大数据与金融治理

2024 Fall

第二章

数据处理与特征工程

Information Sets, Features Scaling, & Dimension Reduction

2

章节目标

01

数据的管理与探索

- 数据管理分析库Pandas
- 数据可视化程序库

02

数据清洗、特征选择与降维技术

- 缺失数据处理、特征提取
- 梯度下降

2.1 章节学习目标

- 熟悉不同数据存储设备的功能及使用场景
- 理解常用数据存储的文件格式及优缺点
- 掌握运用Pandas，seaborn以及matplotlib程序库的能力，并可以使用这些程序对数据进行初步探索以及整理

金融数据管理的挑战与机遇

- 金融市场产生大量的数据给从业人员以及监管机构提供了宝贵的材料。例如，股市中的许多指标可以帮助我们预测公司未来的基本面；银行接触到大量的转账信息可以为金融机构的决策作出非常重要的参考。
- 首先是如何存储以及管理这些数据。我们可以先通过简单的分析理解数据的特性，了解数据中的基本信息。这些信息既能帮助我们掌握数据的特点，也能为我们向数据提出的问题提供一些初步的参考。
- 在本章节中，我们将介绍一些基本的数据存储与读写的概念，并且介绍之后会经常使用的程序库。

1、数据的存储与使用

2、用于数据的管理以及初步分析的Pandas程序包

3、用于数据可视化的matplotlib以及seaborn程序包

数据的存储与使用

内存与硬盘

- 我们的讨论，主要围绕内存与硬盘两个设备，二者区别如下：

1、内存往往速度更快，但是其容量较小（常见的内存在8到64GB之间）。硬盘的容量较大（其容量经常在250GB到几TB之间），但是往往存储与读写的速度偏慢。

2、二者更重要的区别在于：内存属于易失性存储设备，而硬盘属于非易失性存储设备（当关机或重启电脑之后，内存中的数据会被清空，而硬盘中的数据则一般不会受到影响）。

- 因此，我们在存储以及使用数据的过程中，需要这两种设备的协同运作：
  - 通常，我们会将我们得到的数据存储到硬盘之中，便于我们长期保存。
  - 但是，当我们需要进行数据分析的时候，我们则需要将数据读入内存之中（内存存储与读写速度较快，将程序执行中即将要使用的数据暂时存入内存，可以极大加速程序运行的速度）。

数据存储的格式

- 在实际操作中，可以将数据存入各种格式的数据之中,如EXCEL中的CSV和XLSX格式,数据提供商需要程序接触的JOSN文件，大数据存储的Parquet格式。
- **表2.1将以上几种格式进行了比较：**

1、较小的数据常用CSV和XLSX格式保存，阅读这些文件非常简单直接，但最大的问题是读写速度较慢以及其占用的存储空间较大。

2、当数据量巨大时，我们应将数据以Parquet格式保存，其优势在于易于存储与读写。

3、JSON在网络应用中有着非常广泛的应用，由于其格式清晰，我们非常容易能使用程序来抓去这些数据中的关键信息。但是因为该文件以文本形式存储，因此读写速度往往也不是最优。

文件类型	CSV	Parquet	XLSX	JSON
数据结构	平面，表格	列式存储	单元格的电子表格	层次化，键值对
文件大小	一般较大	压缩后较小	由于格式化而较大	变化较大，通常较大
读/写速度	读取快，大文件写入慢	读取操作非常快，特别是在大数据集上	由于格式复杂而较慢	小文件快，大型或复杂结构慢
人类可读性	高（基于文本）	低（二进制格式）	中等（需要软件读取）	高（基于文本）
典型用例	简单的表格数据，快速数据交换	大数据分析，高效存储和快速查询	商务报告，财务分析，带有丰富格式的数据	网络API，配置，具有嵌套结构的数据

表2.1：不同数据格式的比较

Pandas与Matplotlib简介

Pandas简介

- Pandas是一个开源的Python数据分析库，广泛用于快速分析数据，以及数据清洗和准备等任务。它提供了高效的DataFrame对象，这是一种表格型的数据结构，具有对数据操作的各种功能，包括数据过滤、转换、聚合等。Pandas非常适合处理结构化数据，如时间序列、表格数据和任何以行和列组织的数据集。
- 此库的目的是为了提供一个高效、易用的数据结构和数据分析工具。Pandas利用NumPy来提高其数据处理的速度和效率。其还支持读取和写入多种文件格式，如CSV、Excel、JSON、Parquet。
- Pandas是数据科学和数据分析领域中不可或缺的工具之一，它的设计使得Python成为强大且直观的数据分析环境，极大地提高了数据处理的效率和质量。

Pandas的局限性

1、Pandas一般来说需要将一个文件中的所有数据都读入内存才能对其进行高效操作。因此，如果有的数据集过大，那么Pandas将无法高效处理这些数据。

2、另外，在数据处理过程中常用的SQL语言在Pandas没有相应的支持。如果需要在Pandas中使用SQL，我们必须安装额外的程序包。

数据可视化程序库简介

- 介绍两个常用的Python数据可视化程序包：

Matplotlib简介

- Matplotlib是一个用于Python编程语言和其数值数学扩展库NumPy的绘图库。
- Matplotlib广泛应用于数据可视化领域，无论是绘制简单的折线图、条形图、散点图，还是更复杂的图表类型，如误差线图、直方图、3D图形等，Matplotlib都能胜任。
- 它的灵活性和易用性使得它成为Python数据可视化的基石之一，广泛用于学术研究、工业应用以及许多数据分析和数据科学项目。

Seaborn简介

- Seaborn是基于matplotlib的Python数据可视化库，提供了一个高级接口来绘制信息丰富的统计图形。
- Seaborn的目标是使可视化成为探索和理解数据的核心部分。它具有内置主题、支持多种统计图表类型、与Pandas数据框架良好集成、自动统计估计和错误条显示、分面功能以及可高度自定义，非常适合进行统计分析和数据呈现。

程序：数据的管理以及初步探索

使用Pandas读取，整理数据

- 我们用虚拟的投资者收入，性别，以及股票投资组合金额的数据来解释如何使用Pandas模块。
- 导入程序库：
  - import pandas as pd
- 使用pandas程序库读入csv文件中的数据：
  - df = pd.read_csv("income stock.csv")
- 读入数据后，我们需要快速检查数据读入的结果。例如，数据中有哪些信息，以及这些数据的格式。我们可以用 head() 这个方法来查看前几行数据。
  - print(df.head())
  - 结果如下图：

	id	income	gender	stock
0	i_1	252441	F	134719.488149
1	i_2	315818	F	158440.763299
2	i_3	472255	M	244051.884147
3	i_4	286712	M	146488.127383
4	i_5	28696	M	16852.623977

- 我们用df.head(5)来查看df这个数据帧中的前五行。其中，第一列数字是这个数据帧的索引。这个索引便于我们找到数据的位置。该索引是从0开始计数的。这个数据中的每一个行对应的是一个用户。每一列对应的是我们存储的关于该用户的信息。

- 用df.tail(5) 来查看数据帧的最后5行信息。

数据帧的基本信息

接下来，我们来检验一下数据量具体多少。并且数据帧中的变量分别是以什么格式存储的。

- print(df.shape)
- print(df.dtypes)
- 结果如右图：

```
(10, 4)
id          object
income       int64
gender      object
stock      float64
dtype: object
```

- 用 df.shape来查看数据帧的大小。(10,4) 说明这个数据帧中有10行，4列，即有10个样本，每个样本有四个变量。
- df.dtypes说明了数据分别是什么格式。其中income是用int64（64位整数）存储。stock是用float64（双精度，64位浮点数）格式存储。其他数据是object格式，这种格式一般是用来存储字串或混合类型的数据。

获取变量信息

如果我们只要使用一个变量，那么我们可以用方括号来查看数据帧中单个变量。

- stock = df['stock']
- print(stock)
- 结果如下图：

```
0    134719.488149
1    158440.763299
2    244051.884147
3    146488.127383
4     16852.623977
5     73551.575225
6     87849.163065
7    187017.380166
8    178244.890406
9    189905.305184
Name: stock, dtype: float64
```

获取行信息

如果我们需要查看一个用户，那么我们可以续用.iloc[索引]来查看一行。

- user_0 = df.iloc[0]
- print(user_0)
- 结果如下图：

```
id                   i_1
income            252441
gender                 F

stock      134719.488149
Name: 0, dtype: object
```

获取子数据集

我们也可以筛选数据。例如，下面我们将性别为男性的用户筛选出来，存入一个新的数据帧。

- df_male = df[df['gender'] == 'M']
- print(df_male.head())
- 结果如下图：

	id	income	gender	stock
2	i_3	472255	M	244051.884147
3	i_4	286712	M	146488.127383
4	i_5	28696	M	16852.623977
9	i_10	360294	M	189905.305184

- 我们也可以叠加选择条件。例如，我们需要选取用户性别为男，并且收入超过100000的用户。注意，因为我们需要使用两个筛选条件，此时我们需要使用逻辑操作 &（and），以表示我们需要两个条件同时达到才会被选入。
- df_male_highinc = df[(df['gender'] == 'M') & (df['income'] > 100000)]
- print(df_male_highinc)
- 结果如右图：

	id	income	gender	stock
2	i_3	472255	M	244051.884147
3	i_4	286712	M	146488.127383
9	i_10	360294	M	189905.305184

生成新变量

我们也可以通过数据帧中的两个或多个变量来生成新的变量。比如，我们可以生成投资组合与收入的一个比例（income/stock）。

- df['income_stock_ratio'] = df['income'] / df['stock']
- print(df.head(5))
- 结果如下图：

	id	income	gender	stock	income_stock_ratio
0	i_1	252441	F	134719.488149	1.873827
1	i_2	315818	F	158440.763299	1.993288
2	i_3	472255	M	244051.884147	1.935060
3	i_4	286712	M	146488.127383	1.957237
4	i_5	28696	M	16852.623977	1.702762

数据的初步探索

我们将用pandas的内置功能，对数据进行初步探索。首先，我们先对变量的平均值，标准方差，最小值、最大值等一系列描述统计信息进行。

- df['income'].describe()
- 结果如下图：

```
count        10.000000
mean     272333.100000
std      130257.490509
min       28696.000000
25%      183164.250000
50%      301265.000000
75%      356726.000000
max      472255.000000
Name: income, dtype: float64
```

