复习题

1. 数据采集、数据储存、数据管理、数据计算、数据分析、数据展示
2. 数据采集是指从真实世界对象中获得原始数据的过程。

传感器、日志文件和python爬虫

1. 数据管理是利用计算机硬件和软件技术对数据进行有效的收集、存储、处理和应用的过程。

相同点：都对数据进行管理，并充分有效地发挥数据的作用

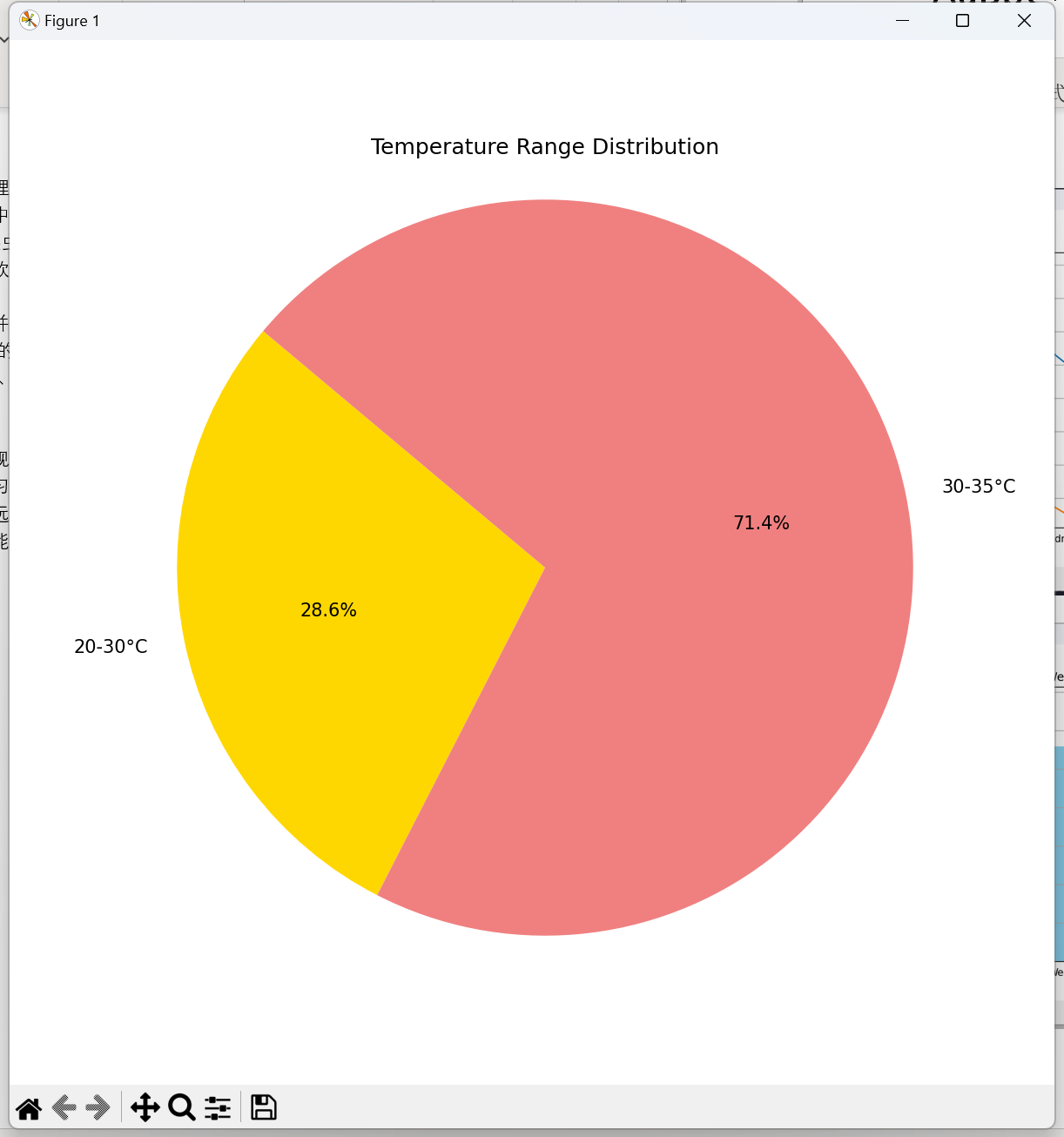
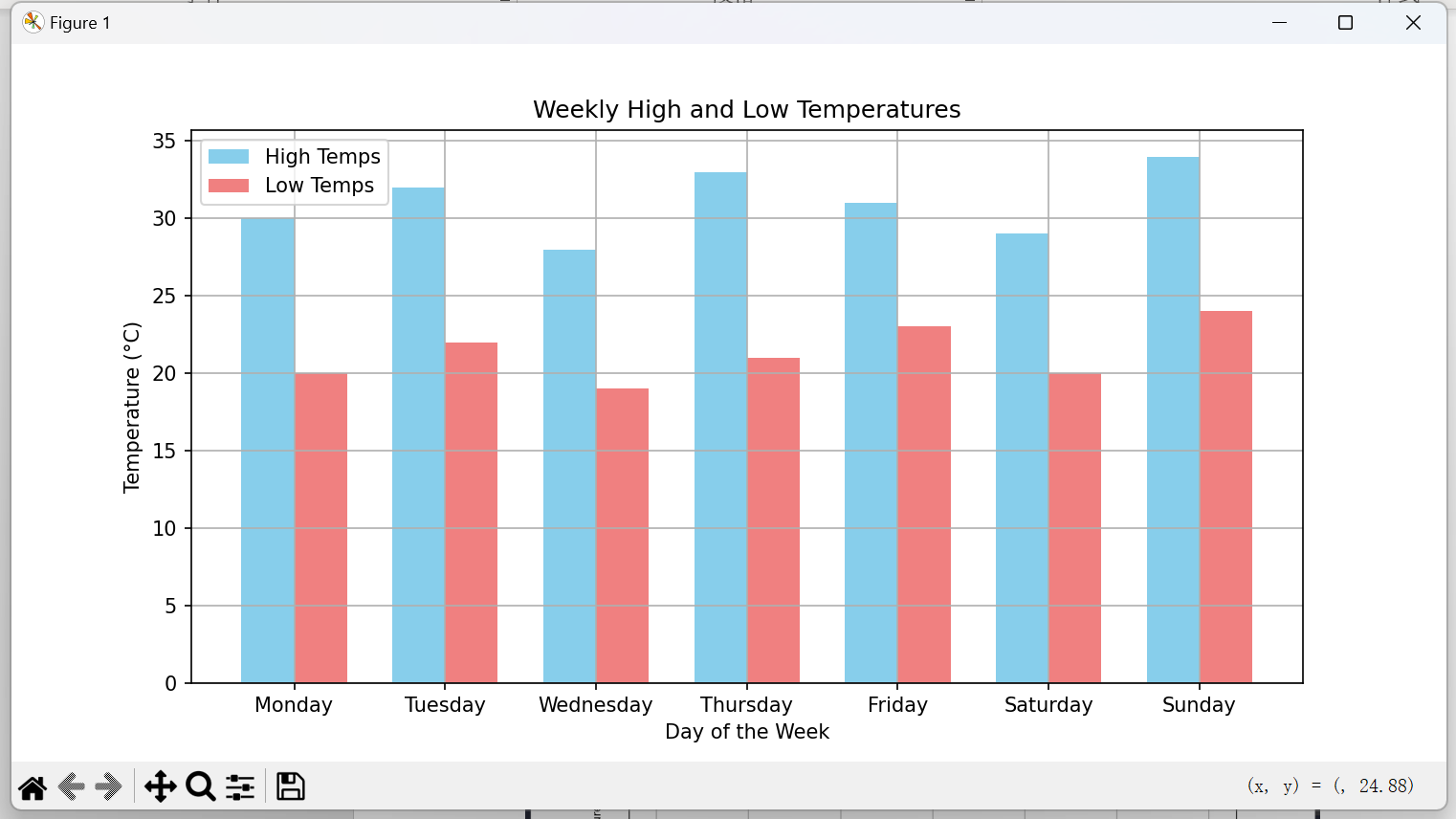
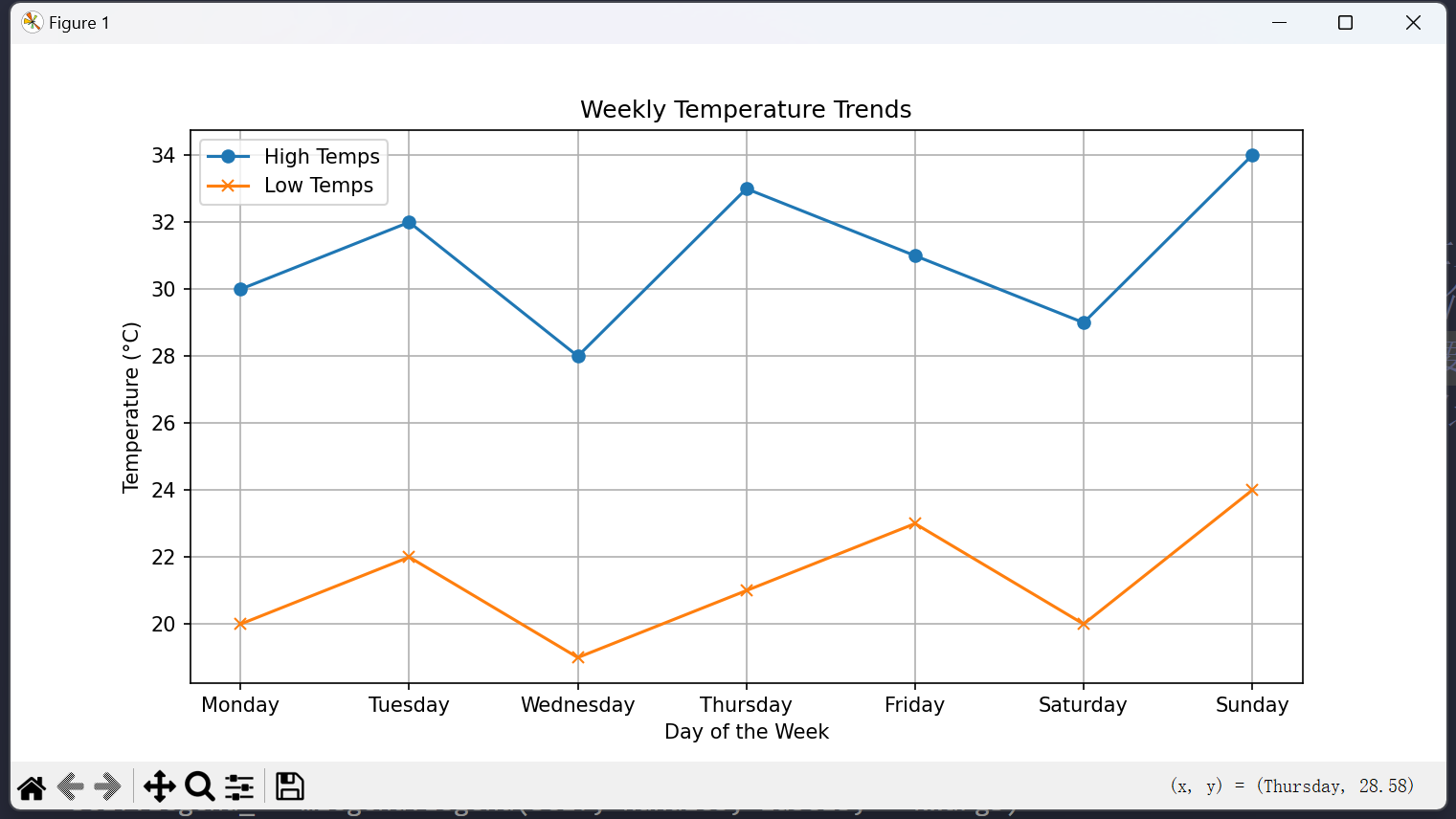
不同点：大数据管理技术处理的数据体量大、结构类型多、来源广、多维度、关联强、及时性、积累久、价值密度低、最终价值大，处理速度快，为用户提供了更加高效、方便、友好的人数交互页面。

1. 批量计算模式、流式计算模式、交互式计算模式和图计算模式
2. 数据分析处理来自对某一兴趣现象的观察、测量或者实验的信息

统计分析、数据挖掘、机器学习和可视化分析

1. 我们利用视觉获取的信息量，远远比别的感官要多得多；数据可视化能够在小空间中展示大规模数据；数据可视化能够帮助我们对数据有更加全面的认识；受人类大脑记忆能力的限制。

践习题

1. 

import matplotlib.pyplot as plt

import numpy as np

# 假设的气温数据

days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

high\_temps = [30, 32, 28, 33, 31, 29, 34]  # 最高气温

low\_temps = [20, 22, 19, 21, 23, 20, 24]    # 最低气温

# 绘制折线图

plt.figure(figsize=(10, 5))

plt.plot(days, high\_temps, label='High Temps', marker='o')

plt.plot(days, low\_temps, label='Low Temps', marker='x')

plt.title('Weekly Temperature Trends')

plt.xlabel('Day of the Week')

plt.ylabel('Temperature (°C)')

plt.xticks(days)

plt.legend()

plt.grid(True)

plt.show()

# 绘制柱状图

plt.figure(figsize=(10, 5))

bar\_width = 0.35

index = np.arange(len(days))

plt.bar(index, high\_temps, bar\_width, label='High Temps', color='skyblue')

plt.bar(index + bar\_width, low\_temps, bar\_width, label='Low Temps', color='lightcoral')

plt.title('Weekly High and Low Temperatures')

plt.xlabel('Day of the Week')

plt.ylabel('Temperature (°C)')

plt.xticks(index + bar\_width / 2, days)

plt.legend()

plt.grid(True)

plt.show()

# 绘制饼状图

temp\_ranges = [(20, 30), (30, 35)]

range\_counts = [sum(1 for temp in high\_temps if low\_bound <= temp < high\_bound) for low\_bound, high\_bound in temp\_ranges]

range\_labels = ['20-30°C', '30-35°C']

