Tips: Fill the area

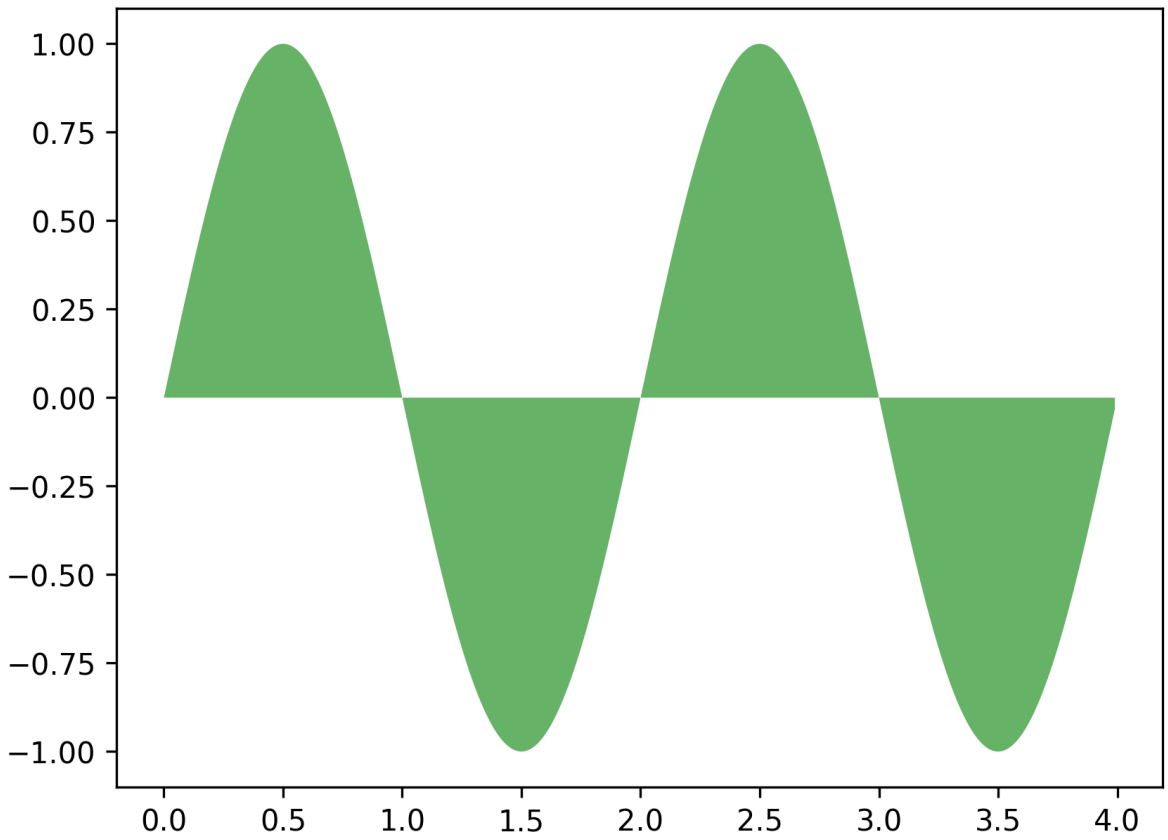
Trong một vài trường một, ta cần mô tả một số khoảng dữ liệu cụ thể trên biểu đồ theo điều kiện nào đó. Ví dụ khi mô tả thuật toán SVM, trên biểu đồ thể hiện các điểm dữ liệu, ta cần mô tả thêm 2 đường boundary dùng để phân loại dữ liệu.

Matplot cung cấp thêm hàm fill_between() để tô màu các khoảng dữ liệu mong muốn giữa 2 đường cong trên biểu đồ. Hàm này có một số thuộc tính như:

- x: tọa độ trục x các điểm của các đường cong
- y1: tọa độ trục y các điểm của đường cong 1
- y2: tạo độ trục y các điểm của đường cong 2
- where: mảng chứa các giá trị của biểu thức logic, dùng để thêm các điều kiện khi muốn tùy chọn các vùng được fill

In [2]:
```
# Ví dụ fill các vùng giữa trục x và đường đồ thị hình sin
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 4, 0.01)
y1 = np.sin(x * np.pi)

fig, axes = plt.subplots(dpi=300)
axes.fill_between(x, y1, facecolor='g', alpha=0.6)
```

