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?

数据采集、数据存储、数据管理、数据计算、数据分析、数据展示。

2 数据采集的概念是什么？都有哪些方法?

数据采集是指从真实世界对象中获得原始数据的过程。
目前，最常用的三种数据采集方法是；传感器、日志文件和 Web 爬虫。（书上 P173 ）

3 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?

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。（书上P181）

	传统的数据管理	**大数据管理技术**
数据处理能力	数据量通常较小，适合关系型数据库（如MySQL、Oracle）进行操作。	能够处理大量和多样化的数据，包括结构化、半结构化和非结构化数据。 采用分布式计算框架以支持实时和批量处理，能够处理PB级别的数据。
数据存储方式	数据一致性和完整性要求较高，适合事务性应用。	更注重数据的可用性和可扩展性，允许更高的灵活性和容错性。
数据分析和应用	应用场景较为固定，多用于业务报告和决策支持。	采用机器学习、数据挖掘等复杂分析技术，能够发现更深层次的模式和趋势。 应用场景多样，涵盖实时数据流分析、社交媒体分析、物联网数据处理等。

4 大数据的计算方式可以分为哪几类?

数据的计算模式大致分为批量计算模式、流式计算模式、交互式计算模式和图计算模式四类。（P183）

5 什么是数据分析？有哪些数据分析的方法或者模型？

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。数据分析目的是从和主题相关的数据中提取尽可能多的信息。主要目标包括:①推测或解释数据并确定如何使用数据;②检查数据是否合法;③给决策提供合理建议;④诊断或推断错误原因;⑤预测未来将要发生的事情。（书上P185）
根据数据分析深度将数据分析分为三个层次:描述性(descriptive)分析,预测性分析和规则性(prescriptive)分析。
数据的分析技术主要依靠四个方面：统计分析、数据挖掘、机器学习和可视化分析。

