



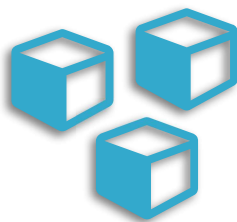
INTERMEDIATE PYTHON FOR DATA SCIENCE

# **Basic Plots with Matplotlib**

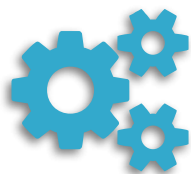
- Visualization



- Data Structures



- Control Structures



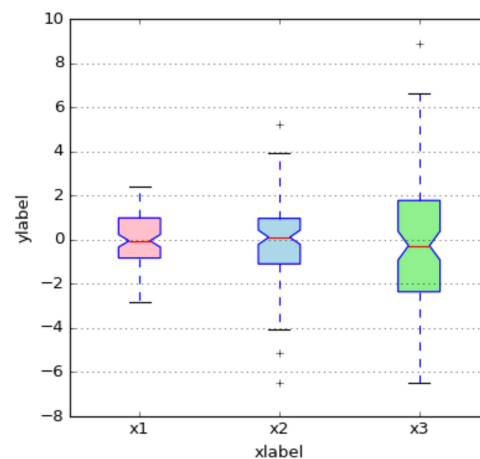
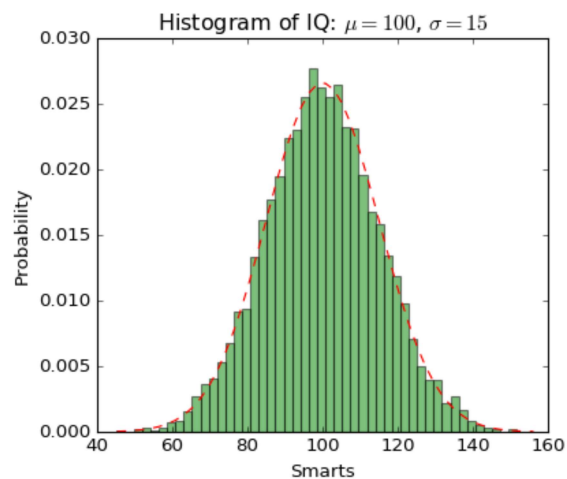
- Case Study

