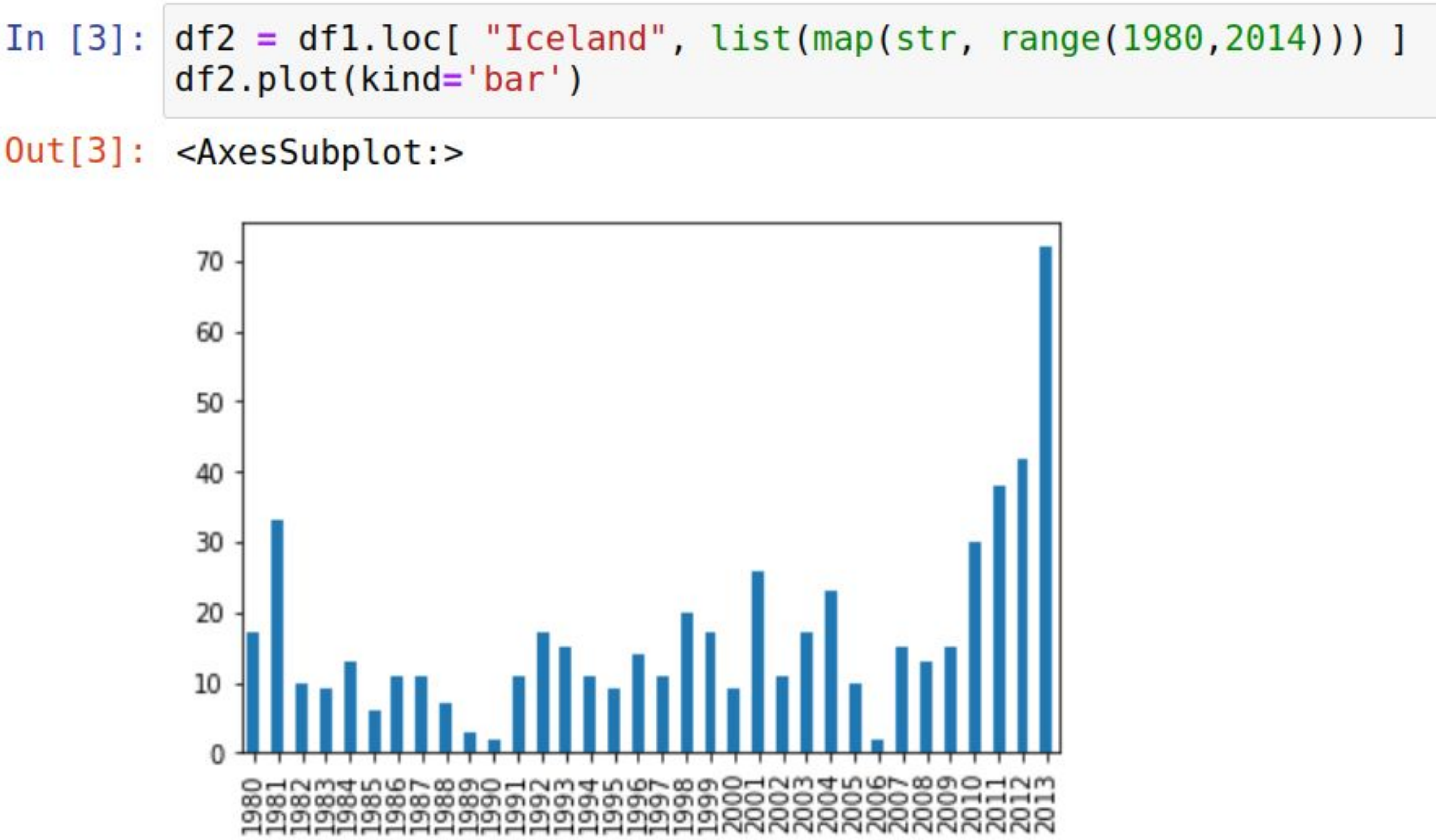
```
range(1,5)
list(range(1,5))
```

```
[1, 2, 3, 4]
```

```
range(1,5)
map(str, range(1,5))
list(map(str, range(1,5)))
```

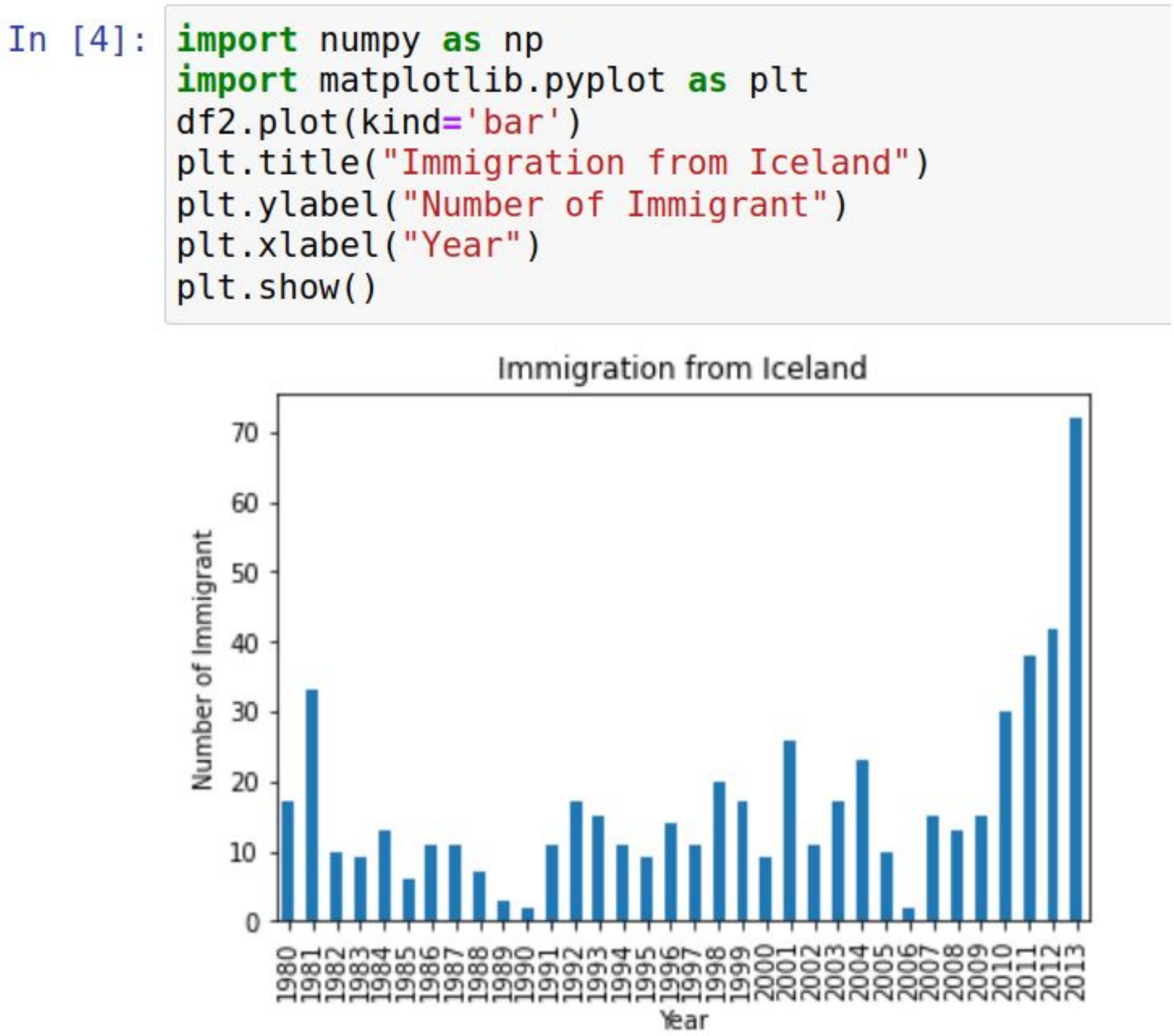
```
['1', '2', '3', '4']
```

Bar Chart Example - Cell 3



```
