01



数据预处理
Data Preprocessing

2026.04

授课教师：[李阳源]

SL02:教学目标



今日学习目标

- 理解数据预处理 (Data Preprocessing) 的概念
- 了解数据预处理的必要性
- 理解数据预处理过程的整体流程
- 理解数据科学管道 (Data Science Pipeline) 中预处理的位置

SL03:讲座概述



讲座构成

1. 什么是数据预处理?
2. 为什么需要数据预处理?
3. 数据预处理中的常见问题
4. 数据预处理的阶段和过程
5. 数据科学管道中的位置
6. 数据预处理的实际案例
7. 总结与问答

SL04:数据预处理的定义



什么是数据预处理?

- 将原始数据 (Raw Data) 转换为适合分析的形式的过程
- 包括数据清洗 (Data Cleaning)、整合 (Integration)、转换 (Transformation)、降维 (Reduction) 等
- 实际分析工作的50-80%用于预处理过程
- "垃圾进，垃圾出"(Garbage In, Garbage Out) 的原则



SL05:数据预处理的必要性



为什么需要数据预处理?

- 现实数据是"不完美的"
  - 缺失值 (Missing Values)
  - 噪声 (Noise)
  - 异常值 (Outliers)
  - 重复 (Duplicates)
  - 不一致 (Inconsistencies)
- 需要整合从各种来源收集的数据
- 满足分析算法的格式要求

SL06:数据质量问题



数据质量的常见问题



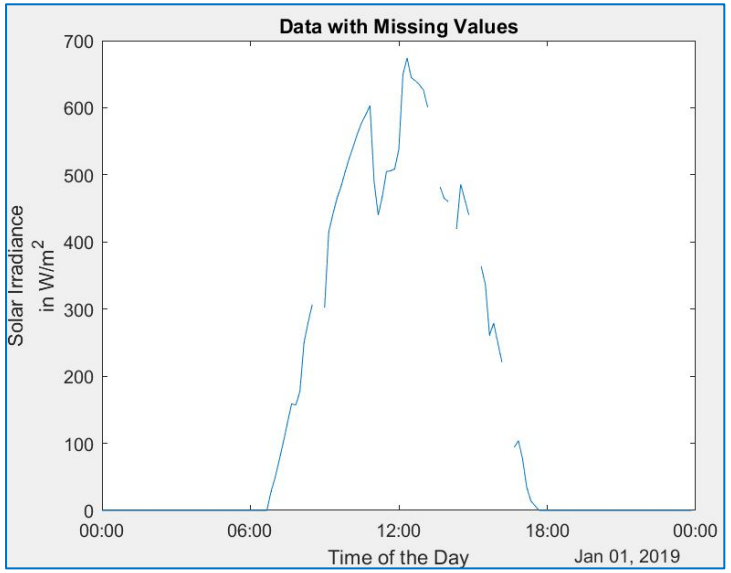
- 不完整性 (Incompleteness): 缺失值，属性缺失
- 噪声 (Noise): 错误，异常值
- 不一致性 (Inconsistency): 代码或名称不一致
- 时效性 (Timeliness): 缺乏最新性
- 重复 (Duplicates): 相同数据重复出现
- 偏差 (Bias): 采样或测量偏差

SL07:实际案例 - 缺失值



缺失值的实际例子

- 问卷调查: 受访者跳过部分问题
- 传感器数据: 因传感器临时故障导致数据丢失
- 医疗记录: 患者未接受特定检查
- 金融数据: 交易记录中部分信息缺失

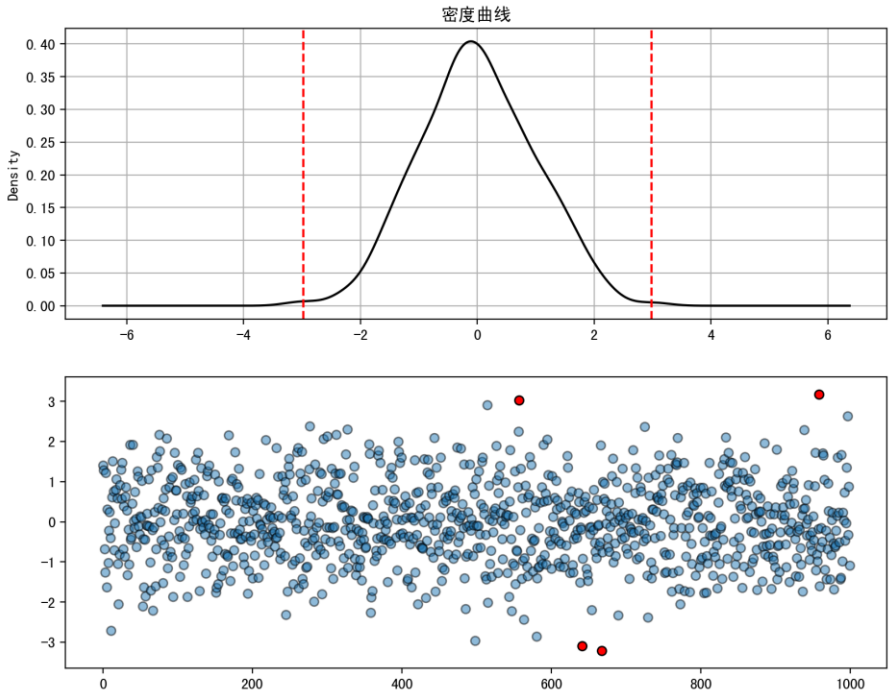


SL08:实际案例 - 异常值



异常值的实际例子

- 温度传感器: 记录到突然的温度急剧上升
- 收入数据: 报告极端高收入
- 网站流量: 异常访问量激增
- 身高/体重数据: 因输入错误导致极端值



SL09:数据预处理阶段



预处理的主要阶段

1. 数据收集 (Data Collection)
2. 数据检查 (Data Inspection)
3. 数据清洗 (Data Cleaning)
4. 数据转换 (Data Transformation)
5. 数据降维 (Data Reduction)
6. 数据整合 (Data Integration)
7. 数据验证 (Data Validation)

数据收集
↓
数据检查
↓
数据清洗
↓
数据转换
↓
数据降维
↓
数据整合
↓
数据验证



数据验证
出餐前最后试味

SL10:数据收集与检查(步骤1,2)



收集与检查阶段

- **数据收集**
  - ➢从各种来源(数据库、文件、网络、传感器等)获取数据
  - ➢了解格式和结构

- **数据检查**
  - ➢确认基本统计(平均值、中位数、最大值、最小值等)
  - ➢识别缺失值、异常值
  - ➢确认数据类型
  - ➢了解分布和相关性

SL11:数据清洗 (步骤3)



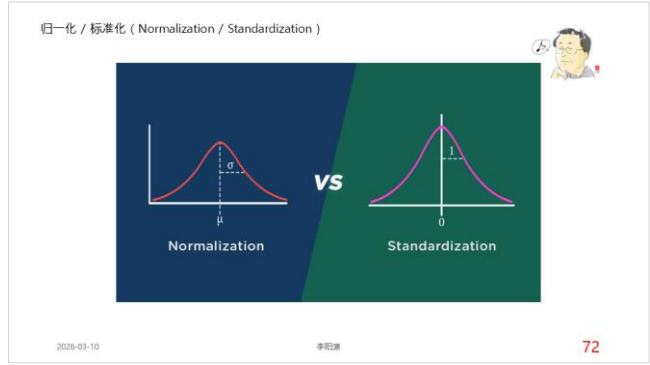
数据清洗 (Data Cleaning)

- 处理缺失值
  - ➢删除 (Deletion)
  - ➢替换 (Imputation): 均值、中位数、众数、预测值等
- 处理异常值
  - ➢识别: 统计方法 (Z-score, IQR等)
  - ➢处理: 移除、转换、单独分析
- 去除重复
- 修正错别字和不一致

SL12:数据转换 (步骤4)



数据转换 (Data Transformation)



- 归一化/标准化 (Normalization/Standardization)
  - Min-Max归一化
  - Z-score标准化
- 对数转换 (Log Transformation)
- 离散化 (Discretization)
- 分类数据编码 (Categorical Encoding)
  - 标签编码 (Label Encoding)
  - 独热编码 (One-Hot Encoding)
- 特征工程 (Feature Engineering)

SL13:数据降维与整合(步骤5,6)



数据降维与整合

- **数据降维 (Data Reduction)**
  - 维度降低 (Dimensionality Reduction)
  - 特征选择 (Feature Selection)
  - 采样 (Sampling)
  - 聚合 (Aggregation)