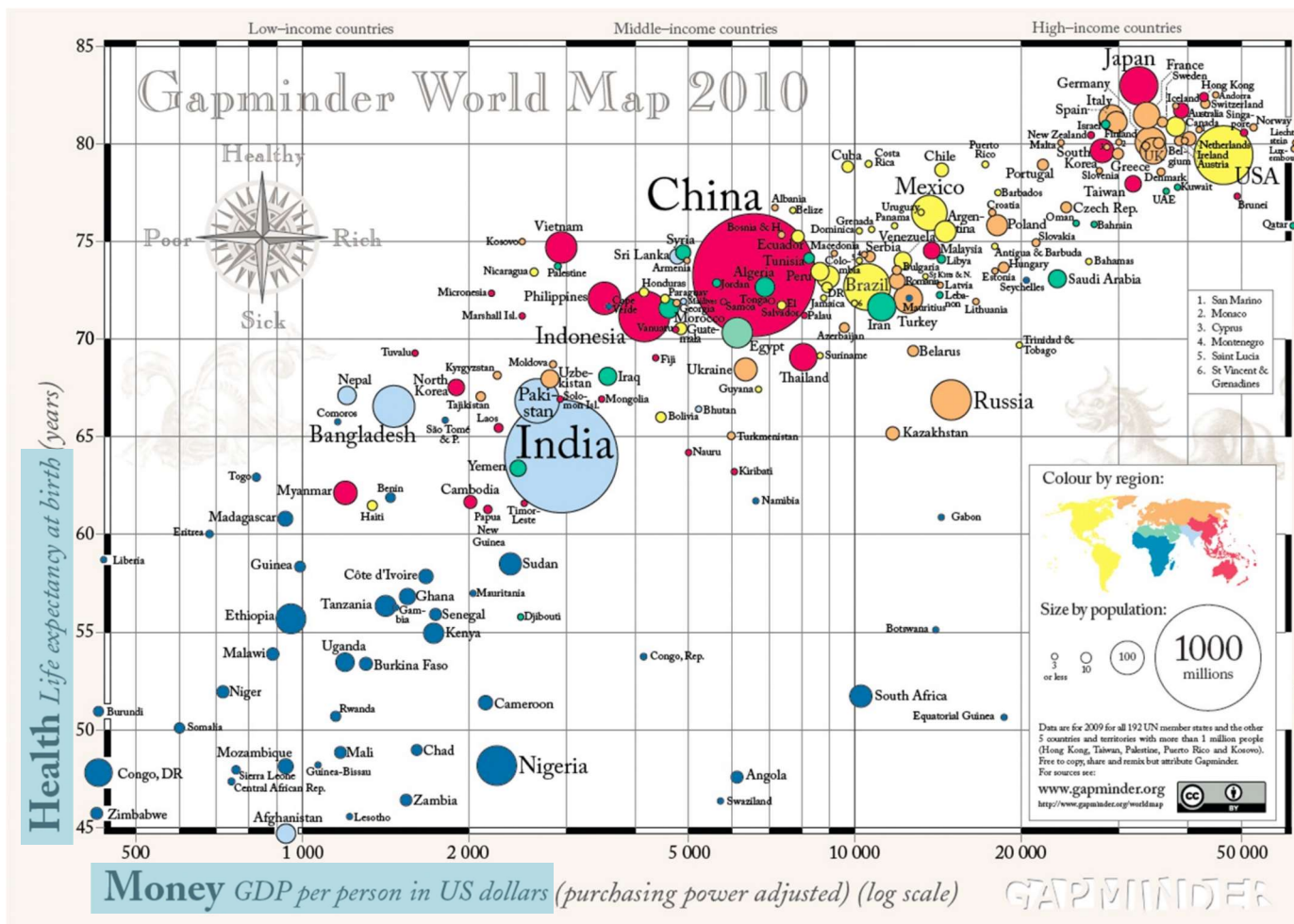




# Data Visualization

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



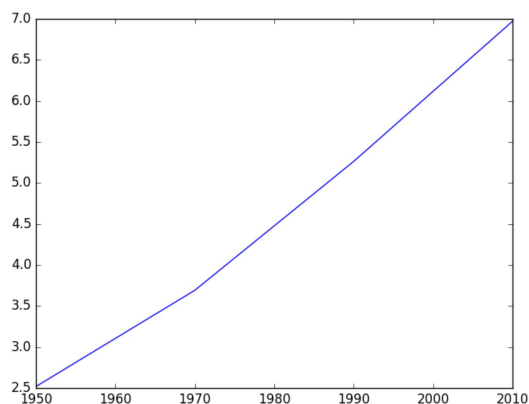


Source: GapMinder, Wealth and Health of Nations



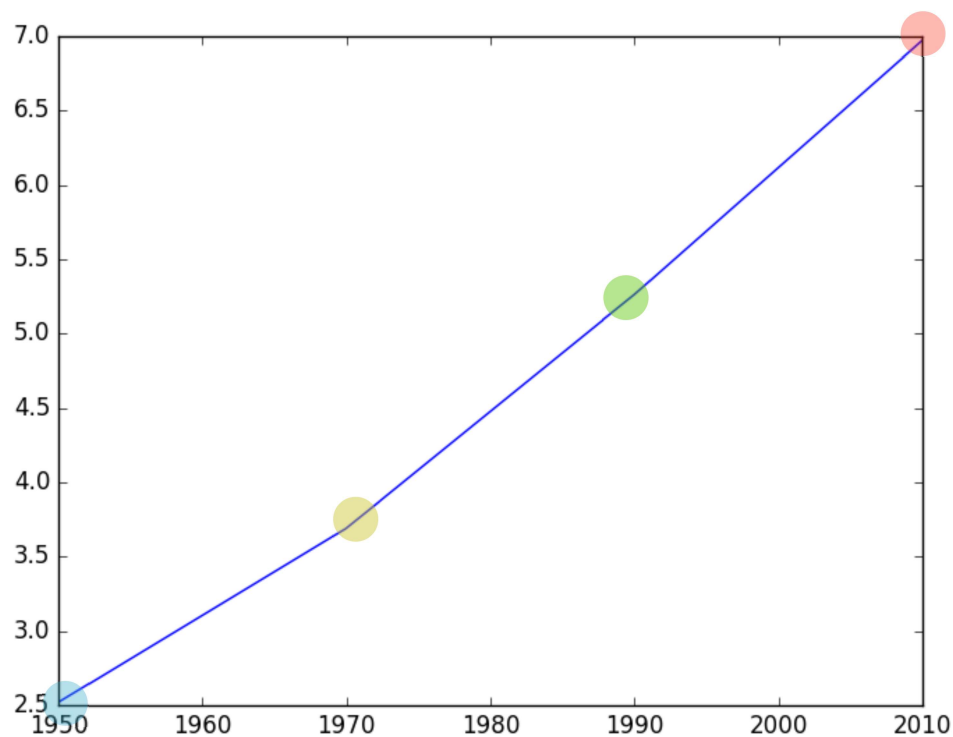
# Matplotlib

```
In [1]: import matplotlib.pyplot as plt  
  
In [2]: year = [1950, 1970, 1990, 2010]  
  
In [3]: pop = [2.519, 3.692, 5.263, 6.972]  
  
In [4]: plt.plot(year, pop)  
           x      y  
  
In [5]: plt.show()
```