6 数据可视化的原因有哪些？

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

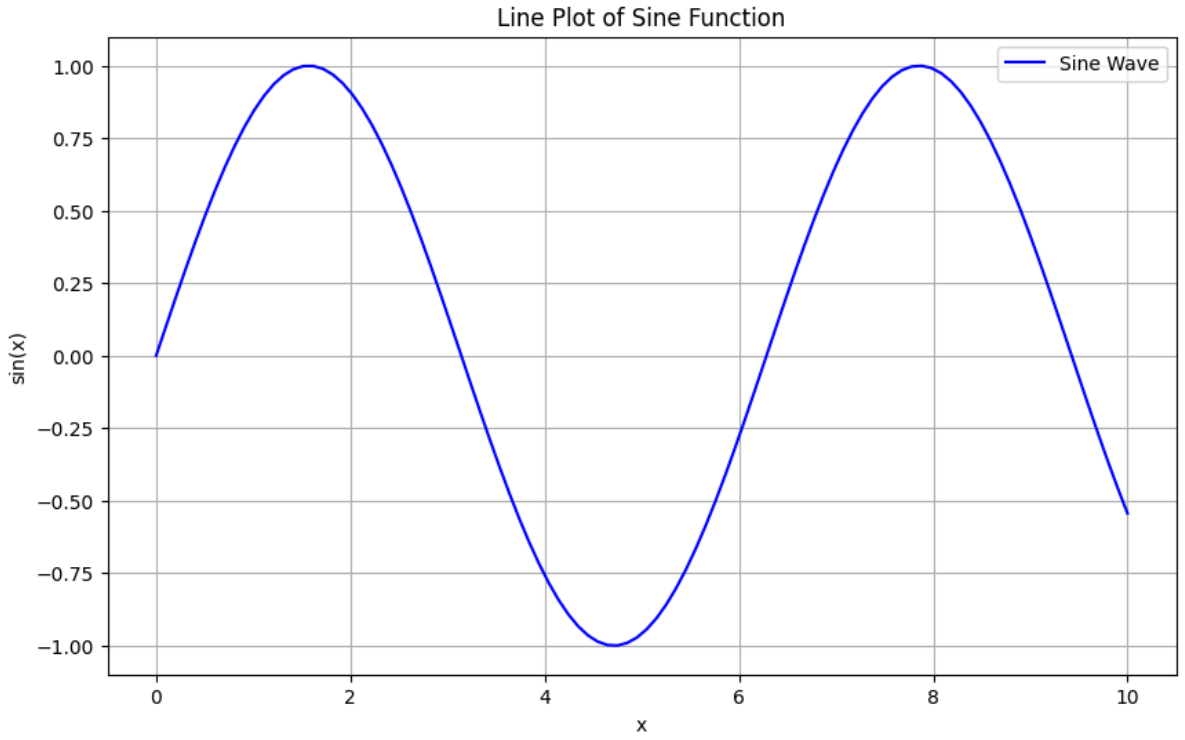
践习题

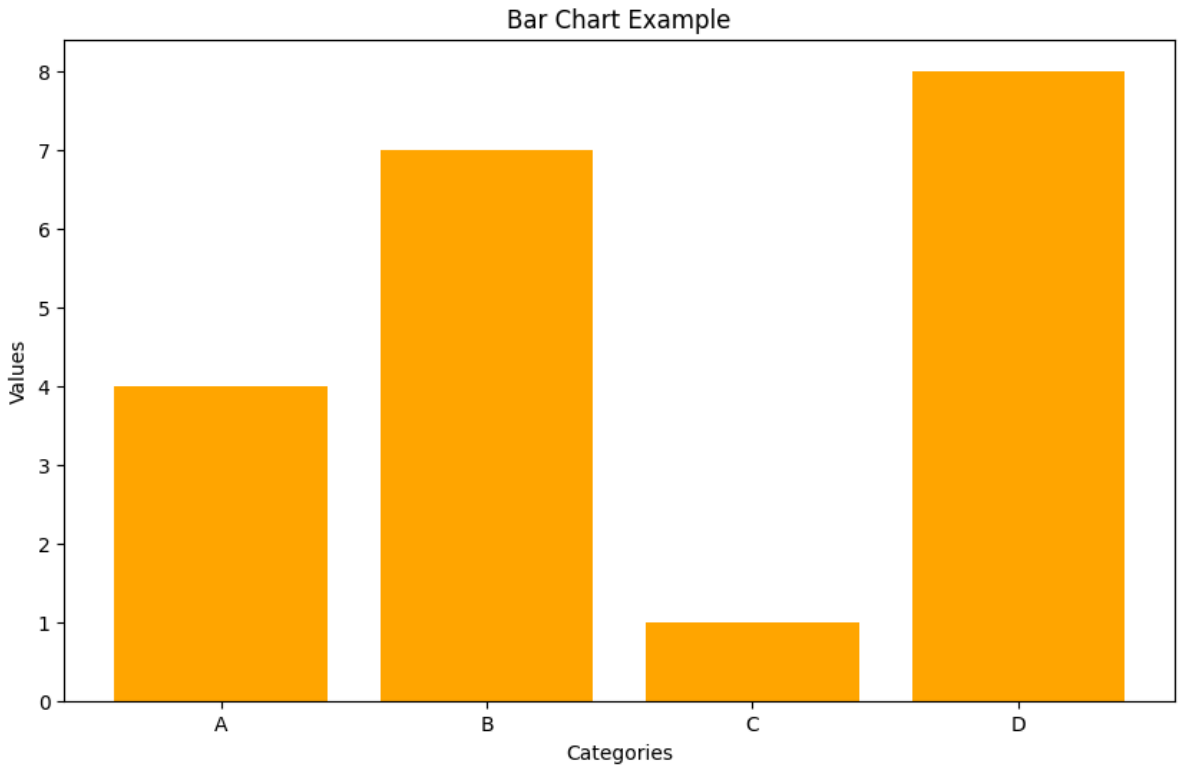
6 Matplotlib绘图

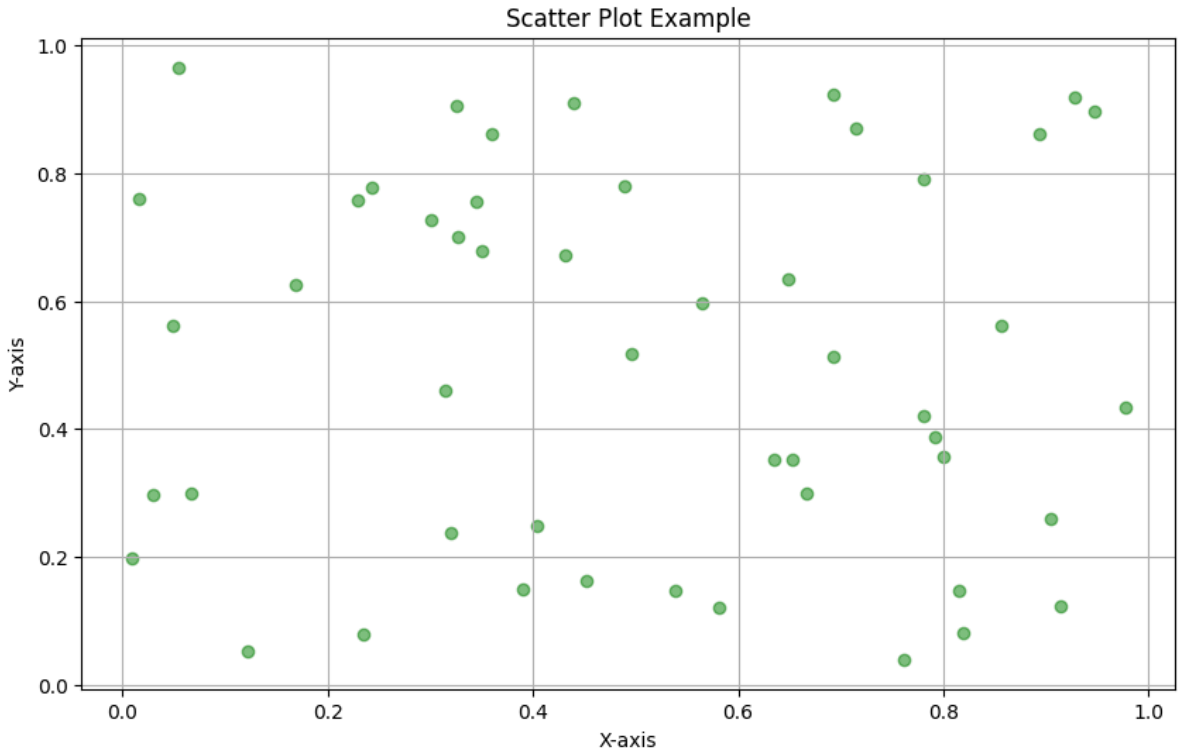
为了图片效果，我一张图使用一个数据集

```
import matplotlib.pyplot as plt
import numpy as np

# 创建示例数据
x = np.linspace(0, 10, 100)  # 生成从0到10的100个点
y = np.sin(x)                 # y 为 x 的正弦值

# 1. 绘制折线图
plt.figure(figsize=(10, 6))
plt.plot(x, y, label='Sine Wave', color='b')
plt.title('Line Plot of Sine Function')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.legend()
plt.grid()
plt.show()

# 2. 绘制柱状图
categories = ['A', 'B', 'C', 'D']
values = [4, 7, 1, 8]

plt.figure(figsize=(10, 6))
plt.bar(categories, values, color='orange')
plt.title('Bar Chart Example')
plt.xlabel('Categories')
plt.ylabel('Values')
```

```
plt.show()

# 3. 绘制散点图
x_scatter = np.random.rand(50)  # 生成50个随机点
y_scatter = np.random.rand(50)

plt.figure(figsize=(10, 6))
plt.scatter(x_scatter, y_scatter, color='green', alpha=0.5)
plt.title('Scatter Plot Example')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.grid()
plt.show()
```





