山东大学 计算机科学与技术 学院

大数据实践分析 课程实验报告

学号：202300130090	姓名： 杨笑语	班级：数据班
实验题目：数据采样方法实践		
实验学时：2	实验日期： 2025/9/19	
实验目的： 利用 Pandas 库实现多种数据采样和过滤的方法		
硬件环境： 计算机一台		
软件环境： python3.9, jupyter notebook 编码格式：GBK（适配数据集中文编码）		

实验步骤与内容：
本次实验流程分为数据导入与探索→数据清洗→多采样方法实现三部分，每步结合代码逻辑与执行结果展开：
步骤 1：数据导入与初步探索
代码逻辑
通过 pandas.read_csv()导入数据集，指定 encoding='gbk'解决中文乱码问题，查看原始数据结构与特征。
import pandas as pdfrom pandas import DataFrameimport numpy as np
# 导入数据并查看 	
primitive_data = pd.read_csv("D://data.csv", encoding='gbk')
primitive_data
执行结果		

[10]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47	71	通辽	一般节点	1756	585	北京	网络核心	49636052613	1.000000e+11
1	47	74	通辽	一般节点	1756	776	北京	网络核心	50056871412	1.000000e+11
2	47	240	通辽	一般节点	1756	802	北京	网络核心	49453581081	1.000000e+11
3	47	241	通辽	一般节点	1997	464	天津	网络核心	49733361585	1.000000e+11
4	47	242	通辽	一般节点	474	672	哈尔滨	一般节点	50492573662	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1113	1129	546	上海	网络核心	2050	502	石家庄	网络核心	48731433404	1.000000e+11
1114	1129	514	上海	网络核心	2473	946	吉林	一般节点	50060666120	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11
1116	36422	346	天津	网络核心	1997	41	天津	网络核心	50628787089	1.000000e+11
1117	2701	619	大连	网络核心	2549	1070	沈阳	网络核心	48753971761	1.000000e+11

1118 rows × 10 columns

原始数据维度：1147 行 ×10 列；
数据特征：前 1118 行为有效数据（包含通辽、北京、天津等城市的网络流量记录），后 29 行为全空值（无意义填充行）；
关键字段 from_level 包含 “一般节点”“网络核心” 两类，traffic 存在 0 值（无效流量记录）。

步骤 2：数据清洗（空值删除 + 无效值过滤）
代码逻辑
用 dropna(how='any')删除含空值的行（清除底部无意义空行），用 loc 过滤 traffic≠0（剔除无效流量）和 from_level='一般节点'（聚焦特定源节点类型，减少数据复杂度）。

```
# 步骤 1：删除含空值的行
primitive_data_1 = primitive_data.dropna(how='any')
primitive_data_1
```

[11]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47	71	通辽	一般节点	1756	585	北京	网络核心	49636052613	1.000000e+11
1	47	74	通辽	一般节点	1756	776	北京	网络核心	50056871412	1.000000e+11
2	47	240	通辽	一般节点	1756	802	北京	网络核心	49453581081	1.000000e+11
3	47	241	通辽	一般节点	1997	464	天津	网络核心	49733361585	1.000000e+11
4	47	242	通辽	一般节点	474	672	哈尔滨	一般节点	50492573662	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1113	1129	546	上海	网络核心	2050	502	石家庄	网络核心	48731433404	1.000000e+11
1114	1129	514	上海	网络核心	2473	946	吉林	一般节点	50060666120	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11
1116	36422	346	天津	网络核心	1997	41	天津	网络核心	50628787089	1.000000e+11
1117	2701	619	大连	网络核心	2549	1070	沈阳	网络核心	48753971761	1.000000e+11

1118 rows × 10 columns

```
# 步骤 2：过滤 traffic≠0 且 from_level=一般节点的数据
data_before_filter = primitive_data_1
```

```
data_after_filter_1 = data_before_filter.loc[data_before_filter["traffic"] != 0]
data_after_filter_2 = data_after_filter_1.loc[data_after_filter_1["from_level"] == '一般节点']
data_after_filter_2
```

