

Python实验报告6--参考答案

1 班级:

姓名:

学号:

要求:

- 在文档开头部分填写自己的信息;
- 在每个题下方的代码块中书写该题的代码，并运行出结果;
- 在2节课的时间内完成前6个题；打印为pdf文件并提交，文件名改为“Python实验6+班级姓名.pdf”。

一、实验目的:

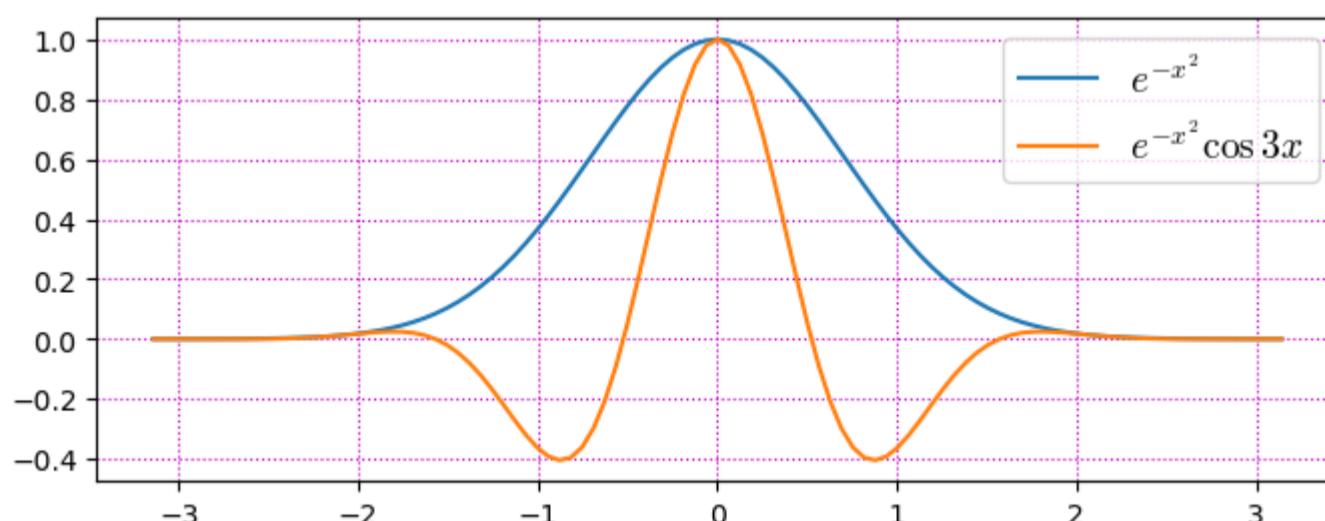
- 掌握曲线图、条形图、饼图、三维曲线、曲面的绘制;
- 掌握简单的图形控制及标注：图形大小、坐标轴范围、标题、图例等；
- 掌握多子图的绘制。

二、实验内容（在各题后书写程序和运行结果）

- 绘制函数 e^{-x^2} 和 $e^{-x^2} \cos 3x$ 在 $[-\pi, \pi]$ 上的图形，并添加适当的文字标注。

In []:

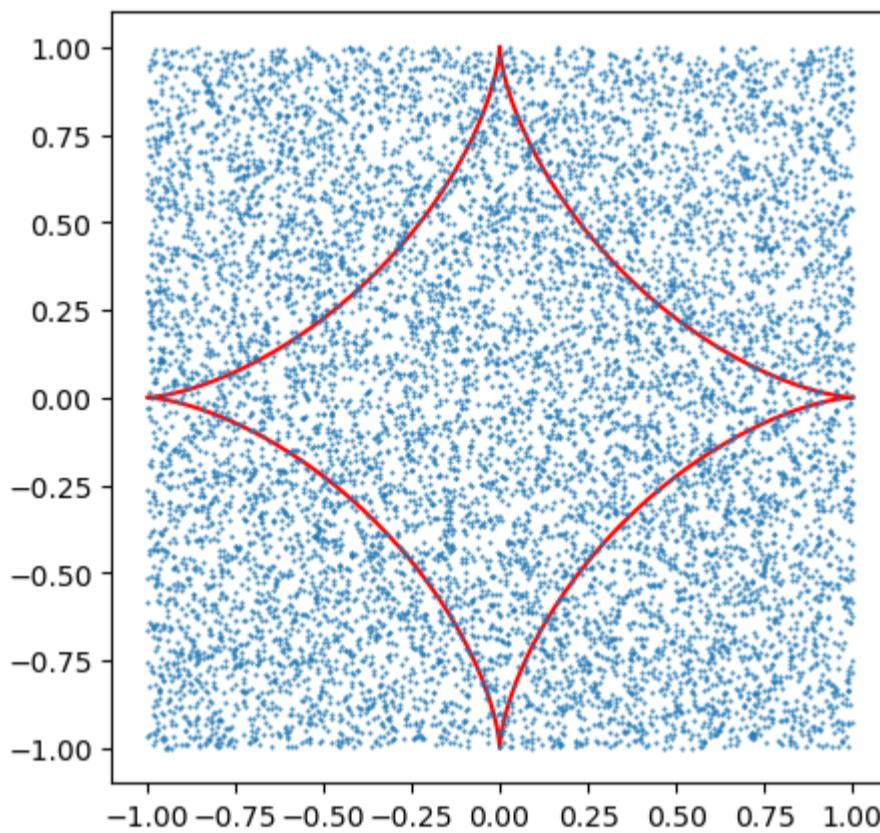
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(-np.pi, np.pi, 101)
5 y1 = np.exp(-x**2)
6 y2 = y1*np.cos(3*x)
7 plt.figure(figsize=(8, 3))
8 plt.plot(x, y1, x, y2)
9 prop = {'math_fontfamily': 'cm', 'size': 14} # 设定数学字体
10 plt.legend(['$e^{-x^2}$', '$e^{-x^2} \cos 3x$'], prop=prop)
11 plt.grid('on', color='m', linestyle=':')
12 plt.show()
```



- 对星形线 $x^{\frac{2}{3}} + y^{\frac{2}{3}} = 1$ 围成的区域，希望用蒙特卡罗方法求其面积。请你绘制一个示意图，以帮助大家理解。

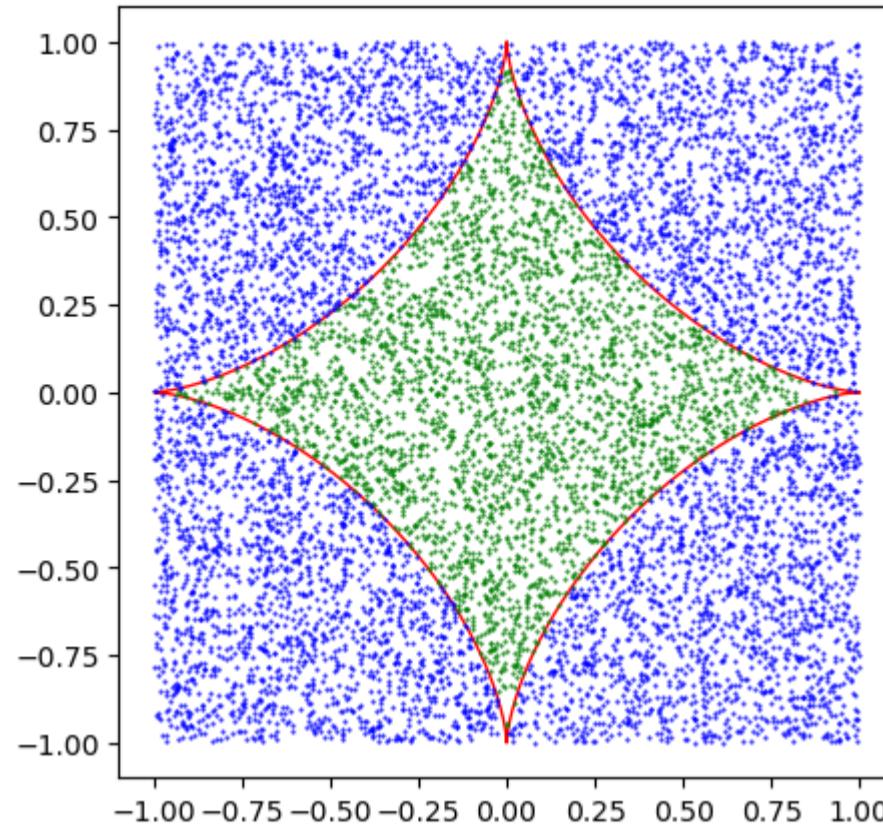
In []:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # 绘制曲线
5 t = np.linspace(0, 2*np.pi, 401)
6 x = np.cos(t)**3
7 y = np.sin(t)**3
8 plt.figure(figsize=(5, 5))
9 plt.plot(x, y, 'r')
10 # 绘制随机点, 注: 不能用random.randn(), randn是正态分布
11 x = np.random.uniform(-1, 1, 10000) # x = np.random.rand(10000)*2-1
12 y = np.random.uniform(-1, 1, 10000) # y = np.random.rand(10000)*2-1
13 plt.plot(x, y, '.', markersize=1)
14 plt.show()
```