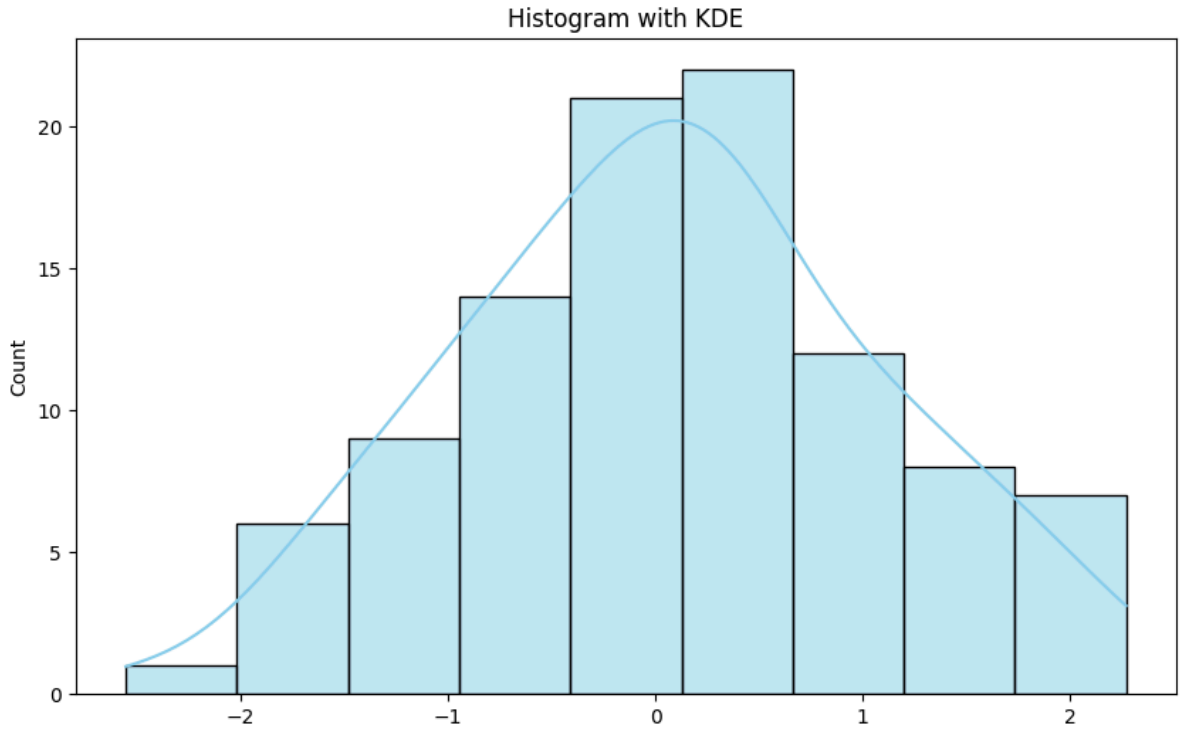


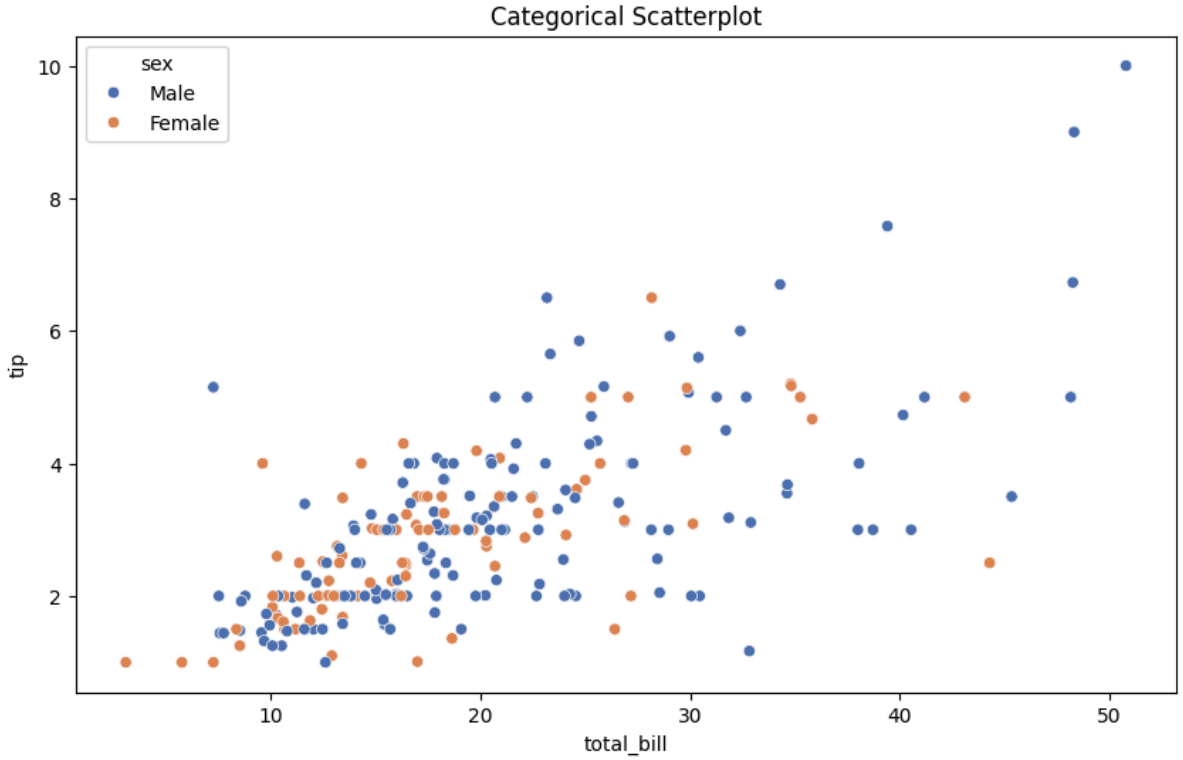
7 Seaborn绘图

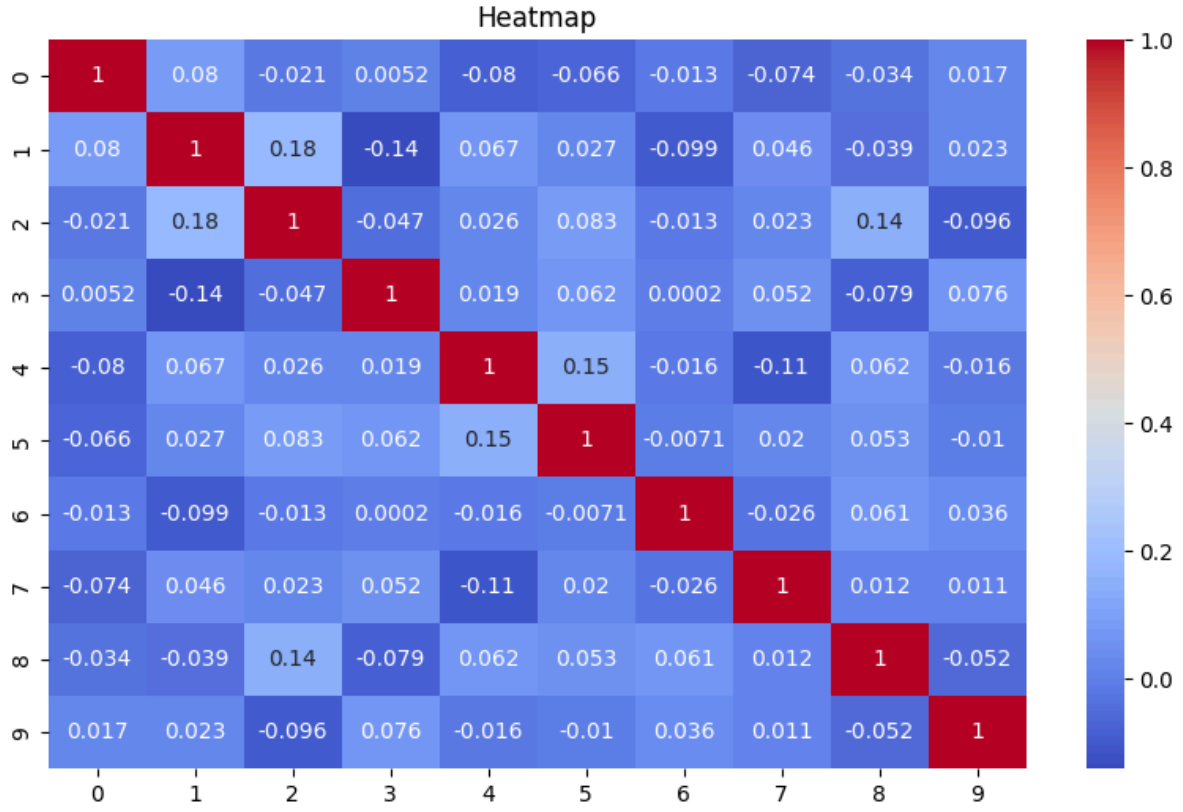
为了图片效果，我一张图使用一个数据集

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```python
# 创建一些示例数据
np.random.seed(0)
data = np.random.randn(100)

# 1. 绘制直方图 (Histogram)
plt.figure(figsize=(10, 6))
sns.histplot(data, kde=True, color='skyblue')
plt.title('Histogram with KDE')
plt.show()

# 2. 绘制分类散点图 (Categorical Scatterplot)
# 使用内置的 tips 数据集
tips = sns.load_dataset("tips")