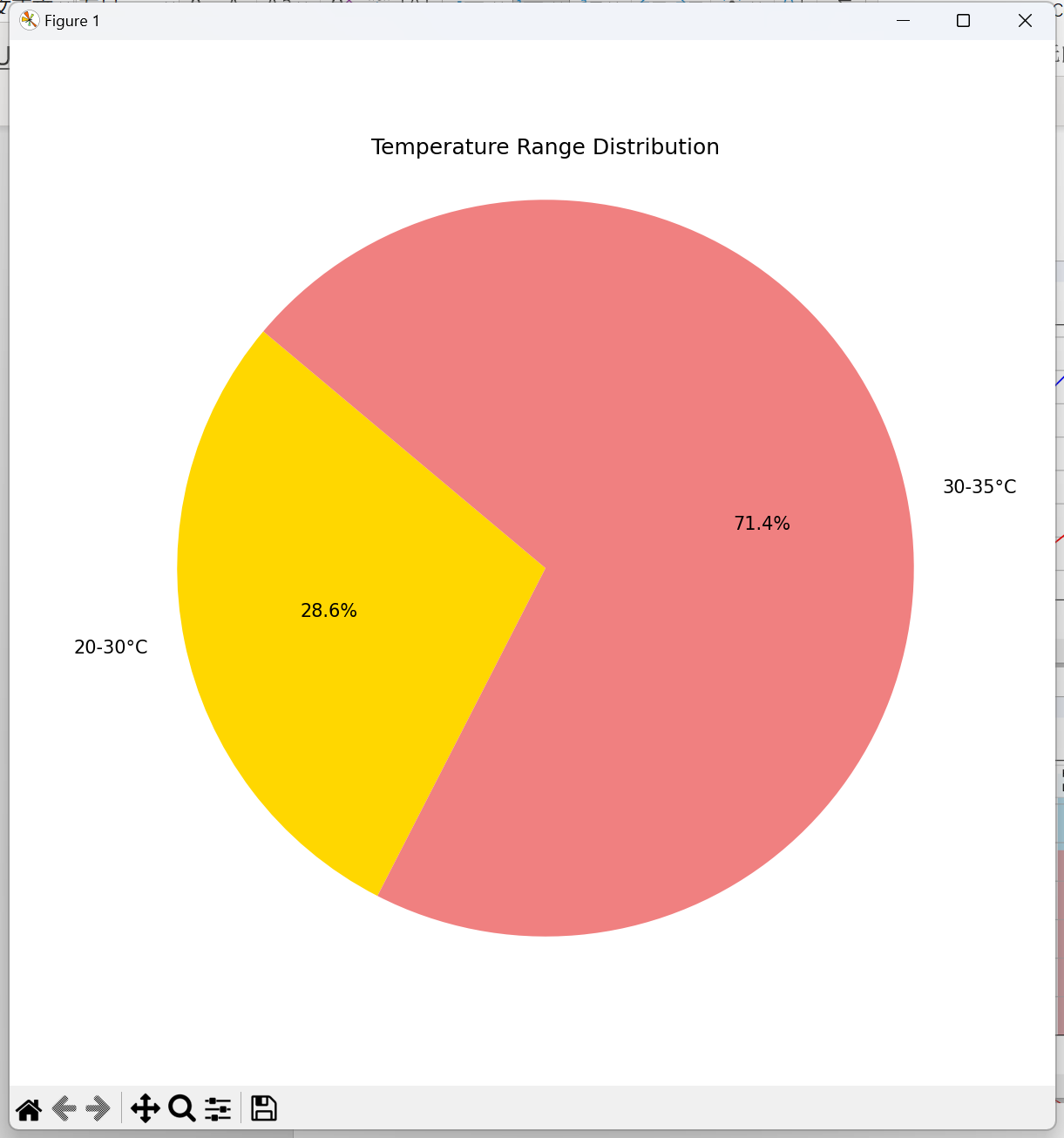
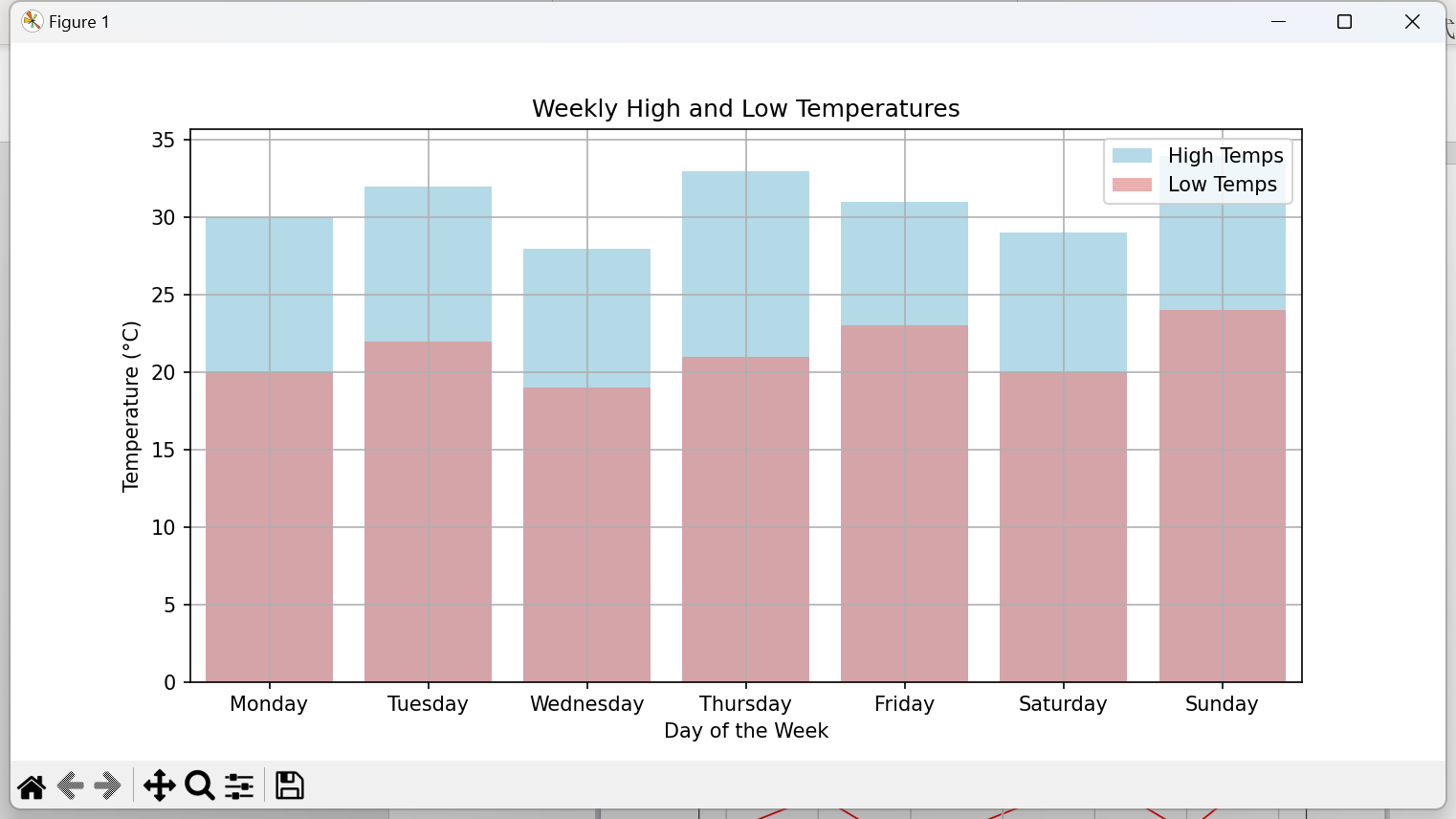