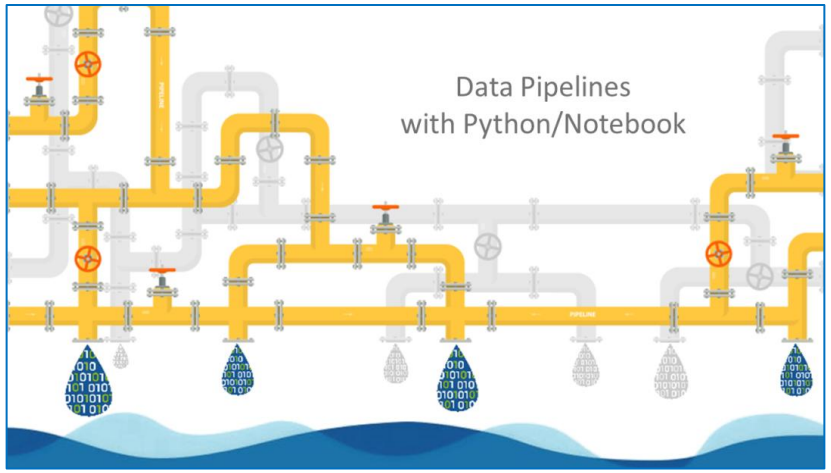
- **数据整合 (Data Integration)**
  - 合并不同来源的数据
  - 模式整合 (Schema Integration)
  - 实体识别 (Entity Identification)
  - 重复检测与解决

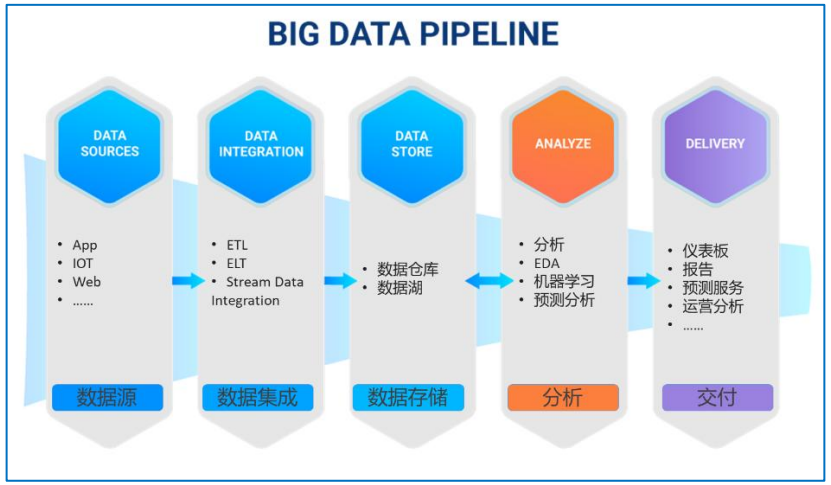
SL14:数据科学管道(PipeLine)



在数据科学管道中的位置

1. 问题定义 (Problem Definition)
2. 数据收集 (Data Collection)
3. **数据预处理 (Data Preprocessing)**
4. 探索性数据分析 (Exploratory Data Analysis)
5. 建模 (Modeling)
6. 评估与解释 (Evaluation & Interpretation)
7. 部署与监控 (Deployment & Monitoring)



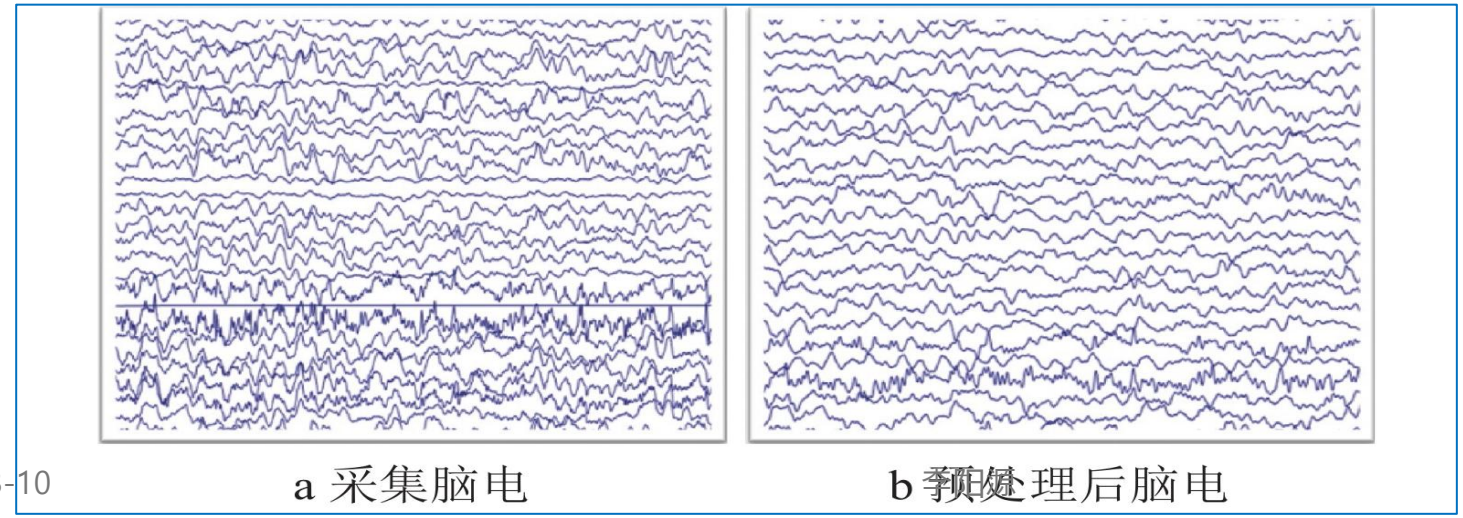


SL15:预处理的重要性



预处理对分析结果的影响

- 提高模型性能: 增加准确性和可靠性
- 提高计算效率: 缩短处理时间
- 增加发现有意义模式的可能性
- 防止得出错误结论



脑电预处理前后波形

SL16:实际案例研究 1



案例研究: 客户数据分析

- **问题**: 构建客户流失预测模型
- **原始数据问题**:
  - 20%的客户年龄信息缺失
  - 支出数据中存在极端异常值
  - 服务使用时长不一致
- **预处理方法**:
  - 年龄缺失值 → 基于其他特征预测
  - 异常值 → 使用IQR方法识别和调整
  - 使用时长 → 统一为标准格式

ID	Type	Value
100	A	2374
101	B	
102	C	24983

→ 결측치

SL17:实际案例研究 2



案例研究: 传感器数据分析

- **问题**: 设备故障预测模型
- **原始数据问题**:
  - 因传感器间歇性故障导致缺失值
  - 不同时间段数据不平衡
  - 使用多种单位
- **预处理方法**:
  - 缺失值 → 使用移动平均插值
  - 数据不平衡 → 重采样
  - 单位 → 转换为标准单位

SL18:预处理工具与库



数据预处理工具

- **Python库**:
  - Pandas: 数据操作与分析
  - NumPy: 数值计算
  - Scikit-learn: 预处理功能
  - Imbalanced-learn: 不平衡数据处理
- **R包**:
  - dplyr, tidyr: 数据操作
  - mice: 处理缺失值
- **其他工具**:
  - Apache Spark: 大规模数据处理
  - Tableau Prep: 可视化数据预处理(GUI)

1. 数据预处理的定义是什么?

A. 将分析结果可视化呈现的过程

B. 将原始数据转换为适合分析形式的过程

C. 仅对数据进行统计描述的过程

D. 将模型输出结果进行评估的过程

2.「GIGO」原则在数据预处理中指的是什么?

A. 好的输入产生好的输出，因此无需预处理

B. 数据量越大，分析结果越准确

C. 垃圾数据输入，结果也是垃圾；强调数据质量的重要性

D. 图形界面操作比代码更安全

3. 实际数据分析工作中，数据预处理大约占用多少比例的时间?

A. 10–20%

B. 20–30%

C. 30–50%

D. 50–80%

4. 以下哪项不属于数据质量问题?

A. 缺失值（Missing Values）

B. 数据可视化（Data Visualization）

C. 重复数据（Duplicates）

D. 噪声（Noise）

5. 下列哪个场景最不可能产生缺失值?

A. 问卷调查中受访者跳过某些问题

B. 传感器因故障导致数据中断

C. 对完整数据集进行Z-score标准化

D. 患者未接受某项特定检查

6. 以下哪种方法属于识别异常值的统计方法?

A. One-Hot 编码

B. IQR（四分位距）方法

C. Min-Max 归一化

D. 模式整合（Schema Integration）

7. 在数据预处理的主要阶段中，「数据检查」（Data Inspection ）紧接在哪个阶段之后?

A. 数据验证（Data Validation ）

B. 数据清洗（Data Cleaning ）

C. 数据收集（Data Collection ）

D. 数据整合（Data Integration ）

8. 处理缺失值的「替换（Imputation）」方法中，以下哪项不是常见的替换策略?

A. 用均值替换

B. 用众数替换

C. 用Z-score替换

D. 用预测值替换

9. Min-Max 归一化的主要目的是什么?

A. 消除数据中的噪声

B. 将数据映射到指定范围（如0到1），消除量纲差异

C. 将类别变量转换为数值型

D. 减少数据集的特征维度

10. 独热编码（One-Hot Encoding）适用于以下哪种情况?

A. 处理连续型数值特征

B. 将无序类别变量转换为数值表示

C. 对时间序列数据进行插值

D. 压缩高维数据的维度

11. 特征选择（Feature Selection）属于数据预处理的哪个阶段?