# Matplotlib

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



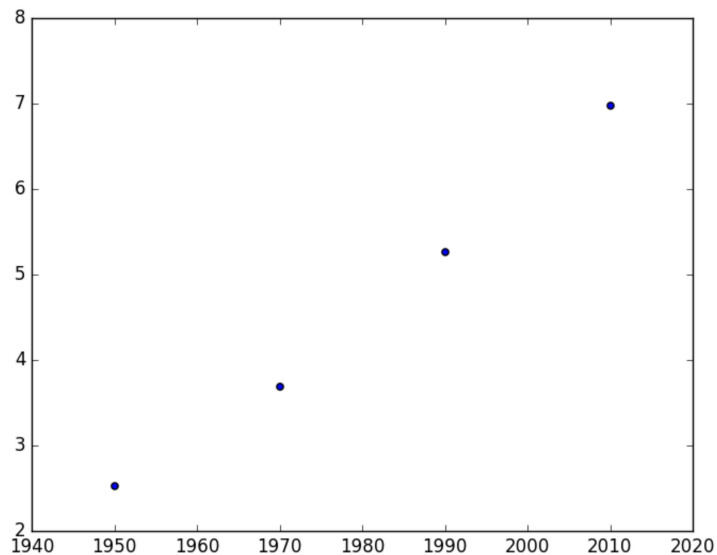
# Scatter plot

```
In [1]: import matplotlib.pyplot as plt
In [2]: year = [1950, 1970, 1990, 2010]
In [3]: pop = [2.519, 3.692, 5.263, 6.972]
In [4]: plt.plot(year, pop)
In [5]: plt.show()
```



# Scatter plot

```
In [1]: import matplotlib.pyplot as plt  
In [2]: year = [1950, 1970, 1990, 2010]  
In [3]: pop = [2.519, 3.692, 5.263, 6.972]  
In [4]: plt.scatter(year, pop)  
In [5]: plt.show()
```







INTERMEDIATE PYTHON FOR DATA SCIENCE

**Let's practice!**



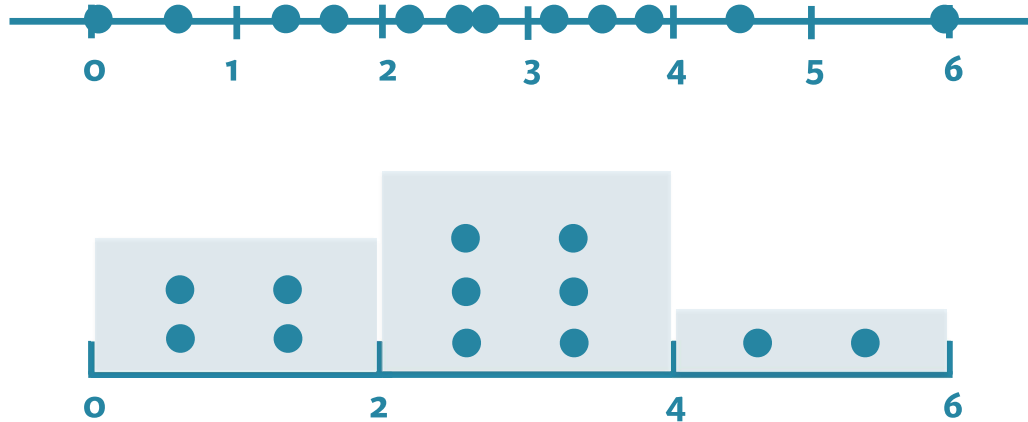
INTERMEDIATE PYTHON FOR DATA SCIENCE

# Histogram



# Histogram

- Explore dataset
- Get idea about distribution





# Matplotlib

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

```
In [2]: help(plt.hist)
```

Help on function hist in module matplotlib.pyplot:

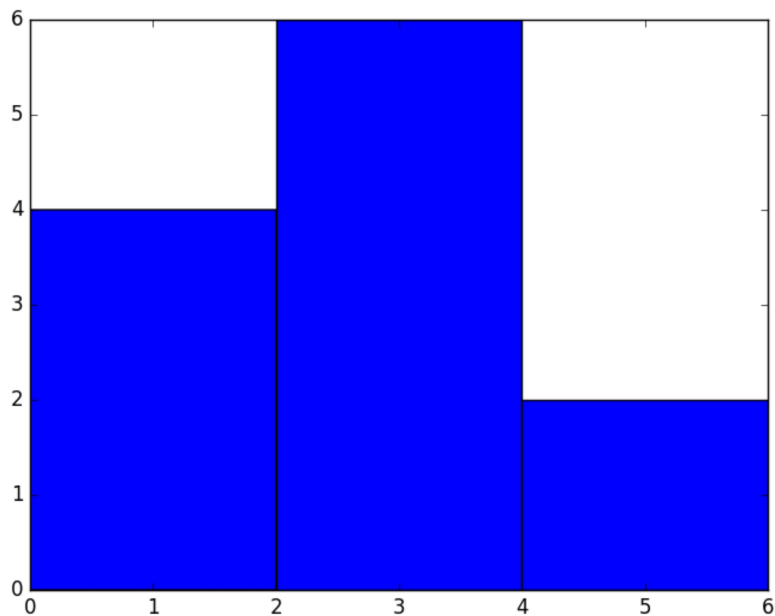
```
hist(x, bins=10, range=None, normed=False, weights=None,
cumulative=False, bottom=None, histtype='bar', align='mid',
orientation='vertical', rwidth=None, log=False, color=None,
label=None, stacked=False, hold=None, data=None, **kwargs)
    Plot a histogram.
```