- 我们可以分别对男性用户以及女性用户的信息进行总结。在此，我们需要用到groupby这个方法。groupby可以帮助我们将数据放到不同的子集中。同时，agg方法帮我们计算汇总统计信息。
- grouped_stats = df.groupby('gender')[['income', 'stock']].agg(['mean', 'median', 'std'])
- print(grouped_stats)
- 结果如右图：

	income			stock	
	mean	median	std	mean	median
gender					
F	262562.333333	284129.5	94719.208110	136637.210052	146580.125724
M	286989.250000	323503.0	188336.989073	149324.485173	168196.716284

	std
gender	
F	47134.921336
M	96913.917135

数据合并

有时我们需要从不同的数据集中取得用户信息，即我们可能需要合并多个数据。使用pandas有许多合并数据的方法，在这里我们只介绍一种较为简单的方法。

- 接下来，将用户的学习成绩（GPA）与他们的收入，股市信息合并。
- df_gpa = pd.read_csv('gpa.csv')
- df_gpa.head(5)
- 结果如图：

	id	gpa
0	i_1	2.81
1	i_2	2.10
2	i_3	3.65
3	i_4	3.46
4	i_5	2.67

- 合并两个数据集。我们要求两个数据集中的用户id对齐。
- df_merged = pd.merge(df, df_gpa, on='id')
- print(df_merged)
- 结果如下图：

	id	income	gender	stock	income_stock_ratio	gpa
0	i_1	252441	F	134719.488149	1.873827	2.81
1	i_2	315818	F	158440.763299	1.993288	2.10
2	i_3	472255	M	244051.884147	1.935060	3.65
3	i_4	286712	M	146488.127383	1.957237	3.46
4	i_5	28696	M	16852.623977	1.702762	2.67
5	i_6	140602	F	73551.575225	1.911611	3.48
6	i_7	160072	F	87849.163065	1.822123	2.62
7	i_8	360419	F	187017.380166	1.927195	3.04
8	i_9	346022	F	178244.890406	1.941273	3.39
9	i_10	360294	M	189905.305184	1.897230	2.58

最后，我们将数据存入csv文件

- df_merged.to_csv('merged.csv')

数据可视化以及数据的探索

- 数据可视化是对数据进行初步了解的重要方法，业界中将数据进行可视化有助于更清晰的传递数据中所蕴含的信息。

- 首先，我们不仅需要使用matplotlib来控制图形大小以及输出，还会使用seaborn来进行数据可视化。因此，我们需要导入这两个库。
- import seaborn as sns
- import matplotlib.pyplot as plt

导入数据

我们将用seaborn程序包自带的tips数据来对其功能进行说明。本数据中有小费的总量（tip），用餐的价格（total bill），还有用餐人数（size）等信息。

- tips = sns.load_dataset('tips')
- tips.head()
- 结果如图：

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

直方图

- 直方图（Histogram）是一种统计报告图，用于展示一系列数据分布的情况。它通过将数据分组（通常是等宽的连续区间），然后计算每个组中的数据点数量（频率或频数），用柱状图的形式来表示。

- 从直方图中，我们可以获得以下信息：

1、分布的形状：直方图可以展示数据是对称分布的，还是偏斜（向左或向右偏斜）。

2、中心的位置：直方图帮助我们估计数据的平均值或中位数所在的大致位置。

3、变异程度：通过观察柱状图的宽度和分布，我们可以了解数据的波动程度或离散程度。

4、异常值：直方图还可以帮助我们发现数据中的异常值或离群点。

- 我们使用直方图来查看数据分布，并着重分析每单小费中小费（tip）的金额。
- plt.figure(figsize=(8, 6))
- sns.histplot(tips['tip'], kde=True, color='blue')
- plt.title('Histogram of Total Tip Amounts')
- plt.xlabel('Tip')
- plt.ylabel('Frequency')
- plt.show()
- 结果如右图所示：

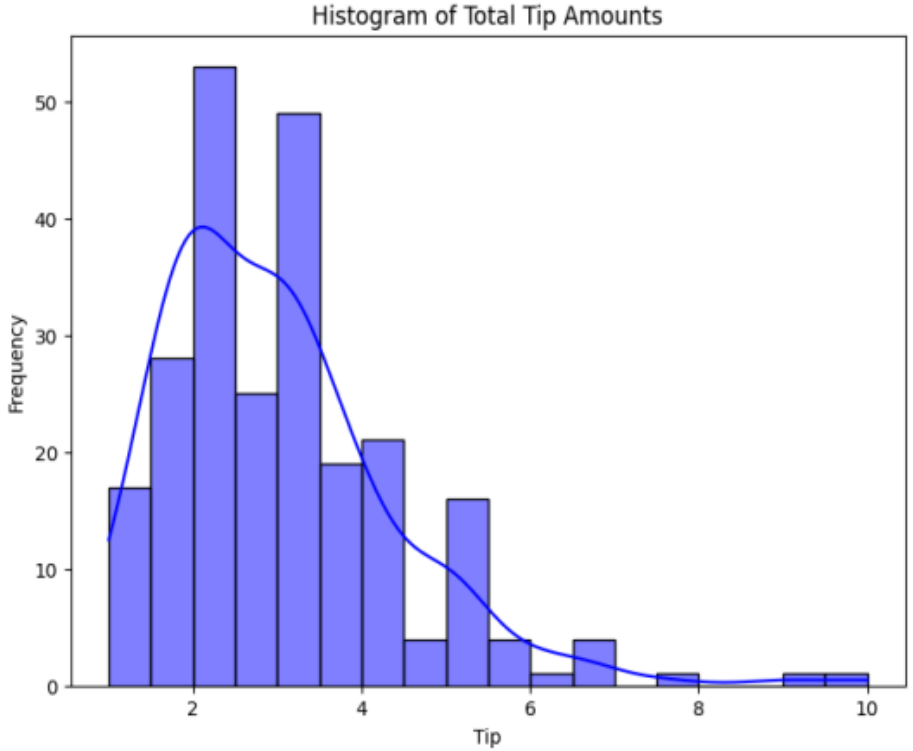


图 1.1: 直方图

箱形图

- 箱形图（Box Plot），也称为盒须图，是另一种用于显示数据分布统计特征的图表。箱形图通过五数概括（最小值、第一四分位数（Q1）、中位数、第三四分位数（Q3）、最大值）来描述数据的分布情况，同时也能展示出数据的异常值。
- 箱形图的构成如下：

1、箱子：箱子的底部和顶部分别表示第一四分位数（Q1）和第三四分位数（Q3），箱子的长度代表了数据的四分位距（IQR），可以用来衡量数据的集中趋势和离散程度。

2、中位线：箱子内部的线表示数据的中位数，反映了数据的中心位置。

3、须：箱子外的两条线（须）延伸至箱子外的最小值和最大值，但不包括异常值。这些线表示数据分布的范围。

4、异常值：用点表示的离群点，通常是那些小于Q_1-1.5IQR或大于Q_3+1.5IQR的值。

- 相较于直方图，箱形图更为简洁。因此可以展示一个变量在不同子集中的分布。
- plt.figure(figsize=(10, 7))
- sns.boxplot(data=tips, x='day', y='tip', palette='rainbow')
- plt.title('Box Plot of Tip by Day of the Week')
- plt.xlabel('Day of the Week')
- plt.ylabel('Tip')
- plt.show()
- 结果如右图，我们发现周四（Thur）与周六（Sat）都有一些异常高的小费。

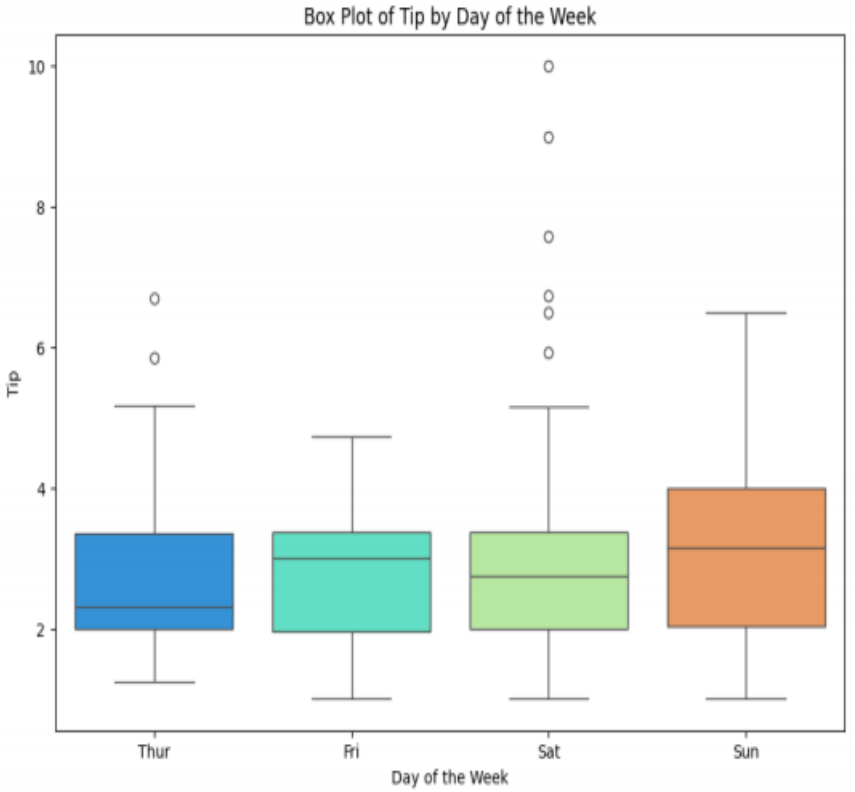


图 1.2: 箱型图

条形图

- 条形图（Bar Plot）是一种在研究中常用的数据可视化类型，它通过条形的长度来表示各类别的数据量或数值的大小，非常适用于展示和比较不同类别或组之间的数量关系。条形图可以是垂直的也可以是水平的，这取决于数据展示的需求。

- 通过条形图，我们可以获得以下信息：

1、类别比较：直观比较不同类别或组的数值大小。

2、数据的分布：观察不同类别中数据的分布情况。

3、趋势识别：识别数据随时间或顺序变化的趋势。

- 有的条形图上有误差棒（Error Bars）。误差棒用来表示数据的变异度、不确定性或置信区间，在下图中，误差棒的长度代表的是95%置信区间，有助于帮助我们理解数据的稳定性和变异范围。
- # Bar plot of average total bill by day
- plt.figure(figsize=(8, 6))
- sns.barplot(data=tips, x='time', y='tip', palette='coolwarm')
- plt.title('Average Tip by Meal')
- plt.xlabel('Meal')
- plt.ylabel('Average Tip')
- plt.show()