A. 数据清洗（Data Cleaning）

B. 数据收集（Data Collection）

C. 数据降维（Data Reduction）

D. 数据验证（Data Validation）

12. 在数据科学管道（Data Science Pipeline）中，数据预处理位于哪个位置?

A. 问题定义之前

B. 数据收集之后、探索性数据分析之前

C. 建模之后、评估之前

D. 部署与监控阶段

13. R语言中的「mice」包主要用于解决什么问题?

A. 大规模数据的并行处理

B. 数据可视化

C. 缺失值的处理

D. 特征重要性的排序

14. 对数转换（Log Transformation）最适合用于以下哪种情况?

A. 数据呈均匀分布

B. 数据存在严重右偏（正偏态）且含有极端大值

C. 数据是类别型变量

D. 数据已完成归一化处理

15. 若某数据集因采样偏差（Sampling Bias）导致数据不均衡，最直接的影响是什么？

A. 增加数据集的特征数量

B. 导致模型对少数类的预测能力下降，产生偏颇结论

C. 使数据集出现大量重复记录

D. 提升模型的训练速度

16. 数据预处理对模型性能的影响中，以下哪项描述是错误的?

A. 提高模型预测的准确性和可靠性

B. 降低计算效率，增加处理时间

C. 增加发现有意义模式的可能性

D. 防止得出错误结论

17. 以下哪种情境最需要进行数据整合（Data Integration）？

A. 单一传感器产生的时间序列数据

B. 来自不同业务系统（如销售、客服、物流）的数据合并分析

C. 对已清洗的单一数据集进行归一化

D. 对数值型特征进行离散化处理

18. 在客户流失预测案例中，针对「服务使用时长不一致」的问题，最合适的预处理方法是?

A. 删除所有包含时长数据的记录

B. 用中位数替换所有时长值

C. 统一转换为标准格式/单位

D. 对时长数据进行独热编码

19. Python中的「Scikit-learn」库在数据预处理中主要用于什么?

A. 大规模分布式数据处理

B. 提供标准化、编码、特征提取等预处理功能

C. 专门处理不平衡数据集

D. 可视化数据预处理结果

20. 综合推断：如果跳过数据预处理直接建模，以下哪种后果最不可能发生？

A. 模型受异常值影响，准确率大幅下降

B. 缺失值导致部分算法无法正常运行

C. 模型训练速度明显加快，结果更精确

D. 不同量纲的特征使模型结果产生偏差

SL20:总结与下节课介绍



总结

- 数据预处理是将原始数据转换为可分析形式的过程
- 数据质量问题: 缺失值、异常值、不一致、重复等
- 主要预处理阶段: 收集、检查、清洗、转换、降维、整合
- 是数据科学管道中的关键阶段
- 良好的预处理可大幅提高分析结果质量

下节课预告

- 理解数据结构和类型
- 结构化/非结构化数据特性
- Pandas数据框架介绍

问答环节

- 有问题吗?


對酒當歌


對酒當歌





[01] 实践课程: Python基础实践环境搭建

PL02: 实践目标



今日实践目标

- 安装和配置 PyCharm IDE
- 配置 Python环境
- 安装数据预处理相关的必要库(pandas, numpy, scikit-learn)
- 进行简单的数据加载和检查实践

PL03: 为什么选择 PyCharm?



PyCharm IDE的优势

- JetBrains开发的专业Python开发环境
- 代码自动完成和语法高亮功能(code automation and syntax highlighting)
- 内置调试工具(Built-in debugging tools)
- 集成版本管理系统(**Integrated Version Control System**)
- 虚拟环境管理功能(**Virtual Environment Management**)
- 支持科学工具 (**Scientific Computing Tools Support**)

Python 安装





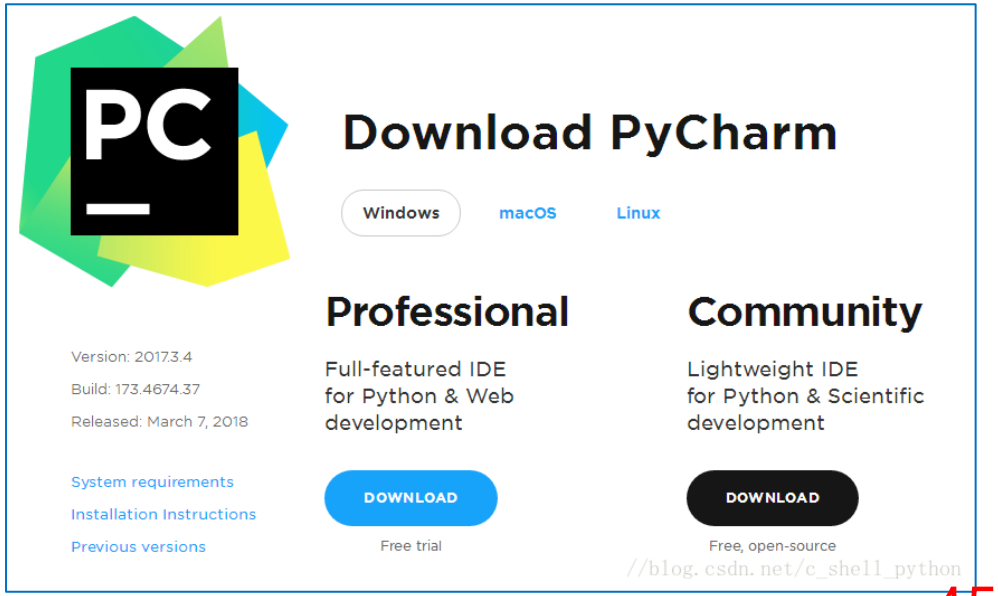
python-3.13.2-amd64.exe

PL04:PyCharm安装 - Windows



在Windows上安装PyCharm

1. 访问JetBrains网站: https://www.jetbrains.com/pycharm/download/
2. 选择Community Edition(免费)或Professional Edition(付费)
3. 下载并运行安装文件
4. 按照安装向导指引进行操作
5. 从开始菜单运行 PyCharm

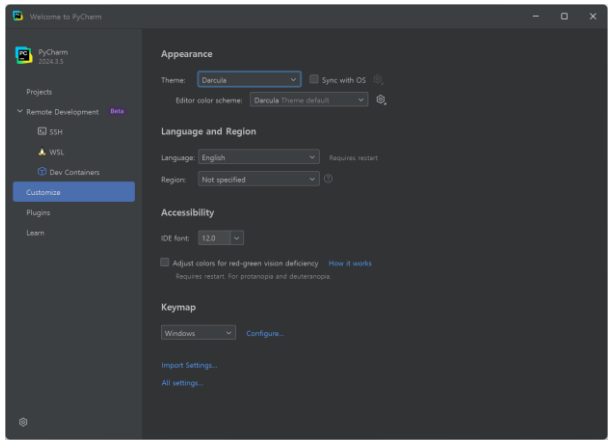


//blog.csdn.net/c_shell_python

PL05: PyCharm首次运行和设置



PyCharm初始设置



1. 首次运行时选择主题 (Light/Darcula)
2. 决定插件安装 (推荐: Markdown, CSV Plugin)
3. 创建新项目 (Project > New Project)
4. 指定项目位置
5. 设置虚拟环境 (New environment using: Virtualenv)
6. 选择Python解释器 (推荐Python 3.13或更高版本)

PL06: 理解虚拟环境(Virtual Environment)



什么是Python虚拟环境 (Virtual Environment)?

- 独立的Python运行环境
- 可以管理项目特定的包依赖
- 防止冲突并提供可重现的环境
- 主要工具: venv, virtualenv, conda

使用虚拟环境的原因

- 防止项目间的包冲突
- 保持部署(bùshǔ)环境的一致性
- 仅安装必要的包以保持轻量化

PL07: 在PyCharm中设置虚拟环境



创建和管理虚拟环境

1. 创建新项目时选择"New environment using"
2. 为现有项目添加: File > Settings > Project > Python Interpreter
3. 添加解释器 (Add Interpreter) > Add Local Interpreter
4. 选择 Virtualenv Environment并指定位置
5. 决定是否安装基本包

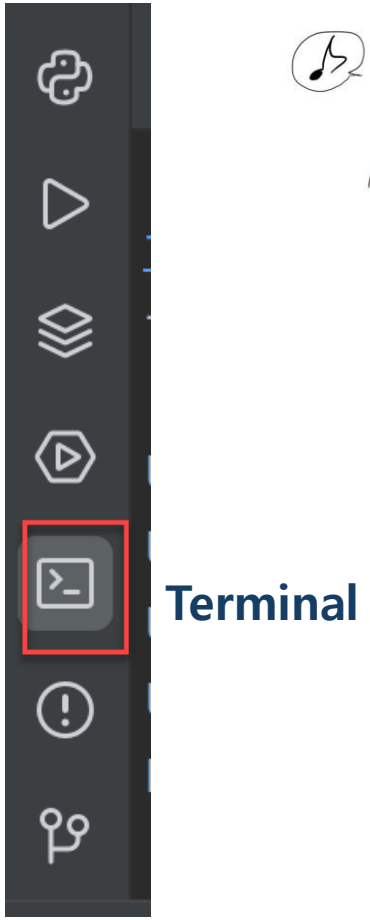
```
# 在终端中激活虚拟环境的方法（参考）
# Windows
.\venv\Scripts\activate
```