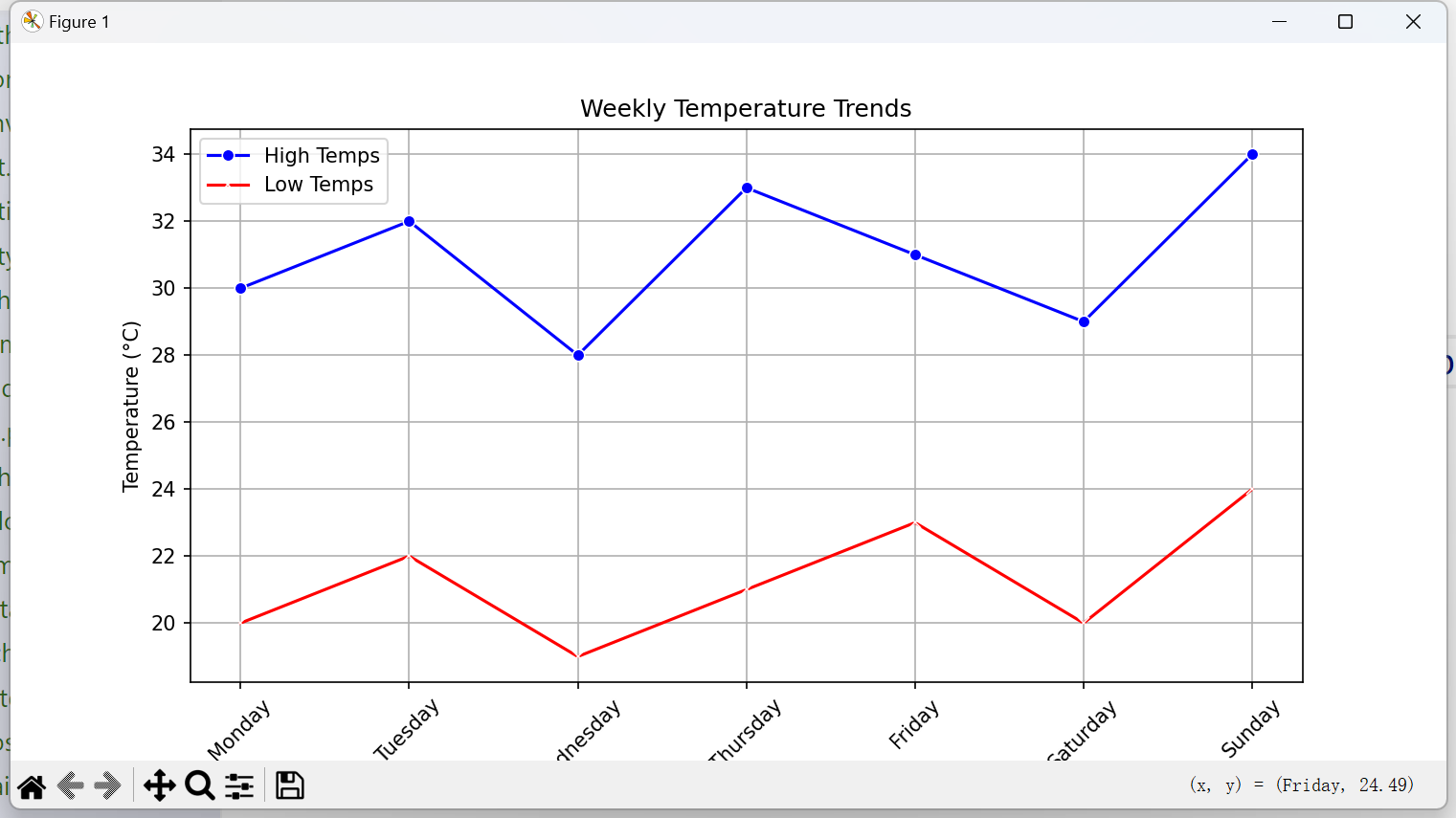
plt.figure(figsize=(8, 8))