- 结果如下图：我们分别描述的是午餐（Lunch）以及晚餐（Dinner）的平均小费量。

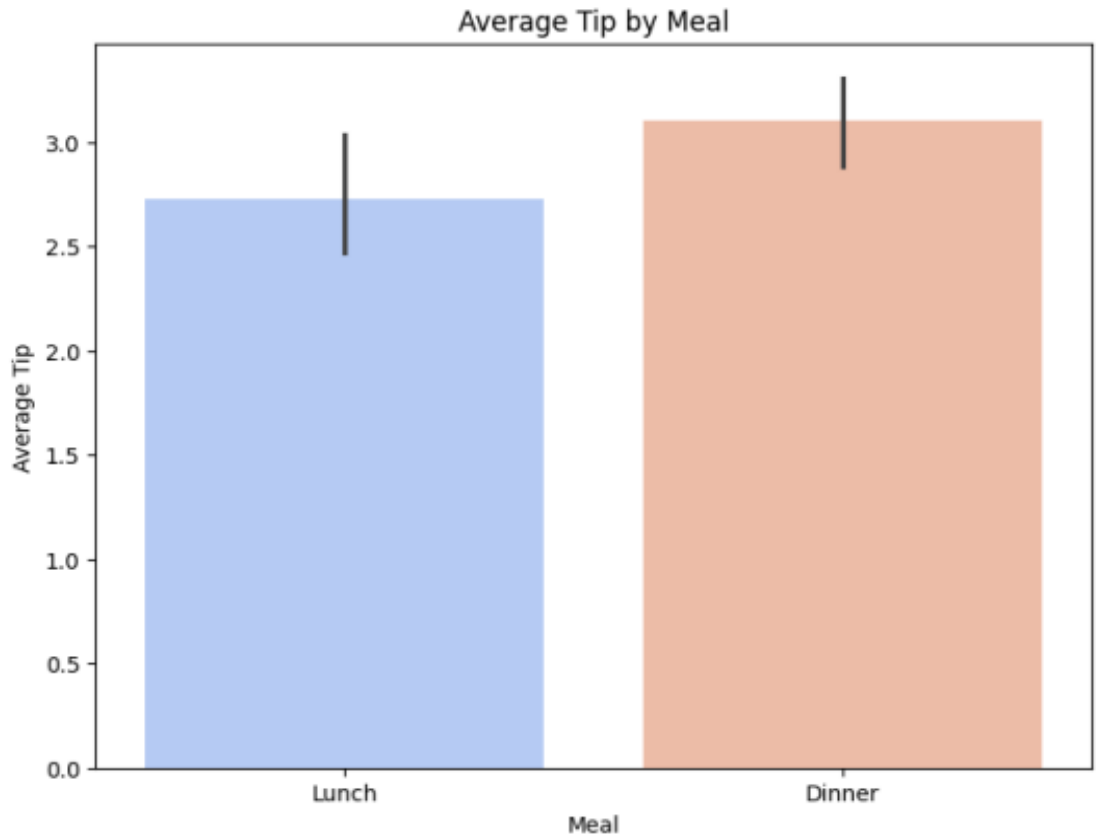


图 1.3: 条形图

散点图

- 散点图（Scatter Plot）是数据可视化中一种基本且常用的图表类型，它通过在二维平面上的点来表示两个数值变量之间的关系。每个点的位置由其所对应的变量值决定，分别在X轴和Y轴上表示。散点图主要用于探索和展示两个连续变量之间是否存在某种相关关系，以及这种关系的强度和方向。
- 通过散点图，我们可以获得以下信息：

1、相关性：观察两个变量之间是否存在线性关系或非线性关系。如果数据点近似分布在一条直线附近，说明变量之间存在线性相关性；如果分布呈现某种曲线模式，说明存在非线性关系。

2、关系的方向：数据点的分布趋势可以显示变量之间的关系是正向的（即一个变量增加时，另一个变量也增加）还是负向的（即一个变量增加时，另一个变量减少）。

3、异常值：在散点图中，远离其他数据点的点可能表明异常值或离群点，这些点可能需要进一步分析。

4、数据的聚集程度：数据点的密集程度可以反映变量之间关系的强度。点越密集，说明两个变量之间的关系越紧密。

- 散点图是探索性数据分析（EDA）中非常重要的工具，可以帮助研究人员发现数据中的模式、趋势和异常值，为后续的深入分析和建模提供线索。
- plt.figure(figsize=(8, 6))
- sns.scatterplot(data=tips, x='total_bill', y='tip')
- plt.title('Scatter Plot of Total Bill vs. Tip')
- plt.xlabel('Total Bill')
- plt.ylabel('Tip')
- plt.show()

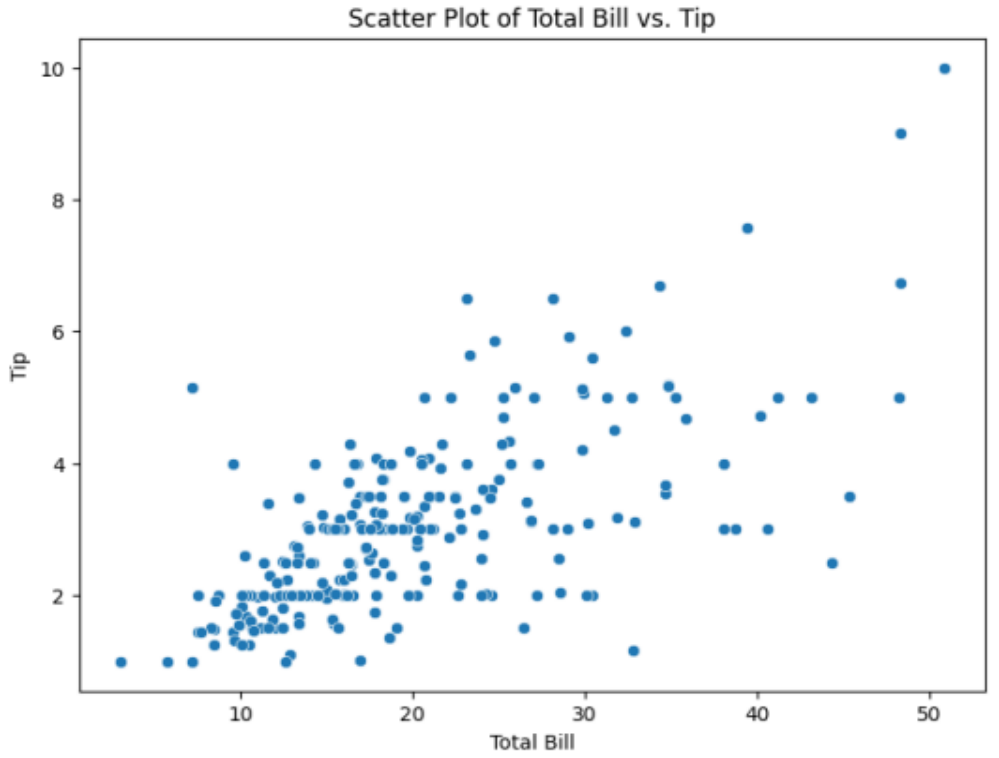


图 1.4: 散点图

结果如图所示：我们使用散点图来可视化用餐费用（total bill）和小费（tip）之间的关系。显而易见，这两个变量成正相关关系。

热图

- 热图（Heatmap）是一种数据可视化技术，用于显示两个类别变量之间的矩阵数据以及这些数据的量值。它通过不同的颜色来表示数据点的大小或数值，颜色的深浅通常与数据值的大小成正比。热图常用于展示数据的分布、变化或者两个变量之间的相关性等。

- 通过热图，我们可以获得以下信息：

1、数据模式：热图可以帮助识别数据中的模式，例如，某些变量组合是否频繁出现或者某些值是否异常高或低。

2、数据的比较：通过颜色的对比，可以快速比较不同类别或组合之间的数据差异。

3、相关性：在展示变量之间的相关系数矩阵时，热图可以直观地反映变量之间的相关性强度，颜色越深表示相关性越强。

4、异常值检测：热图中的颜色异常点可以帮助快速识别数据中的异常值或离群点。

- 在下图中，我们使用热图来直观表示三个连续变量之间的关系。因此，首先我们需要计算不同数据之间的相关性。
- corr = tips[['total_bill','tip','size']].corr()
- plt.figure(figsize=(8, 6))
- sns.heatmap(corr, annot=True, cmap='coolwarm', center=0)
- plt.title('Heatmap of Correlation Matrix')
- plt.show()

