Lab 3 Solutions

Summer Session A, 2023, Ethan M.

July 8, 2023

1 Task 1

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

[2]:
```
np.sin(0)
```

[2]:
```
0.0
```

[3]:
```
import datascience as ds
```

[4]:
```
ds.Table().with_columns(
    "Col1", [1, 2, 3],
    "Col2", [2, 3, 4]
)
```

[4]:
```
Col1 | Col2
1    | 2
2    | 3
3    | 4
```

2 Task 2

[5]:
```
x_list = [1, 2, 3]
x_array = ds.make_array([1, 2, 3])
```

[6]:
```
np.mean(x_list)
```

[6]:
```
2.0
```

[7]:
```
np.mean(x_array)
```

[7]:
```
2.0
```

3 Task 3

`np.ptp()` computes the **range** of a given dataset. The abbreviation `ptp` stands for 'peak-to-peak'.

[10]:
```
np.ptp(x_list) # should be 3 - 1 = 2
```

[10]: 2

[11]:
```
np.ptp(x_array) # should be 3 - 1 = 2
```

[11]: 2

4 Task 4

$$\begin{aligned} s_X^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \\ &= \frac{1}{3-1}\left[(1-2)^2 + (2-2)^2 + (3-2)^2\right] \\ &= \frac{1}{2}(1+1) = 1 \quad \Longrightarrow \quad \boxed{s_X = 1} \end{aligned}$$

[12]:
```
np.std(x_list)
```

[12]: 0.816496580927726O3

This answer does **not** agree with what we obtained by hand.

$$\begin{aligned} (s'_X)^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 \\ &= \frac{1}{3}\left[(1-2)^2 + (2-2)^2 + (3-2)^2\right] \\ &= \frac{1}{3}(1+1) = \frac{2}{3} \quad \Longrightarrow \quad \boxed{s'_X = \sqrt{\frac{2}{3}} \approx 0.8165} \end{aligned}$$

This answer matches with the output of `np.std()`.

[16]:
```
np.std(x_list, ddof = 1)
```

[16]: 1.0

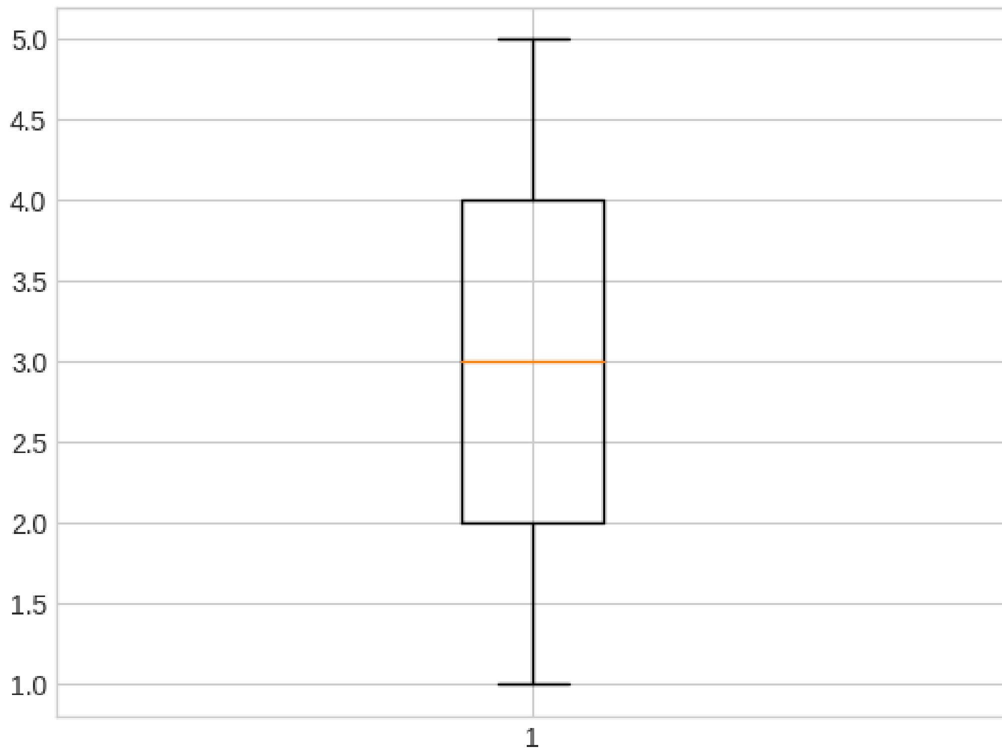
This answer matches with our familiar definition of standard deviation.

