Lab 2 Worksheet
Practicing Visualisation Techniques with Python

Activity 1: Creating a simple visualisation

In this activity, we will create a simple plot using Matplotlib.

1. Import statements

Import the necessary modules and enable plotting within a jupyter notebook.

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

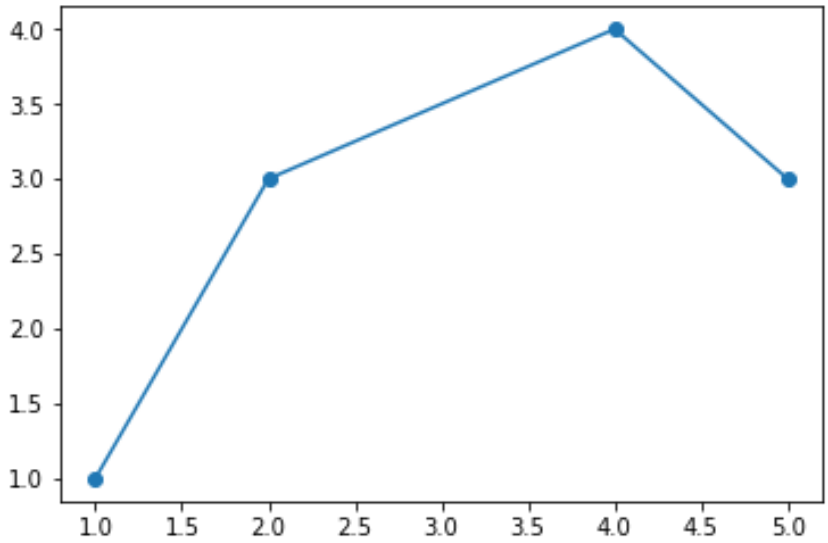
2. Creating a figure

Explicitly create a figure and set the dpi to 200.

```
plt.figure(dpi=200)
```

3. Plotting data pairs

Question 1) Plot the following data pairs (x, y) as circles, which are connected via line segments: (1, 1), (2, 3), (4, 4), (5, 3). Visualise the plot.



Activity 2: Pie Plot

Water usage

1. Use pandas to read the data located in the subfolder data.

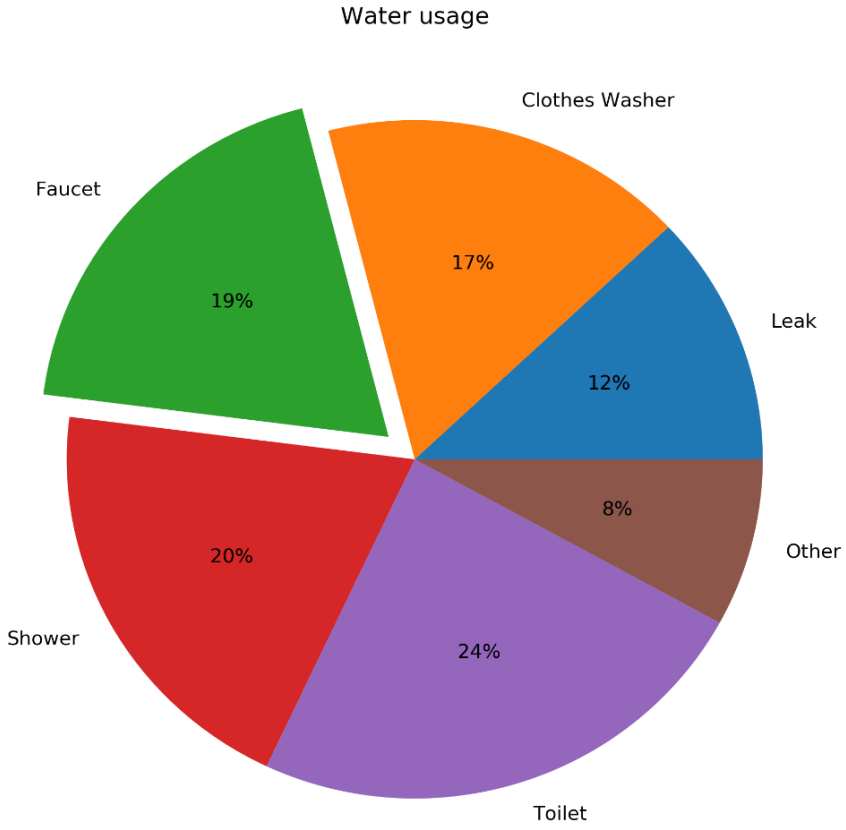
```
# Load dataset
data = pd.read_csv('./data/water_usage.csv')
```

2. Use a pie chart to visualize the water usage. Highlight one usage of your choice using the explode parameter. Show the percentages for each slice and add a title.

```
# Create figure
plt.figure(figsize=(8, 8), dpi=300)
Question 2) Create pie plot

Question 3) Add title

# Show plot
plt.show()
```



Activity 3: Histogram and Box Plots

In this activity, we will visualise the intelligent quotient (IQ) using a histogram and box plots.

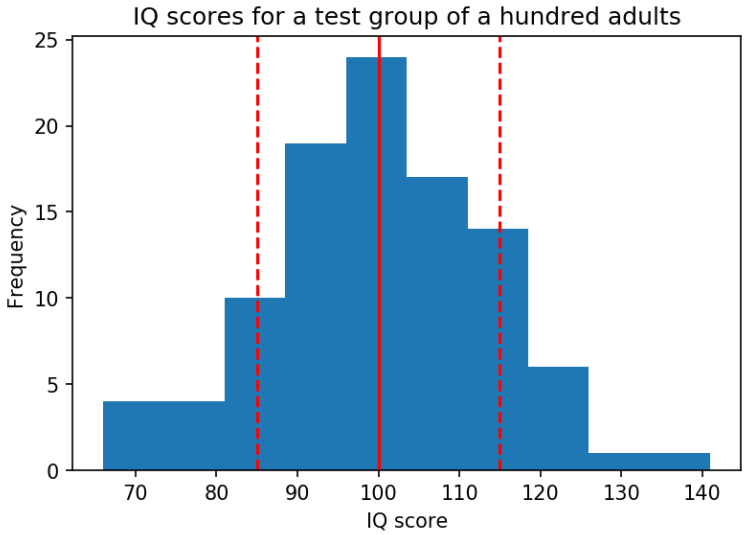
```
# IQ samples
iq_scores = [126,  89,  90, 101, 102,  74,  93, 101,  66, 120, 108,  97,  98,
             105, 119,  92, 113,  81, 104, 108,  83, 102, 105, 111, 102, 107,
             103,  89,  89, 110,  71, 110, 120,  85, 111,  83, 122, 120, 102,
             84, 118, 100, 100, 114,  81, 109,  69,  97,  95, 106, 116, 109,
             114,  98,  90,  92,  98,  91,  81,  85,  86, 102,  93, 112,  76,
             89, 110,  75, 100,  90,  96,  94, 107, 108,  95,  96,  96, 114,
             93,  95, 117, 141, 115,  95,  86, 100, 121, 103,  66,  99,  96,
             111, 110, 105, 110,  91, 112, 102, 112,  75]
```

1. Plot a histogram with ten bins for the given IQ scores. IQ scores are normally distributed with a mean of 100 and a standard deviation of 15. Visualize the mean as a vertical solid red line, and the standard deviation using dashed vertical lines. Add labels and a title.

```
# Create figure
plt.figure(figsize=(6, 4), dpi=150)

Question 3) Create histogram with 10 bins

# Visualize the mean and standard deviation
plt.axvline(x=100, color='r')
plt.axvline(x=115, color='r', linestyle= '--')
plt.axvline(x=85, color='r', linestyle= '--')
# Add labels and title
plt.xlabel('IQ score')
plt.ylabel('Frequency')
plt.title('IQ scores for a test group of a hundred adults')
# Show plot
plt.show()
```

