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：202300130236	姓名： 陈德康	班级： 数据 23
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：

Windows11

Anaconda3 Python 3.8

Jupyter Notebook on VSCode

实验步骤与内容：

1. 库的导入与数据的读入

结果如下：

	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

2. 删除多余的空行并进行过滤

（1）采用 dropna 方法并指定参数为 any 删除多余的空行，结果如下：

	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

（2）过滤得到 traffic 不等于 0 且 from_level=一般节点的数据，结果如下：

	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

3. 对数据进行抽样

（1）加权采样：to_level 的值为一般节点与网络核心的权重之比为 1 : 5，结果如下：

	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

（2）随机抽样，结果如下（部分）：

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
302	63	224	通辽	一般节点	1257	498	上海	网络核心	50870996562	1.000000e+11
432	591	1106	绥化	一般节点	3227	781	济南	网络核心	48568999606	1.000000e+11
984	4360	468	南京	一般节点	96	108	呼和浩特	一般节点	51700762978	1.000000e+11
282	47	250	通辽	一般节点	4953	686	贵阳	一般节点	50250217535	1.000000e+11
660	63	224	通辽	一般节点	2701	71	大连	网络核心	50555895575	1.000000e+11
65	180	20	呼和浩特	一般节点	63	224	通辽	一般节点	50551711152	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
289	47	417	通辽	一般节点	3615	191	长沙	一般节点	50099712071	1.000000e+11
414	591	29	绥化	一般节点	235	1649	北京	网络核心	49268934149	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
822	47	243	通辽	一般节点	474	1311	哈尔滨	一般节点	49029906488	1.000000e+11
39	96	114	呼和浩特	一般节点	2473	769	吉林	一般节点	50350633304	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
331	96	346	呼和浩特	一般节点	1756	1128	北京	网络核心	49834736741	1.000000e+11
386	474	673	哈尔滨	一般节点	1997	464	天津	网络核心	49632718644	1.000000e+11
287	47	260	通辽	一般节点	3213	597	重庆	网络核心	50581039842	1.000000e+11
94	180	485	呼和浩特	一般节点	36422	102	天津	网络核心	52460156321	1.000000e+11
45	96	134	呼和浩特	一般节点	47	252	通辽	一般节点	49416652053	1.000000e+11
33	63	286	通辽	一般节点	180	52	呼和浩特	一般节点	49725190236	1.000000e+11
180	787	360	玉溪	一般节点	3615	191	长沙	一般节点	49629725686	1.000000e+11

（3）分层抽样：根据 to_level 的值进行分层采样，结果如下：

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
43	96	124	呼和浩特	一般节点	47	243	通辽	一般节点	49986988230	1.000000e+11
604	96	134	呼和浩特	一般节点	2473	1460	吉林	一般节点	49201392181	1.000000e+11
743	4069	1195	宁波	一般节点	96	134	呼和浩特	一般节点	50099141709	1.000000e+11
5	47	243	通辽	一般节点	96	124	呼和浩特	一般节点	49942713747	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
766	5058	144	南宁	一般节点	180	30	呼和浩特	一般节点	50481413185	1.000000e+11
785	180	252	呼和浩特	一般节点	180	252	呼和浩特	一般节点	47786098667	1.000000e+11
793	180	20	呼和浩特	一般节点	474	359	哈尔滨	一般节点	50601340670	1.000000e+11
959	36036	939	长春	一般节点	47	260	通辽	一般节点	50593921106	1.000000e+11
830	36036	54	长春	一般节点	591	11	绥化	一般节点	49794381448	1.000000e+11
160	591	1258	绥化	一般节点	4448	127	无锡	一般节点	50322958171	1.000000e+11
984	4360	468	南京	一般节点	96	108	呼和浩特	一般节点	51700762978	1.000000e+11
980	4360	472	南京	一般节点	63	286	通辽	一般节点	49837582425	1.000000e+11
834	180	264	呼和浩特	一般节点	591	19	绥化	一般节点	50578150343	1.000000e+11
21	63	58	通辽	一般节点	36036	54	长春	一般节点	48363382095	1.000000e+11
140	591	56	绥化	一般节点	36036	52	长春	一般节点	48627355195	1.000000e+11
400	474	1374	哈尔滨	一般节点	591	23	绥化	一般节点	49461593438	1.000000e+11
566	591	586	绥化	一般节点	1129	514	上海	网络核心	49995510331	1.000000e+11
486	47	74	通辽	一般节点	1385	133	广州	网络核心	49136084036	1.000000e+11
331	96	346	呼和浩特	一般节点	1756	1128	北京	网络核心	49834736741	1.000000e+11
722	4360	468	南京	一般节点	3443	117	青岛	网络核心	49458752996	1.000000e+11
419	591	98	绥化	一般节点	3227	781	济南	网络核心	50666845945	1.000000e+11

