**《数据统计与分析基础实验》**

实验二报告

|  |  |
| --- | --- |
| 班级： | 计224 |
| 姓名： | 王小康 |
| 学号： | 225432 |

**实验二 可视化绘图**

**一、实验目的**

通过编程和上机实验，掌握实用软件绘图的基本步骤，掌握常见的 绘图函数，正确地依据参数绘制常见的数据统计图形，熟悉特殊图形的绘制和 图形的修饰。

**二、实验内容（本节绘图区间均为 I=[自己学号-15 自己学号+15]）**

1. 一幅图内同时绘制sin、cos、tan函数曲线图，区间长度为I，三条曲线线型、颜色各不相同。
2. 绘制三维图形 x = sint-t\*cost, y = cost-t\*sint, z = t, t区间为I，图形x、y、z轴均加网格线，z轴取对数分布。
3. 随机生成一组有效的50人班级的成绩，并绘制合理的统计直方图。

提示：1、成绩为整数；2、直方图需选择合适区间。

1. 统计自己十月份的各项支出，并利用软件绘制自己的支出饼状图，采用2种以上方式展示。
2. 绘制散点图x = sint-t\*cost, y = cost-t\*sint,, t区间为I。

绘制曲线图、阶梯图x = sint-t\*cost, y = cost-t\*sint,, t区间为I。

**三、实验源程序及运行结果（可附截图）**

1.程序如下：

import numpy as np

import matplotlib.pyplot as plt

# 学号和区间 I

My\_id = 225432

I = [My\_id - 15, My\_id + 15]

t = np.linspace(I[0], I[1], 500)

# 1. 绘制 sin, cos, tan 函数曲线图

plt.figure(figsize=(10, 6))

plt.plot(t, np.sin(t), label='sin(t)', linestyle='-', color='blue')

plt.plot(t, np.cos(t), label='cos(t)', linestyle='--', color='green')

plt.plot(t, np.tan(t), label='tan(t)', linestyle='-.', color='red')

plt.ylim(-10, 10)  # 限制 tan 的纵轴范围，避免无穷值

plt.legend()

plt.title("Trigonometric Functions")

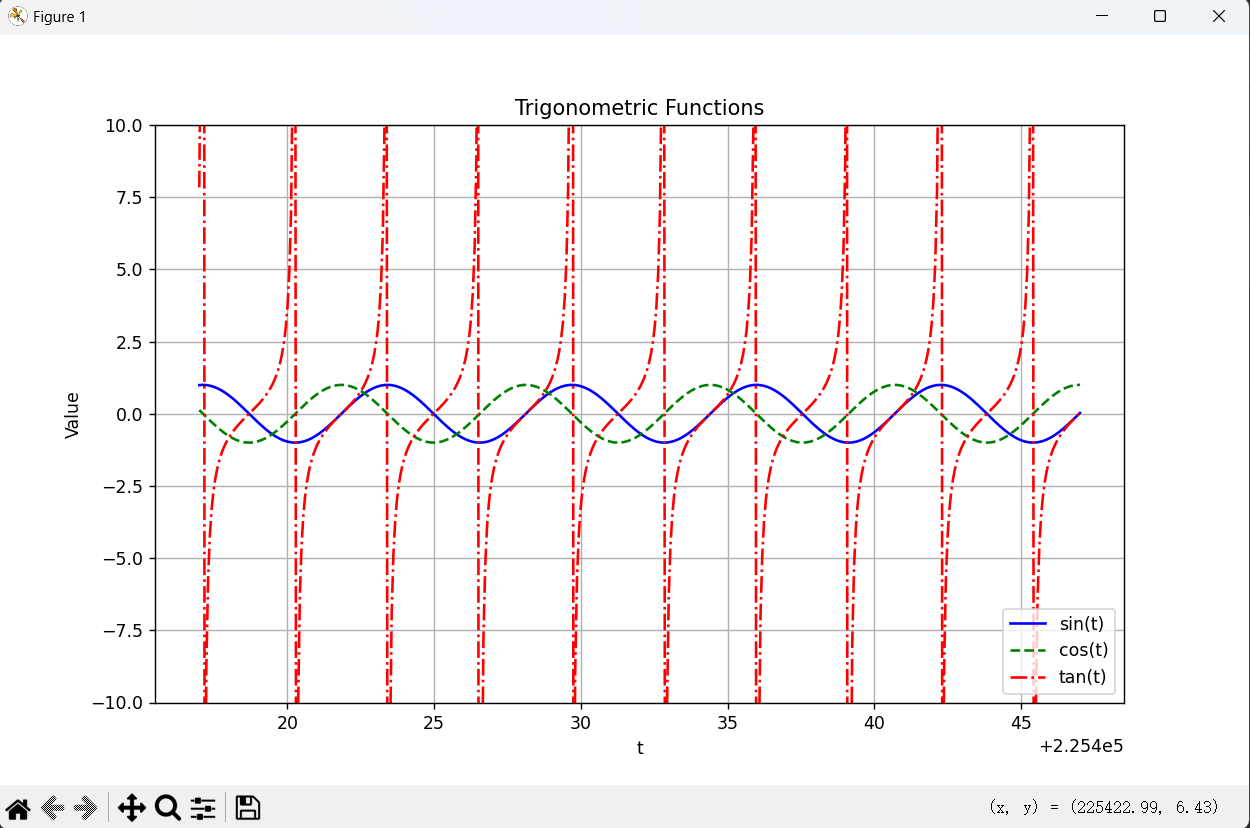
plt.xlabel("t")

