Übungen mit Matplotlib

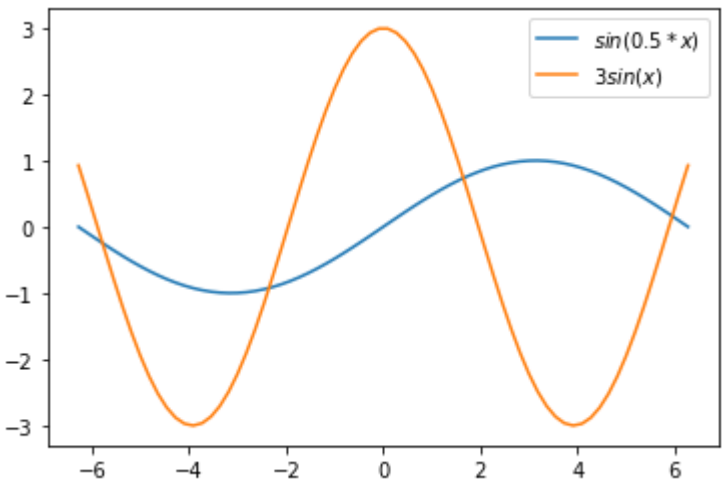
Legenden

Legenden in Matplotlib-Plots kann man mit dem Positionsargument `loc` (zum Beispiel `"upper right"` positionieren. In vielen Fällen weiß man jedoch nicht, wie das Ergebnis aussehen wird, bevor man es geplottet hat und die Legende verdeckt unter Umständen einen (wichtigen) Teil des Plots. Daher kann man alternativ das Positionsargument `best` verwenden. Matplotlib versucht dann, automatisch die bestmögliche Position für die Legende zu finden:

[1]
```
import numpy as np
import matplotlib.pyplot as plt
```

[2]
```
x = np.linspace(0, 25, 1000)
f1 = np.sin(0.5*x)
f2 = 3*np.cos(0.8*x)
```

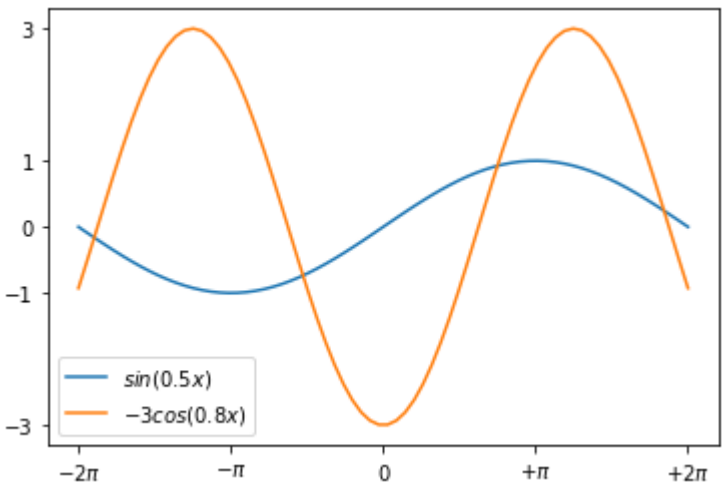
[9]
```
plt.plot(x, f1, label = "$sin(0.5*x)$")
plt.plot(x, f2, label = "$3sin(x)$")
plt.legend(loc = "best")
plt.show()
```



In den folgenden beiden Beispielen kann man sehen, daß `loc = "best"` in der Regel sehr gut funktioniert:

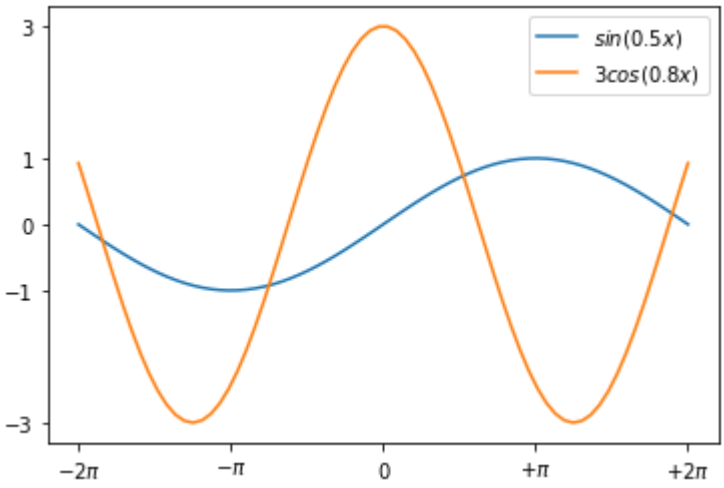
[4]
```
x = np.linspace(-2*np.pi, 2*np.pi, 70, endpoint = True)
f1 = np.sin(0.5*x)
f2 = -3*np.cos(0.8*x)
```

[5]
```
plt.xticks([-6.28, -3.14, 0, 3.14, 6.28],
           [r"$-2\pi$", r"$-\pi$", r"$0$", r"$+\pi$",
r"$+2\pi$"])
plt.yticks([-3, -1, 0, 1, 3])
plt.plot(x, f1, label = "$sin(0.5x)$")
plt.plot(x, f2, label = "$-3cos(0.8x)$")
plt.legend(loc = "best")
plt.show()
```



[6]
```
x = np.linspace(-2*np.pi, 2*np.pi, 70, endpoint = True)
f1 = np.sin(0.5*x)
f2 = 3*np.cos(0.8*x)
```

[7]
```
plt.xticks([-6.28, -3.14, 0, 3.14, 6.28],
           [r"$-2\pi$", r"$-\pi$", r"$0$", r"$+\pi$",
r"$+2\pi$"])
plt.yticks([-3, -1, 0, 1, 3])
plt.plot(x, f1, label = "$sin(0.5x)$")
plt.plot(x, f2, label = "$3cos(0.8x)$")
plt.legend(loc = "best")
plt.show()
```


3
1
0
−1
−3
−2π
−π
0
+π
+2π
sin(0.5x)
3cos(0.8x)

```
[ ]
```