PL08: 安装必要库(library)



数据预处理相关主要库

- **NumPy**: 数值计算库
- **Pandas**: 数据分析和操作库
- **Matplotlib**: 数据可视化库
- **Seaborn**: 高级可视化库
- **Scikit-learn**: 机器学习和预处理库




```
# 在PyCharm终端中安装
pip install numpy pandas matplotlib seaborn scikit-learn
```

- Windows : C:₩Users₩사용자이름₩pip₩pip.ini 파일을 생성



- Pip.ini

```
[global]
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.추
```

pip install numpy pandas matplotlib seaboen scikit-learn -v

Off-line install: pip install --no-index --find-links ./my_packages numpy

pip list

pip install –no-index –find-links=my_packages numpy pandas matplotlib seaborn scikit-learn

pip list

PL09: 通过PyCharm安装包



通过图形界面安装包

1. File > Settings > Project > Python Interpreter
2. 点击"+"按钮
3. 在搜索框中输入所需库名称
4. 选择版本后点击"Install Package"
5. 安装完成后点击OK

![여기는 설명만 하고 통과!!]

PL10: 创建第一个Python脚本



创建新的Python文件:

1. 在项目面板中右击
2. 选择New > Python File
3. 输入文件名 (例如: data_preprocessing_basics.py)
4. 编写基础代码

PL11: 运行代码的方法



在PyCharm中运行代码

1. 在编辑器中编写代码
2. 运行方法:
   ① 右击 > Run 'filename.py'
   ② 点击绿色运行按钮
   ③ 使用Shift + F10快捷键
3. 结果显示在底部Run窗口
4. 调试: 点击行号旁边设置断点后以调试模式运行 (Shift + F9)

PL12: 简单数据创建和检查



1. 使用NumPy创建数据
2. File name: numpy_basics.py

PL13: Pandas基本用法



1. 创建和检查Pandas DataFrame
2. Pandas_basics.py

PL14: 下载样本数据文件 (sample data file)



准备实践数据

1. 公开数据集下载网站:
   ① Kaggle (https://www.kaggle.com/datasets)
   ② UCI ML Repository (https://archive.ics.uci.edu/ml/index.php)
   ③ 公共数据门户 (因地区不同而异)
2. 简单样本数据集:
   ① Iris数据集 (花卉分类)
   ② Titanic数据集 (生存预测)
   ③ Boston Housing数据集 (房价预测)
3. 下载后在项目文件夹中创建'data'文件夹并保存

PL15: 加载CSV文件



1. 读取和检查CSV文件
2. DataFilename: iris.csv



鸢尾花 (Yuānwěi huā)

영어 (English)	중국어 (简体中文)	병음 (Pinyin)	한국어
Iris	鸢尾花	Yuānwěi huā	붓꽃
Sepal	花萼	Huā'è	꽃받침
Petal	花瓣	Huābàn	꽃잎
Stem	花茎	Huājīng	꽃줄기
Leaf	叶子	Yèzi	잎

Iris 品种



영어	중국어	병음	한국어
Setosa	山鸢尾	Shān yuānwěi	세토사
Versicolor	杂色鸢尾	Zásè yuānwěi	버시컬러
Virginica	维吉尼亚鸢尾	Wéijíníyà yuānwěi	버지니카

特征



영어	중국어	병음	한국어
Sepal length	花萼长度	Huā'è chángdù	꽃받침 길이
Petal width	花瓣宽度	Huābàn kuāndù	꽃잎 너비
Flower color	花色	Huāsè	꽃색깔
Blooming season	花期	Huāqī	개화기

PL16: 数据基本探索



1. 数据探索基本命令

PL17: 简单数据可视化



1. 基本可视化代码
2. File name: visualization.py

PL18: 综合实践: 数据基础分析



1. 综合应用所学内容
2. File name:comprehensive_example.py

PL21: 总结与下节课介绍



总结:

- ✓ 今天我们学习了如何设置PyCharm IDE环境
- ✓ 创建并配置了Python虚拟环境
- ✓ 安装了数据分析和预处理所需的基本库
- ✓ 学习了基本的数据加载和探索技术
- ✓ 进行了简单的数据可视化

PL22: 总结与下节课介绍



下节课预告

- Python基础复习
- Pandas和NumPy核心功能
- 更多数据加载和处理方法

问答环节

✓ 有关于今天内容的问题吗?



1. cm：**厘米** (límǐ)
2. kg 千克 (qiānkè)
3. File 文件
4. 한근(500 g)：**1斤** (jīn)


米村®拌饭
米村
拌饭
CHINA





SL04:数据预处理的定义

什么是数据预处理?

- 将原始数据 (Raw Data) 转换为适合分析的形式的过程
- 包括数据清洗 (Data Cleaning)、整合 (Integration)、转换 (Transformation)、降维 (Reduction) 等
- 实际分析工作的50-80%用于预处理过程
- "垃圾进，垃圾出"(Garbage In, Garbage Out) 的原则

GIGO

1. 缺失值（Missing Value）：有属性，但部分样本没值



Table 1

	x1	x2	x3
0	1.0	9.0	1
1	NaN	NaN	2
2	2.0	9.0	3
3	3.0	9.0	4
4	NaN	9.0	5
5	4.0	9.0	6

2. 属性缺失（Missing Attribute）：整列不存在



1. 数据表结构本身缺少该属性

姓名	收入
A	5000
B	6000
C	7000

缺失值 vs 属性缺失 对比表

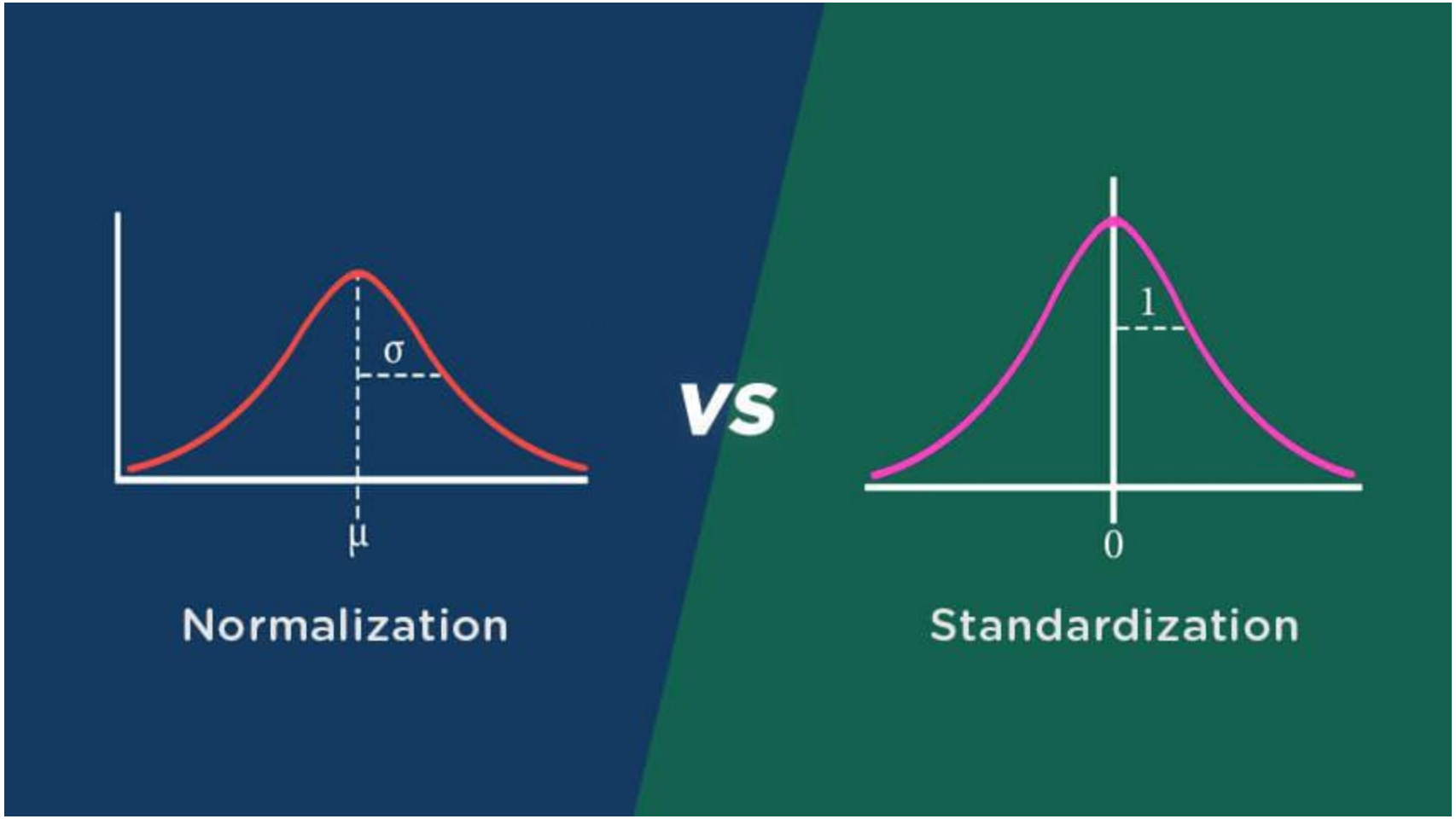
项目	缺失值（Missing Value）	属性缺失（Missing Attribute）
定义	在**已有属性**中，某些样本的取值为空或未记录	整个**属性本身不存在**于数据集中
数据层级	单**元格**层级（**cell level**）	变**量/特征**层级（**feature level**）
结构是否存在	属性存在，只是部分样本**没有**值	属性列本身不存在
表现形式	NaN、NULL、空值、？	数据表中**没有**该列
产生原因	数据采集失败、问卷漏答、传感器故障	数据设计时未收集该变量、不同数据源结**构**不同
对模型的影响	会影响模型训练，需要插补或删除处理	会导致特征空间不一致，需要特征对齐
处理方法	删除样本、均值/中位数插补、模型插补	特征工程（添加默认值、特征对齐、重新建模）
示例	年龄列存在，但某些人的年龄为空	数据集中根本**没有**“年龄”这一列