plt.ylabel("Value")

plt.grid()

plt.show()

结果如下：



2.程序如下：

import numpy as np

import matplotlib.pyplot as plt

# 学号和区间 I

My\_id = 225432

I = [My\_id - 15, My\_id + 15]

t = np.linspace(I[0], I[1], 500)

# 参数方程

x = np.sin(t) - t \* np.cos(t)

y = np.cos(t) - t \* np.sin(t)

z = t

# 绘制三维曲线

fig = plt.figure(figsize=(10, 6))

ax = fig.add\_subplot(111, projection='3d')

ax.plot(x, y, z, color='purple')

ax.set\_title("3D Curve")

ax.set\_xlabel("X-axis")

ax.set\_ylabel("Y-axis")

ax.set\_zlabel("Z-axis (log scale)")

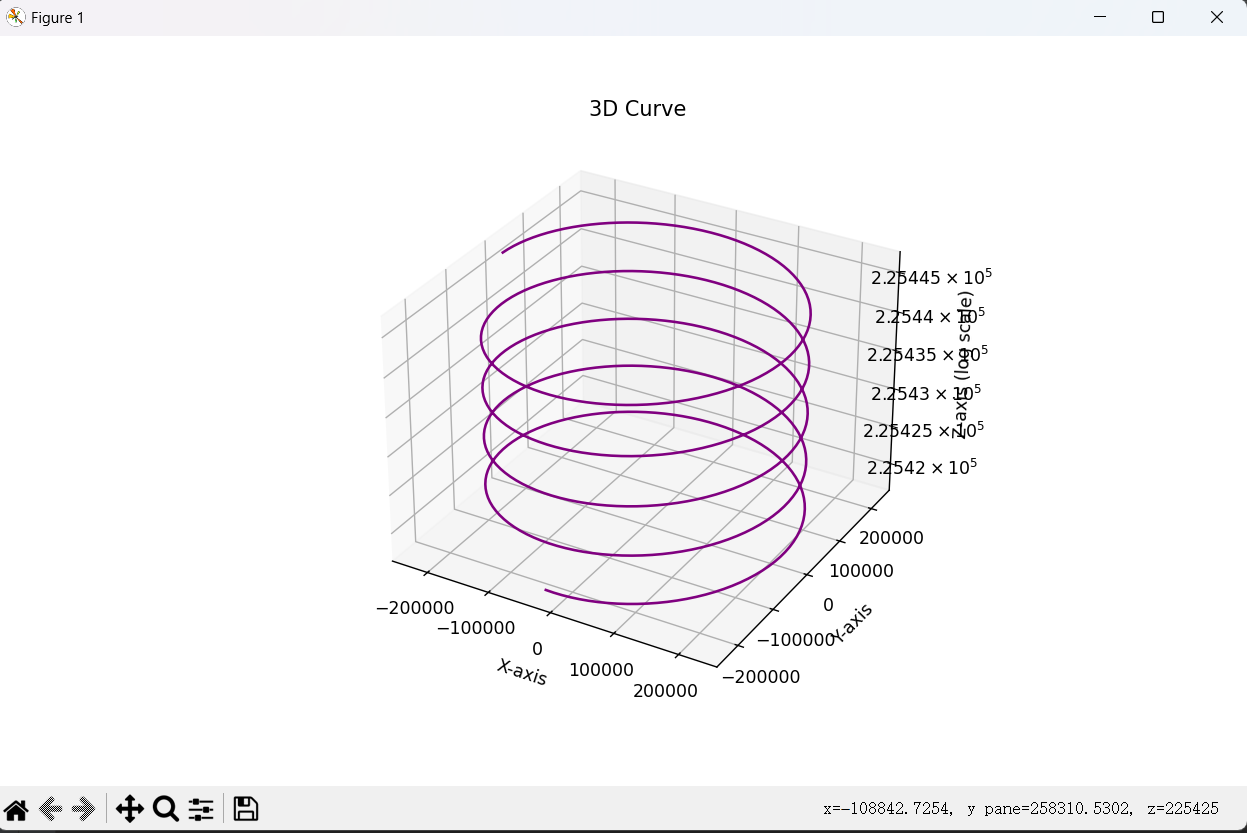
ax.grid(True)

# Z 轴对数刻度

ax.set\_zscale('log')  # Z轴取对数

plt.show()

结果如下：



3.程序如下：

import numpy as np

import matplotlib.pyplot as plt

# 随机生成班级成绩，50个学生的成绩范围在 40 到 100 之间

np.random.seed(42)  # 保持结果一致

scores = np.random.randint(40, 101, 50)  # 生成 50 个成绩，范围为 40 到 100

# 绘制直方图

plt.figure(figsize=(10, 6))

plt.hist(scores, bins=10, edgecolor='black', color='skyblue', range=(40, 100))

plt.title("Class Score Distribution")