Out[2]:
```
<matplotlib.collections.PolyCollection at 0x7f964567fee0>
```

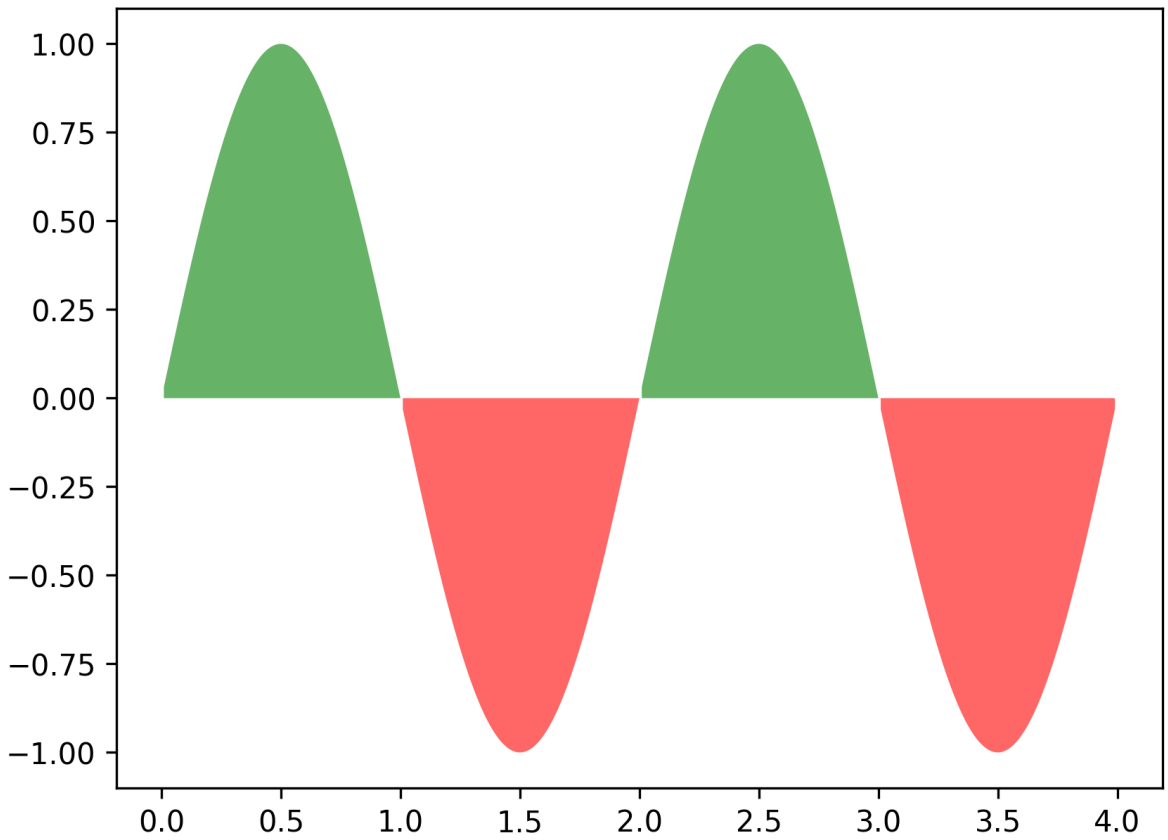


Khi muốn phân biệt các vùng dữ liệu nửa trên và nửa dưới của đồ thị theo trục y, ta có

thể thêm điều kiện trong tham số where như sau

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 4, 0.01)
y1 = np.sin(x*np.pi)