Garbage In,
质量差、不完整数据
创建质量差的特征
数据分析/机器学习
Garbage Out
错误的分析结论
性能差的机器学习模型

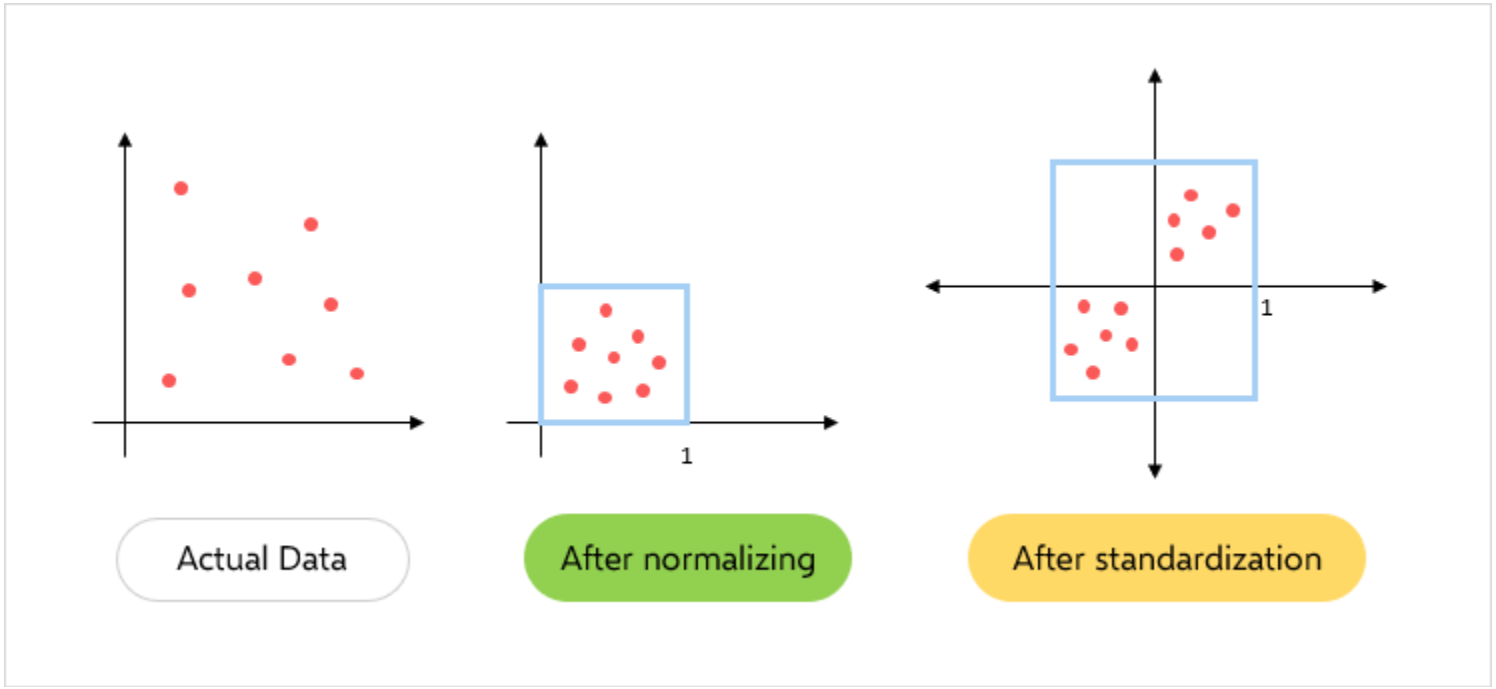
归一化 / 标准化（Normalization / Standardization）