In []:

```
1 # 以下的图中, 见区域内外的点用不同颜色表现
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # 绘制曲线
6 t = np.linspace(0, 2*np.pi, 401)
7 x = np.cos(t)**3
8 y = np.sin(t)**3
9 plt.figure(figsize=(5, 5))
10 plt.plot(x, y, 'r', linewidth=1)
11 # 绘制随机点, 注: 不能用random.randn(), randn是正态分布
12 x = np.random.uniform(-1, 1, 10000) # x = np.random.rand(10000)*2-1
13 y = np.random.uniform(-1, 1, 10000) # y = np.random.rand(10000)*2-1
14 flag = (x*x)**(1/3)+(y*y)**(1/3)<=1 # 区域内
15 plt.plot(x[flag], y[flag], 'g.', markersize=1)
16 plt.plot(x[~flag], y[~flag], 'b.', markersize=1)
17 plt.show()
```



3. 2022年全球主要经济体的工业增加值前10为 ['中国大陆', '美国', '欧盟', '日本', '俄罗斯', '印度', '沙特', '韩国', '印尼', '英国']，数额依次为 [59714.46, 37173.58, 30852.27, 9157.12, 6297.45, 6155.51, 5411.46, 4425.21, 4169.40, 3798.07]；这些经济体也是2021年工业增加值的前10，但顺序有所差异，其数额依次为 [58055.60, 32091.75, 30694.91, 11651.53, 5010.89, 5747.71, 3430.38, 4940.82, 3488.52, 3760.97]。请按要求在同一图形窗口中绘制下列3个图形：(1) 在一组条形图中同时表现2022年和2021年工业增加值的前10；(2) 以两个饼图分别表现2022年和2021年的工业增加值，要求按顺序排列。(3) 条形图放在上方，占一半空间，两个饼图放在下方，各占剩余的一半空间，并添加适当的文字标注。

In []:

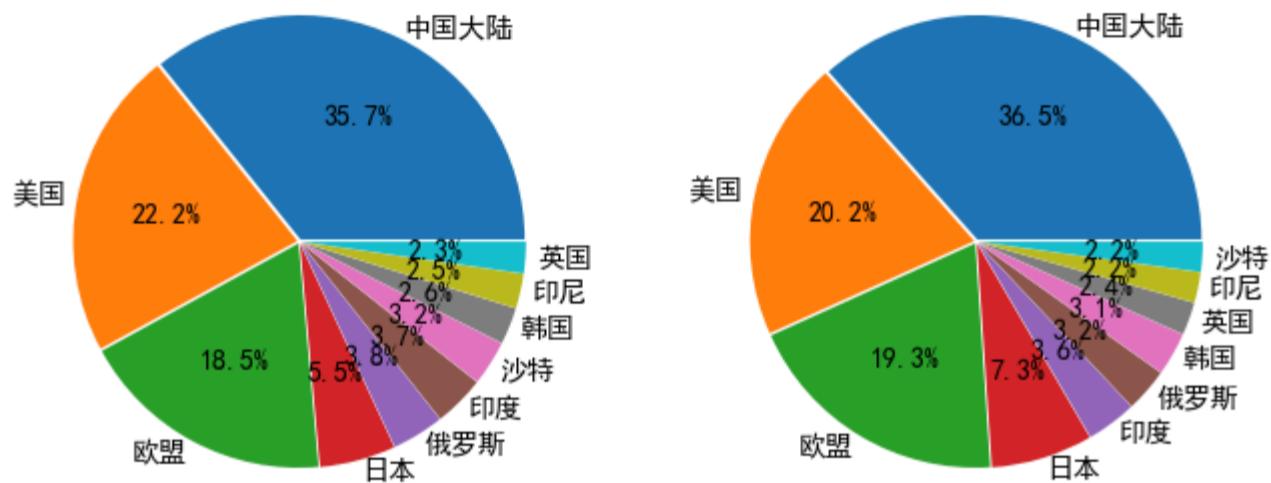
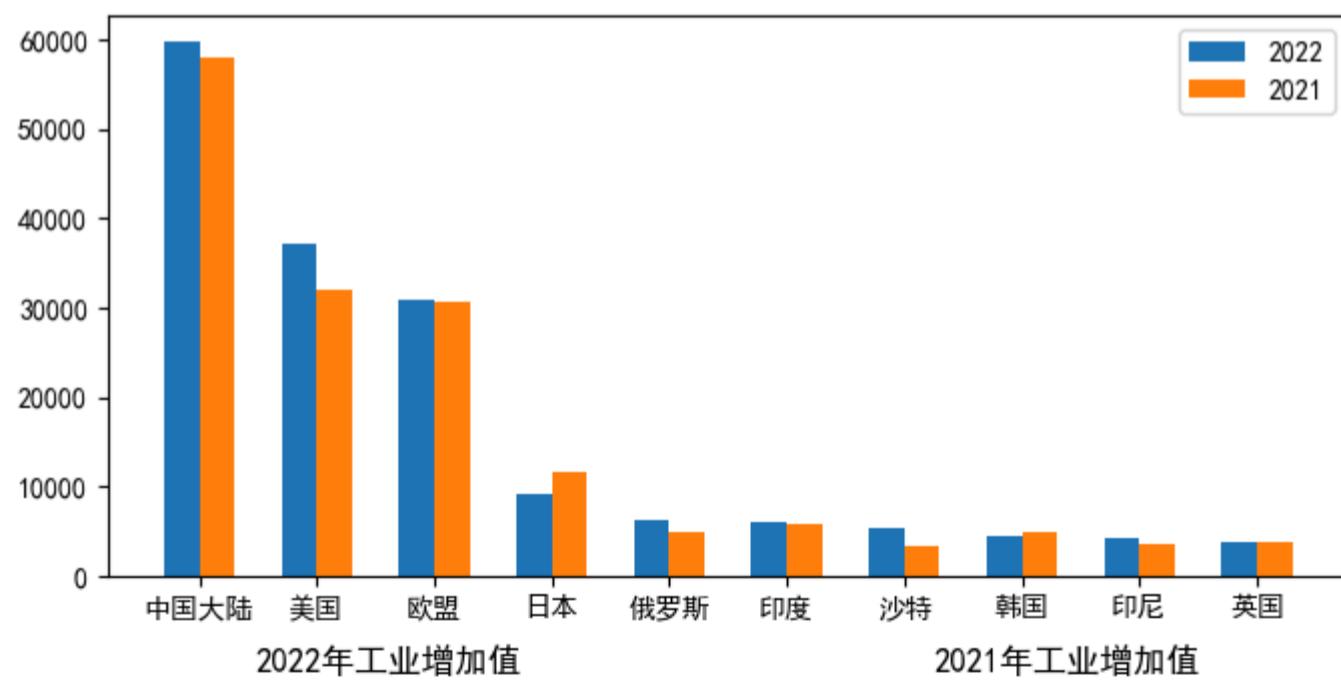
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 economy = np.array(['中国大陆', '美国', '欧盟', '日本', '俄罗斯',
5                     '印度', '沙特', '韩国', '印尼', '英国'])
6 indust2022 = np.array([59714.46, 37173.58, 30852.27, 9157.12,
7                       6297.45, 6155.51, 5411.46, 4425.21, 4169.40, 3798.07])
8 indust2021 = np.array([58055.60, 32091.75, 30694.91, 11651.53,
9                       5010.89, 5747.71, 3430.38, 4940.82, 3488.52, 3760.97])
```

In []:

```

1 plt.figure(figsize=(8, 8))
2 # 条形图
3 plt.subplot(211)
4 plt.bar(np.arange(10)-0.15, indust2022, width=0.3, label='2022')
5 plt.bar(np.arange(10)+0.15, indust2021, width=0.3, label='2021')
6 plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文字体
7 plt.xticks(np.arange(10), economy)
8 plt.legend()
9 # 饼图
10 explode = 0.01 * np.ones(len(indust2022))
11 plt.subplot(223)
12 plt.pie(indust2022, explode=explode, labels=economy,
13         autopct='%.1f%%', labeldistance=1.05)
14 plt.title('2022年工业增加值')
15 plt.subplot(224)
16 ind = np.argsort(indust2021)[::-1] # 由大到小排序
17 plt.pie(indust2021[ind], explode=explode, labels=economy[ind],
18         autopct='%.1f%%', labeldistance=1.05)
19 plt.title('2021年工业增加值')
20 plt.show()

```

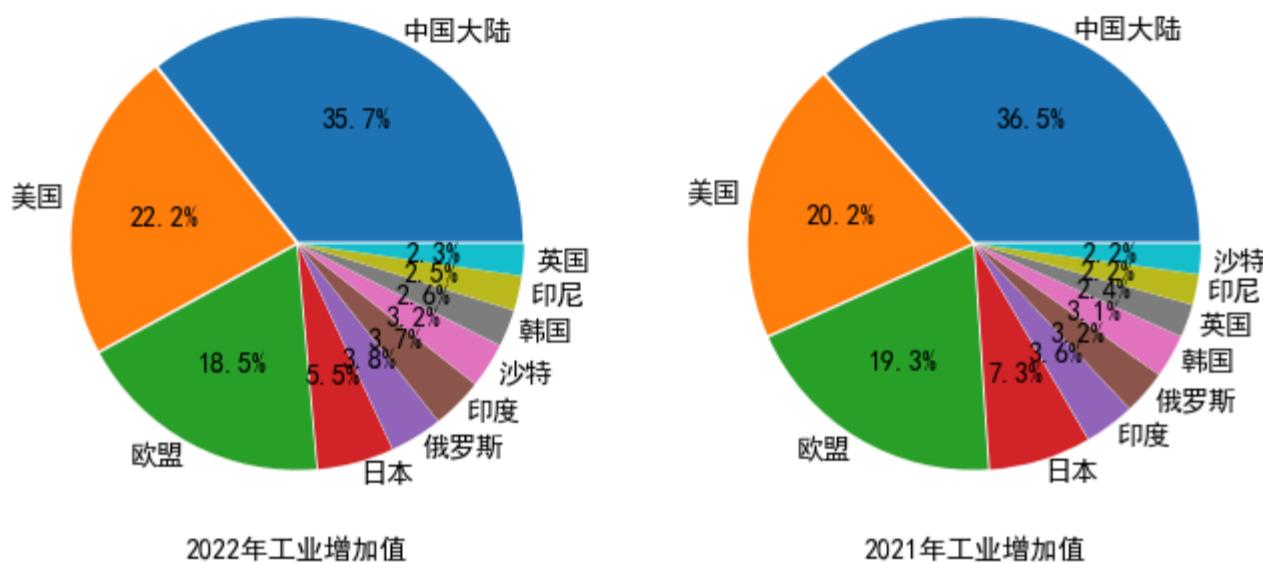
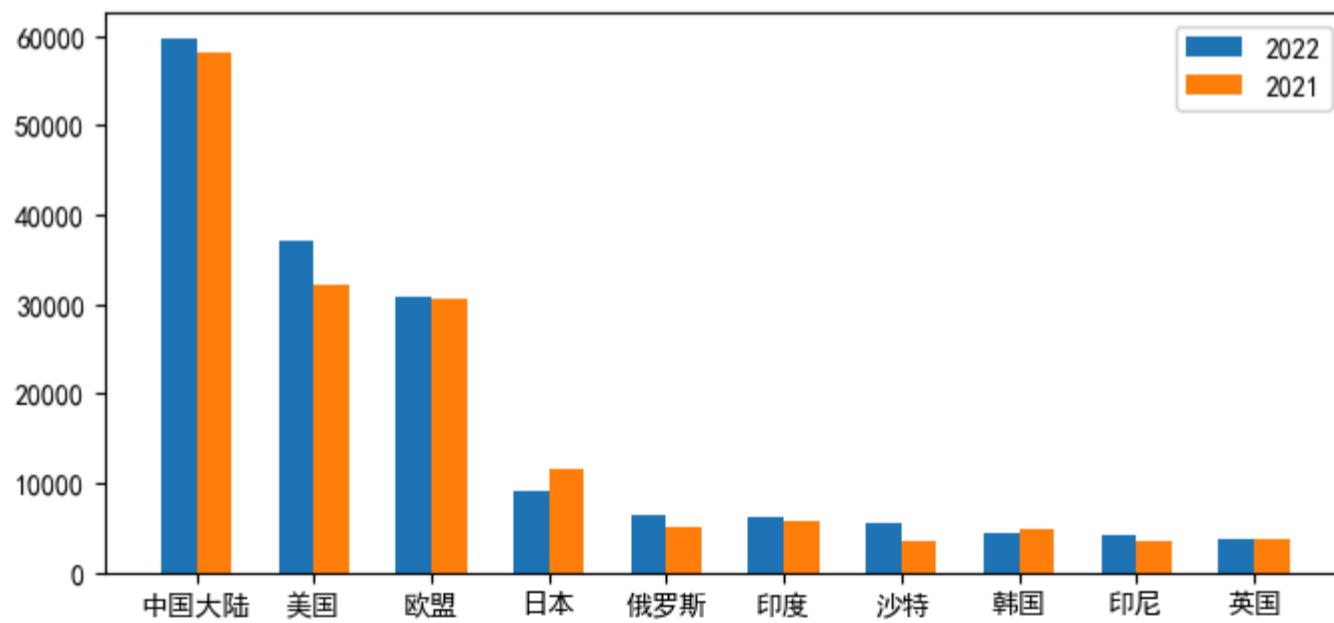


In []:

```

1 # 以下用subplot_mosaic()安装子图的布局
2 layout = [['A', 'A'],
3             ['B', 'C']]
4 fig, ax = plt.subplot_mosaic(layout, figsize=(8, 8))
5 xa, xb, xc = ax.values()
6 # 条形图
7 xa.bar(np.arange(10)-0.15, indust2022, width=0.3, label='2022')
8 xa.bar(np.arange(10)+0.15, indust2021, width=0.3, label='2021')
9 xa.set_xticks(np.arange(10), economy)
10 xa.legend()
11 # 饼图
12 explode = 0.01 * np.ones(len(indust2022))
13 xb.pie(indust2022, explode=explode, labels=economy,
14         autopct='%.1f%%', labeldistance=1.05)
15 xb.set_xlabel('2022年工业增加值')
16 ind = np.argsort(indust2021)[-1:] # 由大到小排序
17 xc.pie(indust2021[ind], explode=explode, labels=economy[ind],
18         autopct='%.1f%%', labeldistance=1.05)
19 xc.set_xlabel('2021年工业增加值')
20 plt.show()

```

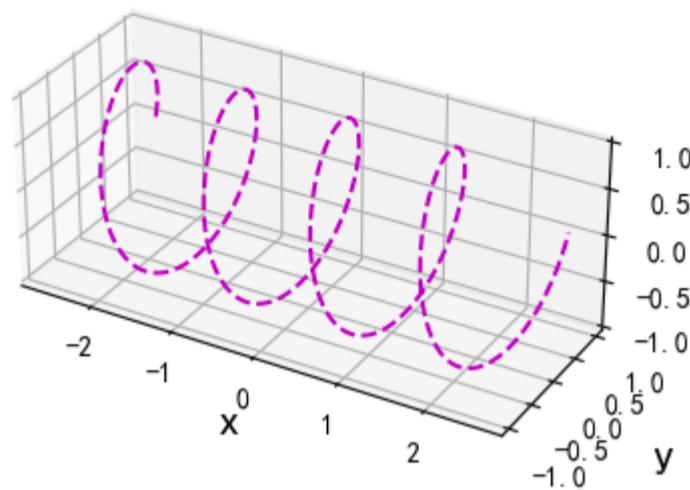


4. 绘制三维曲线 $\begin{cases} x = \frac{t}{5}, \\ y = \cos t, \\ z = \sin t. \end{cases}$, $t \in [-4\pi, 4\pi]$, 要求坐标轴单位长度相等, 并添加适当的文字标注。

In []:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 fig = plt.figure(figsize=(6, 4))
5 ax = fig.add_subplot(projection='3d')
6 t = np.linspace(-4*np.pi, 4*np.pi, 801)
7 x = t/5
8 y = np.cos(t)
9 z = np.sin(t)
10 ax.plot(x, y, z, 'm--')
11 ax.set_box_aspect((np.ptp(x), np.ptp(y), np.ptp(z)))
12 plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文字体
13 plt.rcParams['axes.unicode_minus'] = False # 设置正常显示负号
14 ax.set_title('三维曲线')
15 ax.set_xlabel('x', fontsize=16)
16 ax.set_ylabel('y', fontsize=16)
17 ax.set_zlabel('z', fontsize=16)
18 plt.show()
```