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

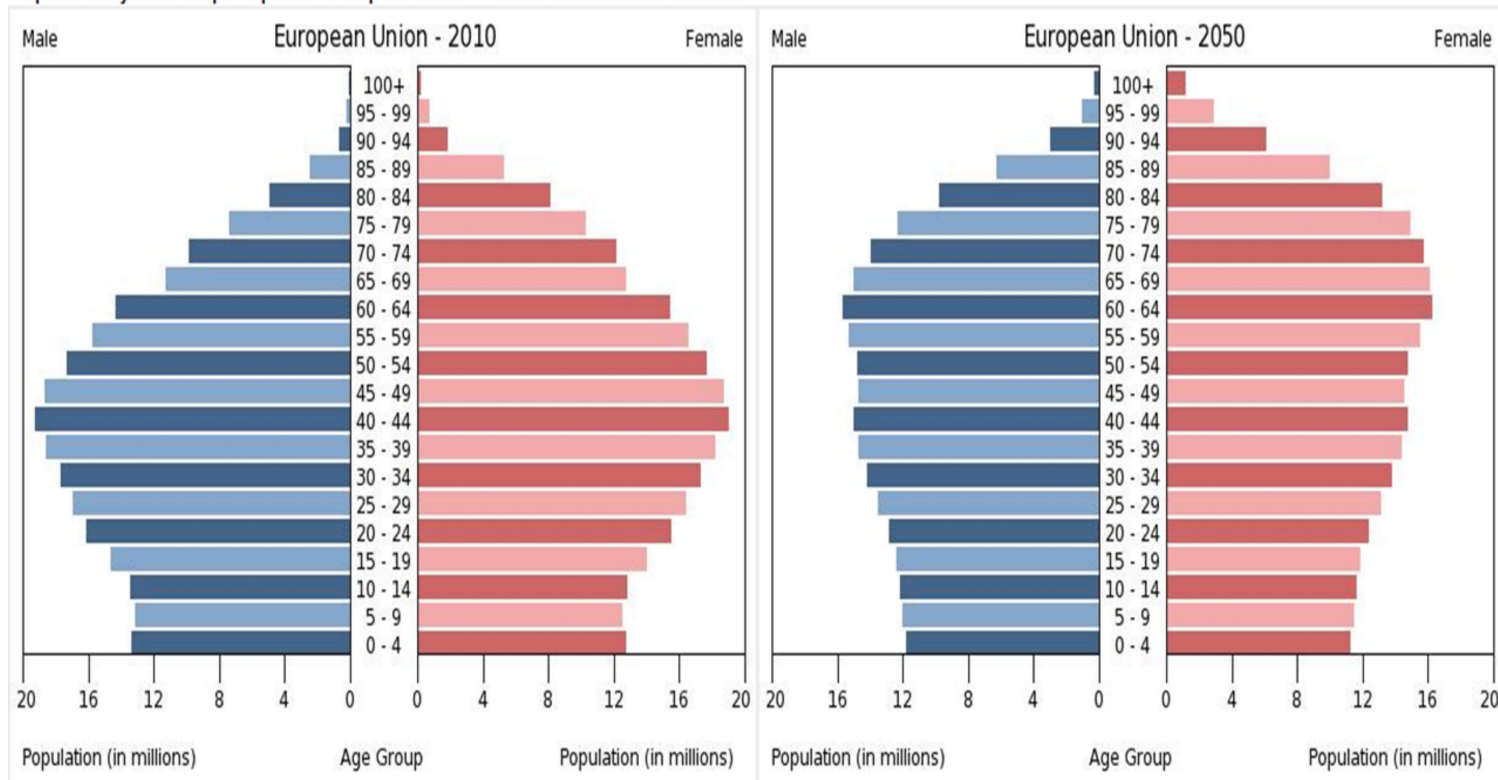
```
In [3]: values = [0,0.6,1.4,1.6,2.2,2.5,2.6,3.2,3.5,3.9,4.2,6]  
In [4]: plt.hist(values, bins = 3)  
In [5]: plt.show()
```





