姓名：李宇豪

学号：21305412

周数：3

成绩：

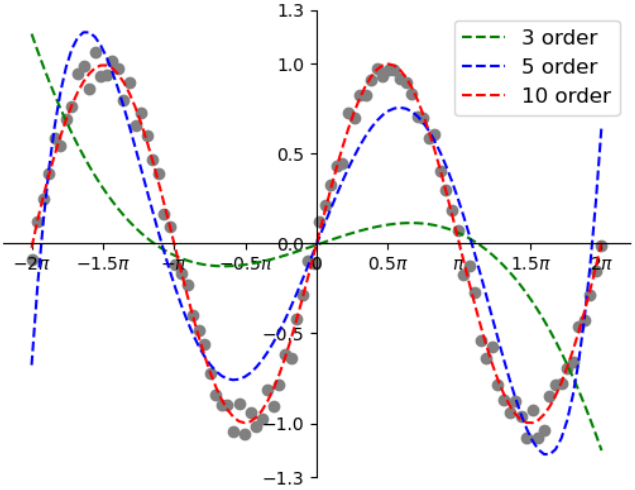
请生成 10000 个在[-2π,2π]区间内的正弦函数上的点，并且给这些点加上一些随机的噪声，把训练集和测试集按 4:1 的比例随机划分，分别用 3、5、10 阶多项式拟合数据集，画出对应的包含真实数据点和预测函数的图片，输出各种情况下训练集特征矩阵的形状，输出各种情况下的训练集和测试集的 R-squared 拟合评分，输出模型拟合的前两项参数，输出在 x=1 这一点的预测值与真实值（sin(1)）的差距。

代码：

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from matplotlib import pyplot as plt

# 数据集
n_dots = 10000
x = np.linspace(-2 * np.pi, 2 * np.pi, n_dots)
y = np.sin(x) + 0.2 * np.random.rand(n_dots) - 0.1
# 由于采样率过高时绘制出来的图像过于稠密，因此单独设置一个用于画图的数据集
x_plot = np.linspace(-2 * np.pi, 2 * np.pi, 100)
y_plot = np.sin(x_plot) + 0.2 * np.random.rand(100) - 0.1

# 划分数据集
Xtrain, Xtest, Ytrain, Ytest = train_test_split(x.reshape(-1, 1),
                                                y, test_size=0.2)
degree_list = [3, 5, 10]

# 一些循环前的准备
color_list = ["green", "blue", "red"]
p = []
x_plot_reshaped = x_plot.reshape(-1, 1)

for i in range(3):
    print("{}阶多项式拟合".format(degree_list[i]))

    # 获得特征矩阵
    featurizer = PolynomialFeatures(degree=degree_list[i])
    Xtrain_transformed = featurizer.fit_transform(Xtrain)
    Xtest_transformed = featurizer.transform(Xtest)
    x_plot_transformed = featurizer.transform(x_plot_reshaped)

    # 进行拟合
    regressor = LinearRegression()
    regressor.fit(Xtrain_transformed, Ytrain)
```

```
    # 画图
    p.append(plt.plot(x_plot, regressor.predict(x_plot_transformed),
             color=color_list[i], linestyle="--")[0])

    # 输出
    print("训练集特征矩阵的形状是：", Xtrain_transformed.shape)
    print("训练集 R-squared 拟合评分：%.3f" %
          regressor.score(Xtrain_transformed, Ytrain))
    print("测试集 R-squared 拟合评分：%.3f" %
          regressor.score(Xtest_transformed, Ytest))
    print("模型的前两项参数是：%.1f %.3f" % (regressor.coef_[0],
          regressor.coef_[1]))
    print("在 x=1 这一点的预测值与真实值 sin(1)的差距是：",
          regressor.predict(featurizer.transform([[1]]))[0] - np.sin(1))
    print("------------------")

    # 销毁对象，进入下一次循环
    del regressor, featurizer

# 画图
plt.scatter(x_plot, y_plot, c="gray")

plt.xticks((-2*np.pi, -1.5*np.pi, -1*np.pi, -0.5*np.pi, 0, 0.5*np.pi,
          np.pi, 1.5*np.pi, 2*np.pi),('$-2\pi$', '$-1.5\pi$', '$-\pi$',
          '$-0.5\pi$', '0', '$0.5\pi$', '$\pi$', '$1.5\pi$', '$2\pi$'))
plt.yticks([-1.3, -1, -0.5, 0, 0.5, 1, 1.3])

ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data', 0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data', 0))

legend = plt.legend([p[0], p[1], p[2]], ["3 order", "5 order",
                    "10 order"],fontsize=12, loc="upper right")
plt.show()
```

输出图像：



输出结果：

```
Python 控制台
3阶多项式拟合
训练集特征矩阵的形状是： (8000, 4)
训练集R-squared拟合评分：0.284
测试集R-squared拟合评分：0.295
模型的前两项参数是：0.0 0.088
在x=1这一点的预测值与真实值sin(1)的差距是： -0.7677074869262251
-----------------
5阶多项式拟合
训练集特征矩阵的形状是： (8000, 6)
训练集R-squared拟合评分：0.900
测试集R-squared拟合评分：0.904
模型的前两项参数是：0.0 0.638
在x=1这一点的预测值与真实值sin(1)的差距是： -0.27684271444771624
-----------------
10阶多项式拟合
训练集特征矩阵的形状是： (8000, 11)
训练集R-squared拟合评分：0.993
测试集R-squared拟合评分：0.993
模型的前两项参数是：0.0 0.990
在x=1这一点的预测值与真实值sin(1)的差距是： -0.005176790893755556
-----------------
```