plt.pie(range\_counts, labels=range\_labels, autopct='%1.1f%%', startangle=140, colors=['gold', 'lightcoral'])

plt.title('Temperature Range Distribution')

plt.axis('equal')

plt.show()

1. 

import matplotlib.pyplot as plt

import numpy as np

import seaborn as sns

import pandas as pd

# 假设的气温数据

days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

high\_temps = [30, 32, 28, 33, 31, 29, 34]  # 最高气温

low\_temps = [20, 22, 19, 21, 23, 20, 24]    # 最低气温

# 创建DataFrame

data = pd.DataFrame({

    'Day': days,

    'High Temps': high\_temps,

    'Low Temps': low\_temps

})

# 绘制折线图

plt.figure(figsize=(10, 5))

sns.lineplot(x='Day', y='High Temps', data=data, marker='o', label='High Temps', color='blue')

sns.lineplot(x='Day', y='Low Temps', data=data, marker='x', label='Low Temps', color='red')

plt.title('Weekly Temperature Trends')

plt.xlabel('Day of the Week')

plt.ylabel('Temperature (°C)')

plt.xticks(rotation=45)

plt.legend()

plt.grid(True)

plt.show()

# 绘制柱状图

plt.figure(figsize=(10, 5))

bar\_width = 0.35

index = np.arange(len(days))

sns.barplot(x='Day', y='High Temps', data=data, color='skyblue', label='High Temps', alpha=0.7)

sns.barplot(x='Day', y='Low Temps', data=data, color='lightcoral', label='Low Temps', alpha=0.7)

plt.title('Weekly High and Low Temperatures')

plt.xlabel('Day of the Week')

plt.ylabel('Temperature (°C)')

plt.legend()

plt.grid(True)

plt.show()

# 绘制饼状图

temp\_ranges = [(20, 30), (30, 35)]

range\_counts = [sum(1 for temp in high\_temps if low\_bound <= temp < high\_bound) for low\_bound, high\_bound in temp\_ranges]

range\_labels = ['20-30°C', '30-35°C']

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

plt.pie(range\_counts, labels=range\_labels, autopct='%1.1f%%', startangle=140, colors=['gold', 'lightcoral'])

plt.title('Temperature Range Distribution')

plt.axis('equal')  # 使饼状图为圆形

plt.show()