Swiss UMEF
UNIVERSITY OF APPLIED SCIENCES INSTITUTE

Big Data Analytics
大数据分析

03: In-Memory Analytics with Pandas. Introduction to Pandas
用Pandas进行内存分析　　Pandas简介

Instructor: Oleh Tymchuk
讲师：奥勒·特姆恰克

#03: Agenda 课程安排

- What is Pandas? 什么是Pandas？
- Pandas Data Structures Pandas数据结构
- Loading Data into Pandas 加载数据到Pandas
- Practical cases 实际案例
- Useful Links 实用链接

What is Pandas?
什么是Pandas？

What is Pandas? 什么是Pandas？

- Fast, flexible, and open-source Python library

  快速、灵活且开源的Python库

- Designed for data manipulation and analysis

  专为数据操作和分析设计

- “Pandas” = Panel Data

  “Pandas”= Panel Data（面板数据）

- Optimized for in-memory data operations

- 针对内存数据操作进行了优化

What Types of Data Can Pandas Handle?
Pandas能处理哪些类型的数据？

Tabular Data 表格数据

- Similar to an SQL table or an Excel spreadsheet
- 类似于SQL表或Excel电子表格
- Supports heterogeneously-typed columns (e.g., numerical + categorical data)
- 支持异构类型列（例如，数值+分类数据）

Time Series Data 时间序列数据

- Works with both ordered and unordered time series
- 支持有序和无序的时间序列

Matrix Data 矩阵数据

- Supports homogeneous or heterogeneous matrix-like structures
- 支持同质或异质矩阵结构

Observational & Statistical Data 观察和统计数据

- Can store any type of dataset, even unlabeled data
- 可以存储任何类型的数据集，甚至是未标记的数据

Key Features of Pandas　Pandas的主要特点

- Easy handling of missing data　轻松处理缺失数据
- Label-based indexing and slicing　基于标签的索引和切片
- Powerful data filtering and transformation 强大的数据过滤和转换
- Efficient group-by operations and aggregation 高效的分组操作和聚合
- Integration with other data analysis tools 与其他数据分析工具集成

Pandas Cohabitation Pandas的共生环境

Jupyter Notebooks & Google Colab:

- Perfect for interactive data analysis and visualization
- 非常适合交互式数据分析和可视化

Plotting with: 绘图：

- matplotlib: Basic plotting 基础绘图
- seaborn | plotly: Nicer statistical plots 更美观的统计图

Numerical analysis with: 数值分析：

- numpy: Efficient numerical computations 高效的数值计算

Modelling with: 建模：

- statsmodels: Statistical modeling 统计建模
- scikit-learn: Machine learning 机器学习

Installing & Setup 安装与设置

```
# Install via pip/conda: 通过pip/conda安装：
!pip install pandas

# Import the library: 导入库：
import pandas as pd

# Check version: 检查版本：
print(pd.__version__)
```

Pandas Data Structures
Pandas数据结构

Data Structures 数据结构

Dimensions 维度	Name 名称	Description 描述
1	Series 系列	1D labeled homogeneously-typed array 一维带标签的同质类型数组
2	DataFrame 数据框	General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed column 通用二维带标签的、大小可变的表格结构，列可能为异质类型

Labeled Data 带标签的数据

Labeled data refers to data that has identifiers (labels) for rows and columns.

带标签的数据指的是行和列都有标识符（标签）的数据。

Key Components in Pandas: Pandas中的关键组件：

- Index: labels for rows (e.g., 0, 1, 2, ... or custom labels like ['A', 'B', 'C']).
- 索引：行的标签（例如，0, 1, 2, ... 或自定义标签如 ['A', 'B', 'C']）。
- Columns: labels for columns (e.g., ['Name', 'Age', 'City']).
- 列：列的标签（例如，['姓名', '年龄', '城市']）。

Why Labeled Data Matters: 为什么带标签的数据很重要：

- Enables intuitive data access using meaningful labels instead of numeric positions
- 允许使用有意义的标签而不是数字位置直观地访问数据
- Supports alignment of data during operations 支持在操作期间对齐数据
- Makes data more readable and interpretable 使数据更易于阅读和解释

Labeled Data 带标签的数据

	Name 姓名	Age 年龄	Salary 收入
0	Alice	25	50000
1	Bob	30	55000
2	Charlie	28	52000

- Column labels ("Name", "Age", "Salary") make data easy to read and manipulate.
- Row index (0, 1, 2) provides structured referencing.