- 结果如图所示，对角线上的值为1，因为变量与自己的相关性为1。同时，我们可以看到餐费总量（total bill）与其他两个变量都有较强关系。

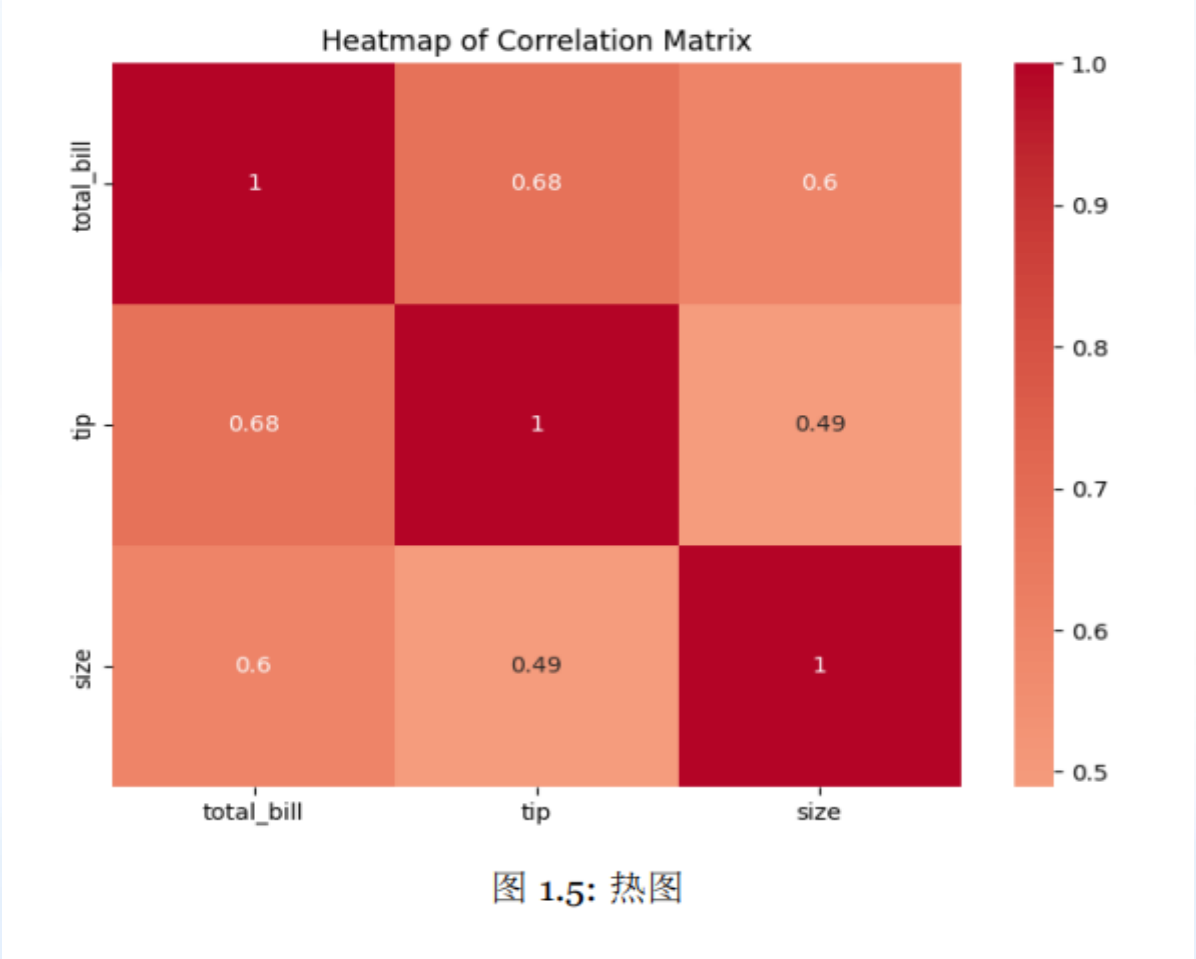
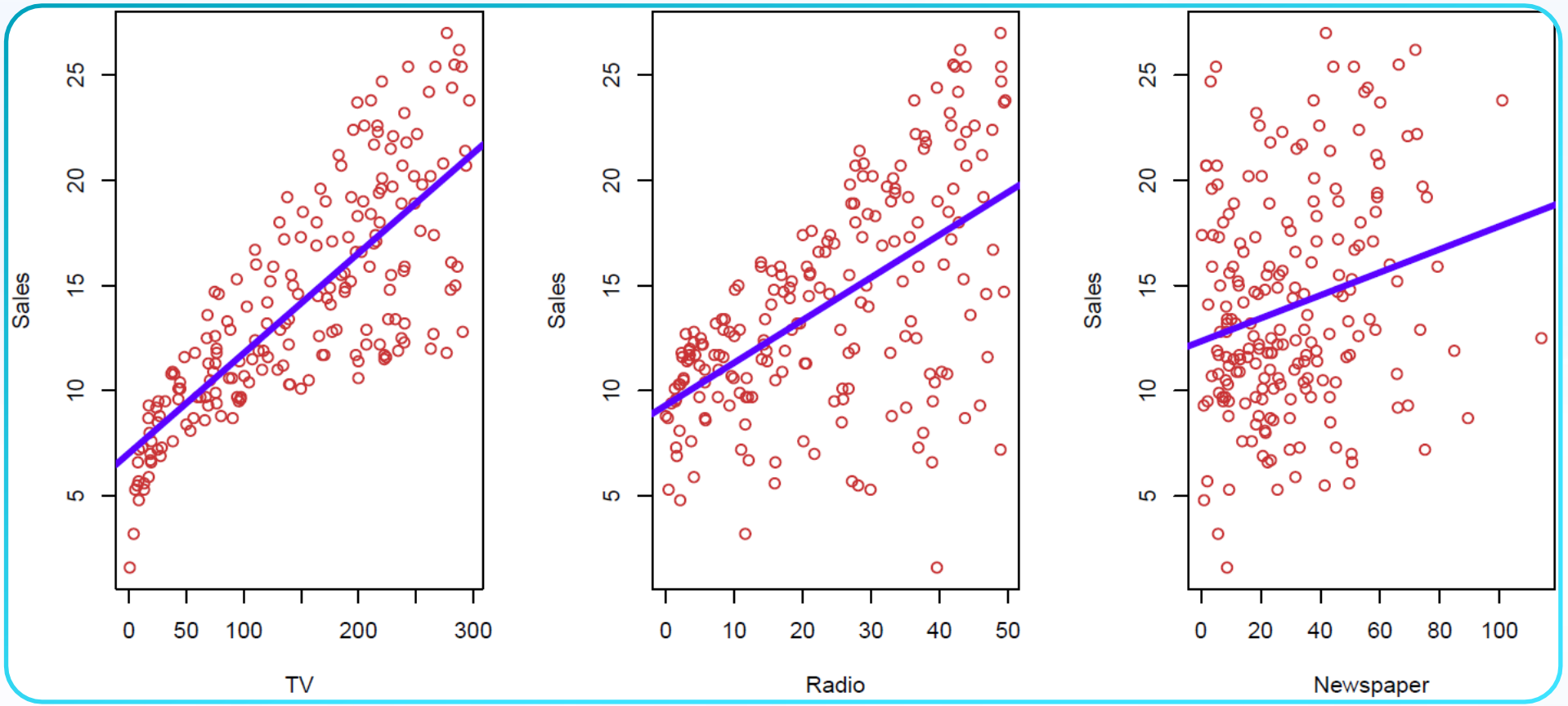


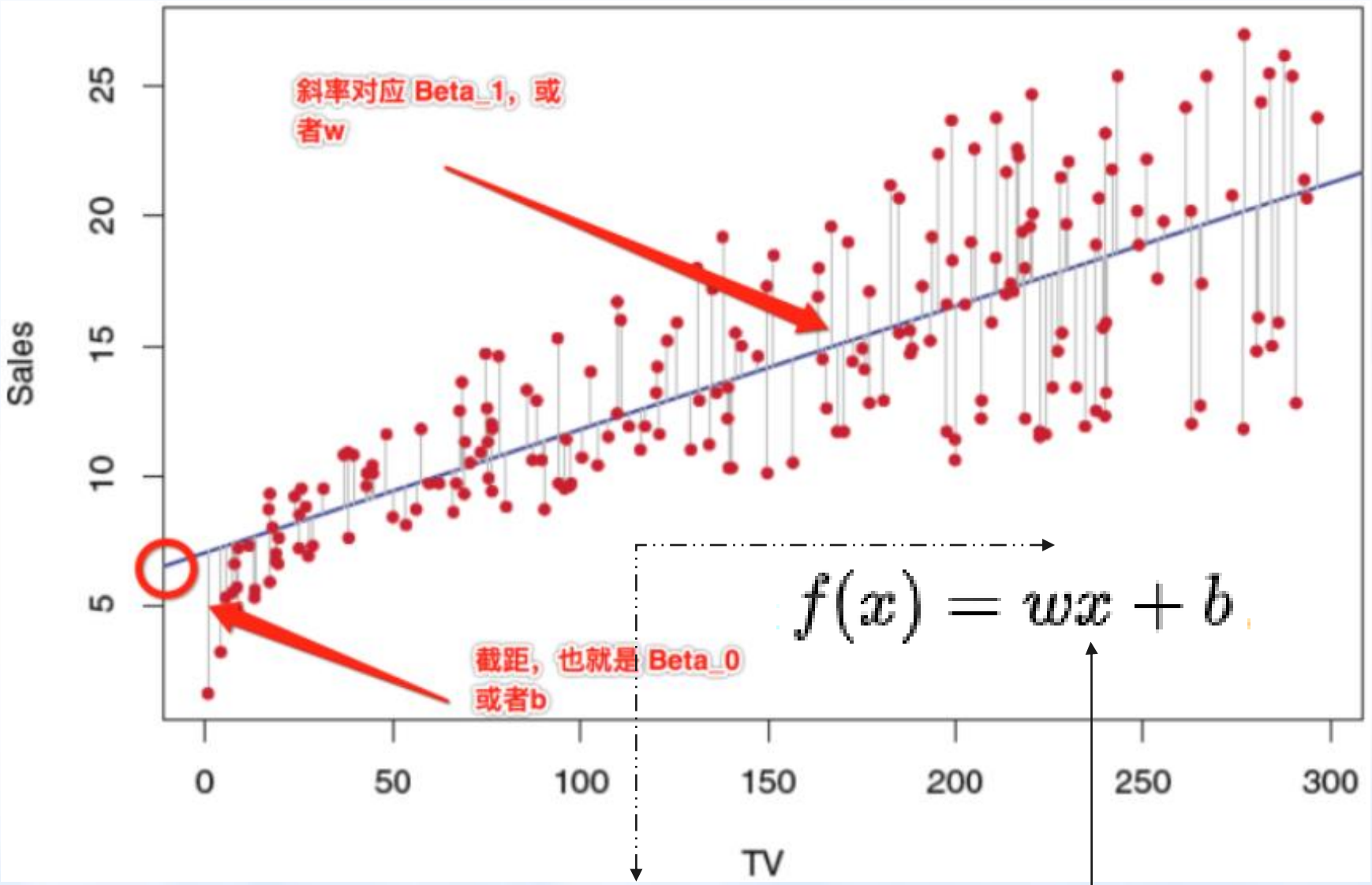
图 1.5: 热图

2.2.1 Data Preprocessing
数据处理进阶

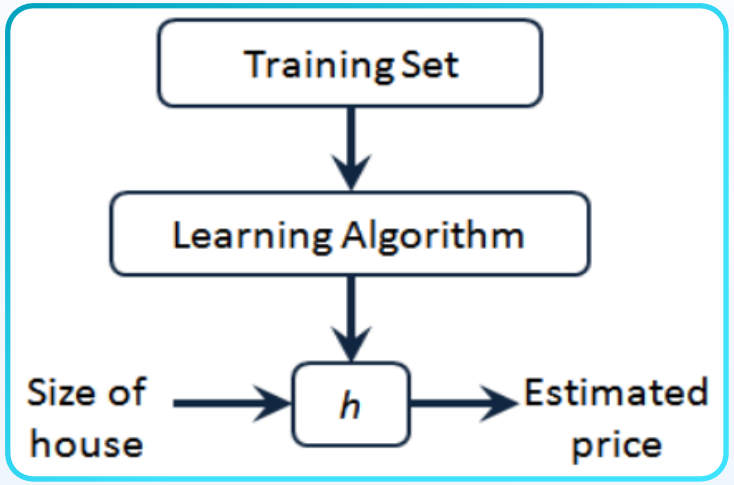
The Advertising data set

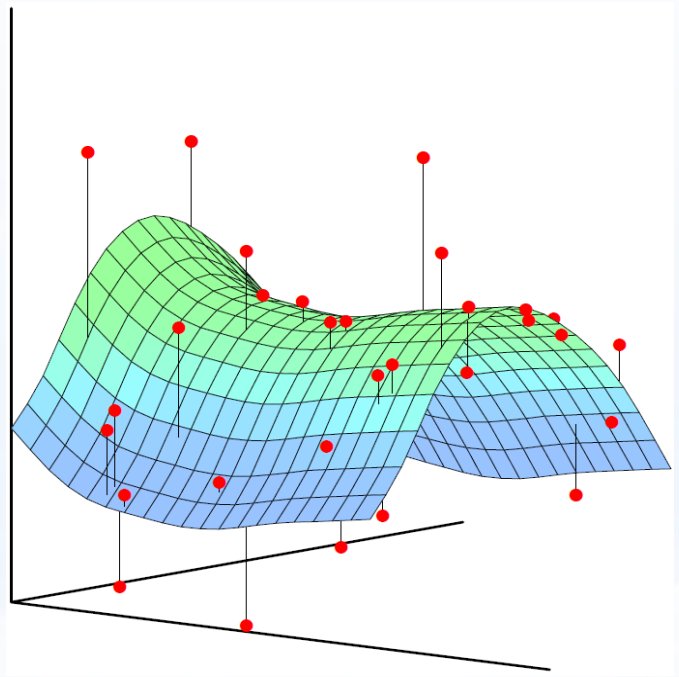


The Advertising data set



Training set Example





Training set of housing prices (Portland, OR)	Size in feet² (x)	Price ($) in 1000's (y)
	2104	460
	1416	232
	1534	315
	852	178
	...	...