5 Task 5

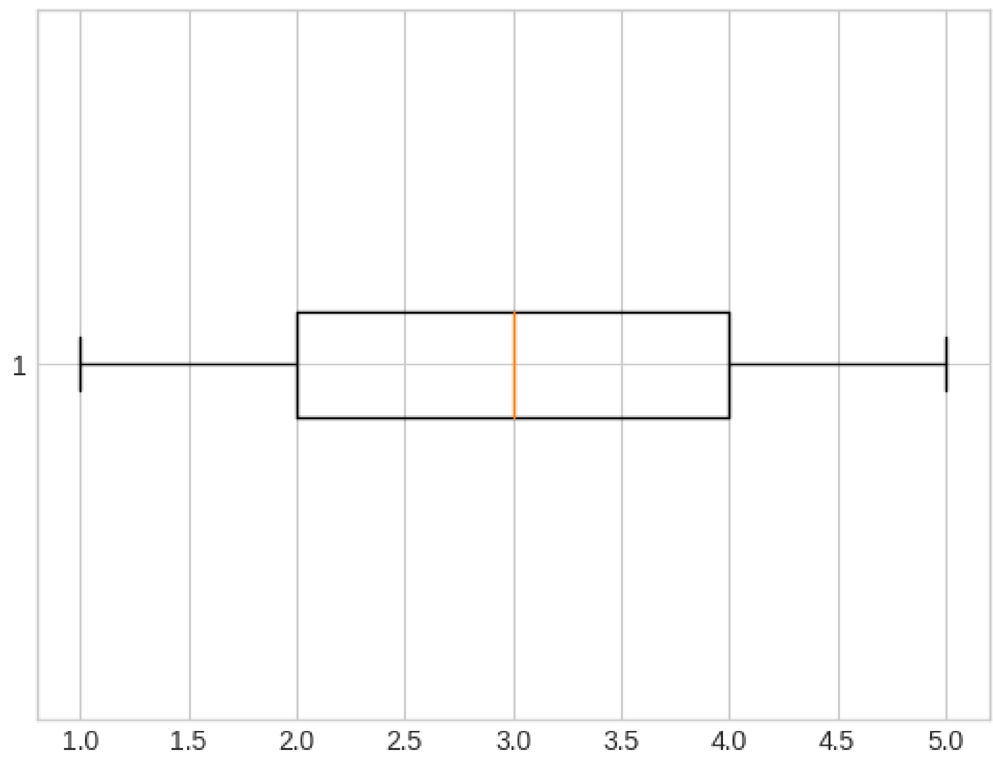
[19]:
```
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
plt.style.use('seaborn-v0_8-whitegrid')
```