行标签（（"姓名"、"年龄"、"收入"）使数据易于阅读和操作。）
行索引（0, 1, 2）提供了结构化的引用。

Series 系列

- A one-dimensional labeled array
- 一维带标签的数组
- Can hold any data type: integers, strings, floats, etc.
- **可以容**纳任何数据类型：整数、字符串、浮点数等
- Each element has a value and a label (index)
- **每个元素都有一个**值和一个标签（索引）
- Can be created from lists, dictionaries, or NumPy arrays
- **可以从列表、字典或**NumPy**数**组创建
- Similar to a single column in a spreadsheet or a list with labels
- 类似于电子表格中的单列或带标签的列表

	销售额 Sales
一月 Jan	250
二月 Feb	420
三月 Mar	390

dtype: int64

数据类型

DataFrame 数据框

- A two-dimensional labeled data structure
- 二维带标签的数据结构
- Similar to a spreadsheet or SQL table
- 类似于电子表格或SQL表
- Consists of rows and columns
- 由行和列组成
- Labeled axes: rows (index) and columns (column names)
- 带标签的轴：行（索引）和列（列名）
- Each column is a Series
- 每列都是一个系列
- Can be created from dictionaries, lists, NumPy arrays, or other DataFrames
- 可以从字典、列表、NumPy数组或其他数据框创建

		产品 Product	价格 Price	数量 Quantity
苹果	0	Apples	1.2	30
香蕉	1	Bananas	0.5	50
樱桃	2	Cherries	2.5	20

Loading Data into Pandas
加载数据到Pandas

Common Data Types in Datasets 数据集中的常见数据类型

Text: Emails, customer reviews, chat logs, social media posts.
文本：电子邮件、客户评价、聊天记录、社交媒体帖子。

Numbers: Statistics, financial transactions, measurements, sensor data.

数字：统计数据、金融交易、测量数据、传感器数据。

Categorical Data: Product categories, survey responses, customer segments.

分类数据：产品类别、调查回应、客户细分。

Date/Time: Timestamps, event logs, transaction dates.

日期/时间：时间戳、事件日志、交易日期。

Boolean: Yes/No, True/False, binary indicators.

布尔值：是/否、真/假、二进制指示器。

Common Data Formats. CSV 常见数据格式 CSV

- CSV (Comma-Separated Values) is a simple file format for storing tabular data
- CSV（**逗号分隔**值）是一种简单的文件格式，用于存储表格数据
- Each line represents a row, and columns are separated by commas
- **每行代表一行数据，列由逗号分隔**
- Commonly used for data exchange between applications
- **常用于**应用程序之间的数据交换
- Example: 举例

```
Name, Age, City
John, 28, Kyiv
Anna, 24, Lviv
```

```
姓名，年龄，城市
John，28，基辅
Anna，24，利沃夫
```

Common Data Formats. JSON　常见数据格式 JSON

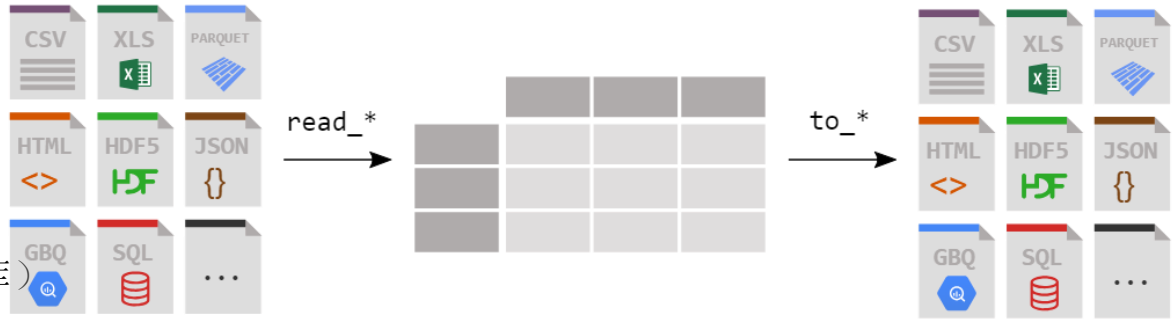
- JSON (JavaScript Object Notation) is a lightweight format for storing and exchanging data
- JSON（JavaScript对象表示法）是一种轻量级格式，用于存储和交换数据
- Uses key-value pairs and supports nested structures
- 使用键值对，并支持嵌套结构
- Commonly used in APIs and configuration files 常用于API和配置文件
- Example: 举例：

```
{
    "name": "John",
    "age": 28,
    "city": "Kyiv”
}
```

Loading Data Functions 加载数据函数

Common data sources:
常见数据源：

- CSV (.csv)
- Excel (.xlsx, .xls)
- SQL databases（SQL数据库）
- JSON (.json)



Functions to read data: 读取数据的函数：

- pd.read_csv()
- pd.read_excel()
- pd.read_sql()
- pd.read_json()

Pandas Practice
Pandas练习

Useful Links 实用链接

Pandas. Getting started 入门

Pandas. Intro to data structures 数据结构简介

Pandas. How do I read and write tabular data? 如何读写表格数据？