1.用Python语言编程实现一个排序算法。数据随机生成。

代码截图如下

```python
1    # 用 Python 语言编程实现一个排序算法。数据随机生成。
2    import random
3
4    #待排序 list
5    nums = []
6
7    #随机生成数据
8    len = random.randint(10,20)
9    for i in range(len):
10       nums.append(random.randint(1,100))
11
12   #打印排序前的数据
13   print("before sort:")
14   print(nums)
15
16   # 冒泡排序
17   for i in range(len):
18       for j in range(0,len-i-1):
19           if(nums[j] > nums[j+1]):
20               nums[j],nums[j+1] = nums[j+1],nums[j]
21
22   # 打印排序后的数据
23   print("after sort:")
24   print(nums)
```

运行结果截图如下

```
before sort:
[30, 29, 32, 21, 17, 55, 94, 94, 23, 35, 8, 8, 29, 77, 98]
after sort:
[8, 8, 17, 21, 23, 29, 29, 30, 32, 35, 55, 77, 94, 94, 98]

Process finished with exit code 0
```

2.用Matplotlib（1）画出高斯分布概率密度函数（2）画出sigmoid 函数 $y = \frac{1}{1+e^{-x}}$
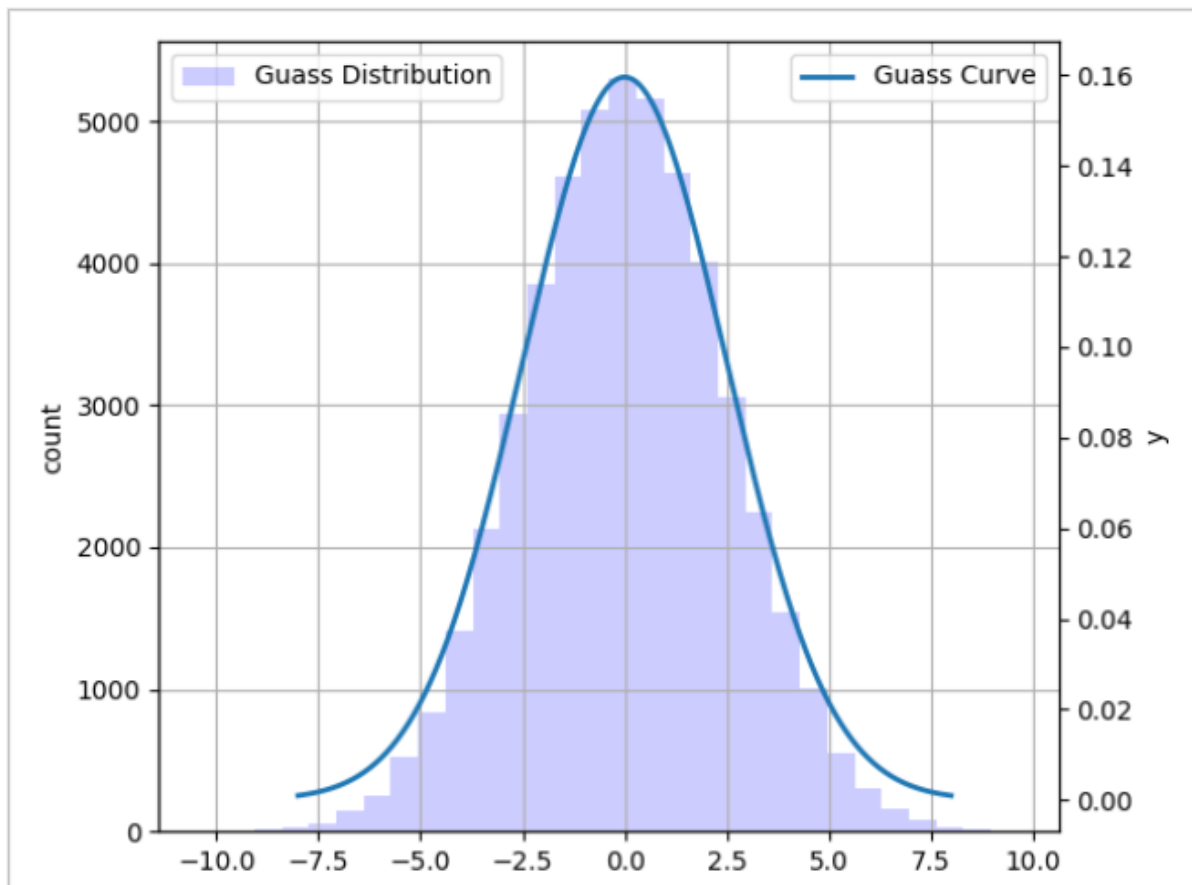
（1）**代码截图如下**

```
1    import numpy as np
2    from matplotlib import pyplot as plt
3
4    #高斯概率分布均值和方差取值
5    mu = 0
6    sigma = 2.5
7
8    #概率分布密度函数
9    x = np.linspace(-8,8,1000)
10   y = np.power(np.e,-((x-mu)*(x-mu)/(2*sigma*sigma)))/(np.sqrt(2*np.pi)*sigma)
11
12   #坐标轴设置
13   fig, ax1 = plt.subplots()
14   ax1.set_ylabel('count')
15   ax2 = ax1.twinx()
16   ax2.set_ylabel('y')
17
18   #高斯概率生成数字
19   f = np.random.normal(mu,sigma,50000)
20   ax1.hist(f,30,color="b",alpha=0.2,label="Guass Distribution")
21   ax1.legend(loc="upper left")
22   ax1.grid()
23
24   #图像展示
25   ax2.plot(x,y,linewidth=2.0,label="Guass Curve")
26   ax2.legend(loc="upper right")
27   plt.show()
```

（1）**运行结果截图如下**

（2）**代码截图如下**

```
1    import numpy as np
2    from matplotlib import pyplot as plt
3
4    #sigmoid函数表达式
5    x = np.linspace(-8,8,1000)
6    y = 1/(1+pow(np.e,-x))
7
8    #坐标轴设置
9    plt.figure()
10   ax = plt.gca()
11   ax.spines['right'].set_color('none')
12   ax.spines['top'].set_color('none')
13   ax.spines['left'].set_position(('data',0))
14   ax.spines['bottom'].set_position(('data',0))
15
16   #图像展示
17   plt.plot(x,y,linewidth=2.0,label="Sigmoid Curve")
18   plt.legend(loc="best")
19   plt.show()
```

（2）**运行结果截图如下**