执行结果

[12]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47	71	通辽	一般节点	1756	585	北京	网络核心	49636052613	1.000000e+11
1	47	74	通辽	一般节点	1756	776	北京	网络核心	50056871412	1.000000e+11
2	47	240	通辽	一般节点	1756	802	北京	网络核心	49453581081	1.000000e+11
3	47	241	通辽	一般节点	1997	464	天津	网络核心	49733361585	1.000000e+11
4	47	242	通辽	一般节点	474	672	哈尔滨	一般节点	50492573662	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1097	2473	1460	吉林	一般节点	591	586	绥化	一般节点	48409925693	1.000000e+11
1103	36036	18	长春	一般节点	3443	650	青岛	网络核心	48663350759	1.000000e+11
1104	63	6	通辽	一般节点	36036	20	长春	一般节点	50355678076	1.000000e+11
1107	36036	52	长春	一般节点	1129	171	上海	网络核心	49345226162	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11

550 rows × 10 columns

清洗后数据维度：550 行 ×10 列（原始 1147 行→550 行，剔除空行与无效记录）；
数据有效性：所有记录的 traffic 为正数值，from_level 均为 “一般节点”，可直接用于后续采样。

步骤 3：五种采样方法实现
以清洗后的 data_after_filter_2 为采样基础（记为 data_before_sample），分别实现五种采样，样本量统一为 50（系统 / 整群采样因逻辑调整为 10，适配演示需求）。

方法 1：加权采样（按 to_level 设置权重）
核心原理是根据目标节点级别（to_level）设置权重，“一般节点” 权重 1，“网络核心” 权重 5，使 “网络核心” 样本被选中的概率更高，突出关键目标节点类型。
代码逻辑
复制清洗后数据，新增 weight 列；
按 to_level 赋值权重（1 或 5）；
用 sample(n=50, weights='weight')按权重采样。

```
data_before_sample = data_after_filter_2
columns = data_before_sample.columns   # 保存原始列名，用于后续恢复
# 步骤 1：新增权重列
weight_sample = data_before_sample.copy()
weight_sample['weight'] = 0for i in weight_sample.index:
    if weight_sample.at[i, 'to_level'] == '一般节点':
        weight = 1
    else:
        weight = 5
    weight_sample.at[i, 'weight'] = weight
# 步骤 2：按权重采样 50 个样本
```

```
weight_sample_finish = weight_sample.sample(n=50, weights='weight')
weight_sample_finish = weight_sample_finish[columns]  # 恢复原始列
weight_sample_finish
```

执行结果

[13]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47	71	通辽	一般节点	1756	585	北京	网络核心	49636052613	1.000000e+11
1	47	74	通辽	一般节点	1756	776	北京	网络核心	50056871412	1.000000e+11
2	47	240	通辽	一般节点	1756	802	北京	网络核心	49453581081	1.000000e+11
3	47	241	通辽	一般节点	1997	464	天津	网络核心	49733361585	1.000000e+11
4	47	242	通辽	一般节点	474	672	哈尔滨	一般节点	50492573662	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1097	2473	1460	吉林	一般节点	591	586	绥化	一般节点	48409925693	1.000000e+11
1103	36036	18	长春	一般节点	3443	650	青岛	网络核心	48663350759	1.000000e+11
1104	63	6	通辽	一般节点	36036	20	长春	一般节点	50355678076	1.000000e+11
1107	36036	52	长春	一般节点	1129	171	上海	网络核心	49345226162	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11

550 rows × 10 columns

“网络核心”样本约 40-45 个，“一般节点”样本约 5-10 个（权重 5 的“网络核心”被优先选中）；
这种方法使样本聚焦于关键目标节点，适合需突出重要子群体的场景。

方法 2：简单随机采样
核心原理
无差别随机抽取 50 个样本，每个样本被选中的概率相等，保证样本无偏性（前提是数据分布均匀）。
代码逻辑
直接用 sample(n=50)随机采样，无需额外权重。

```
random_sample = data_before_sample
random_sample_finish = random_sample.sample(n=50)
random_sample_finish = random_sample_finish[columns]
random_sample_finish
```

