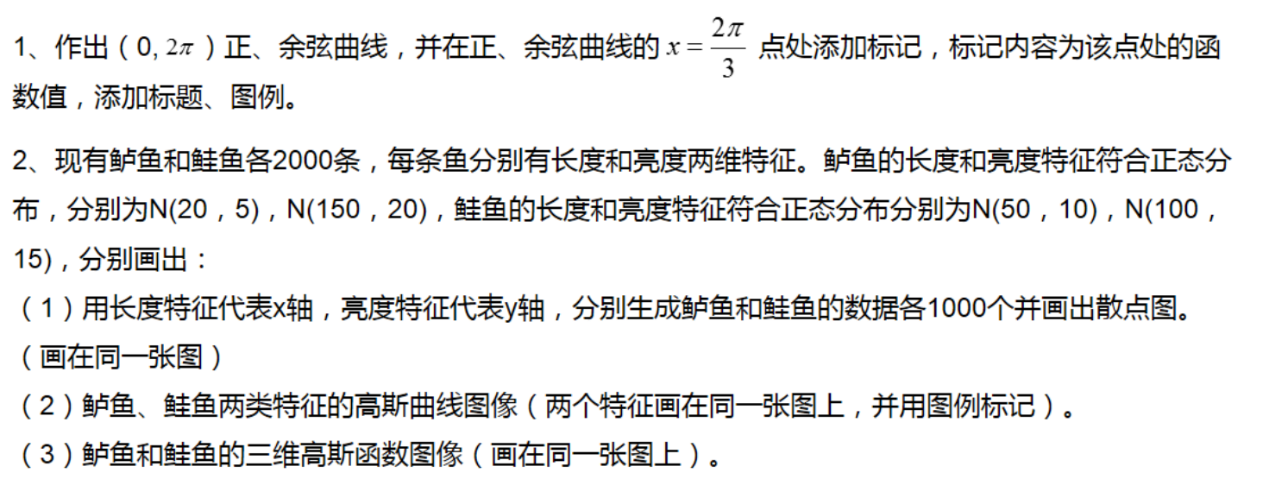
 

**智能信息系统综合实践**

**实验报告**

|  |  |
| --- | --- |
| **题 目：** | matplotlib画图 |
| **年 级：** | **2021** |
| **专 业：** | **软件工程** |
| **学 号：** | **2021117405** |
| **姓 名：** | **孙潇桐** |

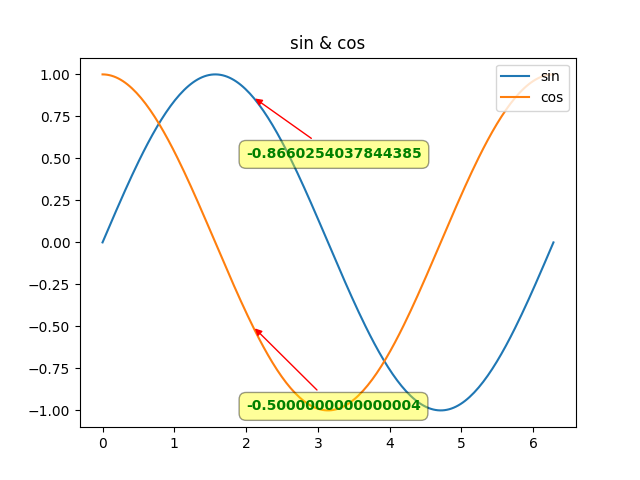
1. **题目**



1. **解题步骤**
   1. **第一题**

**只要计算出(0, 2**)之间200个点的cos和sin值就能用plot函数直接画出图像。

#### **结果**



#### **代码**

import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.linspace(0, 2 \* np.pi, 200)  
y1 = np.sin(x)  
y2 = np.cos(x)  
  
plt.plot(x, y1, label='sin')  
plt.plot(x, y2, label='cos')  
  
plt.annotate(xy=(np.pi \* 2 / 3, np.sin(np.pi \* 2 / 3)), text=str(np.sin(2 \* np.pi \* 2 / 3)), xytext=(2, 0.5),  
 weight='bold', color='green', arrowprops=dict(arrowstyle='-|>', connectionstyle='arc3', color='red'),  
 bbox=dict(boxstyle='round,pad=0.5', fc='yellow', ec='k', lw=1, alpha=0.4))