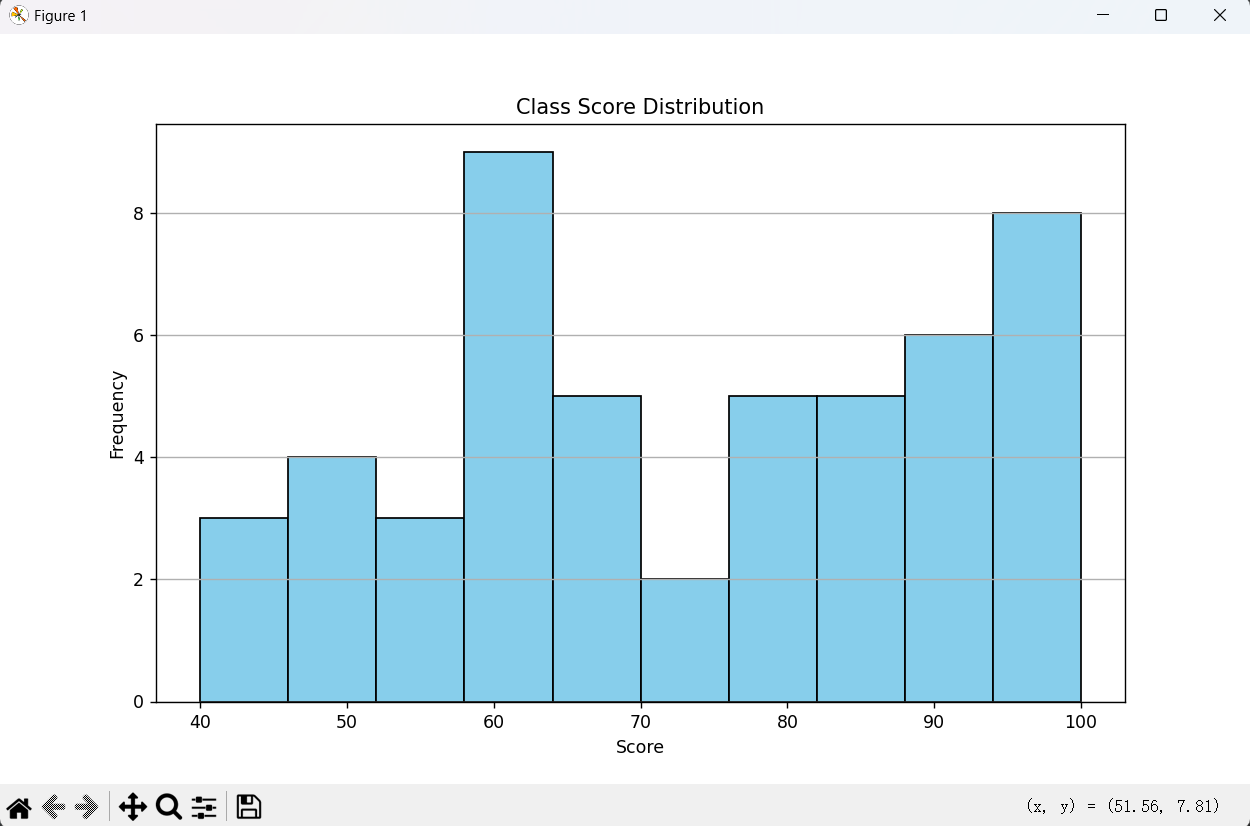
plt.xlabel("Score")

plt.ylabel("Frequency")

plt.grid(axis='y')

plt.show()

结果如下：



4.结果如下：

import matplotlib.pyplot as plt

# 十月份支出数据

expenses = {

    "Food": 900,

    "Transport": 30,

    "books": 100,

    "Other": 300

}

# 提取标签和支出金额

labels = list(expenses.keys())

sizes = list(expenses.values())

colors = ['lightcoral', 'lightskyblue', 'lightgreen', 'purple']

# 饼状图 1：普通样式

plt.figure(figsize=(8, 8))

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)

plt.title("October Expenses")

plt.show()

