数据科学导论第四次作业

复习题一：

数据采集，数据存储，数据管理，数据计算，数据分析，数据展示。

复习题二：

数据采集是指从真实世界对象中获得原始数据的过程

最常用的三种数据采集方法是：

传感器 日志文件和 web爬虫。

复习题三：

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。

数据规模：

传统数据管理：通常处理的数据量相对较小，适合存储在单个数据库中。

大数据管理：设计用于处理PB甚至EB级别的数据，通常分布在多个服务器和地点。

处理速度：

传统数据管理：往往在事务处理系统中使用，优化的是单个事务的响应时间。

大数据管理：强调的是批量数据处理速度，使用如Hadoop和Spark等技术来处理大量数据集。

数据类型：

传统数据管理：主要处理结构化数据，如关系数据库中的表格数据。

大数据管理：能够处理结构化、半结构化和非结构化数据，如文本、图像、视频等。

查询和分析：

传统数据管理：使用SQL等声明性查询语言，优化查询性能和准确性。

大数据管理：使用更复杂的分析工具和算法，如机器学习和统计分析，以发现数据中的模式和趋势。

可扩展性：

传统数据管理：扩展性有限，通常通过垂直扩展（增加单个服务器的资源）来实现。

大数据管理：通过水平扩展（增加更多的服务器到集群中）来实现，以支持数据量的增长。

容错性：

传统数据管理：通常依赖于高可用性硬件和备份系统来提供容错能力。

大数据管理：设计时就考虑到了故障的可能性，通过数据复制和分布式处理来提高容错性。

成本：

传统数据管理：可能需要昂贵的硬件和专业数据库管理系统。

大数据管理：倾向于使用成本较低的硬件，但可能需要更多的服务器和存储资源。

实时处理：

传统数据管理：通常需要实时或近实时的数据处理。

大数据管理：虽然也可以实现实时处理，但很多大数据处理工作是批量处理的。

技术栈：

传统数据管理：可能使用成熟的数据库系统，如Oracle、SQL Server等。

大数据管理：使用Hadoop生态系统、NoSQL数据库、以及各种大数据处理和分析工具。

数据治理：

传统数据管理：数据治理相对简单，因为数据源和数据类型较为有限。

大数据管理：需要更复杂的数据治理策略，以处理数据的多样性和分布性。

复习题四：

1 数据批量计算模式

2 数据流计算模式

3 数据交互式计算模式

4 数据图计算模式

复习题五：

数据分析是指用适当的统计方法、工具和编程语言对收集来的数据进行处理和分析，以从中提取有价值的信息、发现规律或者支持决策制定的过程。数据分析可以帮助我们理解数据的含义，预测未来的趋势，以及为业务决策提供依据。

方法或类型：统计分析 数据挖掘 机器学习和可视化分析

复习题六：

数据可视化的应用价值其多样性和表现力吸引了许多使用者。无论是动态还是静态的可视化图形，都为我们搭建了新的桥梁。让我们能洞察世界的究竟，发现形形色色的关系，感受每时每刻围绕在我们身边的信息变化，还能让我们理解其他形式下不易发掘的事物。

可视化技术把数据变为图形展示给大众，注重技术的实现及其算法优化，通过开 发可视化工具变抽象为具象，便于理解的同时加深印象。

总结起来数据可视化的原因有下面几点：

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

践行题7：

****线图****

python

import matplotlib.pyplot as plt

*# 假设我们有以下数据*

x = [1, 2, 3, 4, 5]

y = [2, 3, 5, 7, 11]

plt.plot(x, y) *# 创建线图*

plt.title('Line Plot') *# 添加标题*

plt.xlabel('X Axis Label') *# 添加X轴标签*

plt.ylabel('Y Axis Label') *# 添加Y轴标签*

plt.show() *# 显示图表*

****条形图****

python

import matplotlib.pyplot as plt

*# 假设我们有以下数据*

categories = ['Category A', 'Category B', 'Category C']

values = [10, 20, 15]

plt.bar(categories, values) *# 创建条形图*

plt.title('Bar Chart') *# 添加标题*

plt.xlabel('Categories') *# 添加X轴标签*

plt.ylabel('Values') *# 添加Y轴标签*

plt.show() *# 显示图表*

****散点图****：

python

import matplotlib.pyplot as plt

*# 假设我们有以下数据*

x = [1, 2, 3, 4, 5]

y = [2, 3, 5, 7, 11]

plt.scatter(x, y) *# 创建散点图*

plt.title('Scatter Plot') *# 添加标题*

plt.xlabel('X Axis Label') *# 添加X轴标签*

plt.ylabel('Y Axis Label') *# 添加Y轴标签*

plt.show() *# 显示图表*

践行题8：

****散点图****

import seaborn as snsimport matplotlib.pyplot as plt*# 假设我们有以下数据*

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

sns.scatterplot(x=x, y=y)

plt.show()

****折线图****

import seaborn as snsimport matplotlib.pyplot as plt*# 假设我们有以下数据*

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

sns.lineplot(x=x, y=y)

plt.show()

****柱状图****

import seaborn as snsimport matplotlib.pyplot as plt*# 假设我们有以下数据*

x = ['A', 'B', 'C', 'D']

y = [10, 15, 7, 12]

sns.barplot(x=x, y=y)

plt.show()

"

"