import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

# 设置字体为支持中文的字体，例如SimHei

plt.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体

plt.rcParams['axes.unicode\_minus'] = False # 解决保存图像时负号'-'显示为方块的问题

# 创建示例学生学情数据

np.random.seed(0)

student\_data = pd.DataFrame({

'学生ID': range(1, 51), # 增加至50人

'周测成绩': np.random.randint(60, 100, 50),

'月测成绩': np.random.randint(65, 95, 50),

'期中成绩': np.random.randint(70, 100, 50), # 添加期中成绩

'期末成绩': np.random.randint(75, 100, 50), # 添加期末成绩

'小节测验': np.random.randint(60, 100, 50), # 添加小节测验成绩

'听课情况': np.random.choice(['好', '一般', '差'], size=50),

'作业情况': np.random.choice(['完成', '未完成'], size=50)

})

# 将听课情况和作业情况映射为数值

student\_data['听课情况\_数值'] = student\_data['听课情况'].map({'好': 3, '一般': 2, '差': 1})

student\_data['作业情况\_数值'] = student\_data['作业情况'].map({'完成': 2, '未完成': 1})

# 折线图：周测成绩和月测成绩趋势

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

sns.lineplot(data=student\_data[['周测成绩', '月测成绩']])

plt.title('周测和月测成绩趋势')

plt.xlabel('学生ID')

plt.ylabel('成绩')

plt.legend(['周测成绩', '月测成绩'])

plt.grid(True)

plt.show()

# 箱线图：不同考试成绩分布情况

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

sns.boxplot(data=student\_data[['周测成绩', '月测成绩', '期中成绩', '期末成绩']])

plt.title('不同考试成绩分布情况')

plt.ylabel('成绩')

plt.xlabel('考试类型')

plt.xticks(ticks=range(4), labels=['周测成绩', '月测成绩', '期中成绩', '期末成绩'])

plt.grid(True)

plt.show()

# 箱线图：听课情况与周测成绩分布情况

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

# 假设听课情况的类别是 ['好', '一般', '差']

sns.boxplot(x='听课情况', y='周测成绩', data=student\_data, order=['好', '一般', '差'])

plt.title('听课情况与周测成绩分布情况')

plt.ylabel('周测成绩')

plt.xlabel('听课情况')

plt.grid(True)

plt.show()

# 箱线图：作业情况与周测成绩分布情况

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

# 假设作业情况的类别是 ['完成', '未完成']

sns.boxplot(x='作业情况', y='周测成绩', data=student\_data, order=['完成', '未完成'])

plt.title('作业情况与周测成绩分布情况')

plt.ylabel('周测成绩')

plt.xlabel('作业情况')

plt.grid(True)

plt.show()

# 柱状图：听课情况分布

plt.figure(figsize=(8, 6))

sns.countplot(x='听课情况', data=student\_data)

plt.title('听课情况分布')

plt.xlabel('听课情况')

plt.ylabel('学生数量')

plt.grid(True)

plt.show()

# 散点图：周测成绩与月测成绩关系

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

sns.scatterplot(x='周测成绩', y='月测成绩', data=student\_data, hue='听课情况', style='作业情况', s=100)

plt.title('周测成绩与月测成绩关系')

plt.xlabel('周测成绩')

plt.ylabel('月测成绩')

plt.legend(title='听课情况')

plt.grid(True)

plt.show()

# 热力图：学习情况与周测成绩、月测成绩的相关性

corr\_matrix = student\_data[['周测成绩', '月测成绩', '期中成绩', '期末成绩', '小节测验', '听课情况\_数值', '作业情况\_数值']].corr()

plt.figure(figsize=(10, 8))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm')

plt.title('学习情况与测试成绩的相关性')

plt.show()

# 雷达图：学生学习情况

categories = ['周测成绩', '月测成绩', '期中成绩', '期末成绩', '小节测验', ]

values = student\_data[categories].mean().tolist()

# 计算雷达图的角度

num\_vars = len(categories)

angles = [n / float(num\_vars) \* 2 \* np.pi for n in range(num\_vars)]

values += values[:1] # 为了闭合雷达图，将第一个值重复添加到末尾

angles += angles[:1] # 同样地，将第一个角度重复添加到末尾

# 绘制雷达图

fig, ax = plt.subplots(figsize=(8, 8), subplot\_kw=dict(polar=True))

ax.plot(angles, values, 'o-', linewidth=2)

ax.fill(angles, values, alpha=0.25) # 填充雷达图区域

ax.set\_yticklabels([]) # 设置Y轴刻度标签为空

ax.set\_xticks(angles[:-1]) # 设置X轴刻度位置

ax.set\_xticklabels(categories) # 设置X轴刻度标签

plt.title('学生学习情况雷达图')

plt.grid(True)

plt.show()