# 饼状图 2：突出某项支出（比如 Rent）

explode = [0.1, 0, 0, 0]  # 突出第一项支出

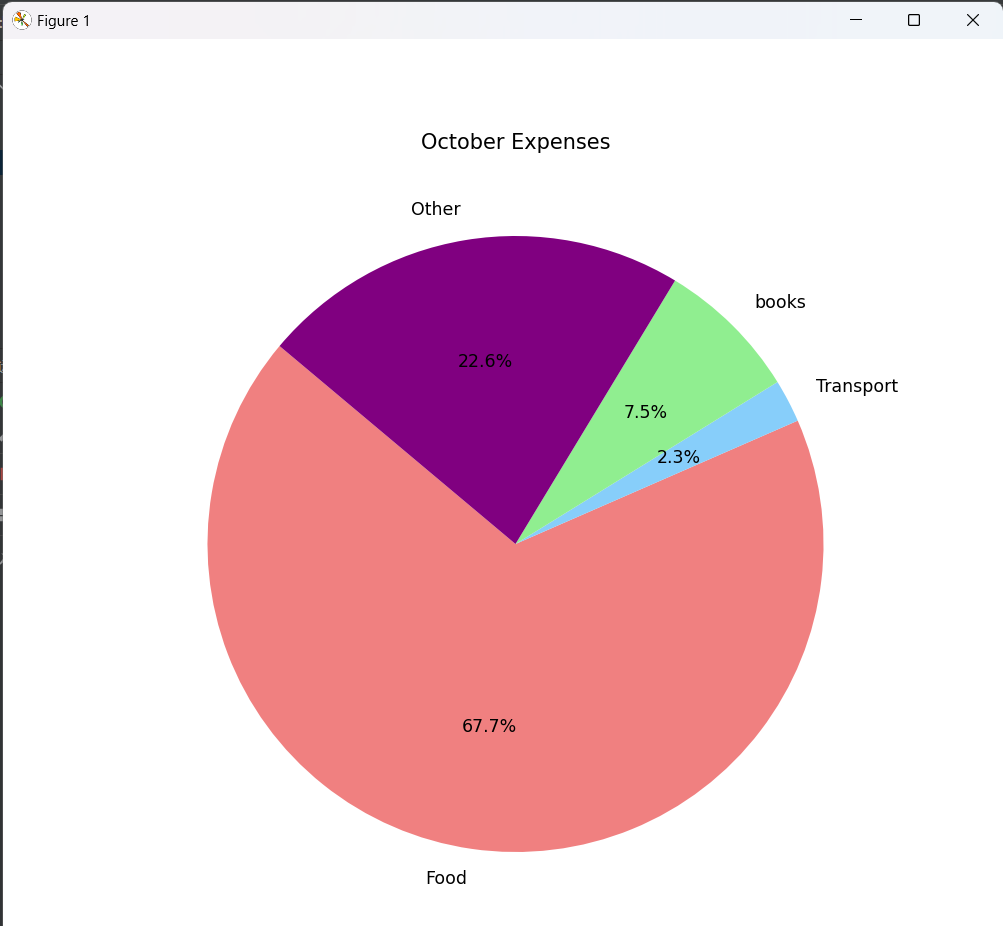
plt.figure(figsize=(8, 8))

