**沈思妤-10235501458-数据科学导论第四次作业**

**复习题**

1. **数据的全生命周期管理包括哪些阶段？**

答：通常包括以下几个阶段：

数据采集：从真实世界对象中获取原始数据的过程。

数据存储：将采集到的数据保存在数据库或其他存储系统中。

数据管理：涉及数据的组织、索引、安全性和维护。

数据计算：执行数据分析和处理，如统计分析、机器学习等。

数据分析：使用统计和逻辑方法来发现数据中的模式和趋势。

数据展示：将分析结果以图形或视觉格式呈现出来，以便更好地理解。

1. **数据采集的概念是什么？都有哪些方法？**

答：数据采集是指从真实世界对象中获得原始数据的过程。数据采集方法的选择不但要依赖于数据源的物理性质，还要考虑数据分析的目标。最常用的三种数据采集方法是：

传感器：用于测量物理环境变量并将其转化为可读的数字信号以待处理。

日志文件：由数据源系统产生，以特殊的文件格式记录系统的活动。

Web爬虫：为搜索引擎下载并存储网页的程序。

1. **什么是数据管理？比较传统的数据管理和大数据管理技术有什么异同？**

答：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。其目的在于充分有效地发挥数据的作用。

传统的数据管理技术通常指的是关系型数据库管理系统，它们在结构化数据和事务处理方面表现出色。而大数据管理技术则包括了非结构化和半结构化数据的处理，需要处理的数据量更大，类型更多样，速度更快。异同点包括：

传统数据管理通常关注结构化数据，大数据管理则需要处理结构化、半结构化和非结构化数据。

传统数据管理在单一服务器或少量服务器上运行，而大数据管理则需要分布式处理，跨越多个服务器和地点。

大数据管理技术通常需要具备高扩展性和容错性，而传统数据管理则可能不需要。

1. **大数据的计算模式可以分为哪几类？**

答：大数据的计算模式可以分为：

批量计算模式：适用于静态数据的离线计算和处理，关注整个计算的吞吐量。

流式计算模式：关注数据处理的实时性，能够快速为决策提供支持。

交互式计算模式：系统与操作人员以人机对话的方式一问一答。

图计算模式：处理图数据，包括图数据的存储、查询、分类、聚类等。

1. **什么是数据分析？有哪些数据分析的方法或者模型？**

答：数据分析是处理和分析数据以发现代表趋势或模式的有用信息的过程。数据分析方法或模型包括：

描述性分析：描述发生了什么，例如使用回归技术发现趋势。

预测性分析：预测未来会发生什么，使用统计模型进行预测。

规范性分析：确定为什么会发生，使用机器学习算法找出数据背后的规律。

聚类分析：将数据点分组，组内成员相似度较高。

分类分析：将数据分配到预定义的类别中。

关联规则学习：发现变量之间的有趣关系。

1. **数据可视化的原因有哪些？**

答：原因包括：

人类通过视觉获取信息比较容易，可视化可以帮助我们更好地传递信息。

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，可视化可以帮助我们更好地记忆和理解数据。

可视化技术可以变抽象为具象，便于理解的同时加深印象。

**践习题**

1. **熟悉可视化包Matplotlib，绘制任一数据集的三种常见图形。**

答：

import matplotlib.pyplot as plt

import numpy as np

# 数据

x = np.arange(1, 6)

y = np.array([10, 20, 15, 25, 30])

# 1. 绘制柱状图

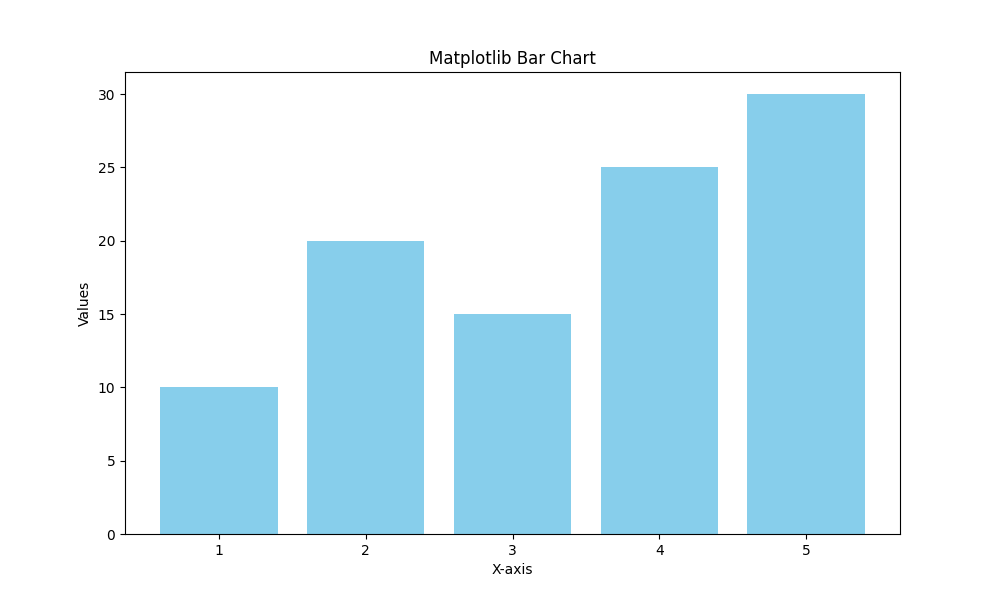
plt.figure(figsize=(10, 6))

plt.bar(x, y, color='skyblue')

plt.title('Matplotlib Bar Chart')

plt.xlabel('X-axis')

plt.ylabel('Values')

plt.show()