# Population Pyramid

Population Pyramid Graph - Special - European Union - TOTAL FOR SELECTED REGION





INTERMEDIATE PYTHON FOR DATA SCIENCE

**Let's practice!**



INTERMEDIATE PYTHON FOR DATA SCIENCE

# Customization



# Data Visualization

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



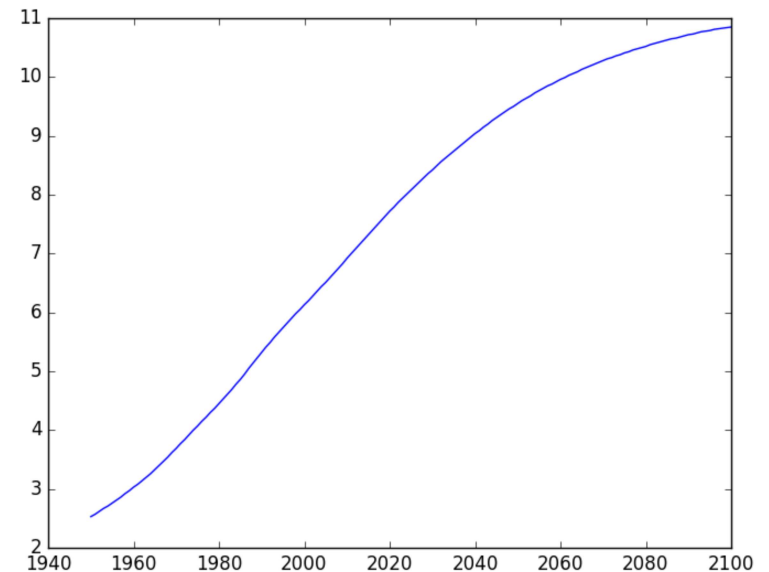
# Basic Plot

 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

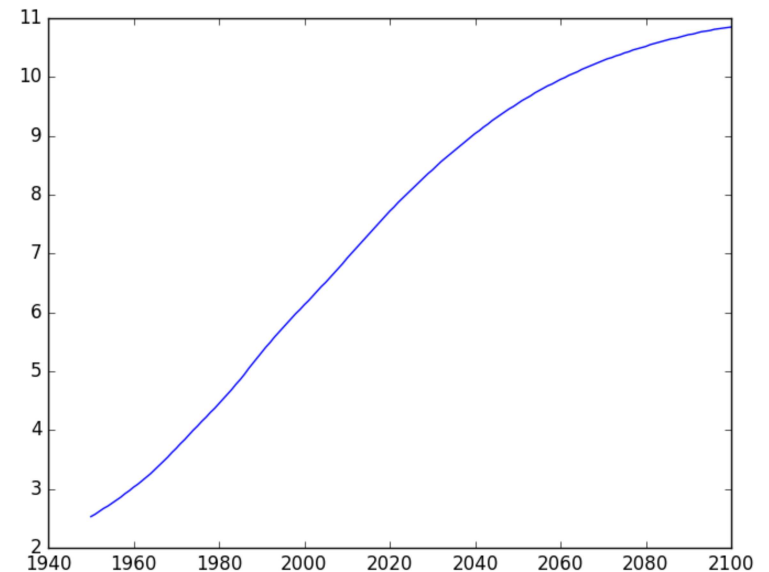
 population.py

```
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

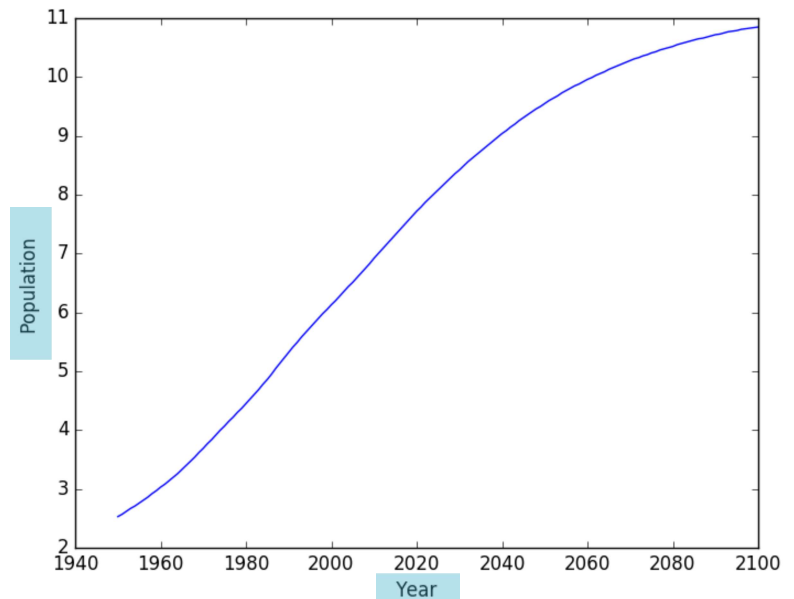
 population.py