[20]:
```
y = [1, 2, 3, 4, 5, 4, 3, 5, 4, 1, 2]
```

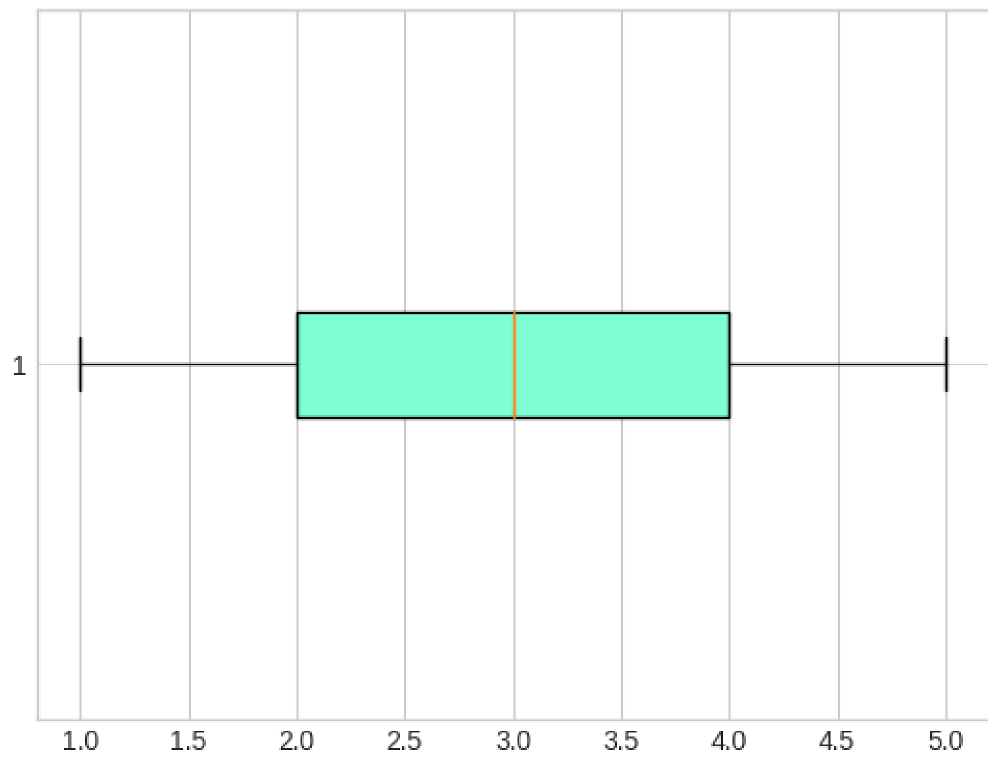
[21]:
```
plt.boxplot(y);
```



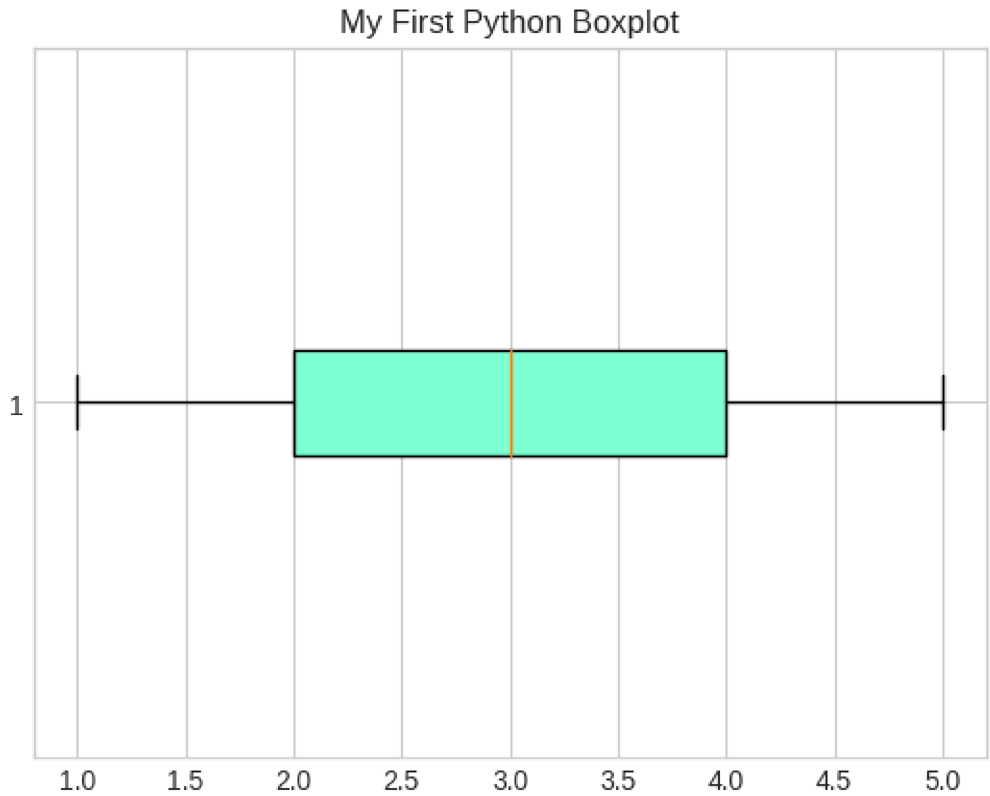
[23]:
```
plt.boxplot(y, vert = False);
```



[24]:
```
plt.boxplot(y, vert = False,
            patch_artist = True,
            boxprops = dict(facecolor = "aquamarine")
            );
```



[25]:
```
plt.boxplot(y, vert = False,
            patch_artist = True,
            boxprops = dict(facecolor = "aquamarine")
            );
plt.title("My First Python Boxplot");
```



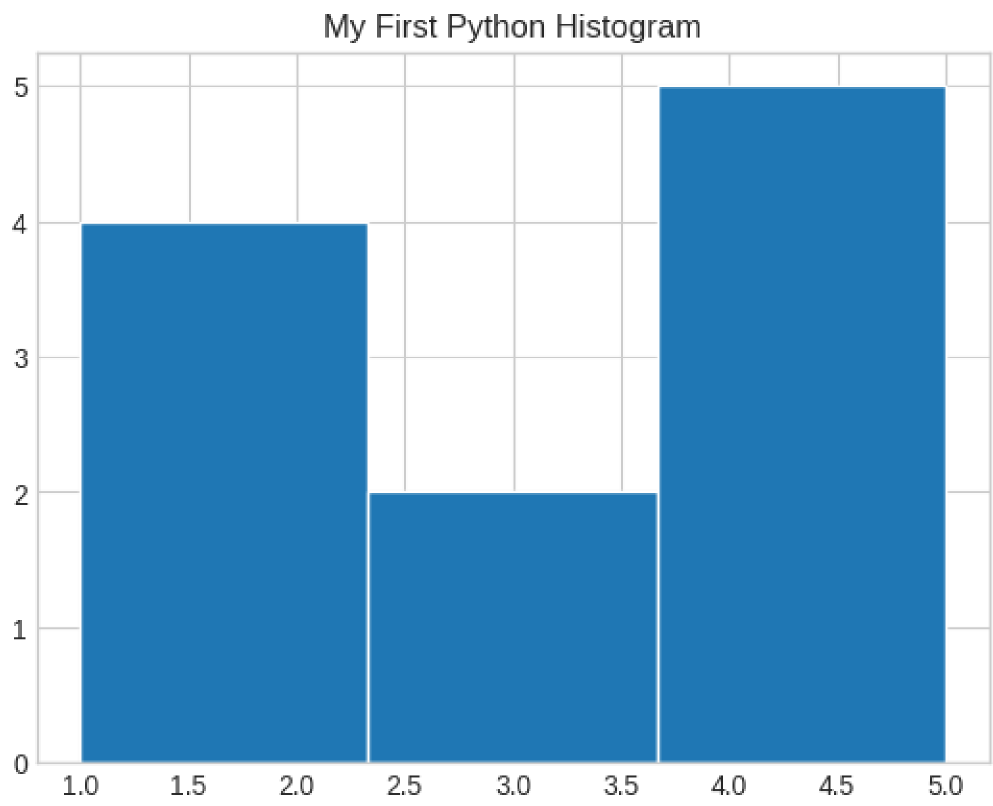
The IQR should be (4 - 2) = 2

[26]:
```
np.diff(np.percentile(y, [25,75]))[0]
```

[26]: 2.0

6 Task 6

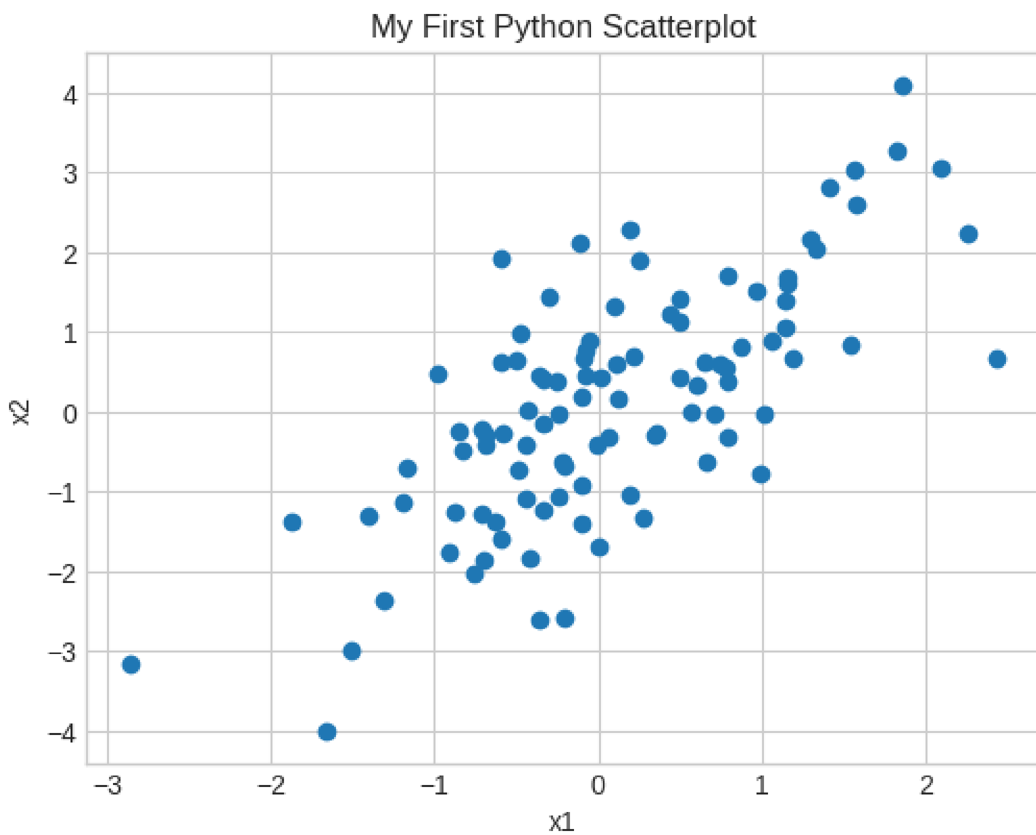
[30]:
```
plt.hist(y,
        bins = 3,
        edgecolor = "white");

plt.title("My First Python Histogram");
```



7 Task 7

[31]:
```
np.random.seed(5)
x1 = np.random.normal(0, 1, 100)
x2 = x1 + np.random.normal(0, 1, 100)
```