归一化 / 标准化（Normalization / Standardization）

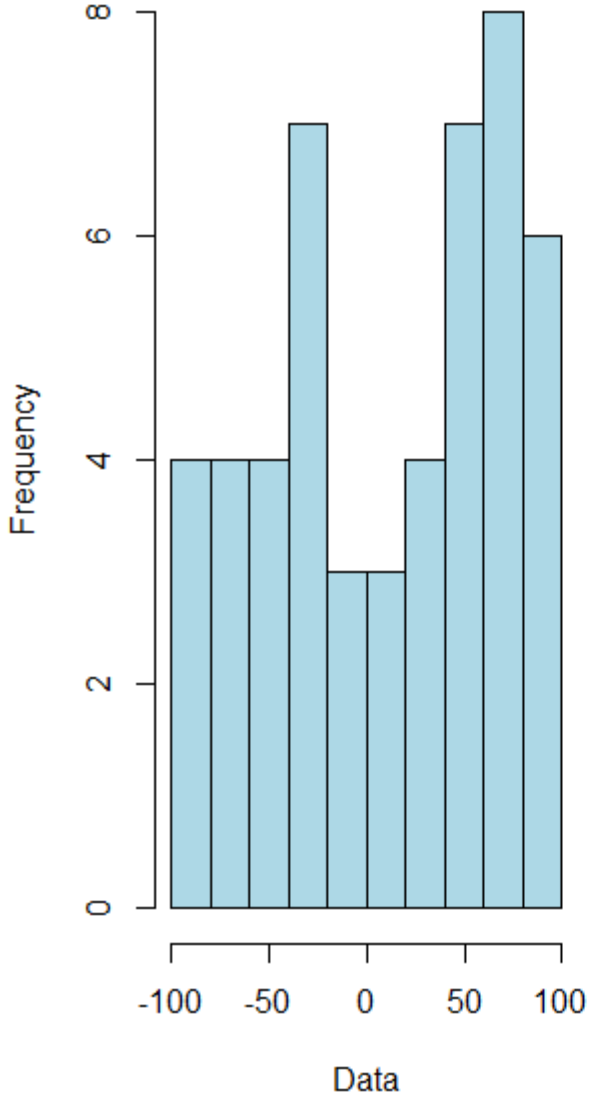


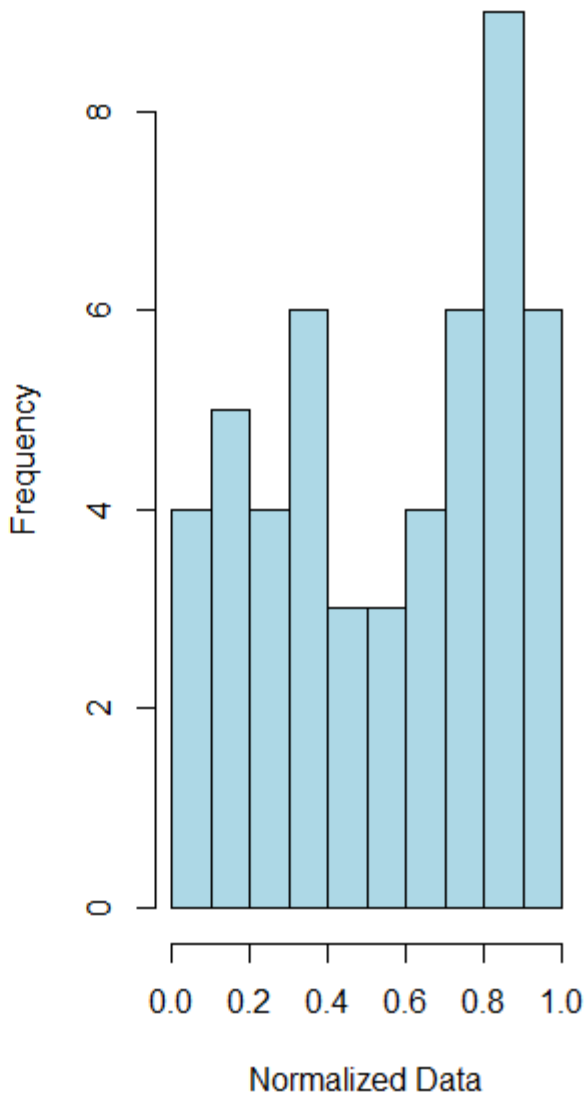




Min-Max 归一化




Frequency
0
2
4
6
8
-100
-50
0
50
100
Data


Frequency
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
Normalized Data

Min-Max 归一化

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$



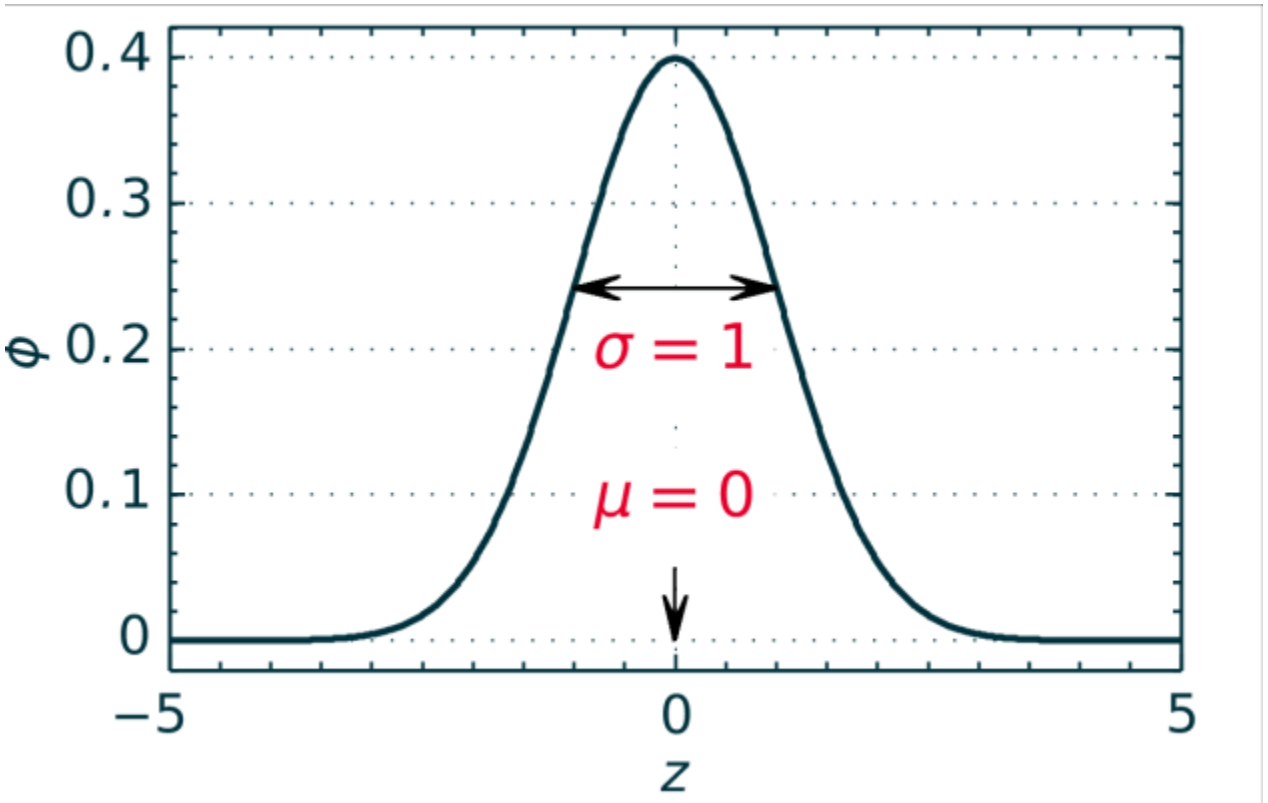
where:

- X is the original value,
- X_{min} is the minimum value of the feature,
- X_{max} is the maximum value of the feature,
- X' is the scaled value.



