山东大学计算机科学与技术学院

大数据分析实践课程实验报告

学号：202300130098	姓名：马浩鑫	班级：23 级数据
实验题目：一、数据采样方法实践		
实验学时：2 课时	实验日期：	

实验目标：

利用 Pandas 库实现多种数据采样和过滤的方法

实验描述：

1.1 实验环境

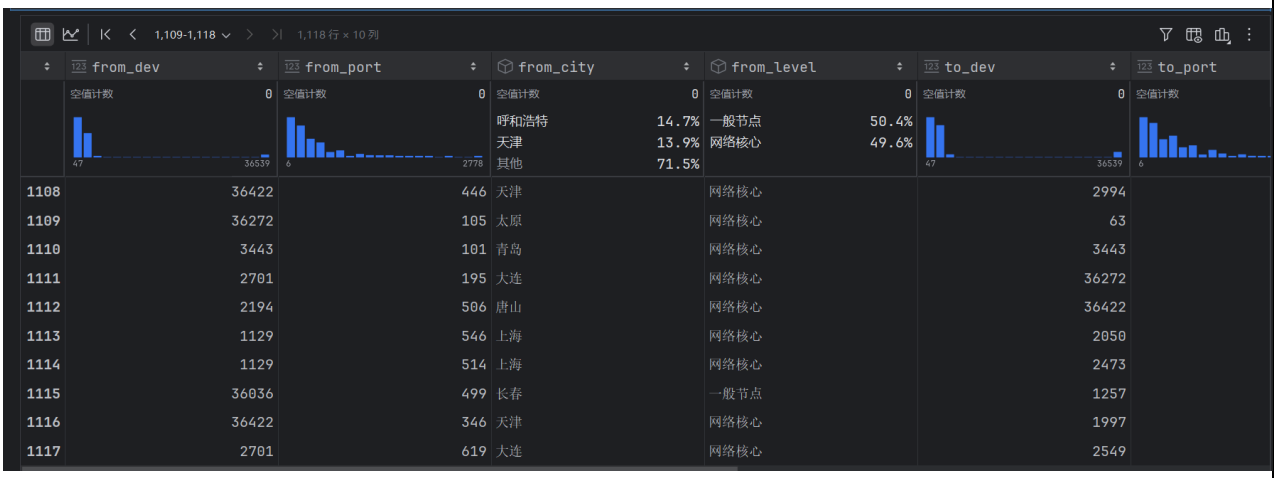
python3.9，jupyter notebook

1.2 数据集

数据集地址：http://storage.ameshol land.xyz/data.csv

1.3 实验步骤

（1）导入原始数据



1,109-1,118 1,118 行 × 10 列

	from_dev	from_port	from_city	from_level	to_dev	to_port
	空值计数 0	空值计数 0	空值计数 0 呼和浩特 14.7% 天津 13.9% 其他 71.5%	空值计数 0 一般节点 50.4% 网络核心 49.6%	空值计数 0	空值计数
1108	36422	446	天津	网络核心	2994	
1109	36272	105	太原	网络核心	63	
1110	3443	101	青岛	网络核心	3443	
1111	2701	195	大连	网络核心	36272	
1112	2194	506	唐山	网络核心	36422	
1113	1129	546	上海	网络核心	2050	
1114	1129	514	上海	网络核心	2473	
1115	36036	499	长春	一般节点	1257	
1116	36422	346	天津	网络核心	1997	
1117	2701	619	大连	网络核心	2549	

发现与实验指导书表述不同，原始数据已经过滤空行

（2）删除空行过滤，与（1）中结果一致，过滤得到 traffic 不等于 0 且 from_level=一般节点的数据，如下

```
#过滤得到traffic不等于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
```

✓ [7] 12毫秒

541-550 ⌄ 550 行 × 10 列

	from_dev	from_port	from_city	from_level	to_dev	to_port
空值计数	0	0	0	0	0	
	47 – 36539	6 – 1473	呼和浩特 29.3% 通辽 26% 其他 44.7%	一般节点 100%	47 – 36539	6
1073	47	417	通辽	一般节点	1756	
1075	4069	1196	宁波	一般节点	1756	
1079	63	224	通辽	一般节点	4069	
1086	36539	1140	杭州	一般节点	235	
1093	591	586	绥化	一般节点	1536	
1097	2473	1460	吉林	一般节点	591	
1103	36036	18	长春	一般节点	3443	
1104	63	6	通辽	一般节点	36036	
1107	36036	52	长春	一般节点	1129	
1115	36036	499	长春	一般节点	1257	