# 2. 绘制折线图

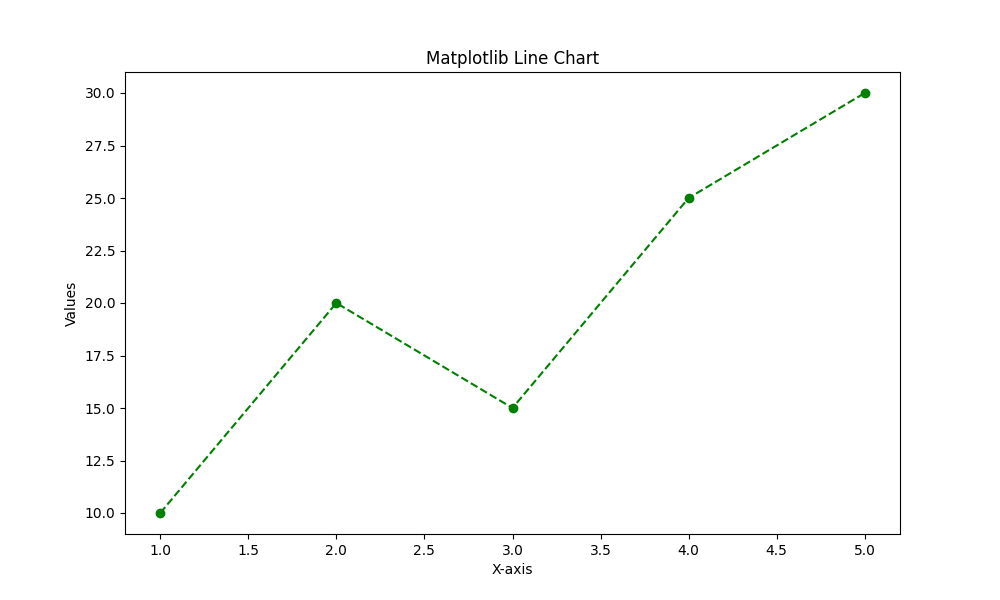
plt.figure(figsize=(10, 6))

plt.plot(x, y, color='green', marker='o', linestyle='--')

plt.title('Matplotlib Line Chart')

plt.xlabel('X-axis')

plt.ylabel('Values')

plt.show()  


# 3. 绘制散点图

x\_scatter = np.random.rand(50)

y\_scatter = np.random.rand(50)

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

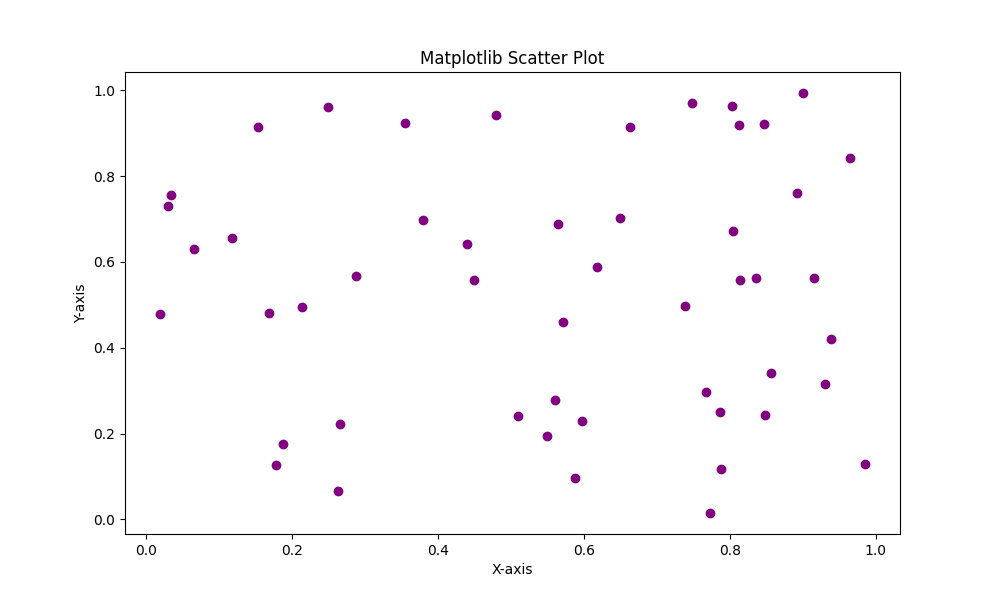
plt.scatter(x\_scatter, y\_scatter, color='purple')

plt.title('Matplotlib Scatter Plot')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.show()