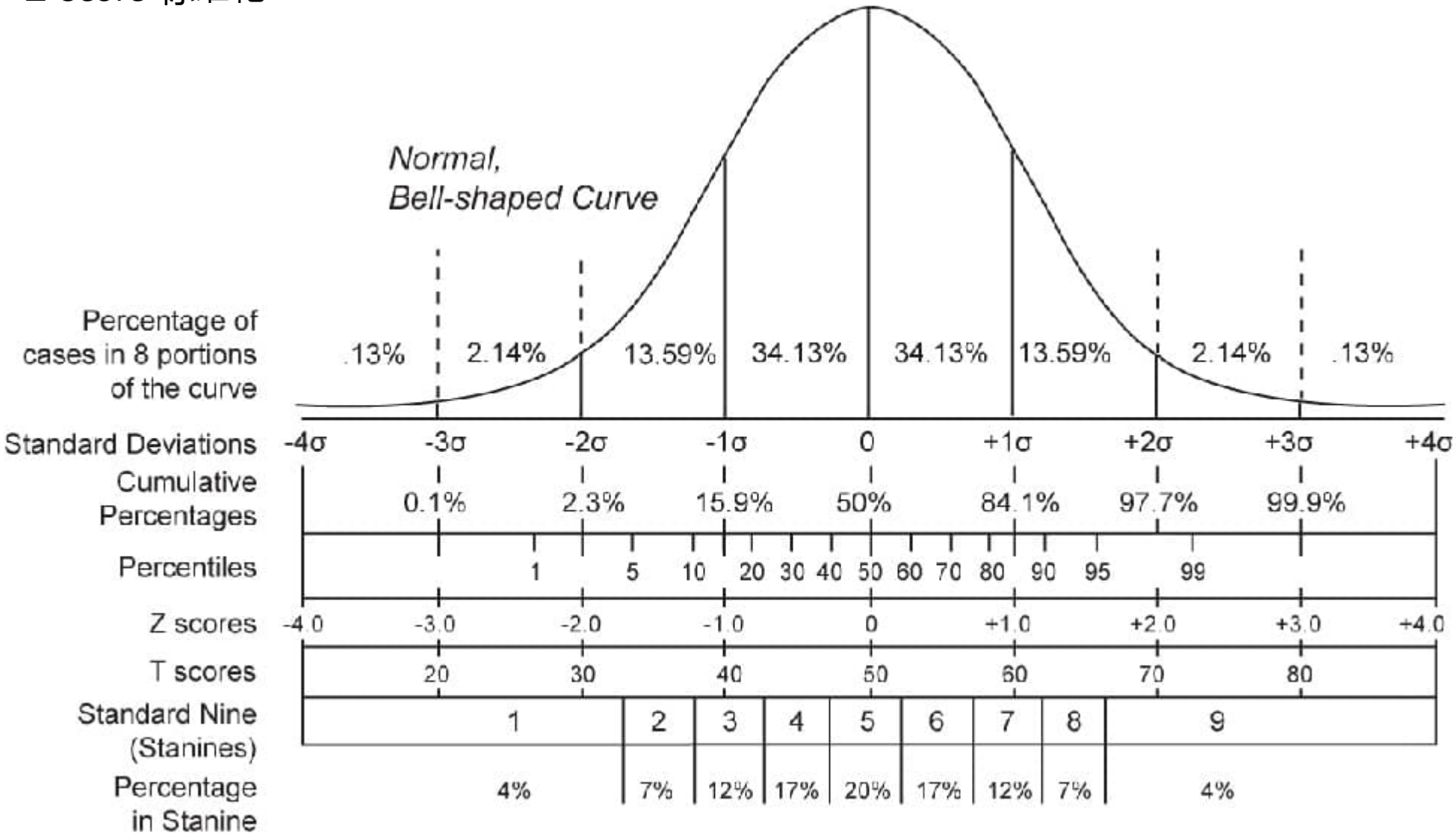
Z-score 标准化



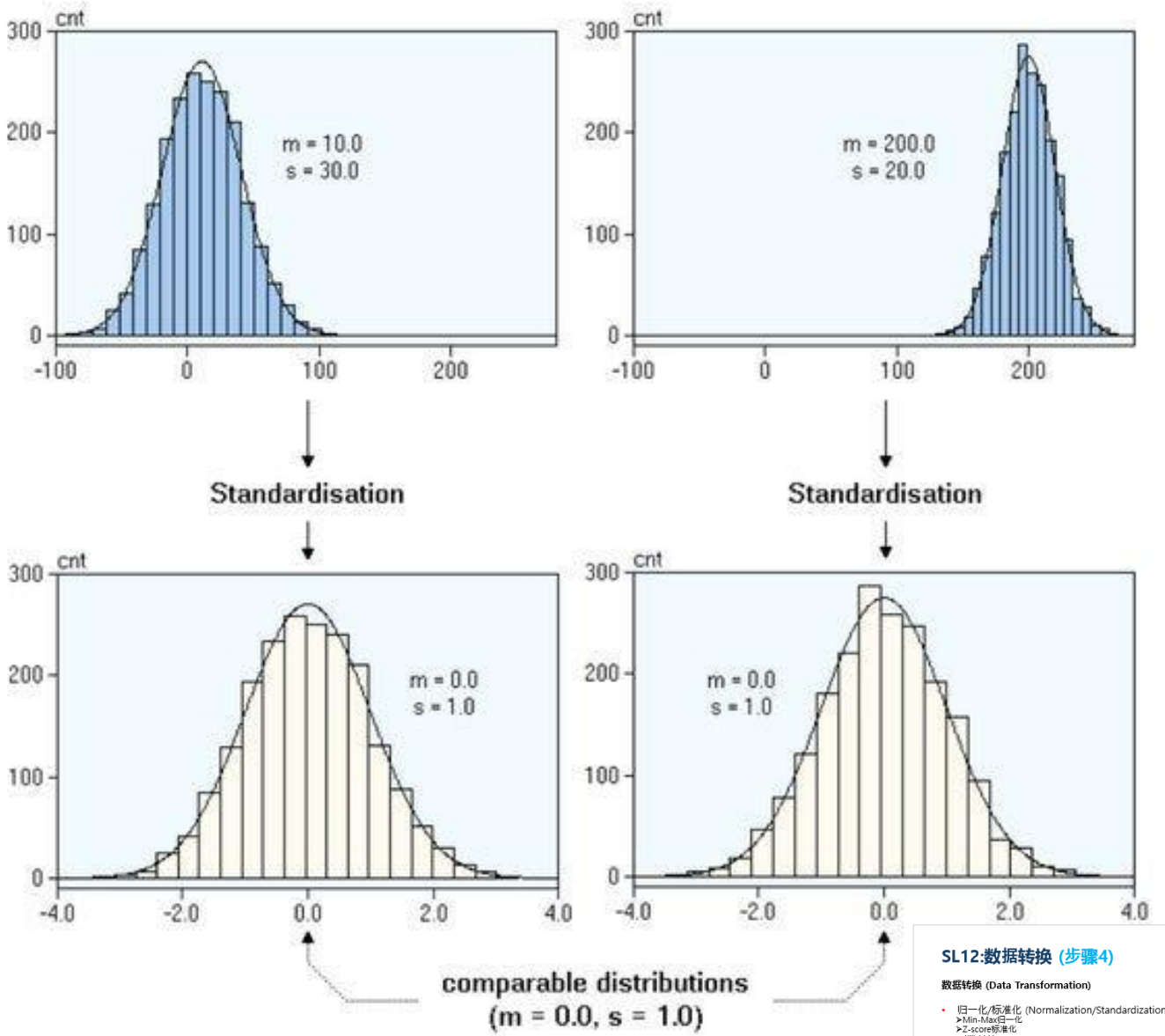


Z-score 标准化




Normal,
Bell-shaped Curve
Percentage of cases in 8 portions of the curve
.13%
2.14%
13.59%
34.13%
34.13%
13.59%
2.14%
.13%
Standard Deviations
-4σ
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ
+4σ
Cumulative Percentages
0.1%
2.3%
15.9%
50%
84.1%
97.7%
99.9%
Percentiles
1
5
10
20
30
40
50
60
70
80
90
95
99
Z scores
-4.0
-3.0
-2.0
-1.0
0
+1.0
+2.0
+3.0
+4.0
T scores
20
30
40
50
60
70
80
Standard Nine (Stanines)
1
2
3
4
5
6
7
8
9
Percentage in Stanine
4%
7%
12%
17%
20%
17%
12%
7%
4%

Z-score 标准化


cnt
m = 10.0
s = 30.0
m = 200.0
s = 20.0
Standardisation
Standardisation
m = 0.0
s = 1.0
m = 0.0
s = 1.0
comparable distributions
(m = 0.0, s = 1.0)

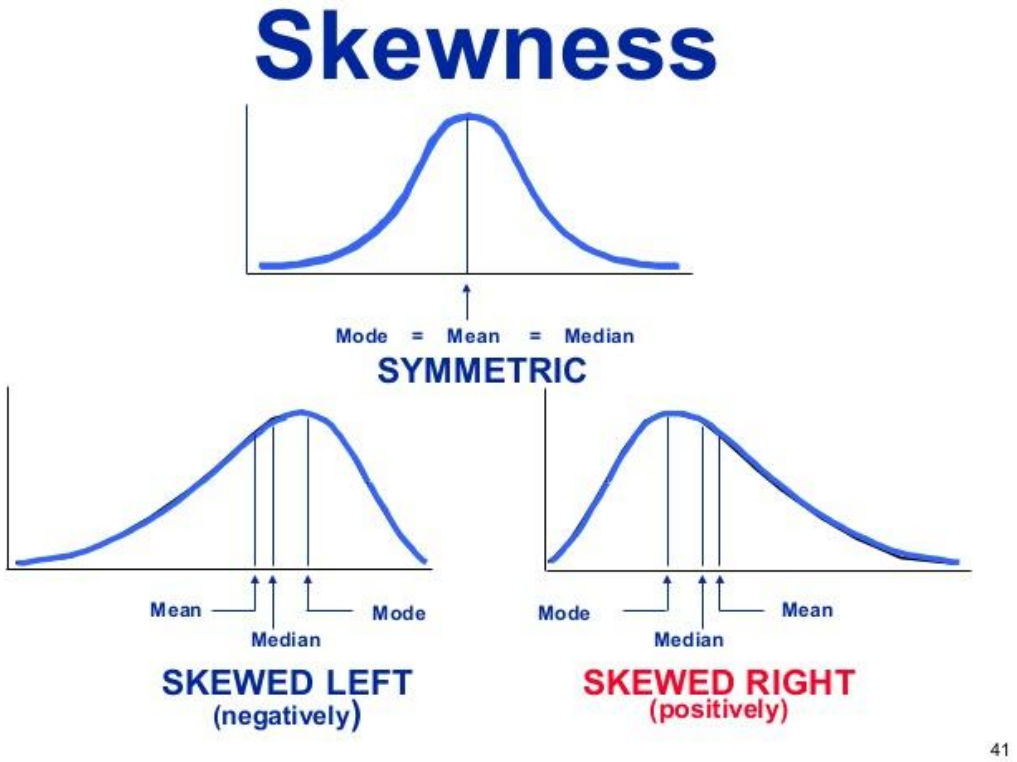



SL12:数据转换 (步骤4)
数据转换 (Data Transformation)
归一化/标准化 (Normalization/Standardization)
Min-Max归一化
Z-score标准化
对数转换 (Log Transformation)
离散化 (Discretization)
分类数据编码 (Categorical Encoding)
标签编码 (Label Encoding)
独热编码 (One-Hot Encoding)
特征工程 (Feature Engineering)
12

对数转换（Log Transformation）和 偏(Skewness)



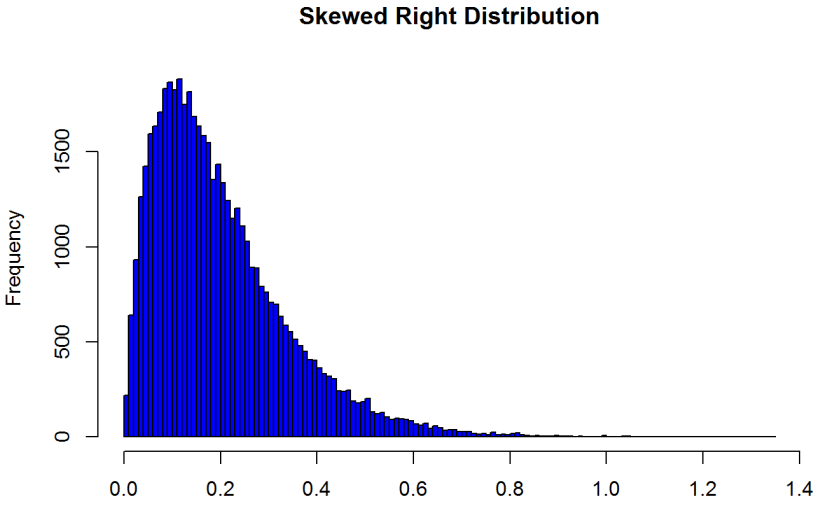
Right-skew (正偏, positive skew)　　Left-skew (负偏, negative skew)



