

第一次作业

刘子安 PB20000069

1. 实验环境

- 操作系统: Windows11
- Python版本: Python 3.10.3
- numpy库版本: 1.22.3
- matplotlib库版本: 3.5.1

2. 实验内容

2.1 实验设计

使用 `pyplot.subplot` 方法将四幅图放在了一起, 为了防止不同图之间的变量相互冲突, 我们定义了四个函数, 再使用其返回的 `Axes` 分别绘制了四种图:

- 函数图, 主要使用 `plot` 方法
- 堆叠图 (条形图), 主要使用了 `bar` 方法
- 散点图, 主要使用了 `scatter` 方法
- 饼图, 主要使用了 `pie` 方法

2.2 核心代码分析

我们首先导入所需的包并起好别名。

```
import matplotlib.pyplot as plt
import numpy as np
```

然后设置参数使得绘制出的图可以正常显示中文和负号。

```
plt.rcParams['font.sans-serif']=['SimHei'] # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False # 用来正常显示负号
```

使用 `pyplot.figure` 函数生成窗口, 通过 `num` 参数设置窗口id为1, 通过 `figsize` 将图片大小设置为长宽都为8英寸, 通过 `dpi` 参数将窗口的分辨率设置为120。

```
plt.figure(num=1, figsize=(8,8), dpi=120)
```

1、初等函数图像

首先我们生成好对应的数据。

```
t = np.arange(-np.pi, np.pi, 0.01)
s = np.sin(t)
c = np.cos(t)
```

再将图片分成两行两列, 并使 `ax1` 为第一个子图。

```
ax1 = plt.subplot(221)
```

使用 `Axes.plot` 方法绘制对应函数图像。不同的图像使用不同的颜色参数 `color` 和不同的线条参数 `linestyle`，并指定对应线条宽度 `linewidth`，还有对应的函数名称 `label`，其中 `label` 参数部分使用了 latex 公式格式。

其中还使用了 `legend` 函数来设置对应标签的显示位置。

```
ax1.plot(t, s, color='red', linestyle='-', linewidth=3, label="$\sin{x}$")
ax1.plot(t, c, color='blue', linestyle='--', linewidth=3, label="$\cos{x}$")
ax1.plot(t, t/2, color='green', linestyle=':', linewidth=3, label="x/2")
ax1.legend(loc="best")
```

接下来我们设置图的标题。我们使用了 `Axes.set_title` 方法，并设置了标题字体 `fontsize` 大小为 14，位置 `loc` 为中间。

```
ax1.set_title('基本初等函数', fontsize=14, loc='center')
```

我们还使用了 `Axes` 中的 `axhline` 和 `axvline` 方法分别绘制对应的直线，即 $x = 0$ 和 $y = 0$ 。

```
ax1.axhline(y=0, ls='--', c='black')
ax1.axvline(x=0, ls='--', c='black')
```

使用 `Axes.scatter` 方法来标注点(0,0)，并且使用 `marker` 参数来指定点形状为圆，`s` 参数来指定标注点的大小。

```
ax1.scatter(0, 0, marker='o', s=80, c='black')
```

使用 `Axes` 的 `set_xlim` 和 `set_ylim` 来指定坐标轴的长度。

```
ax1.set_xlim(-4, 4)
ax1.set_ylim(-1, 1)
```

使用 `Axes` 的 `set_xlabel` 和 `set_ylabel` 方法来指定坐标轴的标签，并用 `size` 参数设置字体大小。

```
ax1.set_xlabel("X", size=12)
ax1.set_ylabel("Y", size=12)
```

2、堆叠图

我们仍然先生成数据并获取第二个子图。再使用 `Axes.bar` 方法绘制条形图，通过 `label` 和 `color` 参数来区分金银铜奖牌的数量，并使用 `bottom` 参数将三个图堆叠在一起。

```
ax2.bar(ind, golds, width=0.5, label='golds', color='gold',
bottom=silvers+bronzes)
ax2.bar(ind, silvers, width=0.5, label='silvers', color='#C0C0C0',
bottom=bronzes)
ax2.bar(ind, bronzes, width=0.5, label='bronzes', color='#B87333')
```

使用 Axes 的 `set_xticks` 和 `set_yticks` 来设置坐标轴的刻度及其大小。将x轴的刻度设置为国家名称。

```
ax2.set_xticks(ind, countries, fontsize=14)
ax2.set_yticks([5, 10, 15, 20, 25, 30, 35, 40], fontsize=14)
```

3、散点图

使用了 Axes.scatter 方法来绘制散点图，数据随机生成，并且通过 `s` 参数设置点大小，`c` 参数设置不同点的颜色，`alpha` 参数设置点的不透明度。

```
ax3.scatter(x, y, s=75, c=t, alpha=0.5)
```

4、饼图

使用 Axes.pie 方法绘制饼图，通过 `explode` 设定各块的偏移量，`label` 参数设置标签，`autopct` 参数设置数值格式，`%.1f%%` 代表保留一位小数。

```
ax4.pie(values, explode=explode, labels=label, autopct='%.1f%%')
```

5、绘制图像

最后我们使用 `plt.show` 函数来显示出我们绘制的图像。

```
plt.show()
```

上述四幅图中重叠的函数部分（如生成数据，取对应子图等）未重复解释。但是我们可以看出，使用 `matplotlib.pyplot` 库绘图的主要步骤为：

- 导入所需的包
- 获取或者生成需要绘制的数据
- 生成窗口并获取图像
- 在对应图像上使用合适的参数绘图
- 显示图像

3. 实验结果