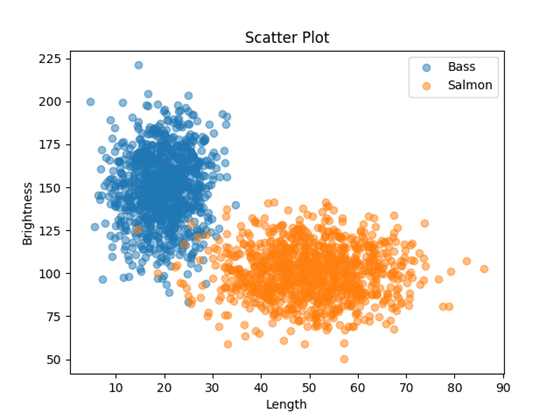
plt.annotate(xy=(np.pi \* 2 / 3, np.cos(np.pi \* 2 / 3)), text=str(np.cos(2 \* np.pi \* 2 / 3)), xytext=(2, -1.0),  
 weight='bold', color='green', arrowprops=dict(arrowstyle='-|>', connectionstyle='arc3', color='red'),  
 bbox=dict(boxstyle='round,pad=0.5', fc='yellow', ec='k', lw=1, alpha=0.4))

plt.title("sin & cos")  
plt.legend(loc='upper right')  
plt.show()

* 1. **第二题**
     1. **第一问**

**根据题目给出的正态分布，随机生成两种鱼的数据。**

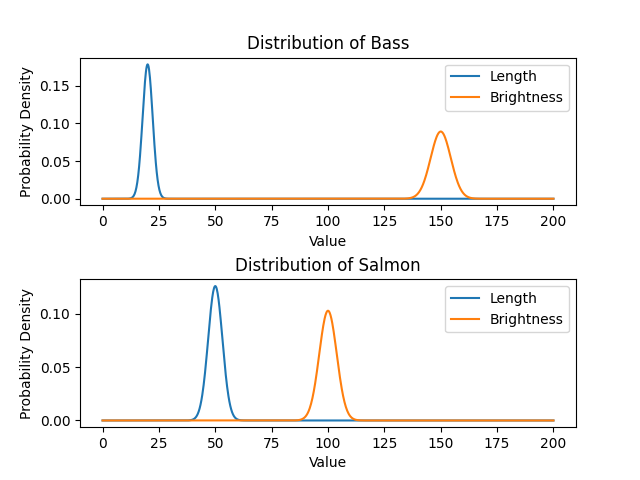
#### **结果：**



* + 1. **第二问**

**分别画出两种鱼的两个正态分布**

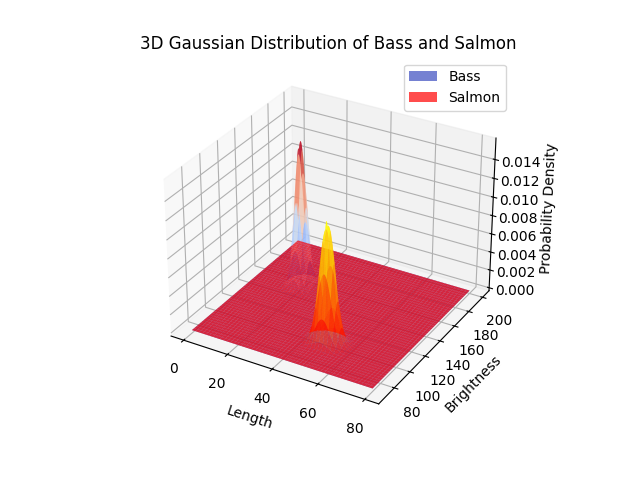
#### **结果：**



* + 1. **第三问**

**根据第一问计算出来的数据，将两个特征的概率相乘就能得到三维图。**

#### **结果：**



* + 1. **代码**

from math import sqrt, pi  
  
import matplotlib.pyplot as plt  
import numpy as np  
  
np.random.seed(2333)  
n\_samples = 1000  
  
# 鲈鱼的数据  
length\_bass = np.random.normal(20, 5, n\_samples)  
brightness\_bass = np.random.normal(150, 20, n\_samples)  
  
# 鲑鱼的数据  
length\_salmon = np.random.normal(50, 10, n\_samples)  
brightness\_salmon = np.random.normal(100, 15, n\_samples)  
  