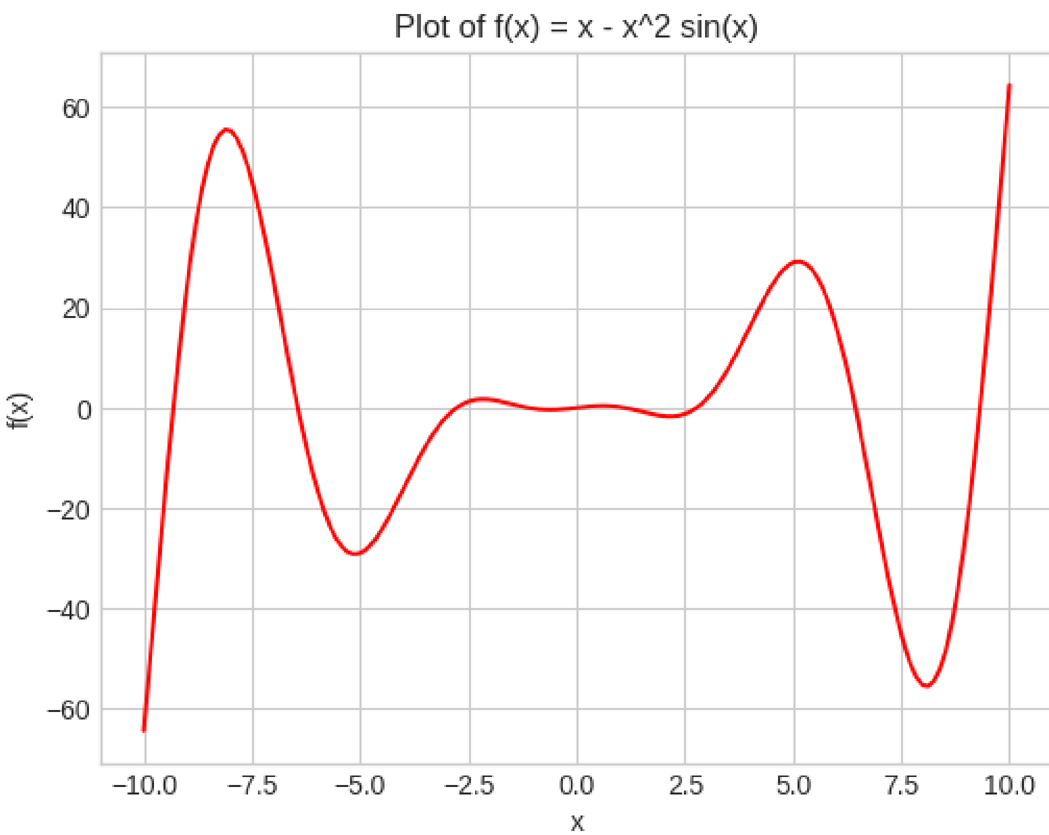
[34]:
```
plt.scatter(x1, x2);
plt.xlabel("x1");
plt.ylabel("x2");
plt.title("My First Python Scatterplot");
```



Yes, there does appear to be a positive linear association between x1 and x2.

8 Task 8

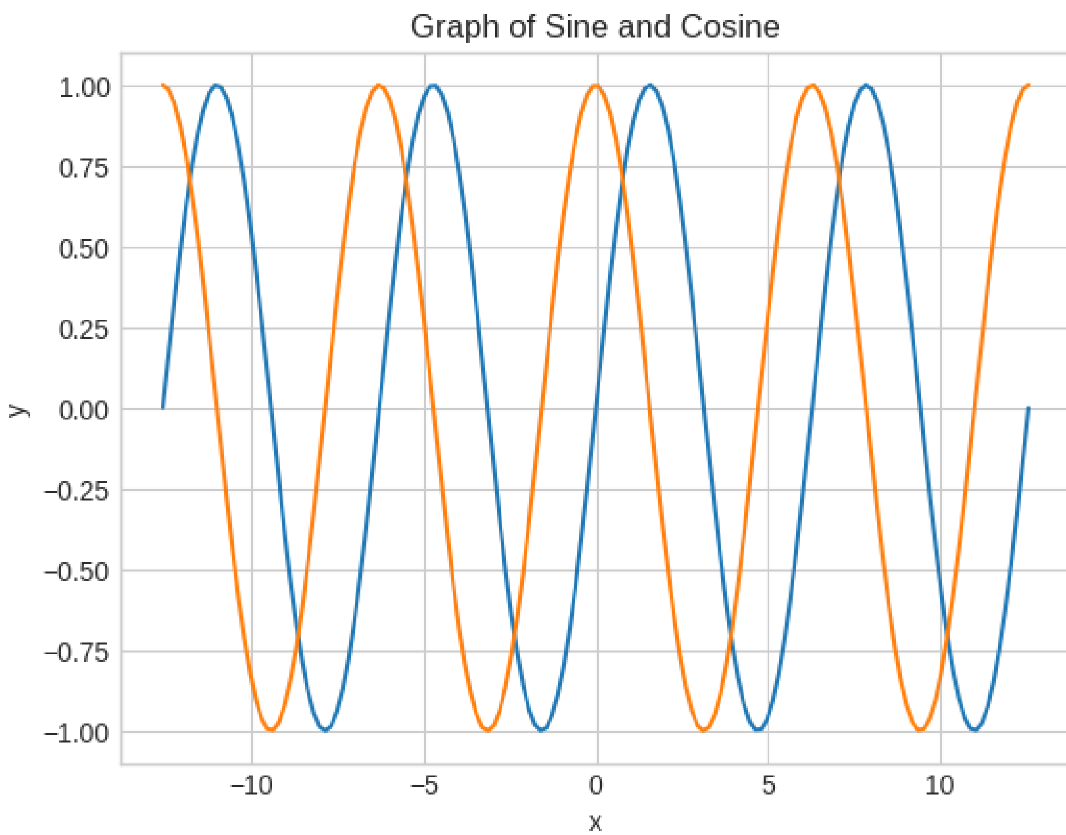
[37]:
```
def f(x):
    """
    returns x - x^2 * sin(x)
    """
    return x - (x ** 2) * np.sin(x)

x = np.linspace(-10, 10, 150);
plt.plot(x, f(x),
        color = "red");
plt.xlabel("x");
plt.ylabel("f(x)");
plt.title("Plot of f(x) = x - x^2 sin(x)");
```



9 Task 9

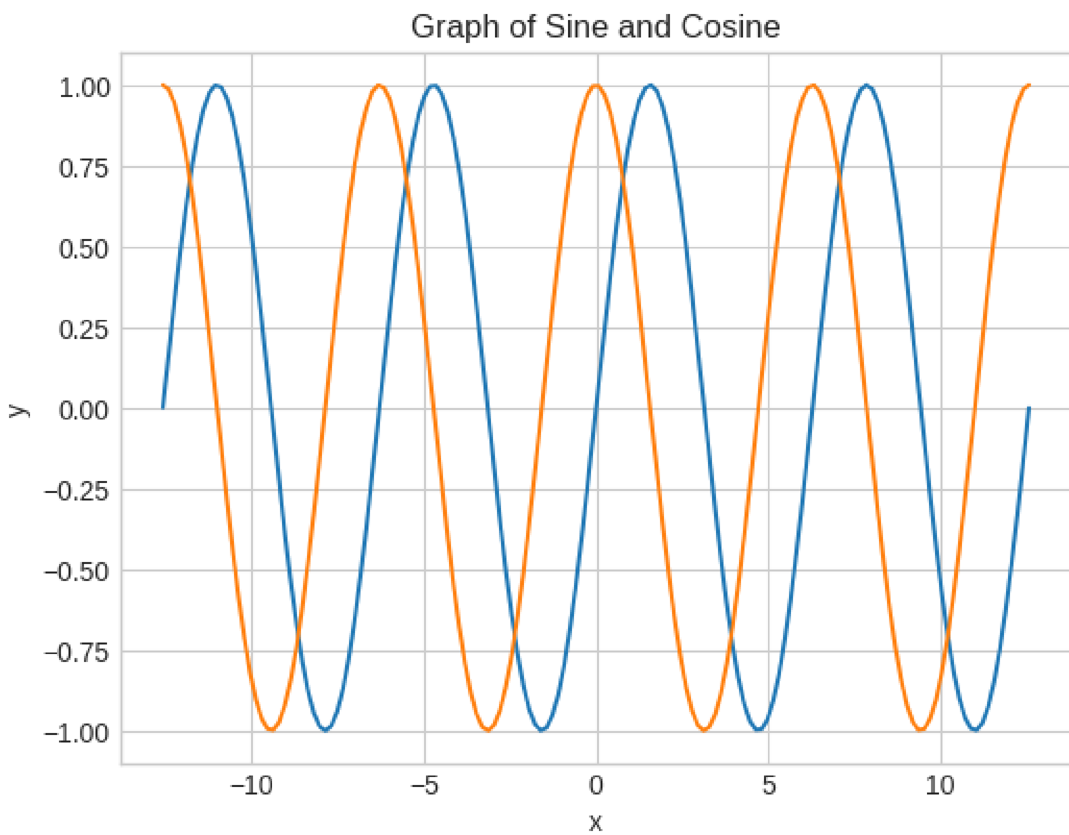
[38]:
```
x = np.linspace(-4 * np.pi, 4 * np.pi, 150);
```

[39]:
```
plt.plot(x, np.sin(x));
plt.plot(x, np.cos(x));

plt.xlabel("x");
plt.ylabel("y");
plt.title("Graph of Sine and Cosine");
```