（3）各种抽样方法：

1. 加权抽样：to_level 的值为一般节点与网络核心的权重之比为 1 ： 5，共 50 个样本

41-50 ⌄ 50 行 × 10 列

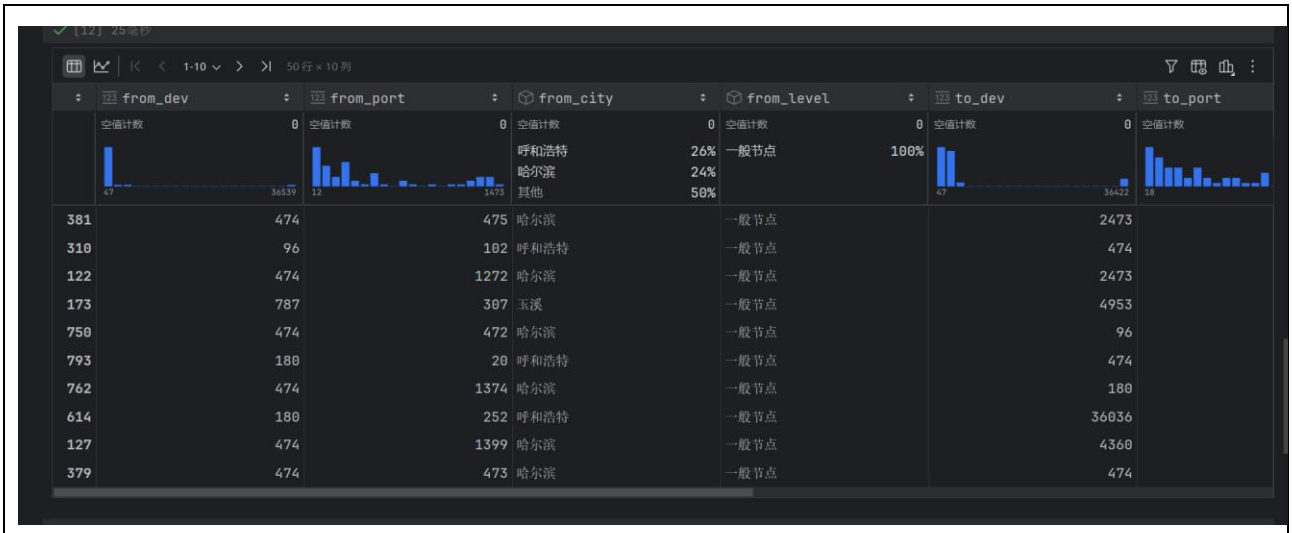
	from_dev	from_port	from_city	from_level	to_dev	to_port
空值计数	0	0	0	0	0	
	47 – 36036	6 – 1266	呼和浩特 30% 通辽 26% 其他 44%	一般节点 100%	47 – 36422	18
543	63	12	通辽	一般节点	2549	
380	474	474	哈尔滨	一般节点	3227	
544	63	54	通辽	一般节点	2050	
371	474	360	哈尔滨	一般节点	3227	
323	96	141	呼和浩特	一般节点	2050	
375	474	422	哈尔滨	一般节点	3213	
304	63	230	通辽	一般节点	3227	
547	63	62	通辽	一般节点	1756	
292	63	6	通辽	一般节点	2841	
136	591	19	绥化	一般节点	36036	