Notation:

- **m** = Number of training examples /rows in the table
- **x**'s = "input" variable / features
- **y**'s = "output" variable / "target" variable

- m 代表训练集中样本的数量
- n 代表特征的数量
- x 代表特征/输入变量
- y 代表目标变量/输出变量
- (x, y) 代表训练集中的样本
- $(x^{(i)}, y^{(i)})$ 代表第i个观察样本
- h 代表学习算法的解决方案或函数也称为假设（**hypothesis**）
- $\hat{y} = h(x)$,代表预测的值

数据预处理：缺失数据

表2-1 缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.015
2	-	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	-	0.024

导致数据缺失的三类原因：

1. **完全随机缺失**（MCAR - Missing Completely At Random）：缺失数据的出现完全是随机的，与任何观察到或未观察到的数据都无关。如由于数据提供商的疏忽，可能没能搜集到一些公司披露过的数据。这些数据与公司的基本面，公司的规模以及公司披露的习惯无关。**最简单的情况，可以使用删除法或插补法。**
2. **随机缺失**（MAR - Missing At Random）：缺失数据的出现与其他观察到的数据有关，但与缺失本身的数据无关。例如在一项问卷调查中，某些问题的回答缺失是因为这些问题与某些人口统计学变量（如年龄、性别等）有关。如果我们能找到并控制这些相关变量，缺失数据可以通过适当的方法（如**多重插补**）进行处理。
3. **非随机缺失**（MNAR - Missing Not At Random）：是指我们在缺失数据的概率与数据集之外的其他变量也有关系。例如在健康调查中，患有某种疾病的患者可能不愿意回答有关健康状况的问题，导致缺失数据与健康状况有关。为**最复杂的情况，需要了解缺失机制，需要建立特定的统计模型来处理缺失数据。**

数据预处理：缺失数据

- 值得注意的是我们很难区分随机缺失与非随机缺失。
- 接下来介绍的大部分方法需要假设数据是**完全随机缺失（MCAR）或随机缺失（MAR）**。

简单处理方法

1. 删除数据（适用于MCAR）

- 最简单的处理方法是直接将具有缺失特征的数据进行删除。
- 这种方法操作简单，但是当有大量数据都有缺失时，可能会造成大量数据点从数据中遭到删除。这种情况下，我们可以考虑将数据缺失较多的特征从模型中移除。

数据预处理：缺失数据

2. 简单数据填充（适用于MCAR）

- 删除数据会造成信息的丢失。因此在许多应用中，我们使用数据填充来对缺失数据进行处理。
- 例如，对于连续变量，我们可以使用未缺失的数据计算变量的**平均值**或者**中位数**。并用这些值对缺失的数据进行填充。这些数据填充方法我们在后续章节的程序中会展开应用。

（1）分类数据

- 如果缺失的数据是分类数据。那么，我们可以考虑**用频率最高的分类用于填充缺失数据**。
- 例如，如果我们将贷款申请人的房产数量当成一个分类数据，那么0 套住房的申请者应该会比拥有房产的人更多，因此我们可以考虑将缺失的数据设置为 0。

（2）时间序列数据

- 如果我们数据中有时间序列，那么我们可以考虑用前一期的非缺失数据对当期的数据进行填充。
- 例如，如果某公司**市净率数据（bm）**在 2020 年缺失，那么我们可以考虑用 2019 年的数据对其进行填充。这种填充方法通常只有在该**变量相对稳定的情况下才能使用**。
- 变化较大的数据（例如**月度股票回报**）则不能用这个方法进行处理。

缺失数据：进阶填充方法

3. 进阶填充方法（适用于MCAR/MAR）

- 虽然将数据用平均值或中位数进行填充的操作简单有效，并且应用广泛，但这些方法也有一些问题。
- 例如这些方法会造成过多的数据出现在平均数或中位数上。另外，我们没有充分运用到数据中的一些相关性信息。
- 接下来我们介绍的数据填充方法主要思路是**利用数据之间的相关性来对缺失数据进行填充**。
- 例如，如果我们数据中有两个相关度较高的变量（比如市净率，市盈率）。

- 某公司的市净率数据可能有缺失，但是我们可以通过该公司未缺失的市盈率数据对缺失的市净率数据进行预测。并用模型的预测值来填充缺失的数据。
- 这一类方法的好处是我们填充的缺失数据相较于平均值/中位数应该更接近于缺失数据原本的数值。
- 因此，这些填充方法所取得的数据应该能帮助我们的模型取得更好的预测效果。

进阶填充方法：多重插补链式方程

4. 多重插补链式方程（MICE，Multiple Imputation by Chained Equations）

MICE是一种用于处理数据集中缺失值的方法。**它通过生成多个完整的数据集来反映缺失数据的不确定性，每个数据集都用不同的填补值来代替缺失值，然后在分析中结合这些填补值。**
具体如下：

a) 将每一个特征中缺失的值用该特征为缺失数据的平均值来暂时替代

b) 重复以下步骤 k 次（需要预先设定 k 值）：

- 对每一个有缺失数据的特征 x_j，我们用回归方法将将x_j作为目标变量，x_j之外的数据作为特征进行回归。
- 用上一步训练的回归模型对 x_j进行预测，并生成$\hat{x}_j$。
- 对于特征x_j中的缺失值$x_j^{(i)}$，我们将$x_j^{(i)}$的原来的值用归模型预测值$\hat{x}_j^{(i)}$取代。

进阶填充方法：多重插补链式方程

4. 多重插补链式方程

- 用表2-1中的数据为例。我们做以下数据填补。右图很好的说明了数据填补的步骤。
- 第一步（如图所示）先用每列的平均值来填补缺失的数据（每股股价的平均值为 18.934，市净率平均值为 0.531）。

第一步：用平均数填补缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.014
2	**18.934**	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	**0.531**	0.024

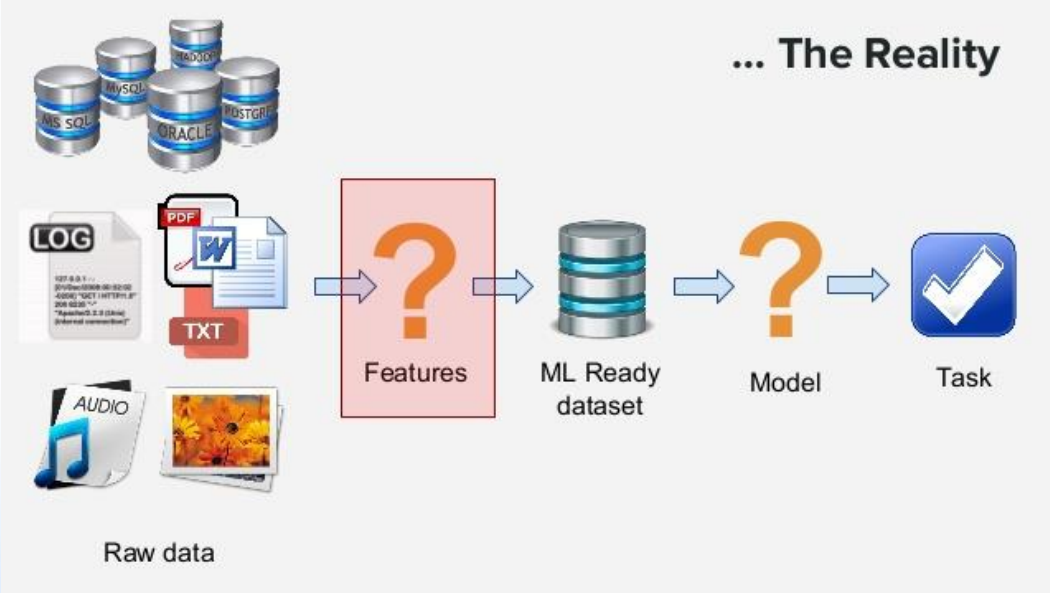
第二步：用其他特征更新缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.014
2	**23.275**	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	**0.523**	0.024

进阶填充方法：多重插补链式方程

4. 多重插补链式方程

下表为 MICE 数据填补 $k=1$ 的情况，我们通过每股股价与其他两个变量之间的关系，将每股价格修正为 23.275（通过回归分析，我们发现 股价 = 16.164 − 1.138 × 市净率 + 201.89 × 资产回报率）。



第一步：用平均数填补缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.014
2	**18.934**	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	**0.531**	0.024

第二步：用其他特征更新缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.014
2	**23.275**	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	**0.523**	0.024

进阶填充方法：多重插补链式方程

4. 多重插补链式方程

- 同时我们将市净率进行同样的调整，得到 **0.523** 的数值。如果我们设置更高的 k 值，那么我们将重复以上步骤来对缺失的数据值进行更新。
- 我们将 MICE 填充方法与平均值填充法进行比较。可以发现 MICE 跟原始数据更为接近。
- 因此，可以得知我们可以通过 MICE 方法得到更好的数据填充效果。

第一步：用平均数填补缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.014
2	**18.934**	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	**0.531**	0.024

第二步：用其他特征更新缺失数据

数据	每股股价	市净率	资产回报率
1	20.170	0.358	0.014
2	**23.275**	0.220	0.036
3	14.400	0.296	0.046
4	10.090	0.290	0.036
5	17.900	**0.523**	0.024

进阶填充方法：多重插补链式方程

- Python 示例代码（使用 statsmodels 库）

python Copy code

```python
import pandas as pd
import numpy as np
from statsmodels.imputation.mice import MICEData

# 创建一个示例数据集，包含缺失值
data = pd.DataFrame({
    'A': [1, 2, np.nan, 4, 5],
    'B': [5, 4, 3, np.nan, 1],
    'C': [2, np.nan, 2, 4, 4]
})

# 使用 MICE 进行多重插补
imp_data = MICEData(data)

# 进行 5 次迭代
for _ in range(5):
    imp_data.update_all()

# 获取插补后的数据
imputed_data = imp_data.data
print(imputed_data)
```

- R 语言示例代码（使用 mice 包）

R Copy code

```r
library(mice)

# 创建一个示例数据集，包含缺失值
data <- data.frame(
  A = c(1, 2, NA, 4, 5),
  B = c(5, 4, 3, NA, 1),
  C = c(2, NA, 2, 4, 4)
)

# 使用 MICE 进行多重插补
imp <- mice(data, m = 5, maxit = 5, method = 'pmm', seed = 500)

# 查看插补后的数据
completed_data <- complete(imp)
print(completed_data)
```

➢ 多重插补链式方程（MICE）是一种强大的数据插补方法，能够有效处理数据集中缺失值的问题，保持数据的完整性和分析结果的可靠性。

进阶填充方法：矩阵补完法

5. 矩阵补完法

- 另一种广泛使用的方法是使用主成分分析法（PCA），在计算主成分的时候我们同时能对缺失数据值进行推算。假设我们原有数据放在X矩阵中，矩阵维度为 n × m (m个特征n个数据点)。

a) 生成一个$\tilde{X}$矩阵，如果x_{ij}未缺失，则$\tilde{x}_{ij} = x_{ij}$，否则，$\tilde{x}_{ij}$取 j 列的平均值。

b) 重复以下步骤，直至目标函数停止下降：

- 用主成分分析方法来计算 $\tilde{X}$ 的 K 个主成分：

$$\min_{A,B}\left\{\sum_{j=1}^{m}\sum_{i=1}^{n}\left(\tilde{x}_{ij}-\sum_{k=1}^{K}a_{ik}b_{ik}\right)\right\}$$