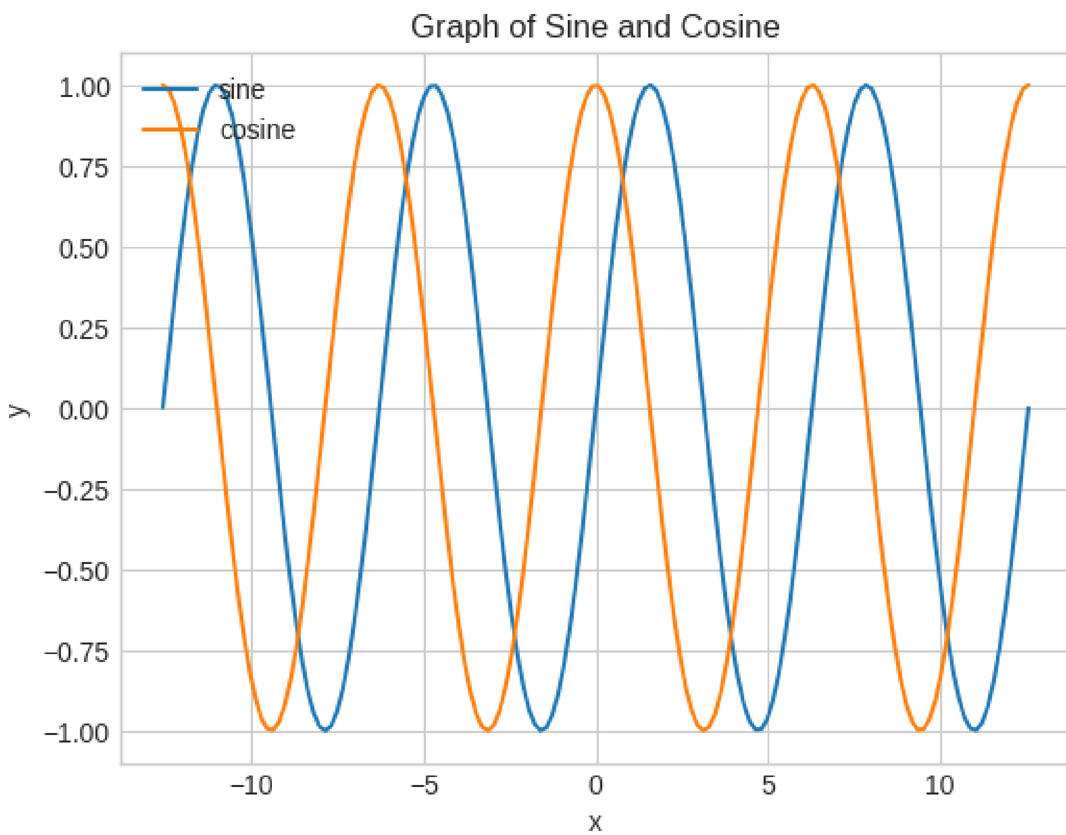
10 Task 10

[40]:
```
plt.plot(x, np.sin(x), label = "sine");
plt.plot(x, np.cos(x), label = "cosine");

plt.xlabel("x");
plt.ylabel("y");
plt.title("Graph of Sine and Cosine");
```