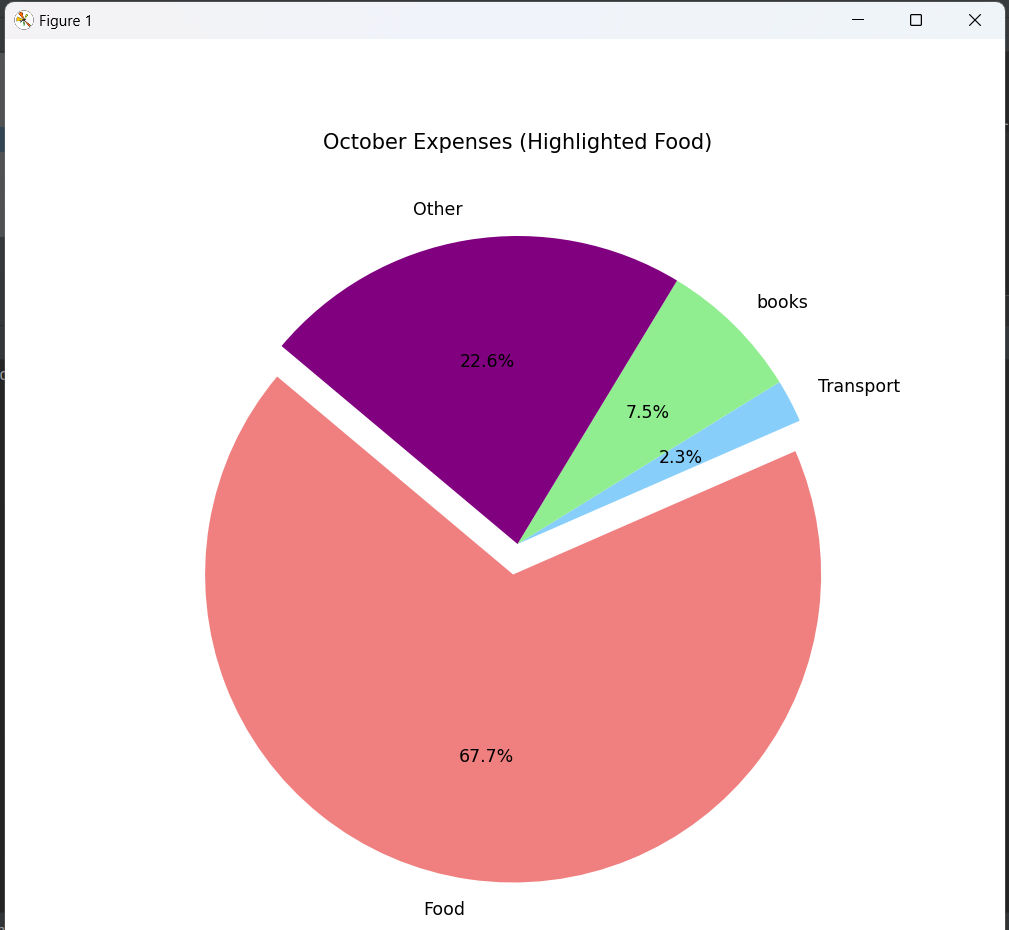
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140, explode=explode)