- 用 K 个主成分的线性组合更新 X 中缺失的数据：

$$\tilde{x}_{ij} := \sum_{k=1}^{K} a_{ik}b_{jk}$$

进阶填充方法：矩阵补完法

- 计算目标函数：

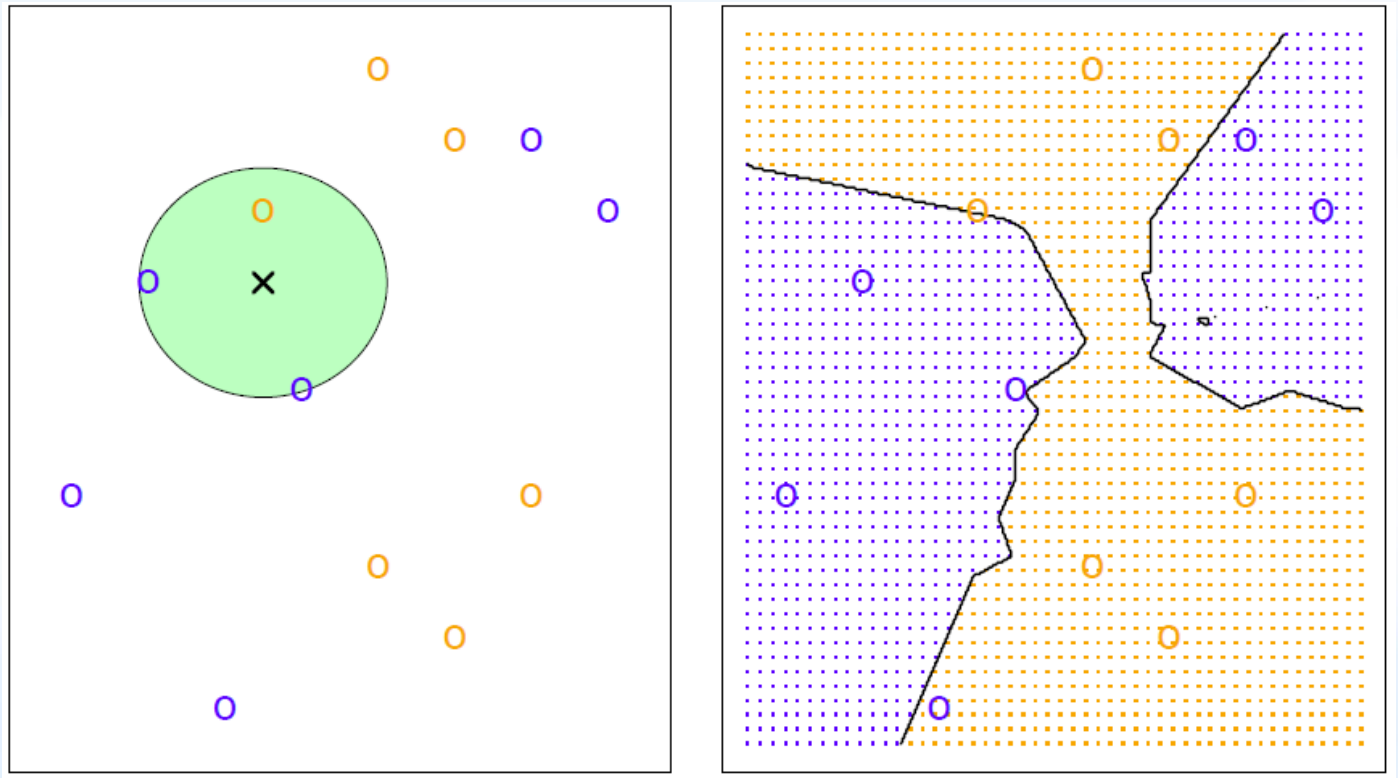
$$\sum_{(i,j)\text{为未缺失数据}} \left(x_{ij} - \sum_{k=1}^{K} a_{ik} b_{jk} \right)^2$$

c) 输出 $\tilde{X}$

进阶填充方法：K近邻法和其他方法

6. K近邻法

- 我们也可以用 *KNN* 方法来对数据进行填补。
- 简单来说，对于一个数据点，我们通过其未缺失的变量来找到离该变量接近的其他数据点，并用这些数据点中为缺失的数据的平均值对该数据点的缺失数据进行填补。



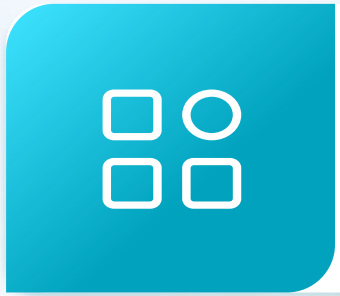
进阶填充方法：K近邻法和其他方法

7. **其他方法**

有些机器学习算法可以自动处理含有缺失值的数据。

- 例如我们之后会提到的一种**提升算法 xgboost**。使用该算法之前不需要对缺失数据进行预处理。 xgboost 算法自动将缺失的数据当成特征中一个特殊的值来进行运用。

- 如果算法无法自动处理缺失数据，而数据填充也不合适，那么我们可以考虑对数据进行如下操作：
- 首先，将缺失数据进行填充（例如将其设置为 0）。然后，再生成一个变量，当发生数据缺失时该变量取值为 1，否则为 0。



2.2.2 Gradient Descent

Gradient Descent

A general algorithm for minimizing the cost function J

Have some function $J(\theta_0, \theta_1)$ $J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$ $\min_{\theta_0 \ldots \theta_n} J(\theta_0, \ldots, \theta_n)$

Outline:

- Start with some θ_0, θ_1 (say $\theta_0 = 0, \theta_1 = 0$)
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

批量梯度下降（Batch Gradient Descent,BGD）

梯度下降的每一步中，都用到了所有的训练样本

随机梯度下降（Stochastic Gradient Descent,SGD）

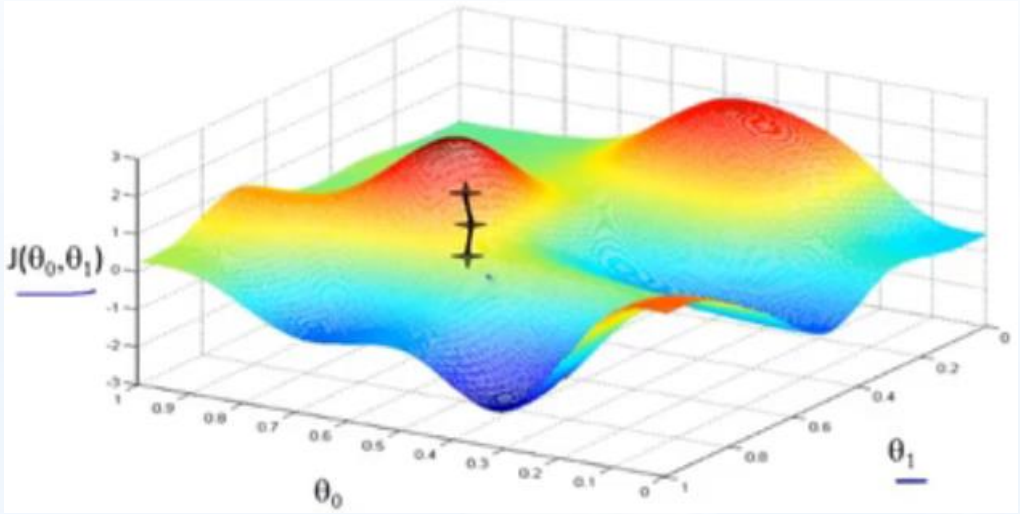
梯度下降的每一步中，用到一个样本，在每一次计算之后便更新参数，而不需要首先将所有的训练集求和

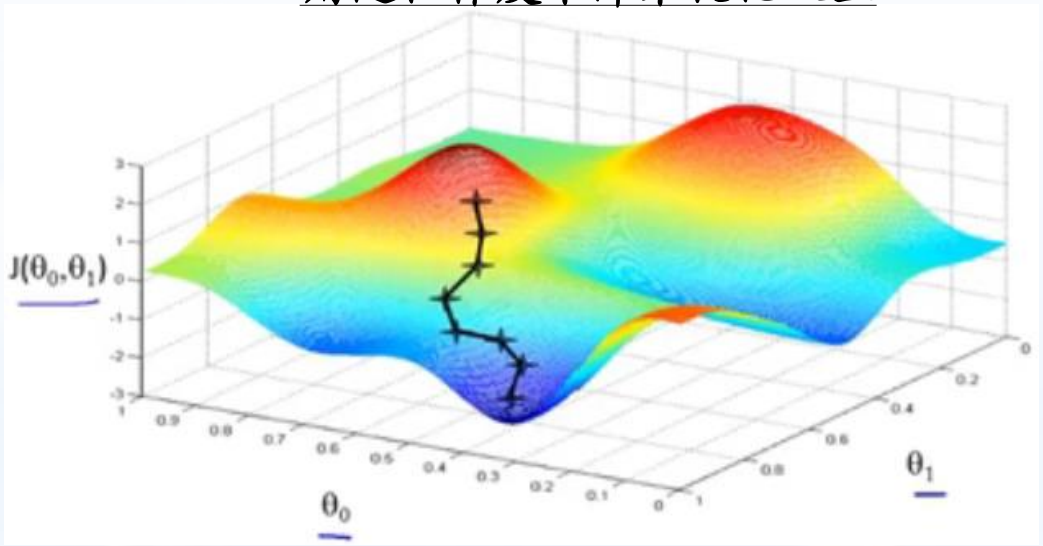
小批量梯度下降（Mini-Batch Gradient Descent,MBGD）