# 绘制散点图  
plt.scatter(length\_bass, brightness\_bass, label='Bass', alpha=0.5)  
plt.scatter(length\_salmon, brightness\_salmon, label='Salmon', alpha=0.5)  
plt.xlabel('Length')  
plt.ylabel('Brightness')  
plt.title('Scatter Plot')  
plt.legend()  
plt.show()  
  
  
def n\_dist(x, miu, sigma):  
 sigma = sqrt(sigma)  
 return np.exp(-(x - miu) \*\* 2 / (2 \* (sigma \*\* 2))) / (sqrt(2 \* pi) \* sigma)  
  
  
# 绘制高斯曲线图像  
x = np.linspace(0, 200, 1000)  
plt.subplot(211)  
plt.plot(x, n\_dist(x, 20, 5), label='Length')  
plt.plot(x, n\_dist(x, 150, 20), label='Brightness')  
plt.xlabel('Value')  
plt.ylabel('Probability Density')  
plt.title('Distribution of Bass')  
plt.legend()  
plt.subplot(212)  
plt.plot(x, n\_dist(x, 50, 10), label='Length')  
plt.plot(x, n\_dist(x, 100, 15), label='Brightness')  
plt.xlabel('Value')  
plt.ylabel('Probability Density')  
plt.title('Distribution of Salmon')  
plt.legend()  
plt.subplots\_adjust(hspace=0.5)  
plt.show()  
  
# 绘制三维高斯函数图像  
ax = plt.subplot(111, projection='3d')  
  
x = np.linspace(0, 80, 1000)  
y = np.linspace(70, 200, 1000)  
X, Y = np.meshgrid(x, y)  
  
Z\_bass = n\_dist(X, 20, 5) \* n\_dist(Y, 150, 20)  
Z\_salmon = n\_dist(X, 50, 10) \* n\_dist(Y, 100, 15)  
  
# 绘制鲈鱼的高斯函数图像  
ax.plot\_surface(X, Y, Z\_bass, cmap='coolwarm', alpha=0.7, label='Bass')  
  
# 绘制鲑鱼的高斯函数图像  
ax.plot\_surface(X, Y, Z\_salmon, cmap='autumn', alpha=0.7, label='Salmon')  
  
ax.set\_xlabel('Length')  
ax.set\_ylabel('Brightness')  
ax.set\_zlabel('Probability Density')  
ax.set\_title('3D Gaussian Distribution of Bass and Salmon')  
ax.legend()  
plt.show()

1. **总结**

第一个题目通过绘制正弦、余弦函数图像和两种鱼的数据分布图，突显了数据可视化在理解数据特征和趋势方面的重要性。通过可视化，我们能够更直观地观察数据间的关系，发现规律和异常，为进一步分析和决策提供依据。

在第二题中，我学习了如何灵活地运用matplotlib绘制函数图像和散点图，并通过annotate函数添加了标注，提升了图像的可读性和信息传递效果。同时，通过subplot函数实现了多个子图的排列，使得图像展示更加清晰。

通过生成两种鱼的数据，并绘制其对应的高斯分布曲线，加深了对高斯分布特性的理解。高斯分布在自然界和工程领域中都有着广泛应用，了解其概念和特性对于数据建模和分析具有重要意义。

实验最后一部分展示了如何将两个特征的高斯分布绘制成三维图像，这不仅展示了matplotlib库在三维可视化方面的功能，也提供了一种直观的方式来观察多维数据之间的关系。

总的来说，通过这次实验，不仅加深了对matplotlib库的应用和数据可视化技术的理解，也拓展了对数据分布特性和多维数据可视化方法的认识，为进一步的数据分析和可视化工作奠定了基础。