In [3]: df2 = df1.loc[ "Iceland", list(map(str, range(1980,2014))) ]
        df2.plot(kind='bar')

Out[3]: <AxesSubplot:>
```

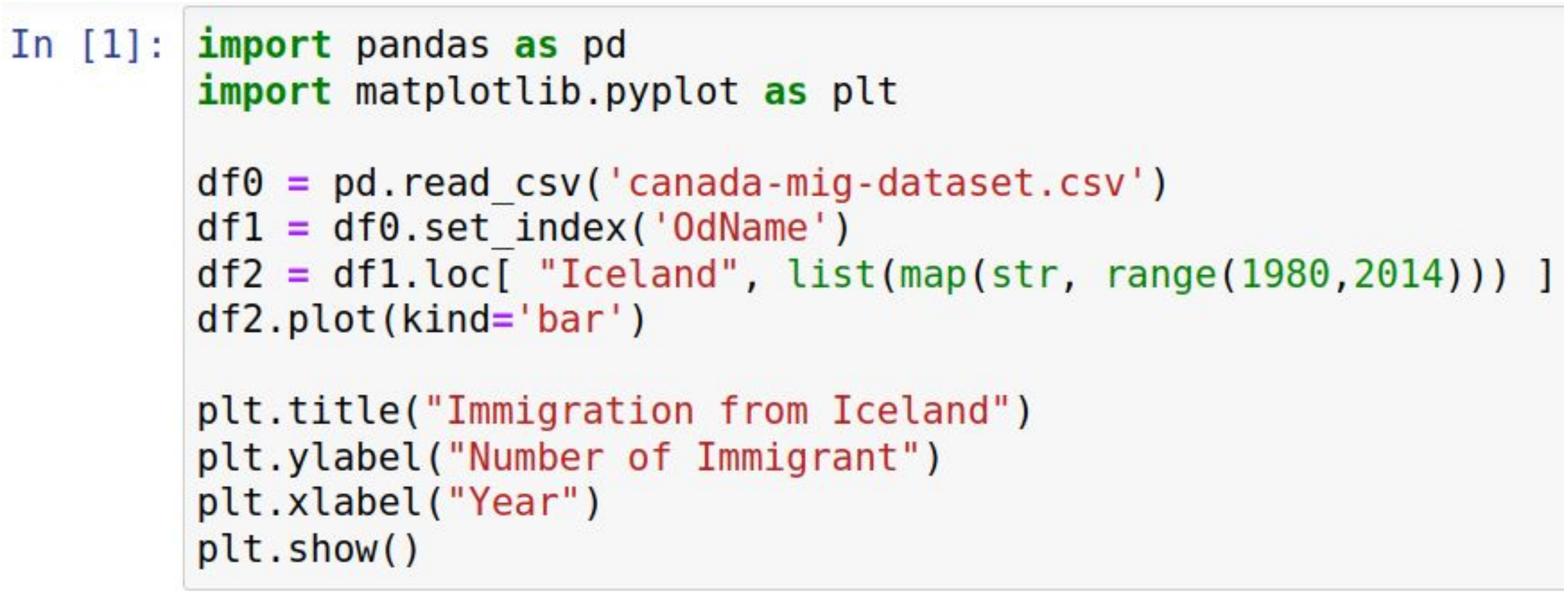
Bar Chart Example - Cell 4



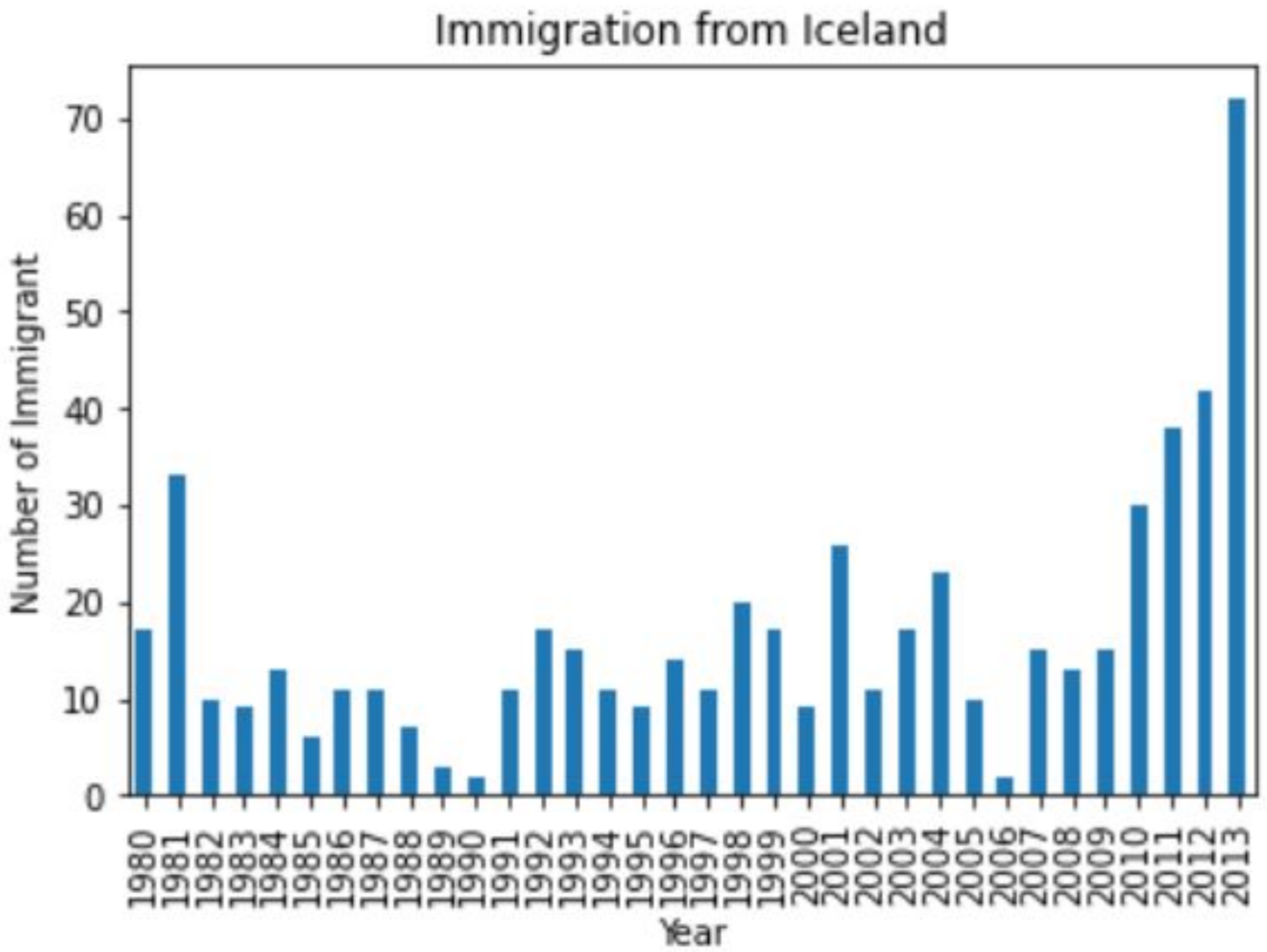
In [4]:

```
import numpy as np
import matplotlib.pyplot as plt
df2.plot(kind='bar')
plt.title("Immigration from Iceland")
plt.ylabel("Number of Immigrant")
plt.xlabel("Year")
plt.show()
```

Bar Chart - 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[ "Iceland", list(map(str, range(1980,2014))) ]
        df2.plot(kind='bar')

        plt.title("Immigration from Iceland")
        plt.ylabel("Number of Immigrant")
        plt.xlabel("Year")
        plt.show()
```



Questions

Links

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

References

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