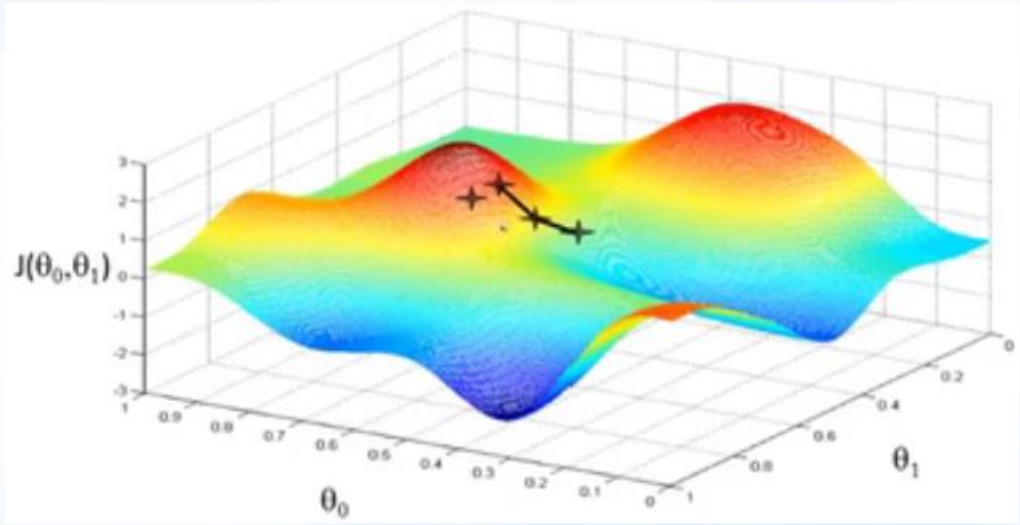
梯度下降的每一步中，用到了一定批量的训练样本

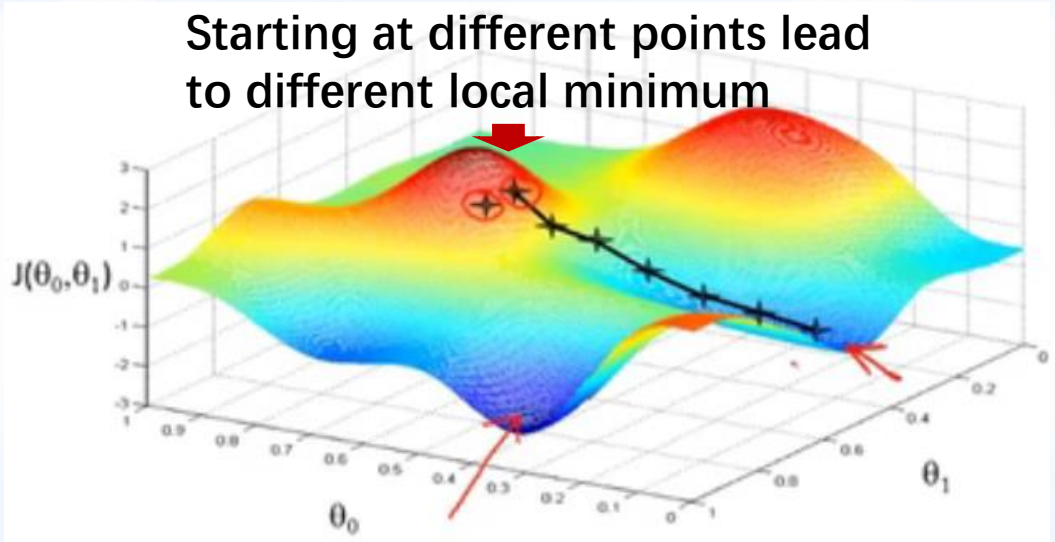
Gradient Descent
可视化讲解

用随机梯度下降来优化人生









梯度下降背后的思想：开始时我们随机选择一个参数的组合计算代价函数，然后我们寻找下一个能让代价函数值下降最多的参数组合。我们持续这么做直到到到一个局部最小值（local minimum），因为我们并没有尝试完所有的参数组合，所以不能确定我们得到的局部最小值是否便是全局最小值（global minimum），选择不同的初始参数组合，可能会找到不同的局部最小值。

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$

}

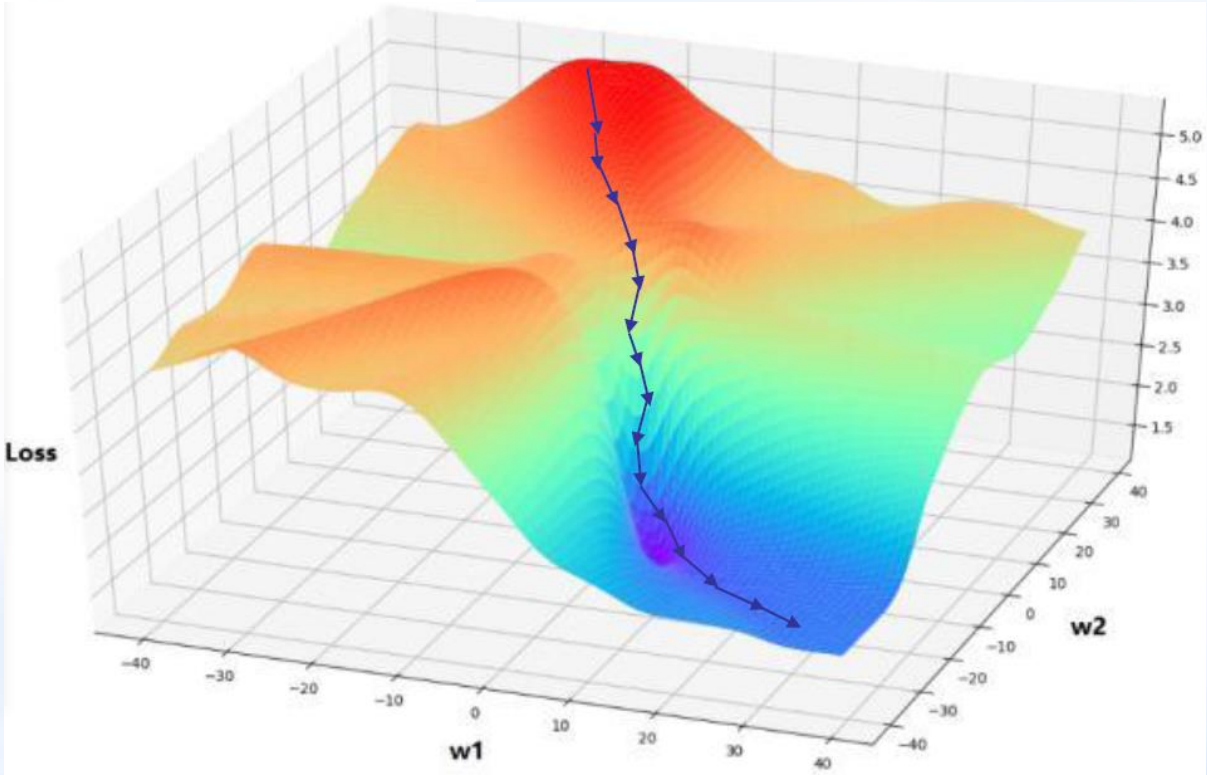
在批量梯度下降中，我们每一次都同时让所有的参数减去学习速率乘以代价函数的导数。

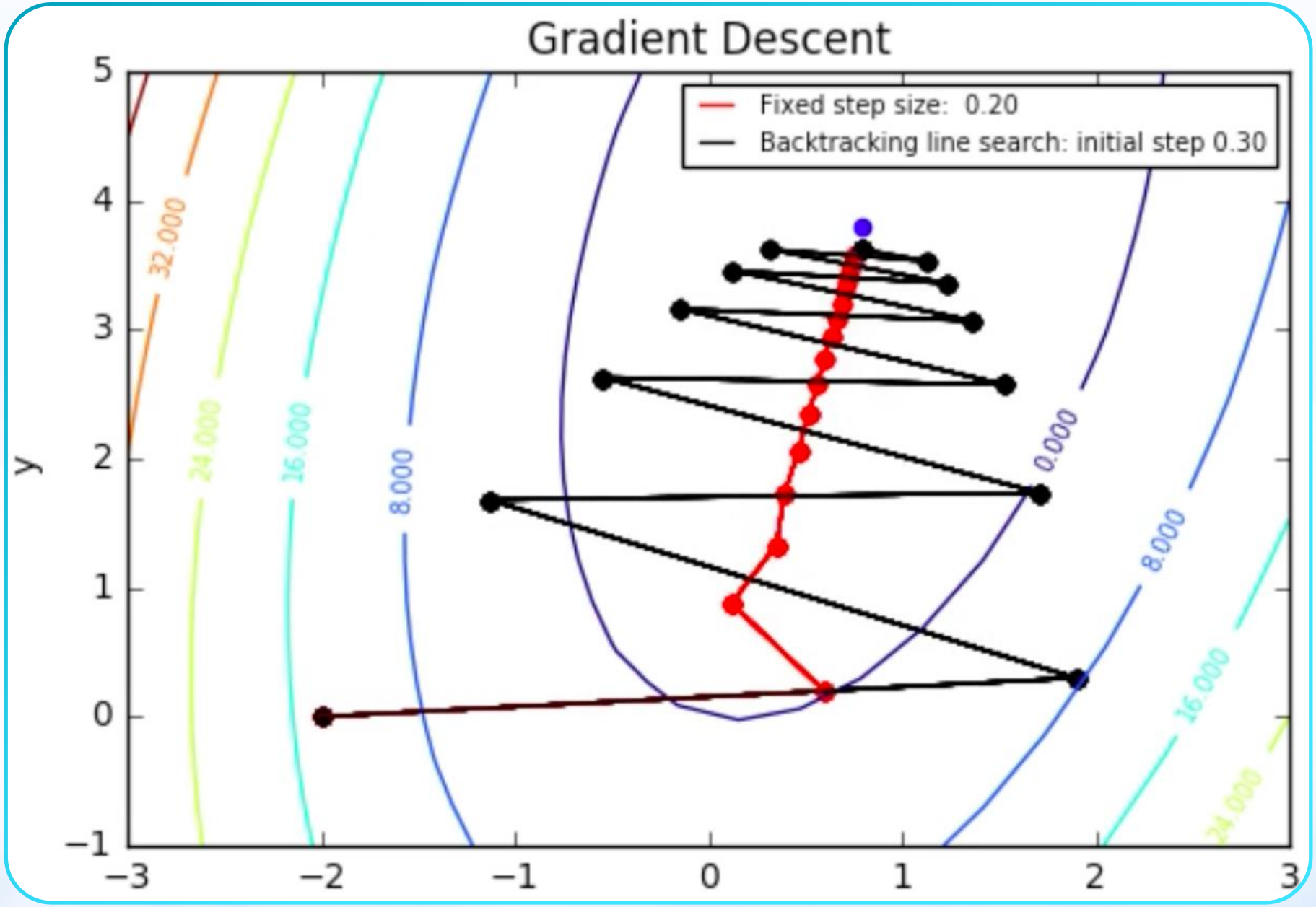
Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Incorrect:

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

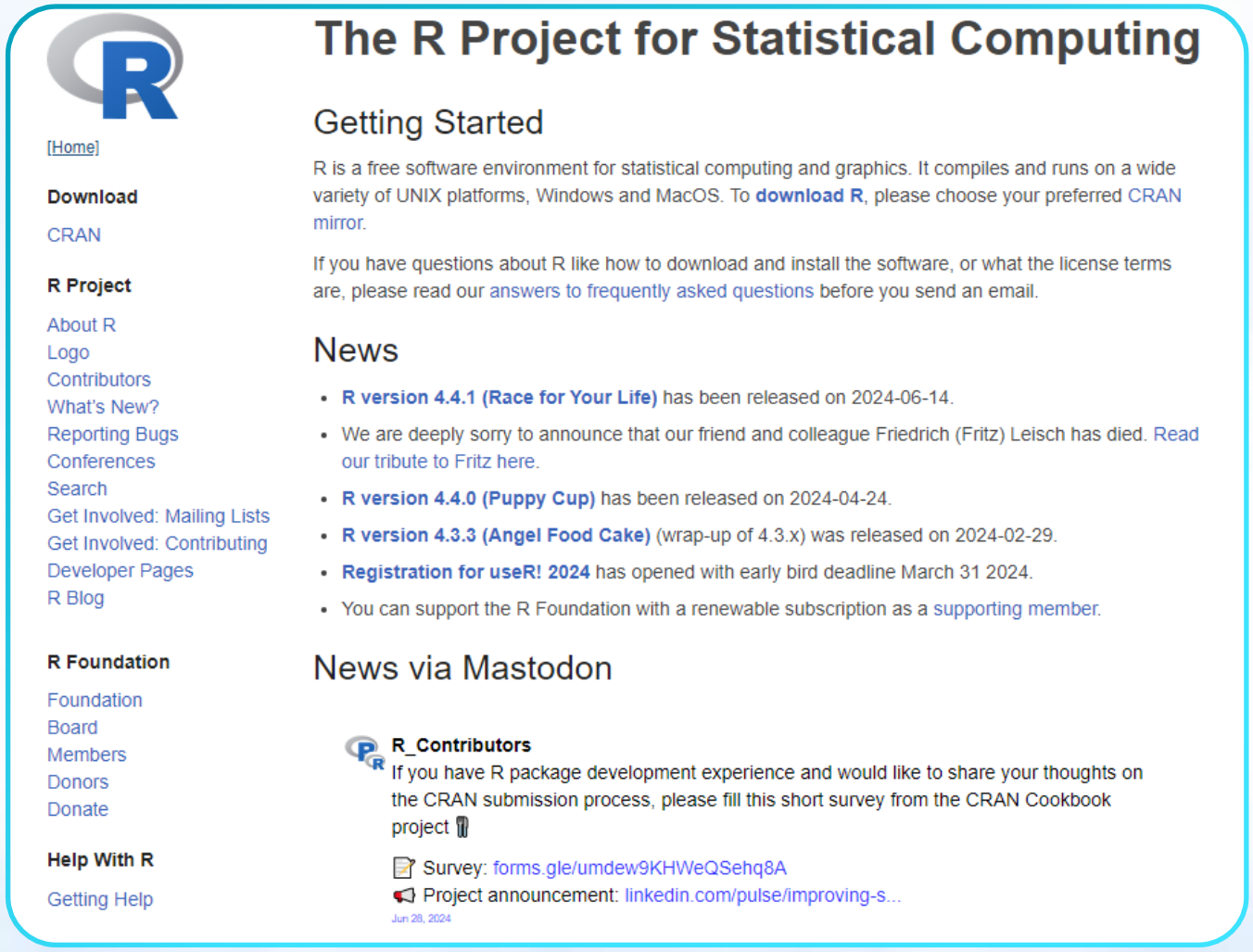



Gradient Descent
Fixed step size: 0.20
Backtracking line search: initial step 0.30
32.000
24.000
16.000
8.000
0.000
8.000
16.000
24.000
y
5
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

2.2.3 Lab Practice

Getting Started

https://www.r-project.org/



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Get Involved: Contributing
Developer Pages
R Blog

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

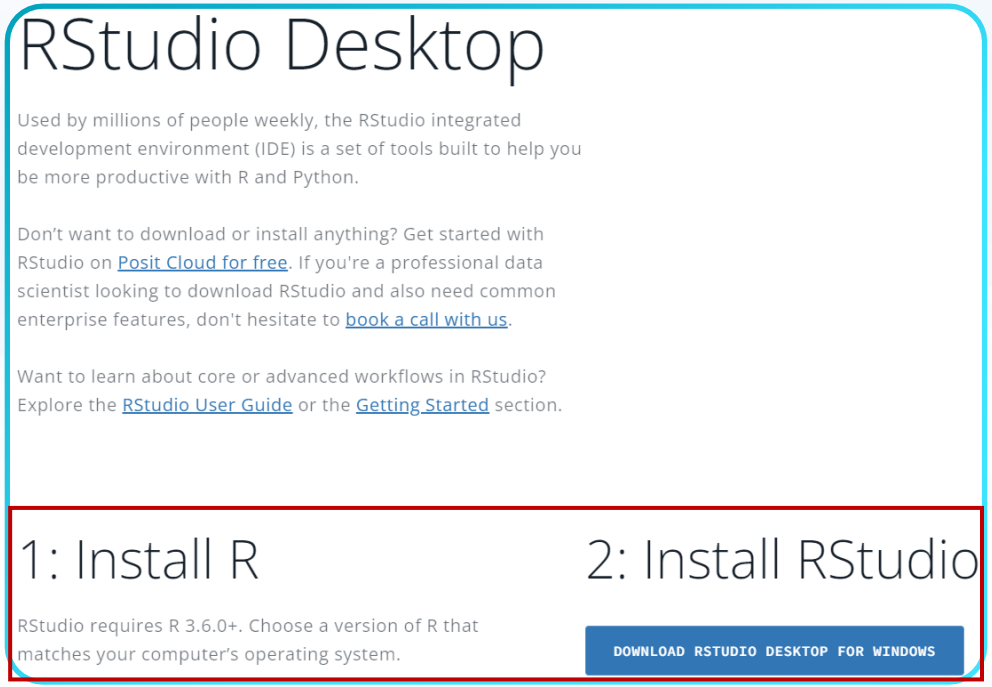
- **R version 4.4.1 (Race for Your Life)** has been released on 2024-06-14.
- We are deeply sorry to announce that our friend and colleague Friedrich (Fritz) Leisch has died. Read our tribute to Fritz here.
- **R version 4.4.0 (Puppy Cup)** has been released on 2024-04-24.
- **R version 4.3.3 (Angel Food Cake)** (wrap-up of 4.3.x) was released on 2024-02-29.
- **Registration for useR! 2024** has opened with early bird deadline March 31 2024.
- You can support the R Foundation with a renewable subscription as a supporting member.

News via Mastodon

R_Contributors
If you have R package development experience and would like to share your thoughts on the CRAN submission process, please fill this short survey from the CRAN Cookbook project

Survey: forms.gle/umdew9KHWeQSehq8A
Project announcement: linkedin.com/pulse/improving-s...
Jun 28, 2024

https://posit.co/download/rstudio-desktop/



RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

Don't want to download or install anything? Get started with RStudio on Posit Cloud for free. If you're a professional data scientist looking to download RStudio and also need common enterprise features, don't hesitate to book a call with us.

Want to learn about core or advanced workflows in RStudio? Explore the RStudio User Guide or the Getting Started section.

1: Install R

RStudio requires R 3.6.0+. Choose a version of R that matches your computer's operating system.

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS

Rstudio界面

脚本编辑器 Script Editor

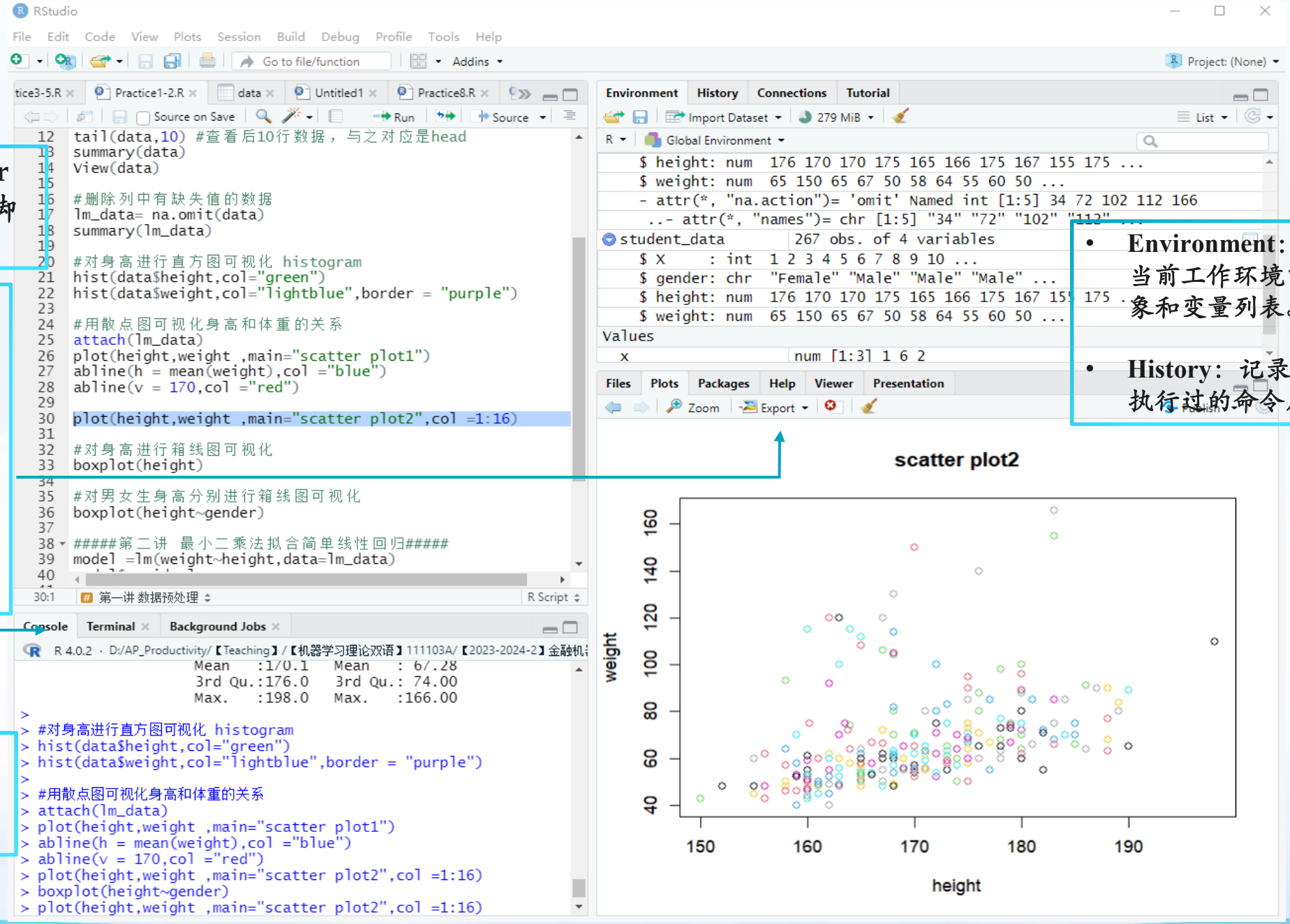
- 功能：编写和编辑R脚本代码。

- **Files**：浏览和管理工作目录中的文件。
- **Plots**：显示绘制的图形和图表。
- **Packages**：查看和管理已安装的R包。
- **Help**：查阅R的帮助文档和函数说明。
- **Viewer**：查看本地或在线的HTML内容。

控制台Console

功能：显示并运行脚本代码的结果。

- **Environment**：显示当前工作环境中的对象和变量列表。
- **History**：记录所有执行过的命令历史。



课程小节

- 数据预处理、数据缺失的原因及解决方案
- 梯度下降的基本概念及应用
- 了解 Rstudio 的四个模块及编程环境，掌握如何进行初步数据分析、绘图和报告生成。