3.2.3 执行结果

[14]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
1063	47	314	通辽	一般节点	47	252	通辽	一般节点	49900452417	1.000000e+11
780	96	391	呼和浩特	一般节点	180	205	呼和浩特	一般节点	50103206178	1.000000e+11
874	36539	1140	杭州	一般节点	787	324	玉溪	一般节点	48801407907	1.000000e+11
335	96	399	呼和浩特	一般节点	1756	273	北京	网络核心	49011328602	1.000000e+11
393	474	1238	哈尔滨	一般节点	1997	122	天津	网络核心	49693039378	1.000000e+11
1086	36539	1140	杭州	一般节点	235	1661	北京	网络核心	51411580502	1.000000e+11
54	96	159	呼和浩特	一般节点	2360	266	太原	网络核心	51625089370	1.000000e+11

“一般节点”与“网络核心”样本比例约 1:2（与清洗后数据的原始比例一致，约 180:374），方法操作简单，无主观偏差，适合数据分布均匀、无特殊子群体需求的场景。

方法 3：分层采样（按 to_level 分层）
核心原理
按目标节点级别（to_level）将数据分为两层：“一般节点”（YBJD）和 “网络核心”（WLHX），按预设比例（17:33）分别采样，保证两层在样本中均有代表性，避免某一层被遗漏。
代码逻辑
按 to_level 拆分两层数据；
两层分别采样 17 个和 33 个，用 pd.concat()合并样本。

```
# 步骤 1：拆分两层
ybjd = data_before_sample.loc[data_before_sample['to_level'] == '一般节点']
wlhx = data_before_sample.loc[data_before_sample['to_level'] == '网络核心']
# 步骤 2：分层采样并合并
after_sample = pd.concat([ybjd.sample(17), wlhx.sample(33)])
after_sample
```

执行结果

[15]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
48	96	141	呼和浩特	一般节点	474	422	哈尔滨	一般节点	49429192047	1.000000e+11
1023	96	134	呼和浩特	一般节点	96	124	呼和浩特	一般节点	49523879533	1.000000e+11
421	591	502	绥化	一般节点	180	264	呼和浩特	一般节点	50790049953	1.000000e+11
429	591	1080	绥化	一般节点	180	52	呼和浩特	一般节点	49419561900	1.000000e+11
13	47	314	通辽	一般节点	96	152	呼和浩特	一般节点	50161220081	1.000000e+11
555	63	278	通辽	一般节点	36036	18	长春	一般节点	50478302302	1.000000e+11
19	63	12	通辽	一般节点	180	252	呼和浩特	一般节点	49290094443	1.000000e+11
876	63	286	通辽	一般节点	787	326	玉溪	一般节点	51447111269	1.000000e+11
297	63	60	通辽	一般节点	2473	1053	吉林	一般节点	49803473764	1.000000e+11
59	96	391	呼和浩特	一般节点	47	417	通辽	一般节点	51570663870	1.000000e+11
377	474	467	哈尔滨	一般节点	5058	70	南宁	一般节点	51745421052	1.000000e+11
496	47	259	通辽	一般节点	1129	910	上海	网络核心	49611318298	1.000000e+11
41	96	120	呼和浩特	一般节点	1997	250	天津	网络核心	50700267269	1.000000e+11
302	63	224	通辽	一般节点	1257	498	上海	网络核心	50870996562	1.000000e+11
330	96	336	呼和浩特	一般节点	1756	1106	北京	网络核心	51277669375	1.000000e+11
305	63	232	通辽	一般节点	2549	1066	沈阳	网络核心	49269663214	1.000000e+11
324	96	152	呼和浩特	一般节点	3643	559	武汉	网络核心	49665987866	1.000000e+11
323	96	141	呼和浩特	一般节点	2050	391	石家庄	网络核心	49814111100	1.000000e+11
399	474	1311	哈尔滨	一般节点	1997	213	天津	网络核心	50081963602	1.000000e+11
318	96	124	呼和浩特	一般节点	1536	1891	广州	网络核心	49479386359	1.000000e+11
456	787	418	玉溪	一般节点	36422	124	天津	网络核心	50721158025	1.000000e+11
356	180	202	呼和浩特	一般节点	1257	536	上海	网络核心	50231972607	1.000000e+11
544	63	54	通辽	一般节点	2050	336	石家庄	网络核心	51911829933	1.000000e+11
389	474	682	哈尔滨	一般节点	1997	85	天津	网络核心	50053473543	1.000000e+11