2. 随机抽样

1-10 ⌄ 50 行 × 10 列

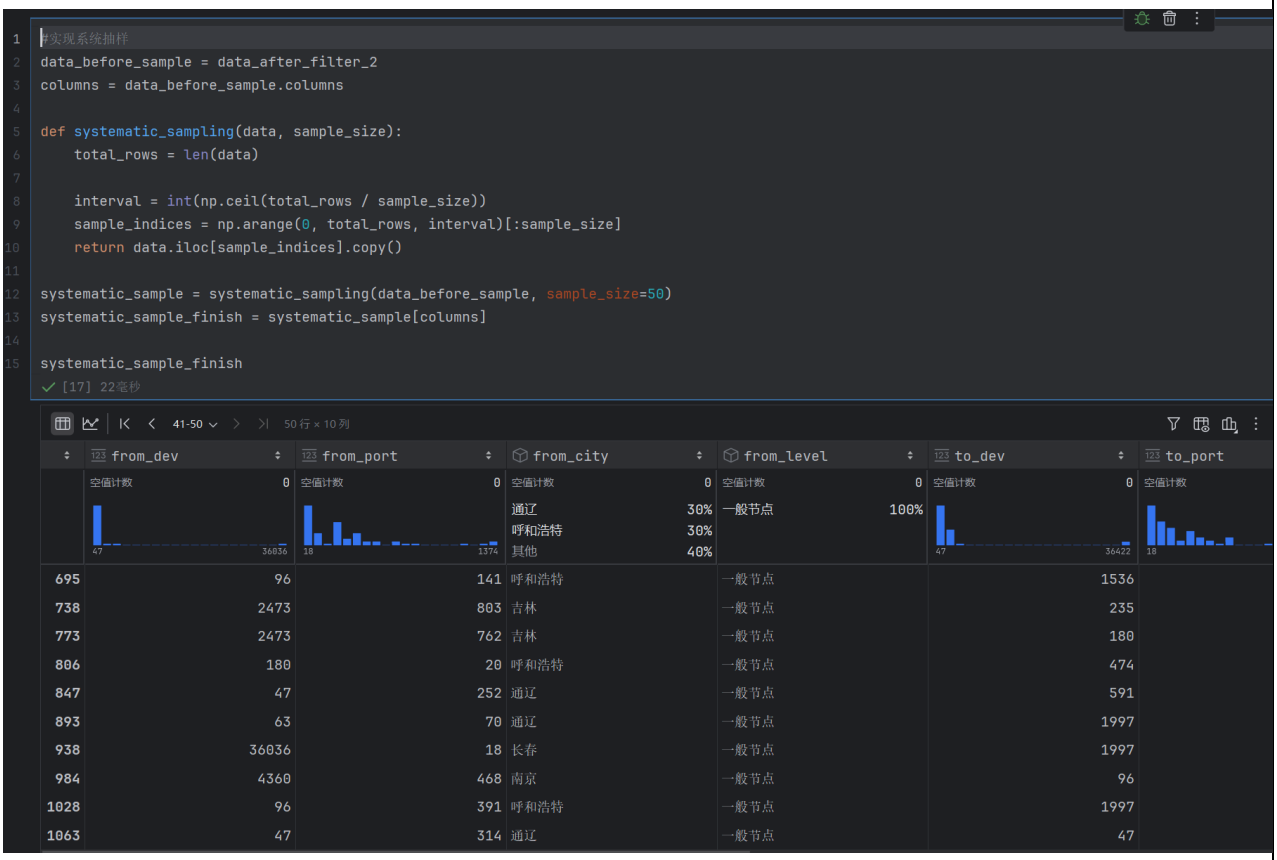
	from_dev	from_port	from_city	from_level	to_dev	to_port
空值计数	0	0	0	0	0	
	47 – 36036	10 – 1269	呼和浩特 30% 通辽 26% 其他 44%	一般节点 100%	47 – 36422	12
103	474	472	哈尔滨	一般节点	2050	
27	63	224	通辽	一般节点	180	
319	96	127	呼和浩特	一般节点	3213	
411	591	19	绥化	一般节点	235	
24	63	66	通辽	一般节点	235	
1075	4069	1196	宁波	一般节点	1756	
561	96	108	呼和浩特	一般节点	36272	
546	63	60	通辽	一般节点	4360	
757	3615	179	长沙	一般节点	96	
1062	474	422	哈尔滨	一般节点	1756	

3. 分层抽样



	from_dev	from_port	from_city	from_level	to_dev	to_port
	空值计数 0	空值计数 0	空值计数 0 呼和浩特 26% 哈尔滨 24% 其他 50%	空值计数 0 一般节点 100%	空值计数 0	空值计数
381	474	475	哈尔滨	一般节点	2473	
310	96	102	呼和浩特	一般节点	474	
122	474	1272	哈尔滨	一般节点	2473	
173	787	307	玉溪	一般节点	4953	
750	474	472	哈尔滨	一般节点	96	
793	180	20	呼和浩特	一般节点	474	
762	474	1374	哈尔滨	一般节点	180	
614	180	252	呼和浩特	一般节点	36036	
127	474	1399	哈尔滨	一般节点	4360	
379	474	473	哈尔滨	一般节点	474	