plt.title("October Expenses (Highlighted Food)")

plt.show()

结果如下：





5.程序如下：

import numpy as np

import matplotlib.pyplot as plt

# 学号和区间 I

My\_id = 225432

I = [My\_id - 15, My\_id + 15]

t = np.linspace(I[0], I[1], 500)

# 参数方程

x = np.sin(t) - t \* np.cos(t)

y = np.cos(t) - t \* np.sin(t)

# 绘制散点图

plt.figure(figsize=(10, 6))

# 散点图

plt.scatter(x, y, label="Scatter Points", color='red', s=10)

# 设置标题和标签

plt.title("Points")

plt.xlabel("x = sin(t) - t\*cos(t)")

plt.ylabel("y = cos(t) - t\*sin(t)")

# 显示图例

plt.legend()

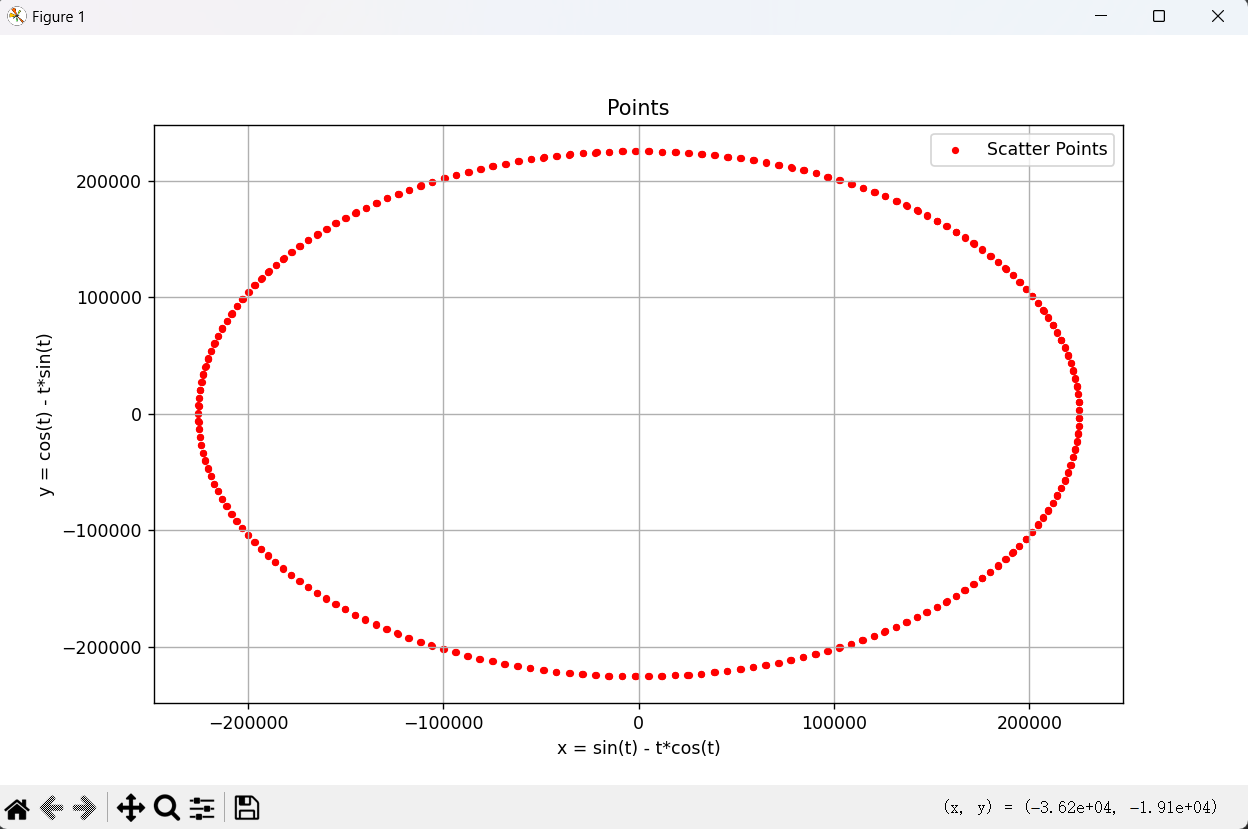
# 显示网格

plt.grid()

# 展示图形

plt.show()

结果如下：



6.程序如下：

import numpy as np

import matplotlib.pyplot as plt

# 学号和区间 I

student\_id = 225432

I = [student\_id - 15, student\_id + 15]

t = np.linspace(I[0], I[1], 500)

# 参数方程

x = np.sin(t) - t \* np.cos(t)

y = np.cos(t) - t \* np.sin(t)

# 绘制曲线图和阶梯图

plt.figure(figsize=(10, 6))

# 曲线图

plt.plot(x, y, label="Line Plot", linestyle='-', color='blue')

# 阶梯图

plt.step(x, y, label="Step Plot", linestyle='--', color='green')

# 设置标题和标签

plt.title("Line and Step Plots")

plt.xlabel("x = sin(t) - t\*cos(t)")

plt.ylabel("y = cos(t) - t\*sin(t)")

# 显示图例

plt.legend()