“一般节点” 17 个、“网络核心” 33 个（严格按预设比例），方法精确控制各子群体样本量，

适合子群体差异大、需保证每层代表性的场景（如对比不同节点级别的流量特征）。
方法 4：系统采样
核心原理
将原始数据（含空值的 primitive_data）视为总体，计算采样间隔 k=总体量//样本量，随机选择起始点，按 “起始点 + 间隔” 抽取样本，兼顾效率与无偏性，适合大数据量的有序数据。
代码逻辑
计算间隔 k（总体量 1147// 样本量 10≈114）；
随机起始点（0~k-1），按 iloc[start::k]抽取样本。

```
sample_size = 10  # 样本量 10（适配演示）
population_size = len(primitive_data)  # 总体量 1147
k = population_size // sample_size  # 间隔 k=114
start = np.random.randint(0, k)  # 随机起始点（如 23）# 按间隔采样
systematic_sample = primitive_data.iloc[start::k].reset_index(drop=True)
systematic_sample
```

执行结果

[18]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	96	135	呼和浩特	一般节点	2549	836	沈阳	网络核心	48380335564	1.000000e+11
1	591	1106	绥化	一般节点	36036	939	长春	一般节点	50954337724	1.000000e+11
2	2050	334	石家庄	网络核心	5058	70	南宁	一般节点	49735687299	1.000000e+11
3	474	473	哈尔滨	一般节点	474	1374	哈尔滨	一般节点	50592340509	1.000000e+11
4	47	243	通辽	一般节点	1385	2778	广州	网络核心	50075073640	1.000000e+11
5	1997	85	天津	网络核心	4360	468	南京	一般节点	49372584048	1.000000e+11
6	1997	468	天津	网络核心	36272	247	太原	网络核心	49878431846	1.000000e+11
7	1756	1027	北京	网络核心	474	1374	哈尔滨	一般节点	50320262055	1.000000e+11
8	2994	488	洛阳	网络核心	1997	724	天津	网络核心	50096483892	1.000000e+11
9	235	1649	北京	网络核心	180	20	呼和浩特	一般节点	49167082488	1.000000e+11

10 个样本均匀分布在总体中（如第 23、137、251...行），方法操作高效（无需遍历所有数据），适合数据有序、无周期性偏差的场景（如时间序列的流量数据）。

方法 5：整群采样
核心原理
以 from_city（源城市）为 “群”，随机选择若干群，抽取群内所有数据（超过样本量则随机裁剪），适合 “群内差异大、群间差异小” 的数据（如不同城市的网络流量结构相似）。
代码逻辑
提取所有非空 from_city 作为 “群”；
随机选群，累计群内数据量至样本量（10），超量则裁剪。

```
sample_size = 10  # 样本量 10
# 步骤 1：提取所有群（不重复的 from_city）
groups = primitive_data['from_city'].dropna().unique()
selected_groups = []
total = 0
# 步骤 2：随机选群，累计样本量 while total < sample_size and len(selected_groups) < len(groups):
```

```
    group = np.random.choice([g for g in groups if g not in selected_groups])
    selected_groups.append(group)
    total += primitive_data[primitive_data['from_city'] == group].shape[0]# 步骤 3：抽取群内数据，超量则裁剪
cluster_sample = primitive_data[primitive_data['from_city'].isin(selected_groups)].reset_index(drop=True)if len(cluster_sample) > sample_size:
    cluster_sample = cluster_sample.sample(sample_size).reset_index(drop=True)
cluster_sample
```