fig, axes = plt.subplots(dpi=300)
axes.fill_between(x, y1, where=(y1 > 0), facecolor='g', alpha=0.6)
axes.fill_between(x, y1, where=(y1 < 0), facecolor='r', alpha=0.6)
```

Out[3]:
```
<matplotlib.collections.PolyCollection at 0x7f96404d7790>
```



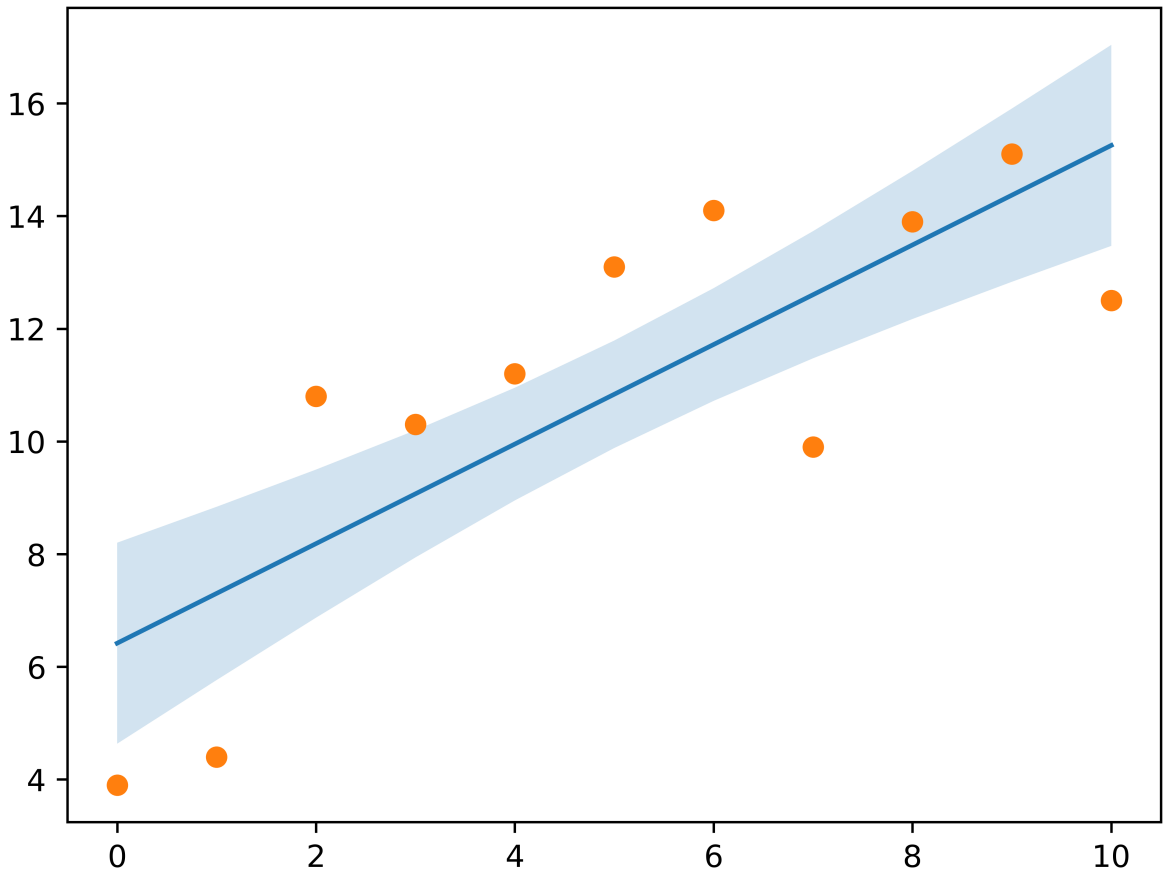
Khoảng tin cậy

Trong những bài toán phân loại, cần mô tả khoảng tin cậy chứa các điểm dữ liệu được chọn với một sai số nhất định của mô hình ta có thể dùng fill_between để mô tả trên biểu đồ như sau:

In [5]:
```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 11)
y = [3.9, 4.4, 10.8, 10.3, 11.2, 13.1, 14.1,  9.9, 13.9, 15.1, 12.5]

a, b = np.polyfit(x, y, deg=1)

# Fake giá trị dự đoán
y_est = a * x + b

# Sai số
y_err = x.std() * np.sqrt(1/len(x) + (x - x.mean())**2 / np.sum((x - x.mean(
```

```
fig, axes = plt.subplots(dpi=800)
axes.plot(x, y_est, '-')
axes.fill_between(x, y_est - y_err, y_est + y_err, alpha=0.2)
axes.plot(x, y, 'o')
```

Out[5]: [<matplotlib.lines.Line2D at 0x7f9640441d30>]



In [ ]: