Basic plots with Matplotlib

INTERMEDIATE PYTHON

Hugo Bowne-Anderson
Data Scientist at DataCamp

datacamp

Basic plots with Matplotlib

- Visualization

- Data Structure

- Control Structures

- Case Study

Data visualization

- Very important in Data Analysis
  - Explore data
  - Report insights

Gapminder World Map 2010
Low–income countries
Middle–income countries
High–income countries
Healthy
Poor
Rich
Sick
Health Life expectancy at birth (years)
Money GDP per person in US dollars (purchasing power adjusted) (log scale)
85
80
75
70
65
60
55
50
45
500
1 000
2 000
5 000
10 000
20 000
50 000
China
India
Japan
USA
Russia
Brazil
Indonesia
Nigeria
Mexico
Bangladesh
Pakistan
Iran
Turkey
Egypt
Thailand
Ukraine
Philippines
Vietnam
Germany
France
Italy
Spain
Poland
Canada
Australia
South Africa
Kenya
Ethiopia
Congo, DR
Zimbabwe
Afghanistan
Angola
Zambia
Mozambique
Madagascar
Cameroon
Chad
Mali
Niger
Burkina Faso
Uganda
Malawi
Sudan
Saudi Arabia
Kazakhstan
Belarus
Venezuela
Argentina
Colombia
Peru
Algeria
Tunisia
Syria
Sri Lanka
Nepal
North Korea
Myanmar
Cambodia
Yemen
Guinea
Côte d'Ivoire
Ghana
Tanzania
Senegal
Rwanda
Burundi
Somalia
Liberia
Togo
Eritrea
Haiti
Comoros
Lesotho
Swaziland
Botswana
Namibia
Gabon
Congo, Rep.
Equatorial Guinea
Mauritania
Djibouti
Guinea-Bissau
Sierra Leone
Central African Rep.
Papua New Guinea
Timor-Leste
Solomon Isl.
Mongolia
Bolivia
Bhutan
Turkmenistan
Nauru
Kiribati
Guyana
Suriname
Azerbaijan
Jamaica
Mauritius
Lithuania
Latvia
Estonia
Seychelles
Slovakia
Croatia
Czech Rep.
Portugal
Greece
Netherlands
Ireland
Austria
Belgium
Denmark
Sweden
Norway
Finland
Israel
New Zealand
Malta
Taiwan
UAE
Kuwait
Brunei
Qatar
Bahrain
Oman
Bahamas
Trinidad & Tobago
Antigua & Barbuda
Hungary
Malaysia
Libya
Serbia
Macedonia
Ecuador
Bosnia & H.
Albania
Belize
Cuba
Costa Rica
Chile
Puerto Rico
Uruguay
Panama
Grenada
Dominica
Iceland
Hong Kong
Andorra
Switzerland
Singapore
Liechtenstein
Luxembourg
Kosovo
Nicaragua
Palestine
Micronesia
Marshall Isl.
Tuvalu
Kyrgyzstan
Moldova
Uzbekistan
Tajikistan
Laos
São Tomé & P.
Fiji
Vanuatu
Guatemala
Honduras
Paraguay
Jordan
Tonga
Samoa
El Salvador
Palau
Cape Verde
Maldives
Georgia
Morocco
Iraq
Libya
Romania
Latvia
Lebanon
Sri Lanka
Sri Kim & N.
Armenia
Belize
Albania
Dominica
Panama
Uruguay
Argentina
Venezuela
Colombia
Peru
Algeria
Tunisia
Syria
Brazil
Mexico
Japan
France
Germany

[1] Source: GapMinder, Wealth and Health of Nations

[1] Source: GapMinder, Wealth and Health of Nations

Matplotlib

```
import matplotlib.pyplot as plt
year = [1950, 1970, 1990, 2010]
pop = [2.519, 3.692, 5.263, 6.972]
plt.plot(year, pop)
plt.show()
```

Matplotlib

Matplotlib

```
year = [1950 , 1970 , 1990 ,  2010]
pop  = [2.519, 3.692, 5.263, 6.972]
```

Scatter plot

```
import matplotlib.pyplot as plt
year = [1950, 1970, 1990, 2010]
pop = [2.519, 3.692, 5.263, 6.972]
plt.plot(year, pop)
plt.show()
```

Scatter plot

```
import matplotlib.pyplot as plt
year = [1950, 1970, 1990, 2010]
pop = [2.519, 3.692, 5.263, 6.972]
plt.scatter(year, pop)
plt.show()
```

Let's practice!

INTERMEDIATE PYTHON

Histogram

INTERMEDIATE PYTHON

Hugo Bowne-Anderson
Data Scientist at DataCamp

datacamp

Histogram

- Explore dataset
- Get idea about distribution

Histogram

- Explore dataset
- Get idea about distribution

Histogram

- Explore dataset
- Get idea about distribution

Histogram

- Explore dataset
- Get idea about distribution

Histogram

- Explore dataset
- Get idea about distribution

Histogram

- Explore dataset
- Get idea about distribution

Histogram

- Explore dataset
- Get idea about distribution

Matplotlib

```python
import matplotlib.pyplot as plt
```

```python
help(plt.hist)
```

```
Help on function hist in module matplotlib.pyplot:
hist(x, bins=None, range=None, density=False, weights=None,
cumulative=False, bottom=None, histtype='bar', align='mid',
orientation='vertical', rwidth=None, log=False, color=None,
label=None, stacked=False, *, data=None, **kwargs)
    Plot a histogram.

    Compute and draw the histogram of *x*.  The return value is a
    tuple (*n*, *bins*, *patches*) or ([*n0*, *n1*, ...],
    *bins*, [*patches0*, *patches1*, ...]) if the input contains
    multiple data.
```

Matplotlib example

```
values = [0,0.6,1.4,1.6,2.2,2.5,2.6,3.2,3.5,3.9,4.2,6]
plt.hist(values, bins=3)
plt.show()
```

Population pyramid

Population pyramid

Let's practice!

INTERMEDIATE PYTHON

Customization

INTERMEDIATE PYTHON

Hugo Bowne-Anderson
Data Scientist at DataCamp

datacamp

Data visualization

- Many options
  - Different plot types
  - Many customizations
- Choice depends on
  - Data
  - Story you want to tell

Basic plot

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.show()
```

Axis labels

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')

plt.show()
```

Axis labels

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')

plt.show()
```

Title

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')

plt.show()
```

Title

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')

plt.show()
```

Ticks

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10])

plt.show()
```

Ticks

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10])

plt.show()
```

Ticks (2)

`population.py`

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10],
           ['0', '2B', '4B', '6B', '8B', '10B'])

plt.show()
```

Ticks (2)

`population.py`

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10],
           ['0', '2B', '4B', '6B', '8B', '10B'])

plt.show()
```

Add historical data

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

# Add more data
year = [1800, 1850, 1900] + year
pop = [1.0, 1.262, 1.650] + pop

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10],
           ['0', '2B', '4B', '6B', '8B', '10B'])

plt.show()
```

Add historical data

population.py

```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

# Add more data
year = [1800, 1850, 1900] + year
pop = [1.0, 1.262, 1.650] + pop

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10],
           ['0', '2B', '4B', '6B', '8B', '10B'])

plt.show()
```

Before vs. after

Let's practice!

INTERMEDIATE PYTHON