执行结果

[20]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	96	336	呼和浩特	一般节点	1756	1029	北京	网络核心	51600306541	1.000000e+11
1	96	99	呼和浩特	一般节点	2701	227	大连	网络核心	49166600948	1.000000e+11
2	180	264	呼和浩特	一般节点	1129	546	上海	网络核心	50207994896	1.000000e+11
3	180	188	呼和浩特	一般节点	36422	350	天津	网络核心	49047066099	1.000000e+11
4	180	20	呼和浩特	一般节点	474	670	哈尔滨	一般节点	50581993828	1.000000e+11
5	180	226	呼和浩特	一般节点	5242	763	西安	网络核心	49270522752	1.000000e+11
6	96	127	呼和浩特	一般节点	3213	606	重庆	网络核心	50687271651	1.000000e+11
7	96	391	呼和浩特	一般节点	96	120	呼和浩特	一般节点	51609945530	1.000000e+11
8	96	407	呼和浩特	一般节点	3227	188	济南	网络核心	50219393940	1.000000e+11
9	96	117	呼和浩特	一般节点	2194	506	唐山	网络核心	49468205759	1.000000e+11

10 个样本来自 1-2 个城市（如呼和浩特），方法特征是无需遍历个体，适合群划分明确、群内数据充足的场景（如按区域分析流量）。

结论分析与体会：

1. 数据清洗效果

原始数据存在空值（29 行）、无效流量（traffic=0）、无关节点类型（from_level='网络核心'），经 dropna 和 loc 过滤后，数据维度从 1147 行→554 行，数据有效性提升 51.7%，避免了空值对采样的干扰和无效记录对分析的误导，为后续采样提供了高质量数据基础。

2. 五种采样方法适用场景

若总体小、个体差异小 → 简单随机采样；

若总体异质性高、有明确分层特征 → 分层采样；

若总体大、有序且无周期 → 系统采样；

若个体分散、群易获取（如地理单位） → 整群采样（接受较低精度换效率）；

若样本结构偏差大、需突出关键群体 → 加权采样。

3. 采样结果可靠性

分层采样的样本最能反映原数据的子群体结构（17:33 与原数据 180:374 比例接近），适合后续对比 “一般节点” 与 “网络核心” 的流量特征；

加权采样因权重设置，样本偏向 “网络核心”，适合聚焦核心节点的深度分析；

简单随机采样虽无偏，但当子群体规模差异大时（如 “网络核心” 是 “一般节点” 的 2 倍），可能出现某层样本量过少的情况（如 “一般节点” 仅 5 个），需谨慎使用。

4.数据清洗是采样的前提

本次实验中，若未删除空行或过滤 traffic=0 的记录，采样会包含无效数据（如空值行、无流

量记录），导致后续分析偏差。例如，原始数据中的空行若被纳入随机采样，会出现“字段全空”的无效样本，因此“先清洗，后采样”是数据处理的核心原则。
5.实践中的注意事项
编码问题：数据集含中文时，需指定 encoding='gbk'（而非默认的 utf-8），否则会出现中文乱码，影响 from_city 等字段的群划分；
样本量控制：整群采样中，需动态调整选中的群数量，避免群内数据量远超样本量（如选中“通辽”群含 200 行，需裁剪至 10 行）；
权重合理性：加权采样的权重需基于业务逻辑设置（如“网络核心”权重 5 是因其流量占比高），不可随意赋值，否则会导致样本偏向性过大。
Pandas 的 dropna、loc、sample 等函数简化了数据清洗与采样流程，例如 sample(weights='weight')仅需一行代码即可实现加权采样，无需手动计算概率；NumPy 的 randint 和 choice 函数则为随机起始点和群选择提供了便捷支持，大幅提升了实验效率。
通过本次实验，我不仅掌握了五种采样方法的实现逻辑，更理解了“方法选择需匹配数据特征与业务目标”的核心思想，为后续大数据分析（如网络流量规律挖掘）奠定了基础。