plt.figure(figsize=(10, 6))
sns.scatterplot(x='total_bill', y='tip', hue='sex', data=tips, palette='deep')
plt.title('Categorical Scatterplot')
plt.show()

# 3. 绘制热力图 (Heatmap)
# 创建随机量的相关系数矩阵
corr_matrix = np.corrcoef(np.random.randn(10, 200))
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap')
plt.show()
```




Categorical Scatterplot
sex
Male
Female
tip
total_bill
10
8
6
4
2
10
20
30
40
50


Heatmap
0 1 2 3 4 5 6 7 8 9
1 0.08 -0.021 0.0052 -0.08 -0.066 -0.013 -0.074 -0.034 0.017
0.08 1 0.18 -0.14 0.067 0.027 -0.099 0.046 -0.039 0.023
-0.021 0.18 1 -0.047 0.026 0.083 -0.013 0.023 0.14 -0.096
0.0052 -0.14 -0.047 1 0.019 0.062 0.0002 0.052 -0.079 0.076
-0.08 0.067 0.026 0.019 1 0.15 -0.016 -0.11 0.062 -0.016
-0.066 0.027 0.083 0.062 0.15 1 -0.0071 0.02 0.053 -0.01
-0.013 -0.099 -0.013 0.0002 -0.016 -0.0071 1 -0.026 0.061 0.036
-0.074 0.046 0.023 0.052 -0.11 0.02 -0.026 1 0.012 0.011
-0.034 -0.039 0.14 -0.079 0.062 0.053 0.061 0.012 1 -0.052
0.017 0.023 -0.096 0.076 -0.016 -0.01 0.036 0.011 -0.052 1
1.0
0.8
0.6
0.4
0.2
0.0