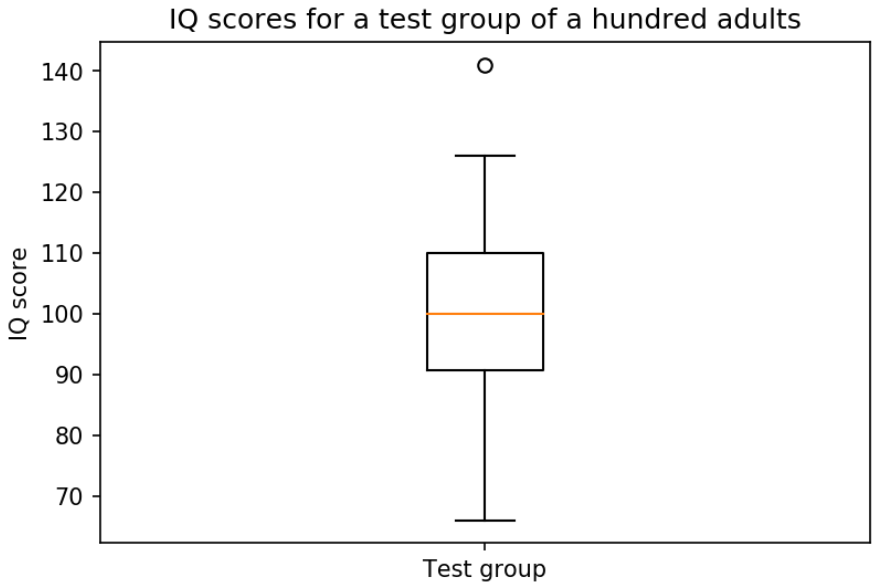


3. Create a box plot to visualise the same IQ scores. Add labels and a title.

```
Question 4) Create figure of size 6x4 and 150 dpi


Question 5) Create histogram

# Add labels and title
ax = plt.gca()
ax.set_xticklabels(['Test group'])
plt.ylabel('IQ score')
plt.title('IQ scores for a test group of a hundred adults')
# Show plot
plt.show()
```




```
group_a = [118, 103, 125, 107, 111,  96, 104,  97,  96, 114,  96,  75, 114,
       107,  87, 117, 117, 114, 117, 112, 107, 133,  94,  91, 118, 110,
       117,  86, 143,  83, 106,  86,  98, 126, 109,  91, 112, 120, 108,
       111, 107,  98,  89, 113, 117,  81, 113, 112,  84, 115,  96,  93,
       128, 115, 138, 121,  87, 112, 110,  79, 100,  84, 115,  93, 108,
       130, 107, 106, 106, 101, 117,  93,  94, 103, 112,  98, 103,  70,
       139,  94, 110, 105, 122,  94,  94, 105, 129, 110, 112,  97, 109,
       121, 106, 118, 131,  88, 122, 125,  93,  78]
group_b = [126,  89,  90, 101, 102,  74,  93, 101,  66, 120, 108,  97,  98,
           105, 119,  92, 113,  81, 104, 108,  83, 102, 105, 111, 102, 107,
           103,  89,  89, 110,  71, 110, 120,  85, 111,  83, 122, 120, 102,
           84, 118, 100, 100, 114,  81, 109,  69,  97,  95, 106, 116, 109,
           114,  98,  90,  92,  98,  91,  81,  85,  86, 102,  93, 112,  76,
           89, 110,  75, 100,  90,  96,  94, 107, 108,  95,  96,  96, 114,
           93,  95, 117, 141, 115,  95,  86, 100, 121, 103,  66,  99,  96,
           111, 110, 105, 110,  91, 112, 102, 112,  75]
group_c = [108,  89, 114, 116, 126, 104, 113,  96,  69, 121, 109, 102, 107,
       122, 104, 107, 108, 137, 107, 116,  98, 132, 108, 114,  82,  93,
```