```
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

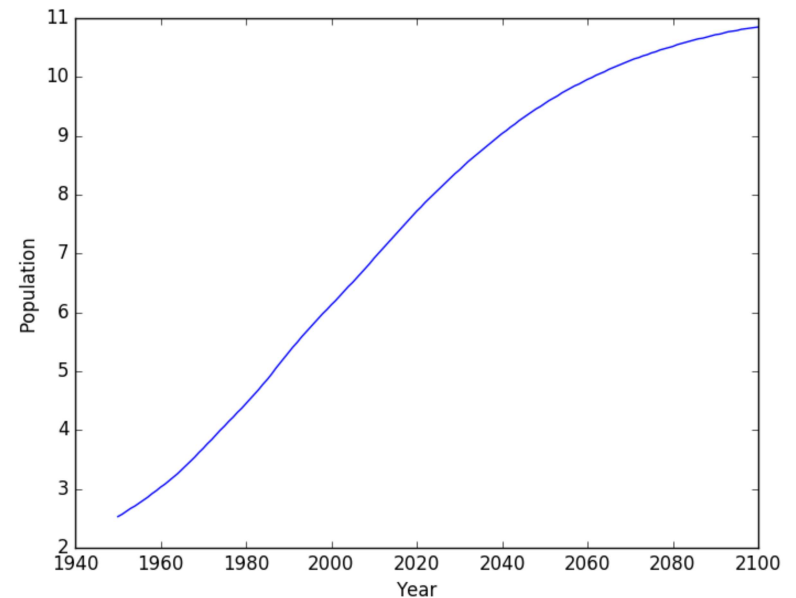
 population.py

```
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

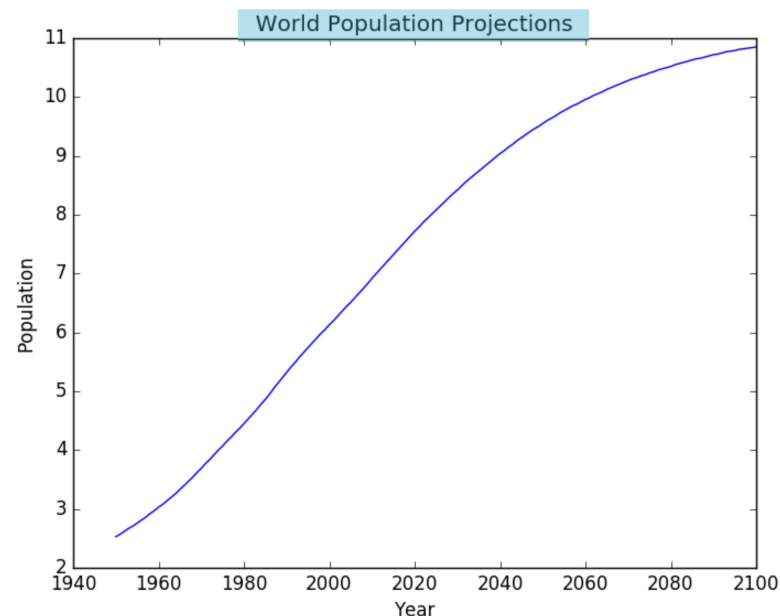
 population.py

```
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

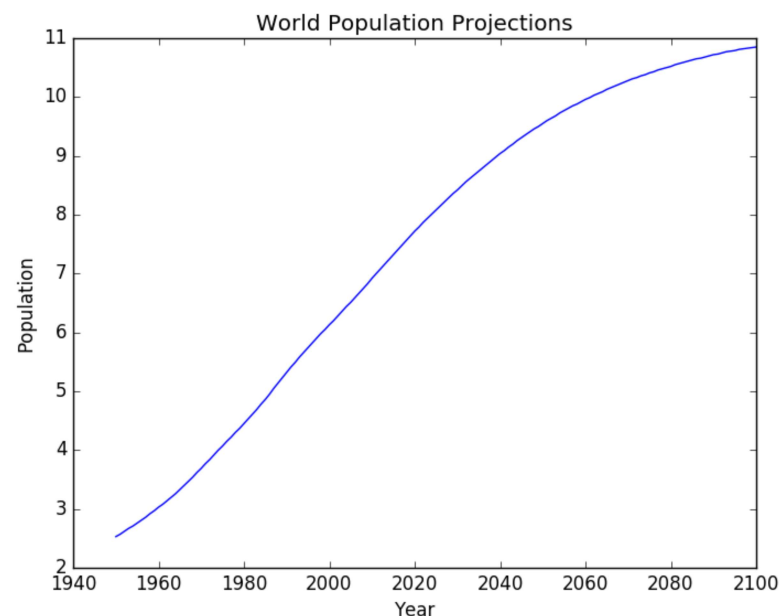
 population.py

```
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

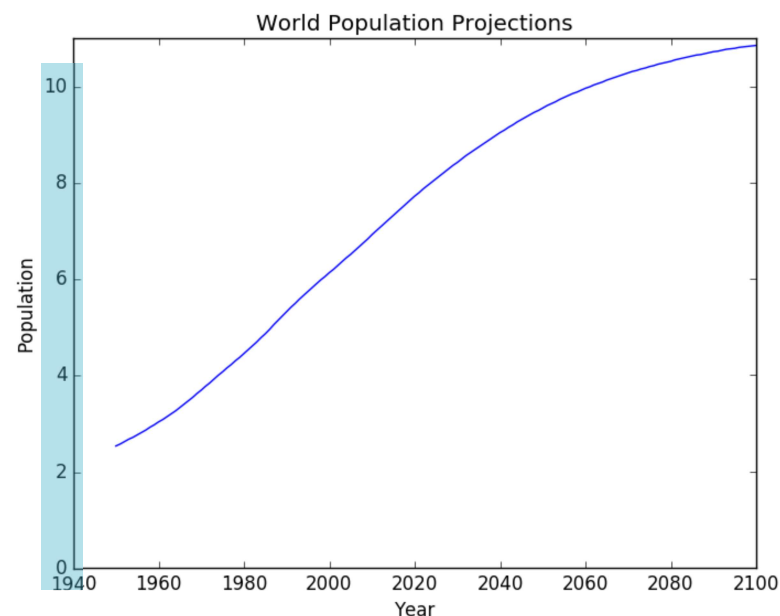
 population.py