三维曲线



5. 分左右子图，绘制曲面 $z = xe^{-x^2-y^2}$ 在区域 $x, y \in [-2, 2]$ 上的图形，以及在区域 $x^2 + y^2 \leq 4$ 上的图形.

In []:

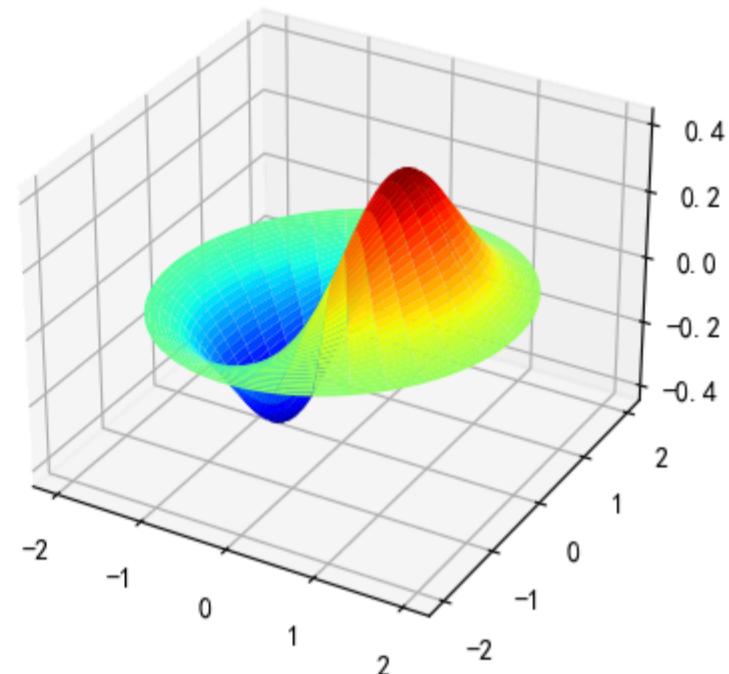
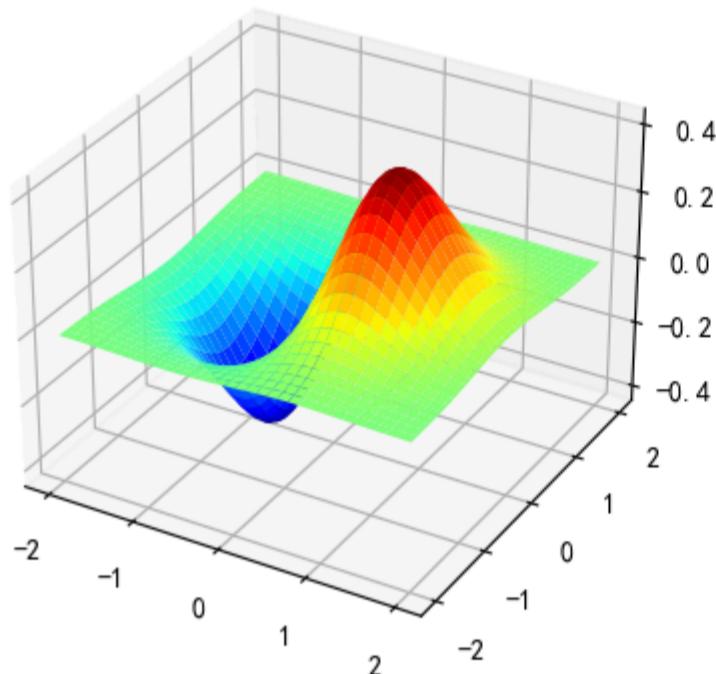
```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # 左子图绘制方域上的图形
5 x = np.linspace(-2, 2, 101)
6 X, Y = np.meshgrid(x, x)
7 Z = X*np.exp(-X*X-Y*Y)
8 fig = plt.figure(figsize=(10, 5))
9 ax1 = fig.add_subplot(121, projection='3d')
10 ax1.plot_surface(X, Y, Z, cmap='jet')
11 prop = {'math_fontfamily': 'cm', 'size': 14} # 设定数学字体
12 ax1.set_title('$z=xe^{-x^2-y^2}, \forall x, y \in [-2, 2]$', fontdict=prop)
13 # 右子图绘制圆域上的图形
14 t = np.linspace(0, 2*np.pi, 101)
15 r = np.linspace(0, 2, 41)
16 r, t = np.meshgrid(r, t)
17 x = r*np.cos(t)
18 y = r*np.sin(t)
19 z = x*np.exp(-r*r)
20 ax2 = fig.add_subplot(122, projection='3d')
21 ax2.plot_surface(x, y, z, cmap='jet')
22 ax2.set_title('$z=xe^{-x^2-y^2}, \forall x^2+y^2 \leq 4$', fontdict=prop)
23 plt.show()