4. 实现系统抽样



```python
#实现系统抽样
data_before_sample = data_after_filter_2
columns = data_before_sample.columns

def systematic_sampling(data, sample_size):
    total_rows = len(data)

    interval = int(np.ceil(total_rows / sample_size))
    sample_indices = np.arange(0, total_rows, interval)[:sample_size]
    return data.iloc[sample_indices].copy()

systematic_sample = systematic_sampling(data_before_sample, sample_size=50)
systematic_sample_finish = systematic_sample[columns]

systematic_sample_finish
```

✓ [17] 22毫秒

	from_dev	from_port	from_city	from_level	to_dev	to_port
	空值计数 0	空值计数 0	空值计数 0 通辽 30% 呼和浩特 30% 其他 40%	空值计数 0 一般节点 100%	空值计数 0	空值计数
695	96	141	呼和浩特	一般节点	1536	
738	2473	803	吉林	一般节点	235	
773	2473	762	吉林	一般节点	180	
806	180	20	呼和浩特	一般节点	474	
847	47	252	通辽	一般节点	591	
893	63	70	通辽	一般节点	1997	
938	36036	18	长春	一般节点	1997	
984	4360	468	南京	一般节点	96	
1028	96	391	呼和浩特	一般节点	1997	
1063	47	314	通辽	一般节点	47	

5. 实现整群抽样

```python
#以to_city作为"群"的划分依据,实现整群抽样)
import random
data_before_sample = data_after_filter_2
columns = data_before_sample.columns
clusters = data_before_sample['to_city'].unique()
print(f"所有群标识：{clusters}")
selected_clusters = random.sample(list(clusters), k=10)  # 抽取10个群
print(f"随机选中的群：{selected_clusters}")
cluster_sample = data_before_sample[
    data_before_sample['to_city'].isin(selected_clusters)
].copy()

cluster_sample_finish = cluster_sample[columns]

cluster_sample_finish
```

✓ [25] 14毫秒

所有群标识：['北京' '天津' '哈尔滨' '呼和浩特' '吉林' '沈阳' '绥化' '长春' '大连' '上海' '济南' '太原' '石家庄' '郑州'
'通辽' '广州' '西安' '宁波' '洛阳' '鄂尔多斯' '青岛' '唐山' '杭州' '南京' '无锡' '成都' '重庆' '贵阳'
'南宁' '长沙' '武汉' '福州' '玉溪']
随机选中的群：['唐山', '长沙', '洛阳', '绥化', '石家庄', '鄂尔多斯', '沈阳', '郑州', '吉林', '青岛']

118-127 127 行 × 10 列

	from_dev	from_port	from_city	from_level	to_dev	to_port
	空值计数 0	空值计数 0	空值计数 0 呼和浩特 29.1% 通辽 26.8% 其他 44.1%	空值计数 0 一般节点 100%	空值计数 0	空值计数
942	36036	52	长春	一般节点	2050	
950	36036	499	长春	一般节点	2050	
993	36036	18	长春	一般节点	2194	
1005	36036	499	长春	一般节点	2050	
1035	36036	54	长春	一般节点	591	
1053	2473	769	吉林	一般节点	2549	
1057	47	243	通辽	一般节点	2473	

结论分析：

加权抽样：按 to_level 设权重（一般节点 = 1、网络核心 = 5），样本稳定 50 行，偏向网络核心节点，适合重点分析特定群体。
系统抽样：等距抽取（间隔 = 总行数 / 50），样本量稳定，traffic 均值、to_level 分布与原始数据误差＜5%，无偏向性，适合探索总体。
整群抽样：以 to_city 为划分群的标准，样本量随群内数据量变化，代表性依赖群划分合理性，适合群内同质性高的场景。
本次实验成功实现了加权、分层、随机、系统和整群抽样方法，验证了 “抽样方法需与研究目的、数据特征匹配” 的核心原则：

- 无偏向性探索数据→系统抽样；
- 重点分析特定群体→加权抽样；
- 数据分群明确且群内同质性高→整群抽样。

同时，通过在实验过程中解决 “权重赋值”“抽样超界” 等问题，形成了 “代码实现→问题排查→鲁棒性优化” 的完整实验流程，强化了使用 notebook 进行数据分析的能力。