（4）系统抽样

系统抽样又称等距抽样，是一种概率抽样方法，其核心思想是：按照某种确定的规则和间隔，从总体中抽取样本，伪代码如下：

$$\text{抽样间隔 } (k) = \text{总体大小 } (N) / \text{样本量 } (n)$$

$$\text{抽样位置} = \text{随机起点} + i \times k\text{，直到遍历完成}$$

代码如下：

```
def systematic_sampling(data, sample_size, random_start=True):
    """系统抽样方法"""
    n = len(data)
    k = n // sample_size

    if random_start:
        start = np.random.randint(0, k)
    else:
        start = 0

    # 生成抽样索引
    indices = [start + i * k for i in range(sample_size) if (start + i * k) < n]

   return data.iloc[indices].reset_index(drop=True)
```

```
systematic_sample = systematic_sampling(data_before_sample, 50, random_start=Tru
e)
systematic_sample
```

结果如下（部分）：

	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	259	通辽	一般节点	1756	245	北京	网络核心	50703793815	1.000000e+11
2	63	60	通辽	一般节点	36422	258	天津	网络核心	49920786706	1.000000e+11
3	63	286	通辽	一般节点	180	52	呼和浩特	一般节点	49725190236	1.000000e+11
4	96	127	呼和浩特	一般节点	1756	1027	北京	网络核心	50087522340	1.000000e+11
5	96	336	呼和浩特	一般节点	1756	1029	北京	网络核心	51600306541	1.000000e+11
6	180	26	呼和浩特	一般节点	36272	133	太原	网络核心	51023900961	1.000000e+11
7	180	98	呼和浩特	一般节点	1129	910	上海	网络核心	50330801190	1.000000e+11
8	180	254	呼和浩特	一般节点	235	1663	北京	网络核心	51477333650	1.000000e+11
9	474	422	哈尔滨	一般节点	96	141	呼和浩特	一般节点	48084671443	1.000000e+11
10	474	682	哈尔滨	一般节点	1536	585	广州	网络核心	50262691915	1.000000e+11
11	474	1374	哈尔滨	一般节点	2050	336	石家庄	网络核心	50242784823	1.000000e+11
12	591	19	绥化	一般节点	36036	18	长春	一般节点	49524524277	1.000000e+11
13	591	526	绥化	一般节点	1129	514	上海	网络核心	49318922185	1.000000e+11
14	591	1266	绥化	一般节点	235	1950	北京	网络核心	48085896120	1.000000e+11
15	787	63	玉溪	一般节点	1536	1882	广州	网络核心	50068630781	1.000000e+11
16	47	74	通辽	一般节点	4561	1033	成都	网络核心	50819524115	1.000000e+11
17	47	260	通辽	一般节点	3213	597	重庆	网络核心	50581039842	1.000000e+11
18	63	66	通辽	一般节点	47	249	通辽	一般节点	48811485662	1.000000e+11
19	96	102	呼和浩特	一般节点	474	678	哈尔滨	一般节点	49006847943	1.000000e+11
20	96	136	呼和浩特	一般节点	3227	389	济南	网络核心	50541979348	1.000000e+11
21	96	383	呼和浩特	一般节点	1536	766	广州	网络核心	50062726803	1.000000e+11

（5）整群抽样

整群分组是按照某个特征（df 中的某一列）把总体分成若干个互不重叠的种群，然后随机抽取部分群组，并取出这些群组的所有个体，适合总体分布广泛或具有自然分组特征的情形，代码如下：

```
def cluster_sampling(data, cluster_column, n_clusters, random_state=None):
    """整群抽样方法"""
    if random_state is not None:
        np.random.seed(random_state)

    # 获取所有唯一的群
    unique_clusters = data[cluster_column].unique()
```

```
    # 随机选择指定数量的群
    selected_clusters = np.random.choice(unique_clusters, n_clusters, replace=False)

    # 抽取选中群的所有样本
    cluster_sample = data[data[cluster_column].isin(selected_clusters)]

   return cluster_sample.reset_index(drop=True)


# 以from_city为依据分组进行整群抽样
cluster_sample = cluster_sampling(data_before_sample, 'from_city', 2, random_state=42)
cluster_sample
```

结果如下：

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	4360	472	南京	一般节点	1385	1490	广州	网络核心	48195505413	1.000000e+11
1	4360	468	南京	一般节点	3443	117	青岛	网络核心	49458752996	1.000000e+11
2	3757	122	福州	一般节点	96	407	呼和浩特	一般节点	47597054356	1.000000e+11
3	4360	472	南京	一般节点	1997	251	天津	网络核心	48414179107	1.000000e+11
4	4360	468	南京	一般节点	1997	464	天津	网络核心	49145116989	1.000000e+11
5	4360	472	南京	一般节点	63	286	通辽	一般节点	49837582425	1.000000e+11
6	4360	468	南京	一般节点	96	108	呼和浩特	一般节点	51700762978	1.000000e+11

结论分析与体会：

通过本次实验，我熟悉并掌握了使用 Pandas 库实现多种数据采样和过滤的方法，体会到了数据过滤对数据分析的重要作用；也体会到了不同抽样方法的作用：加权抽样能够更好地反映总体中不同类别的重要性差异，随机抽样保证了每个样本被选中的公平性，而分层抽样则在保持总体结构的同时提高了抽样效率，等等，需要根据数据特征和分析目标选择合适的抽样方法，这位我后续进行大数据分析实践的学习和进行更复杂的实验奠定基础