```
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)

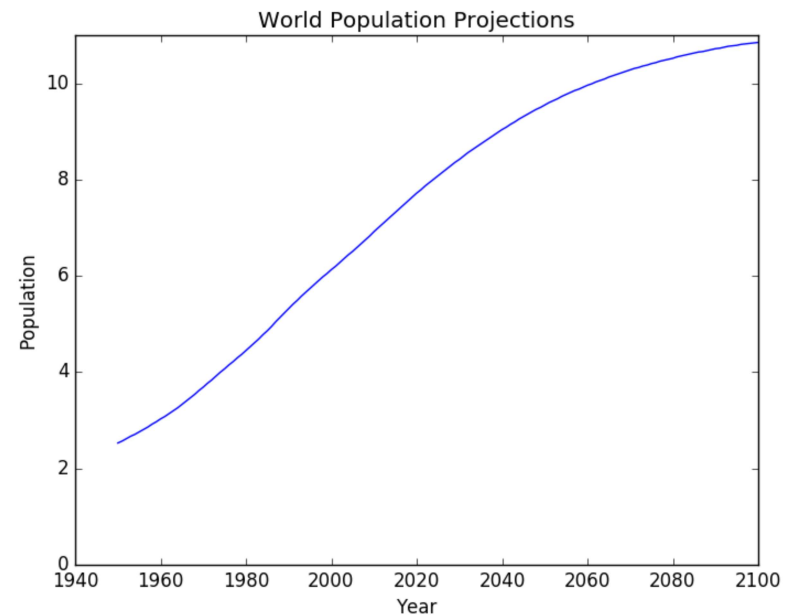
 population.py

```
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)

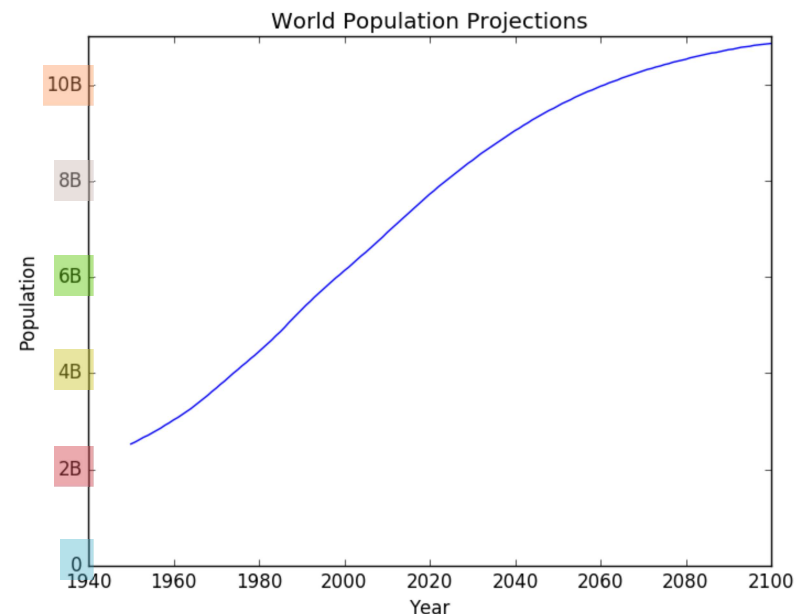
`population.py`