```

$$z = xe^{-x^2 - y^2}, \quad x, y \in [-2, 2]$$

$$z = xe^{-x^2 - y^2}, \quad x^2 + y^2 \leq 4$$



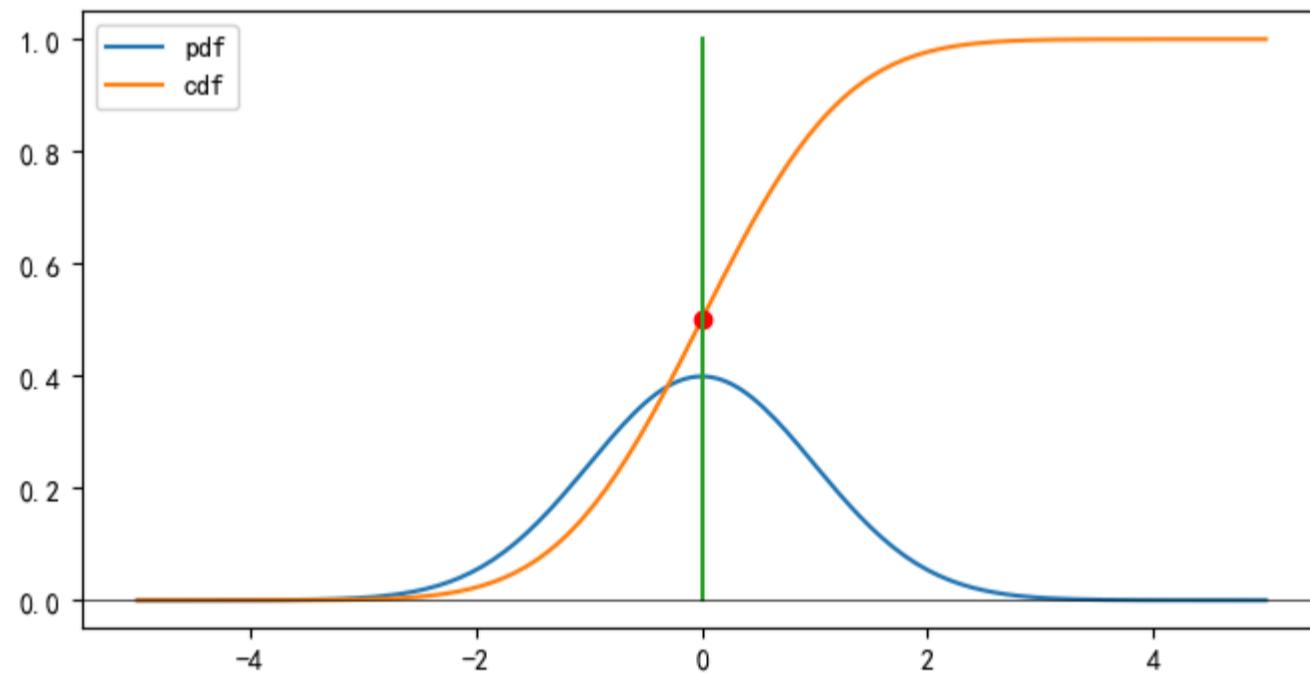
6. 在同一个坐标系中，绘制标准正态分布的概率密度函数 $\rho(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, x \in (-\infty, +\infty)$ 的曲线，以及其分布函数 $f(x) = \int_{-\infty}^x \rho(t)dt$ 的曲线， x 的范围可取 $x \in [-5, 5]$ 。分布函数可以用数值积分计算。添加适当文字标注。

In []:

```

1 a, b, n = -5, 5, 201
2 x = np.linspace(a, b, n)
3 p = np.exp(-x*x/2)/np.sqrt(2*np.pi) # 标准正态分布的概率密度函数
4 # 以下用梯形积分求分布函数
5 dx = (b-a)/(n-1) # 步长
6 F = np.cumsum(p[0:-1] + p[1:])/2*dx
7 F = np.insert(F, 0, 0) # 在头部插入0
8 # 方法2：分布函数也可以用scipy中的integrate模块的cumtrapz函数如下求解：
9 # from scipy.integrate import cumtrapz
10 # F = cumtrapz(p, x, initial=0)
11 # 方法3：分布函数也可以用scipy中的integrate模块的quad函数如下求解：
12 # from scipy.integrate import quad
13 # def pdf(x):
14 #     return np.exp(-x*x/2)/np.sqrt(2*np.pi)
15 # def cdf(x):
16 #     return quad(pdf, -np.inf, x)[0]
17 # F = np.array([cdf(i) for i in x])
18 # 方法4：正态分布函数也可以使用Scipy的特殊函数 ndtr
19 # from scipy.special import ndtr
20 # F = ndtr(x)
21 # 绘制图形，添加标注
22 fig = plt.figure(figsize=(8, 4))
23 plt.plot(x, p, x, F)
24 plt.legend(['pdf', 'cdf'])
25 plt.plot(0, 0.5, 'ro', [0, 0], [0, 1]) # 绘制对称轴
26 plt.axhline(y=0, color='k', linewidth=0.5) # x轴
27 plt.show()