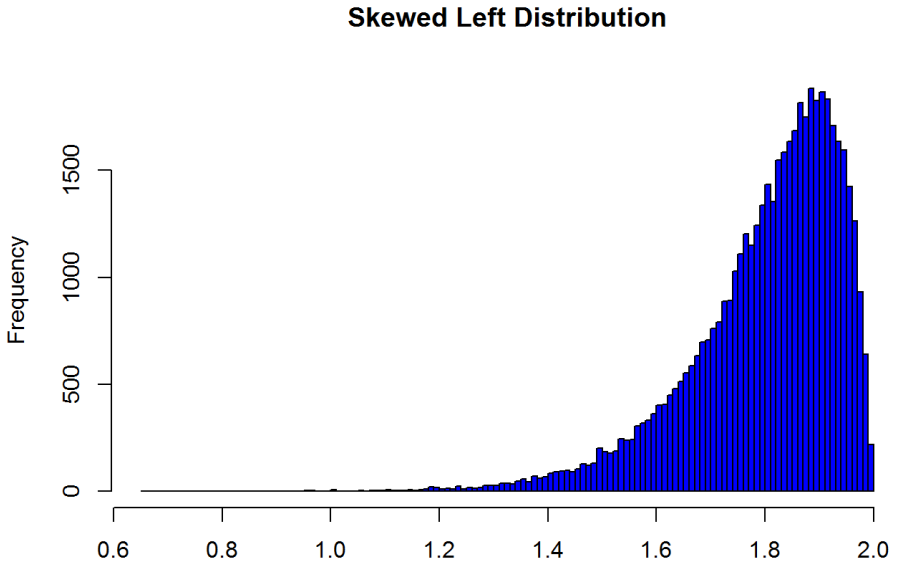
对数转换（Log Transformation）和 偏(Skewness)



Right-skew (正偏, positive skew)

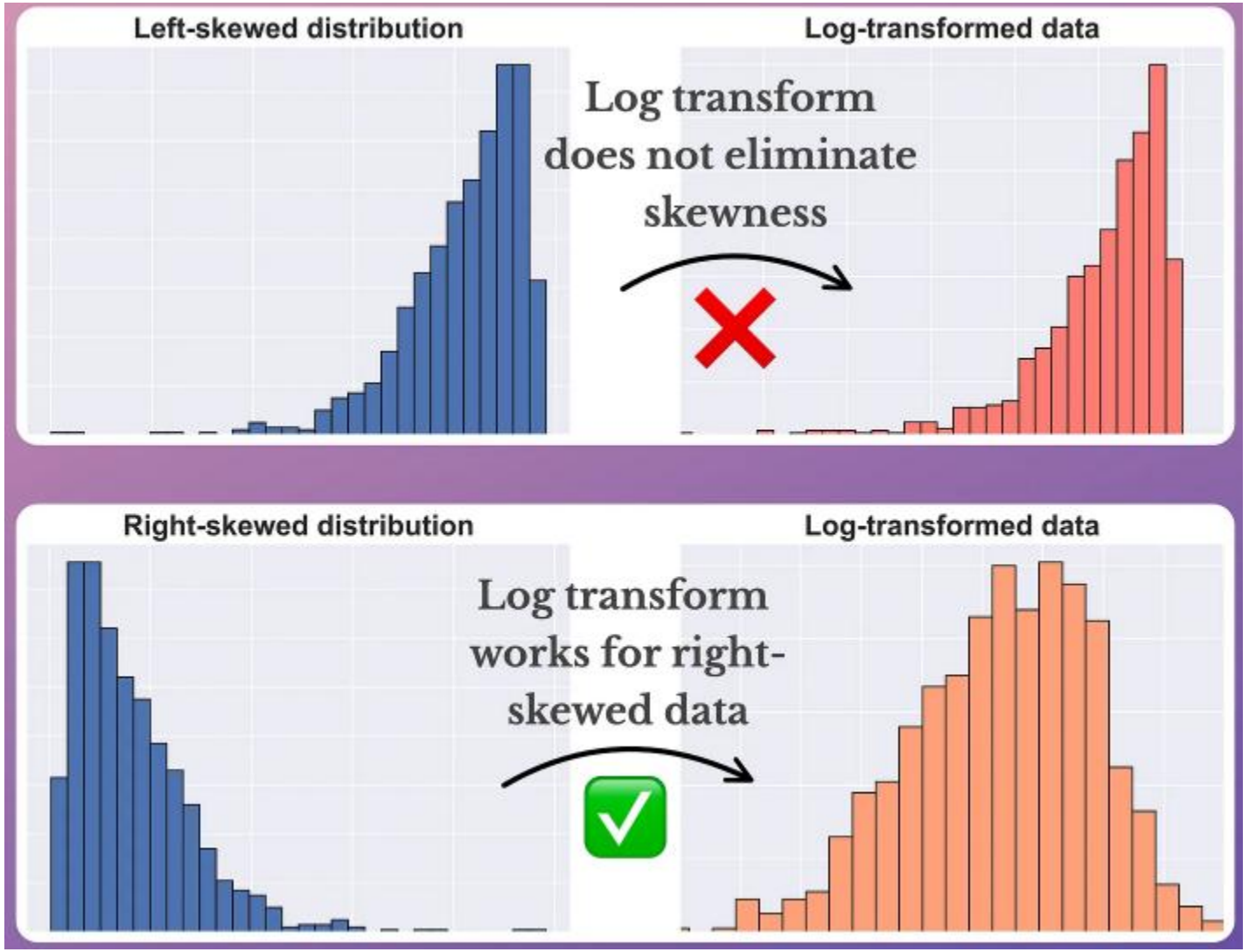


Left-skew (负偏, negative skew)



对数转换（Log Transformation）和 偏(Skewness)

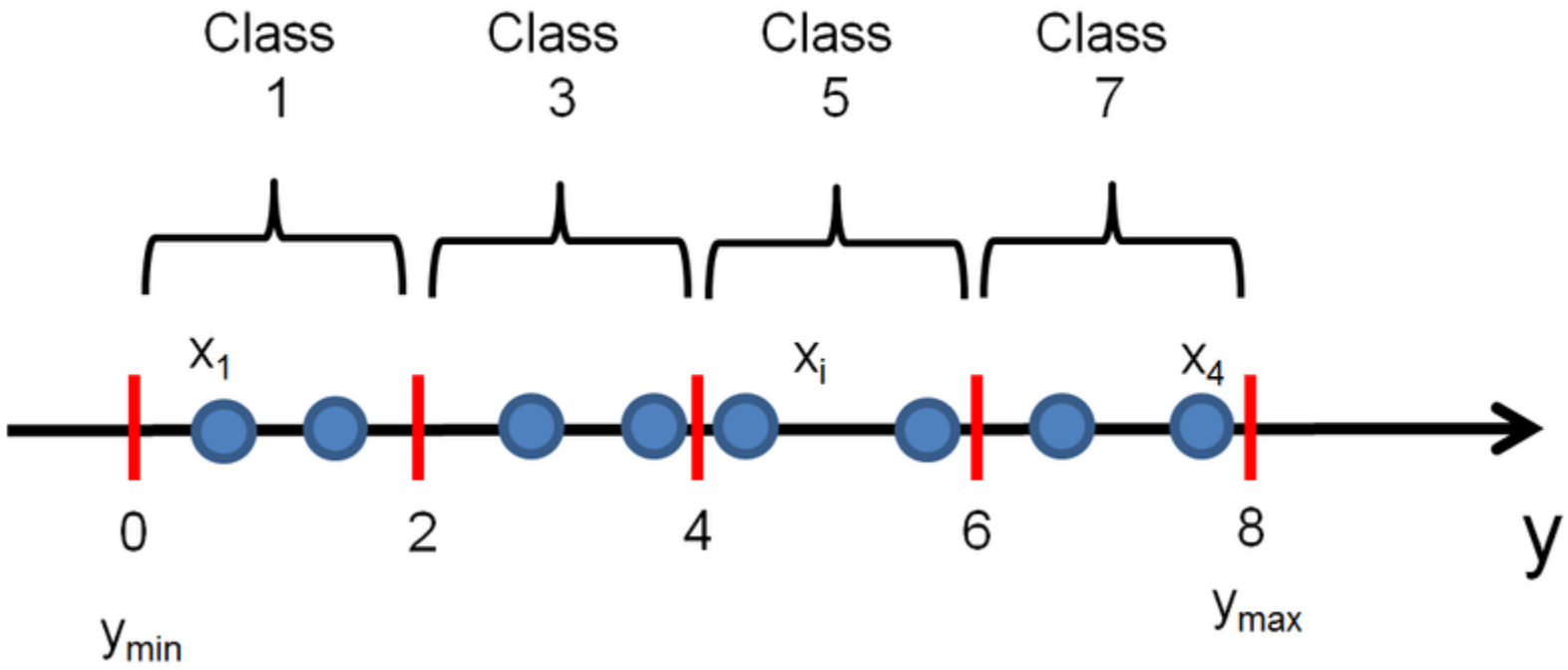






离散化（Discretization）






SL12:数据转换 (步骤4)

数据转换 (Data Transformation)

- 归一化/标准化 (Normalization/Standardization)
  - Min-Max归一化
  - Z-score标准化
- 对数转换 (Log Transformation)
- 离散化 (Discretization)
- 分类数据编码 (Categorical Encoding)
  - 标签编码 (Label Encoding)
  - 独热编码 (One-Hot Encoding)
- 特征工程 (Feature Engineering)

标签编码（Label Encoding）



Size
S
XS
L
M

Ordinal encoding →

Encoding
2
1
4
3

*data with order

独热编码（One-Hot Encoding）



id	color
1	red
2	blue
3	green
4	blue

One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

独热编码（One-Hot Encoding）



Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