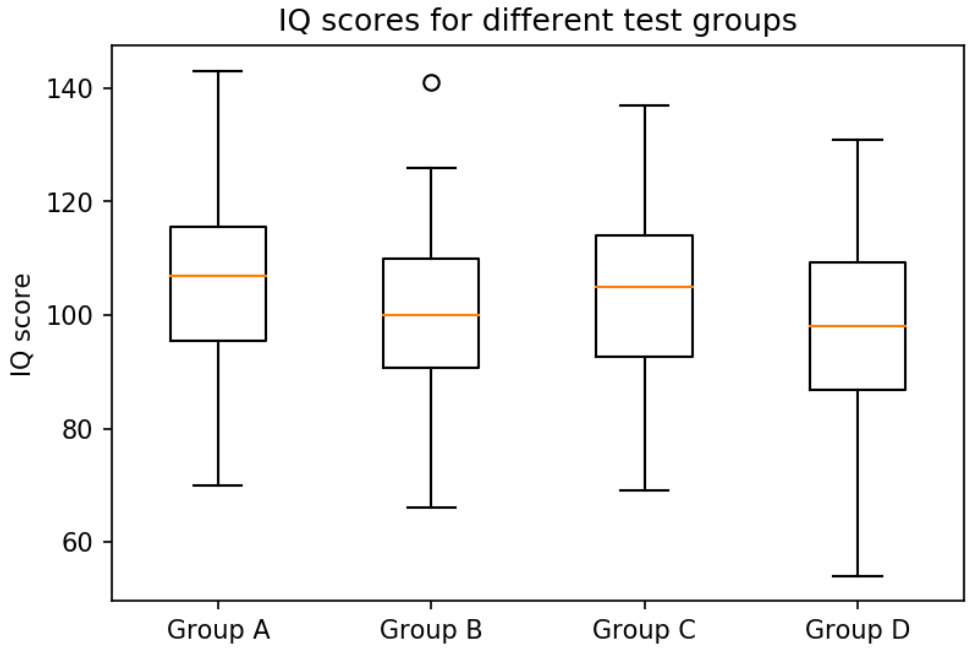
```
         89,  90,  86,  91,  99,  98,  83,  93, 114,  96,  95, 113, 103,
         81, 107,  85, 116,  85, 107, 125, 126, 123, 122, 124, 115, 114,
         93,  93, 114, 107, 107,  84, 131,  91, 108, 127, 112, 106, 115,
         82,  90, 117, 108, 115, 113, 108, 104, 103,  90, 110, 114,  92,
        101,  72, 109,  94, 122,  90, 102,  86, 119, 103, 110,  96,  90,
        110,  96,  69,  85, 102,  69,  96, 101,  90]
group_d = [ 93,  99,  91, 110,  80, 113, 111, 115,  98,  74,  96,  80,  83,
        102,  60,  91,  82,  90,  97, 101,  89,  89, 117,  91, 104, 104,
        102, 128, 106, 111,  79,  92,  97, 101, 106, 110,  93,  93, 106,
        108,  85,  83, 108,  94,  79,  87, 113, 112, 111, 111,  79, 116,
        104,  84, 116, 111, 103, 103, 112,  68,  54,  80,  86, 119,  81,
         84,  91,  96, 116, 125,  99,  58, 102,  77,  98, 100,  90, 106,
        109, 114, 102, 102, 112, 103,  98,  96,  85,  97, 110, 131,  92,
         79, 115, 122,  95, 105,  74,  85,  85,  95]
```

3. Create a box plot for each IQ score of different test groups. Add labels and a title.

```
# Create figure
plt.figure(figsize=(6, 4), dpi=150)

Question 6) Create histogram

# Add labels and title
ax = plt.gca()
ax.set_xticklabels(['Group A', 'Group B', 'Group C', 'Group D'])
plt.ylabel('IQ score')
plt.title('IQ scores for different test groups')
# Show plot
plt.show()
```



Activity 4: Scatter Plot

Animal statistics

In this activity, we will use a scatter plot to show correlation within a dataset. Let's look at the following scenario: You are given a dataset containing information about various animals. Visualise correlation between animal attributes.

```
# Load dataset
data = pd.read_csv('./data/anage_data.csv')
```

1. The given dataset is not complete. Filter the data so you end up with samples containing a body mass and maximum longevity. Sort the data according to the animal class.

```
# Preprocessing
longevity = 'Maximum longevity (yrs)'
mass = 'Body mass (g)'
data = data[np.isfinite(data[longevity]) & np.isfinite(data[mass])]
# Sort according to class
amphibia = data[data['Class'] == 'Amphibia']
aves = data[data['Class'] == 'Aves']
mammalia = data[data['Class'] == 'Mammalia']
reptilia = data[data['Class'] == 'Reptilia']
```

2. Create a scatter plot visualising the correlation between the body mass and maximum longevity. Use different colours for grouping data samples according to their class. Add a legend, labels and a title. Use a log scale for both the x-axis and y-axis.