```
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

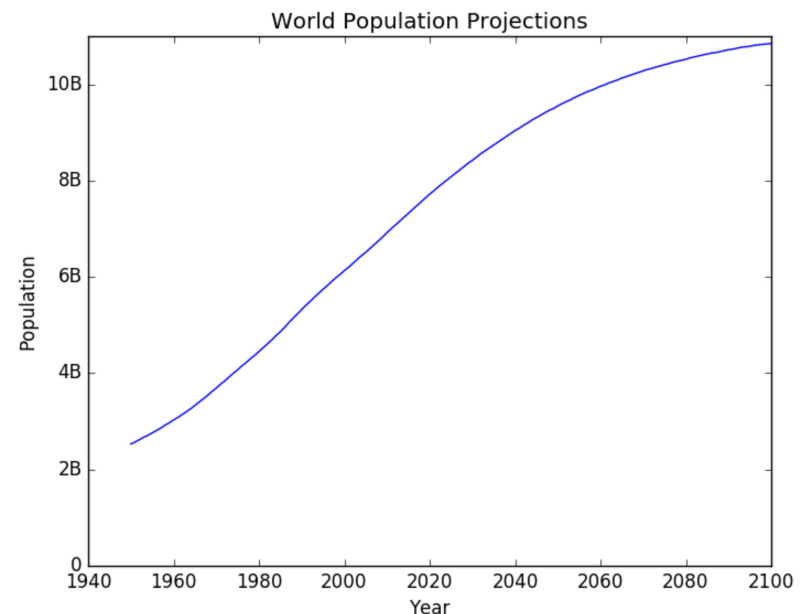
```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

# 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

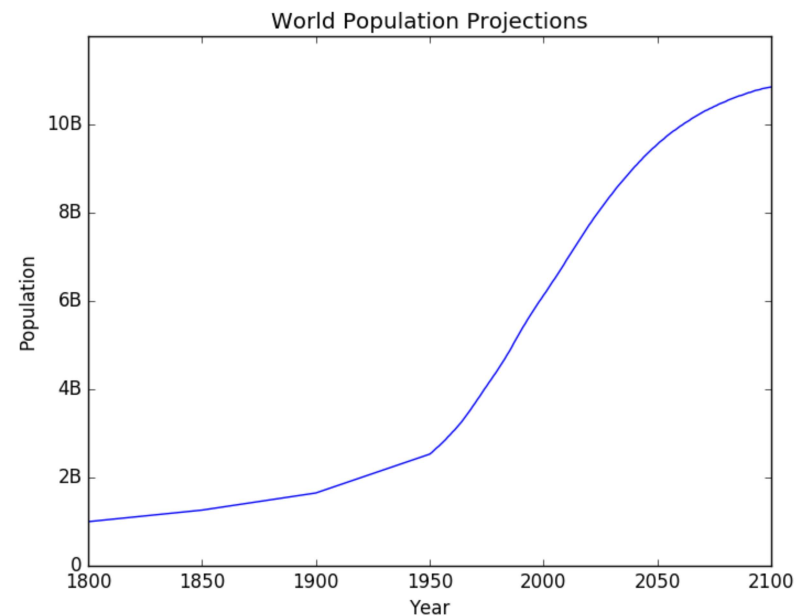
```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

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

plt.plot(year, pop)

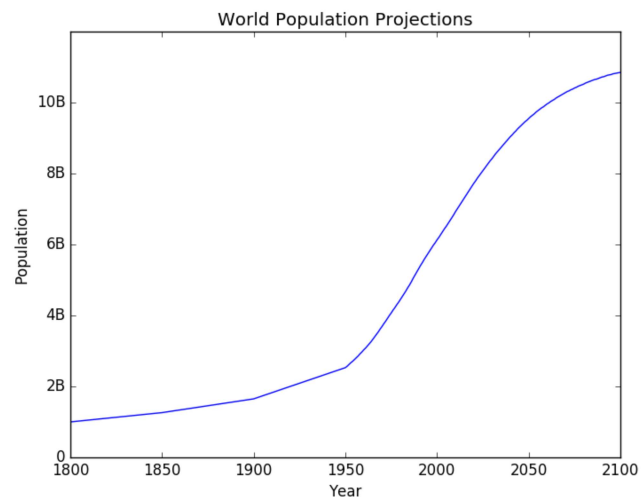
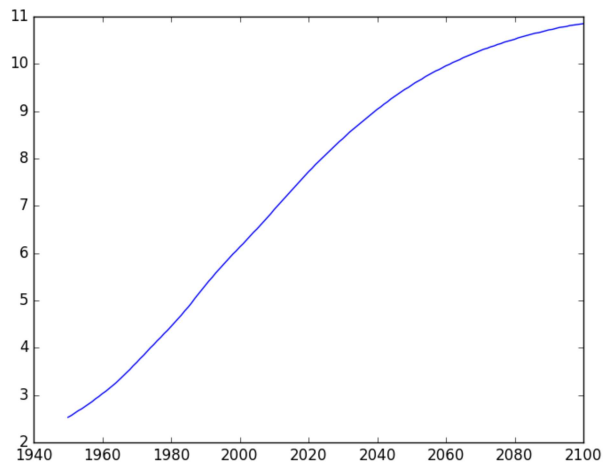
plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10],
           ['0', '2B', '4B', '6B', '8B', '10B'])

plt.show()
```





# Before vs After





INTERMEDIATE PYTHON FOR DATA SCIENCE

**Let's practice!**