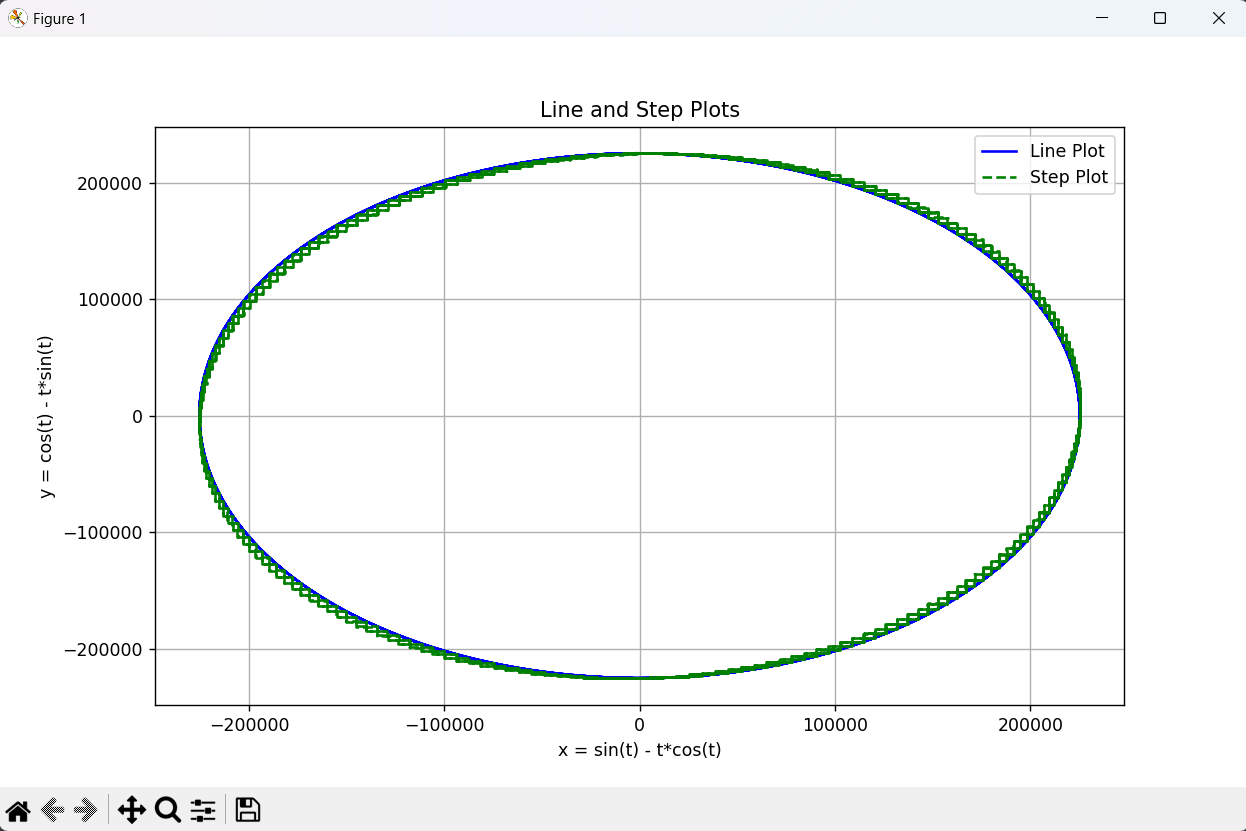
# 显示网格

plt.grid()

# 展示图形

plt.show()

结果如下：



**四、实验感想**

这次实验涵盖了多种数据可视化的内容，通过使用 Python 的 matplotlib 和 numpy 库，我深入理解了图形绘制的技巧以及如何通过图形更直观地表达数据。在进行实验时，我不仅加深了对数据可视化工具的掌握，也体会到了图形化展示数据的重要性。以下是我对每个实验部分的具体感想：

**三角函数的绘制**： 绘制 sin、cos、tan 函数的曲线让我对三角函数的图形变化有了更直观的认识。尤其是 tan 函数，由于其无限增长的特性，限制纵轴范围使得图形看起来更清晰，也让我学到了如何在实际绘图中处理类似的特殊情况。

**三维曲线的绘制**： 使用三维图形展示了数学方程的多维空间，这让我对三维数据的可视化有了更深入的理解。对 z 轴应用对数刻度后，我意识到通过合适的轴变换，可以让数据展示更符合实际需求，尤其是当数据范围跨度很大时。

**班级成绩的直方图**： 通过随机生成成绩并绘制直方图，我体会到了数据分布的可视化意义。直方图清楚地展示了成绩的分布情况，这有助于我理解如何选择合适的区间（bins）来划分数据，以便得出有用的结论。

**十月份支出饼状图**： 在绘制支出饼状图时，我体会到了数据分类的重要性。通过饼状图清晰地看到各项支出所占比例，不仅能够帮助我分析自己支出的结构，也让我意识到数据展示方式的选择对于分析结果的影响。

**散点图、曲线图和阶梯图**： 通过绘制不同类型的图形，我了解了如何展示数据之间的关系。散点图适合展示离散的数据点，曲线图能够清晰地表示数据的趋势，而阶梯图则很好地展示了数据变化的分段过程。这些图形不仅让我学到了如何选择不同的图形类型，也让我意识到如何通过图形来更好地表达数据。

总的来说，通过这次实验，我不仅加深了对 Python 可视化库的掌握，也更加明白了数据可视化在分析和展示中的重要性。不同的图形类型和展示方式能够帮助我们从不同角度理解和分析数据，这对于后续的学习和实际应用都具有重要的意义。