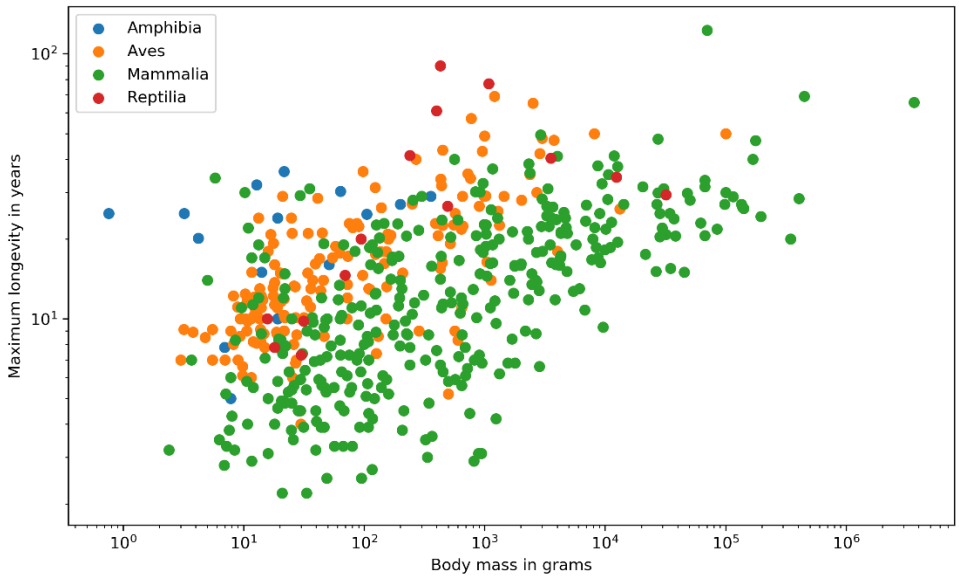
```
# Create figure
plt.figure(figsize=(10, 6), dpi=300)

Question 7) Create scatter plot

# Add legend
plt.legend()

Question 8) set x and y Log scales


# Add labels
plt.xlabel('Body mass in grams')
plt.ylabel('Maximum longevity in years')
# Show plot
plt.show()
```



Activity 5: Plotting Trends

Visualising stock trends

In this activity, we will create a line plot to show stock trends. Let's look at the following scenario: You are interested in investing in stocks. You downloaded the stock prices for the "big five": Amazon, Google, Apple, Facebook, and Microsoft. In this activity, we will create a line plot to show stock trends. Let's look at the following scenario: You are interested in investing in stocks. You downloaded the stock prices for the "big five": Amazon, Google, Apple, Facebook, and Microsoft.

1. Use pandas to read the data located in the subfolder data.

```
# load datasets
google = pd.read_csv('./data/GOOGL_data.csv')
facebook = pd.read_csv('./data/FB_data.csv')
apple = pd.read_csv('./data/AAPL_data.csv')
amazon = pd.read_csv('./data/AMZN_data.csv')
microsoft = pd.read_csv('./data/MSFT_data.csv')
```

2. Use Matplotlib to create a line chart visualising the closing prices for the past five years (whole data sequence) for all five companies. Add labels, titles, and a legend to make the visualisation self-explanatory. Use plt.grid() to add a grid to your plot.

In [3]:

```
# Create figure
plt.figure(figsize=(16, 8), dpi=300)
# Plot data
```

```
plt.plot('date', 'close', data=google, label='Google')
plt.plot('date', 'close', data=facebook, label='Facebook')
plt.plot('date', 'close', data=apple, label='Apple')
plt.plot('date', 'close', data=amazon, label='Amazon')
plt.plot('date', 'close', data=microsoft, label='Microsoft')

Question 9) Specify ticks for x- and y-axis


# Add title and label for y-axis
plt.title('Stock trend', fontsize=16)
plt.ylabel('Closing price in $', fontsize=14)
# Add grid
plt.grid()

Question 10) Add legend

# Show plot
plt.show()
```