1. **熟悉可视化包Seaborne，绘制任一数据集的三种常见图形。**

答：import seaborn as sns

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# 数据

x = np.arange(1, 6)

y = np.array([10, 20, 15, 25, 30])

# 创建数据集用于 Seaborn

df = pd.DataFrame({'X': x, 'Y': y})

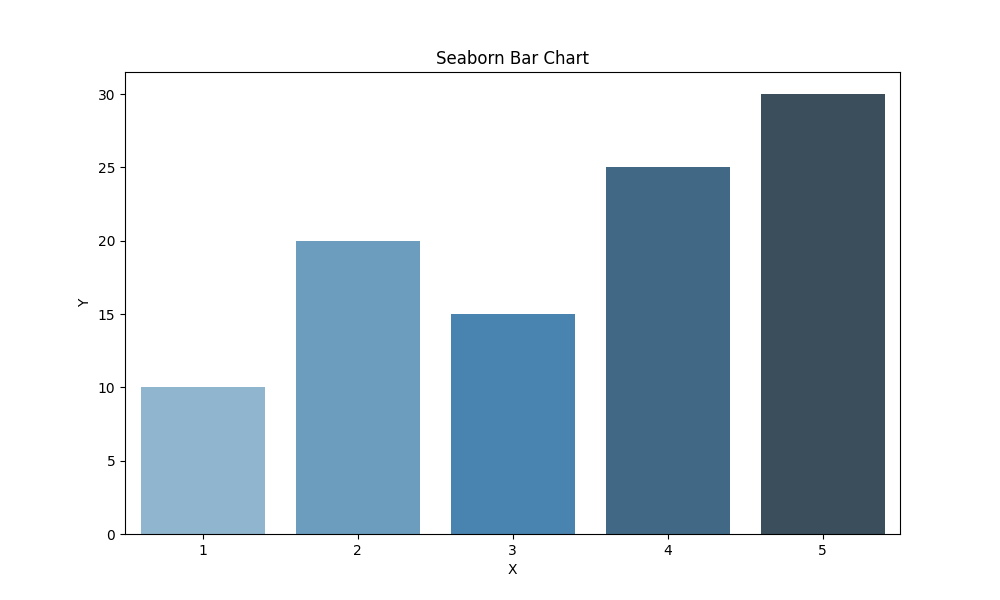
# 1. 绘制柱状图

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

sns.barplot(x='X', y='Y', data=df, palette='Blues\_d')

plt.title('Seaborn Bar Chart')

plt.show()



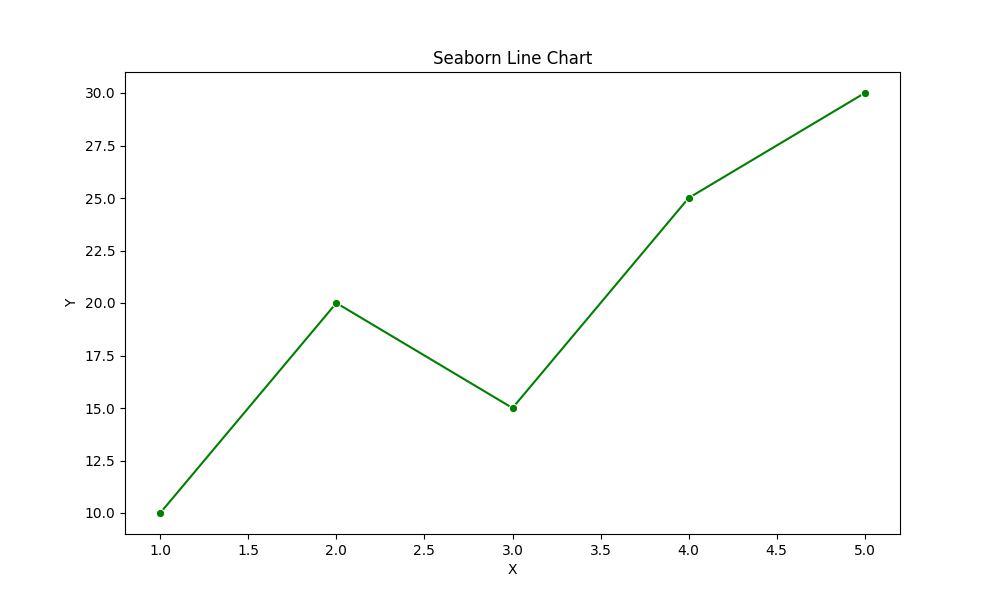
# 2. 绘制折线图

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

sns.lineplot(x='X', y='Y', data=df, marker='o', color='green')

plt.title('Seaborn Line Chart')

plt.show()



# 3. 绘制散点图

x\_scatter = np.random.rand(50)

y\_scatter = np.random.rand(50)

df\_scatter = pd.DataFrame({'X': x\_scatter, 'Y': y\_scatter})

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

sns.scatterplot(x='X', y='Y', data=df\_scatter, color='purple')

plt.title('Seaborn Scatter Plot')

plt.show()