```

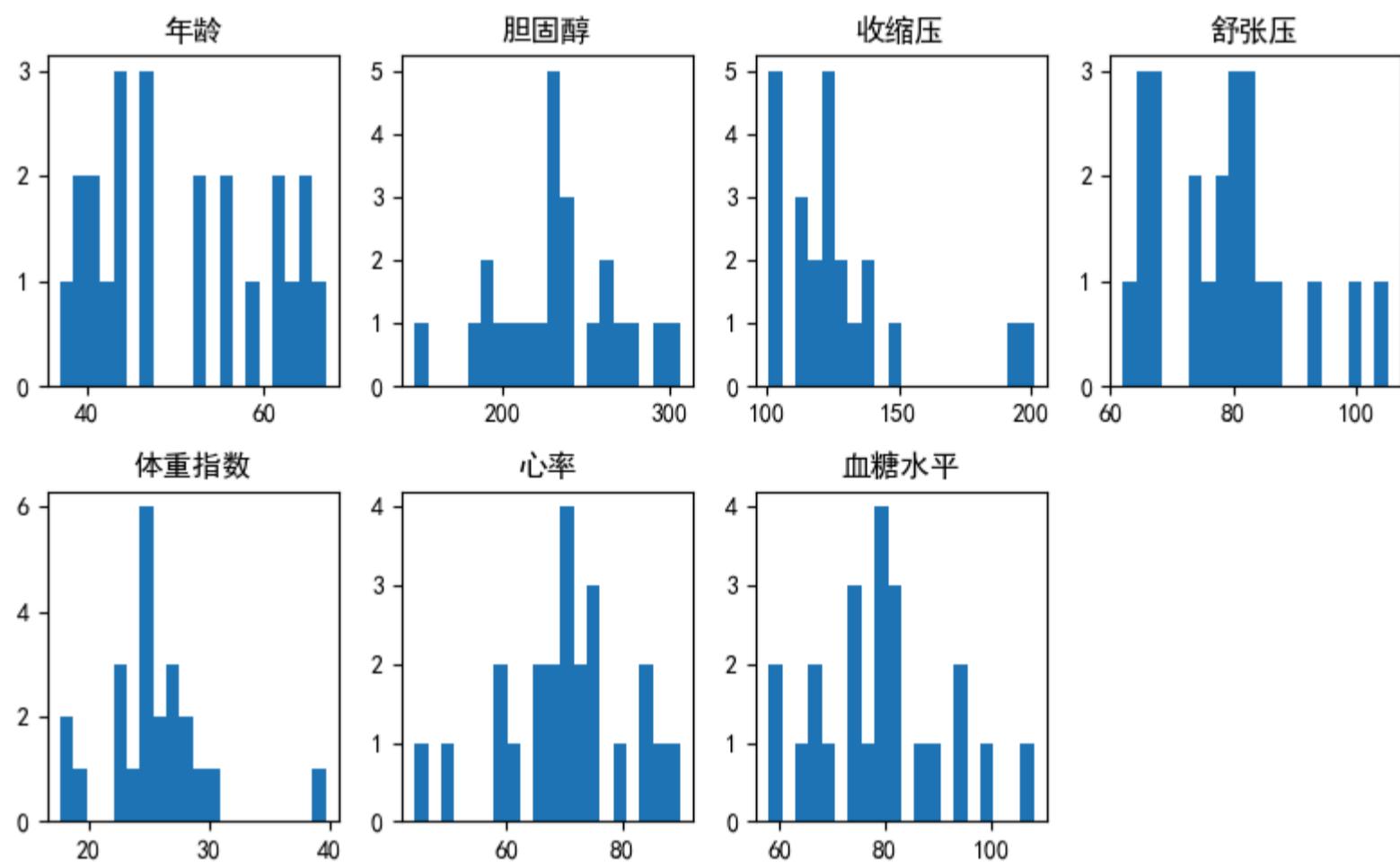


7. 一组样本的多个特征之间可能存在不同程度的相关性，其大小可以用热力图做可视化的呈现。现有来自志愿者的一组健康数据'health.csv'，各列分别为年龄、胆固醇、收缩压、舒张压、体重指数、心率、血糖水平等特征。试用直方图表现各特征的分布情况，然后求各特征间的相关系数，并将相关系数矩阵用热力图表现。

数据集已在下方导入。

In []:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 label = ['年龄', '胆固醇', '收缩压', '舒张压', '体重指数', '心率', '血糖水平']
6 with open('health.csv', 'r') as file:
7     health = pd.read_csv(file, skiprows=1, names=label)
8 # 绘制直方图
9 fig = plt.figure(figsize=(8, 5))
10 for k, feature in enumerate(label, 1):
11     plt.subplot(2, 4, k)
12     plt.hist(health[feature], bins=20)
13     plt.title(feature)
14 plt.tight_layout()
15 plt.show()
```

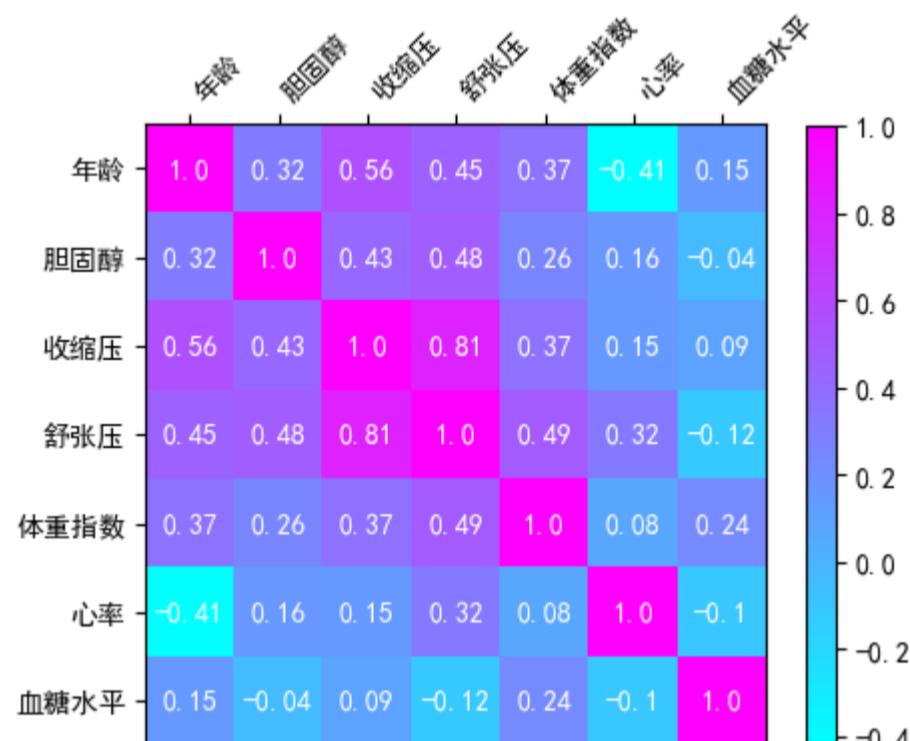


In []:

```

1 # 求相关系数矩阵，并绘制热力图
2 corr = np.corrcoef(health, rowvar=False)
3 fig = plt.figure(figsize=(5, 5))
4 plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文字体
5 plt.rcParams['axes.unicode_minus'] = False # 设置正常显示负号
6 im = plt.imshow(corr, cmap='cool')
7 plt.colorbar(im, shrink=0.8)
8 r, c = corr.shape
9 for i in range(r):
10     for j in range(c):
11         plt.text(j, i, np.round(corr[i, j], 2),
12                  ha="center", va="center", color="w")
13 ax = plt.gca()
14 ax.xaxis.set_label_position('top') # 坐标轴上置
15 ax.xaxis.tick_top() # 刻度朝上
16 plt.xticks(range(len(label)), label, rotation=45, ha='left') # 刻度标签旋转并左对齐
17 plt.yticks(range(len(label)), label)
18 plt.show()

```



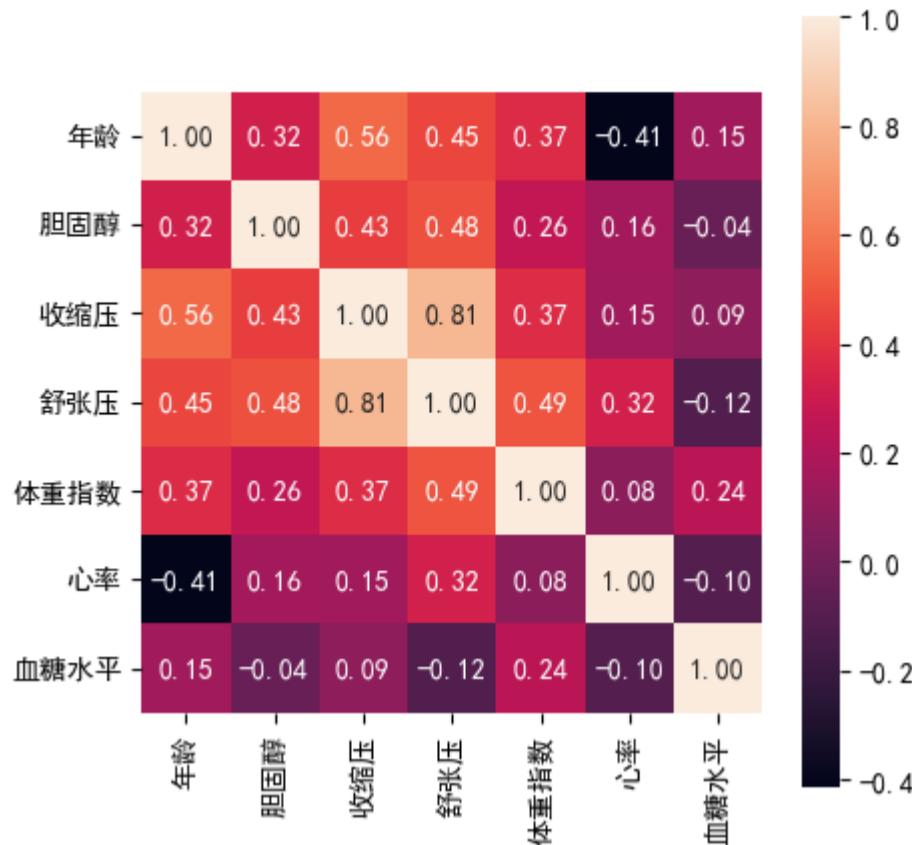
1 以上的程序中直接用matplotlib.pyplot中的imshow()以伪彩色显示矩阵，就可以得到热力图，使用不同的cmap就能得到不同的颜色配置。另一个常用的方法是使用seaborn中的heatmap()，能更方便地显示热力图。

In []:

```

1 # 承上，使用seaborn中的heatmap()绘制热力图
2 import seaborn as sns
3 fig = plt.figure(figsize=(5, 5))
4 sns.heatmap(corr, annot=True, square=True, fmt='.2f',
5             xticklabels=label, yticklabels=label)
6 plt.show()

```



8. 鸢尾花数据集'iris.csv'常用于演示分类问题。其中共有3个品种: Iris Setosa (山鸢尾)、Iris Versicolour (杂色鸢尾) , Iris Virginica (维吉尼亚鸢尾) , 各品种均有50个样本, 4个特征: Sepal.Length (花萼长度)、Sepal.Width (花萼宽度)、Petal.Length (花瓣长度)、Petal.Width (花瓣宽度)。数据集的前4列为特征数据, 最后一列为品种代号。请对这些特征两两组合 (如花萼长度做x轴, 花萼宽度做y轴) 绘制散点图, 对不同品种用不同颜色的点表示。

数据集已在下方导入。

- 1 以上要求将特征两两组合, 分品种绘制散点图, 可以使用seaborn中的pairplot()绘制成对关系图, 以品种为分类依据即可。以下纯以matplotlib绘制。

In []:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 label = ['萼长', '萼宽', '瓣长', '瓣宽', '品种']
6 # 导入数据集
7 iris = pd.read_csv('iris.csv', skiprows=1, names=label)
8 iris.head()
9 # 绘制多个散点图
10 target = iris['品种'].values
11 num = np.unique(target) # 不重复的品种代号
12 fig = plt.figure(figsize=(10, 6)) # 创建图形窗口
13 plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文字体
14 plt.rcParams['axes.unicode_minus'] = False # 设置正常显示负号
15 color = ['blue', 'orange', 'green']
16 marker = ['.', '*', '+']
17 i = 0 # 子图序号
18 for k in range(len(num)):
19     var1 = iris.iloc[:, k] # 取第 1 个特征
20     for n in range(k + 1, 4):
21         var2 = iris.iloc[:, n] # 取第2个特征
22         # 以下画第 i 个子图
23         i = i + 1
24         plt.subplot(2, 3, i)
25         plt.xlabel(label[k]) # 设置x轴标注
26         plt.ylabel(label[n]) # 设置y轴标注
27         for m in range(len(num)):
28             ft1 = var1[target == num[m]]
29             ft2 = var2[target == num[m]]
30             plt.scatter(ft1, ft2, color=color[m], marker=marker[m])
31         plt.legend(num) # 设置图例
32 plt.tight_layout()
33 plt.show()
```