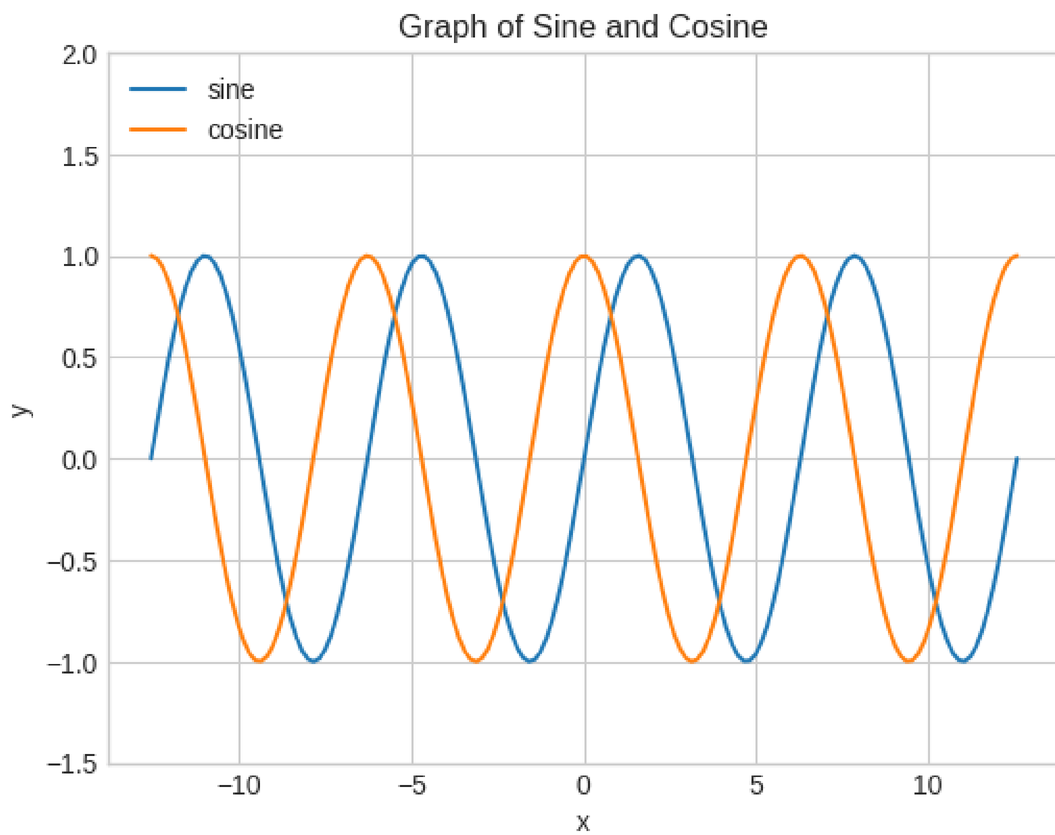
11 Task 11

[41]:
```
plt.plot(x, np.sin(x), label = "sine");
plt.plot(x, np.cos(x), label = "cosine");

plt.xlabel("x");
plt.ylabel("y");
plt.title("Graph of Sine and Cosine");
plt.legend(loc = "upper left");
```



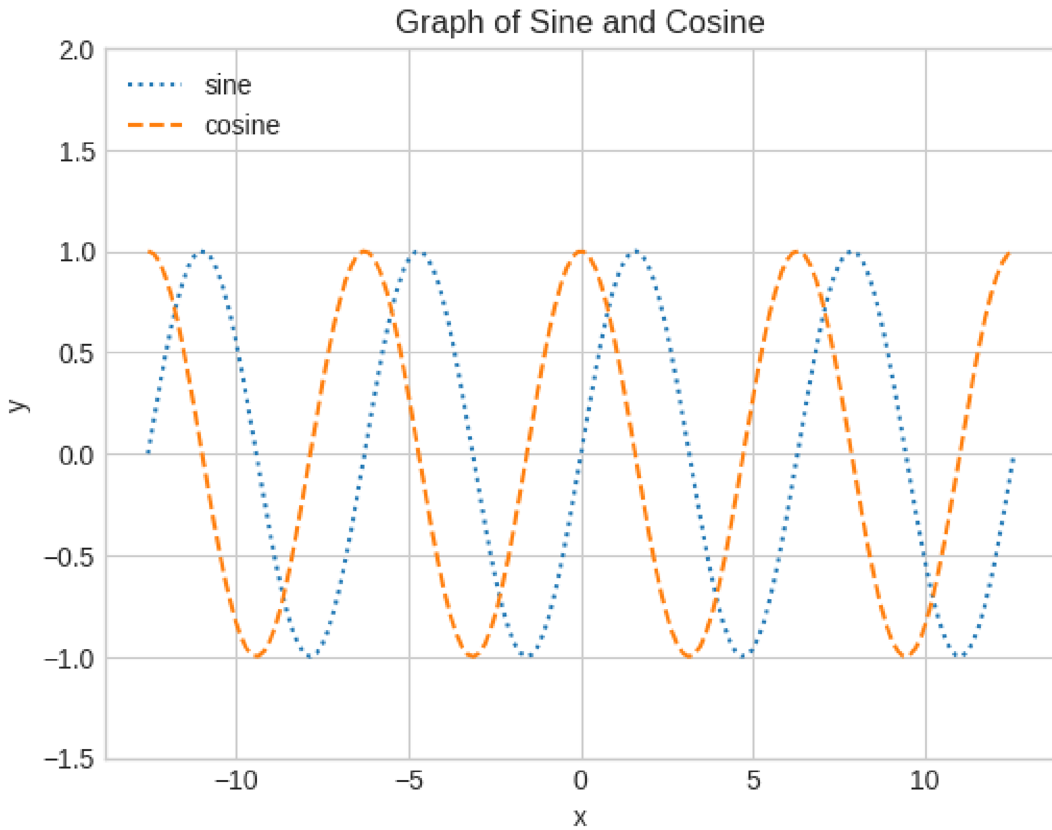
12 Task 12

[42]:
```
plt.plot(x, np.sin(x), label = "sine");
plt.plot(x, np.cos(x), label = "cosine");

plt.xlabel("x");
plt.ylabel("y");
plt.title("Graph of Sine and Cosine");
plt.legend(loc = "upper left");
plt.ylim([-1.5, 2]);
```



13 Task 13

[43]:
```
plt.plot(x, np.sin(x), label = "sine", linestyle = "dotted");
plt.plot(x, np.cos(x), label = "cosine", linestyle = "dashed");

plt.xlabel("x");
plt.ylabel("y");
plt.title("Graph of Sine and Cosine");
plt.legend(loc = "upper left");
plt.ylim([-1.5, 2]);
```


Graph of Sine and Cosine
sine
cosine
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
y
−10
−5
0
5
10
x

```
[ ]:
```