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：202300130051	姓名： 汤冉	班级：数据班
实验题目：实验一		
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		
软件环境： python3.9，jupyter notebook		

实验步骤与内容：

1、库的导入与数据的读入

```
import pandas as pd
from pandas import DataFrame
import numpy as np

primitive_data=pd.read_csv("D:\data.csv",encoding='gbk')
primitive_data
```

[5] ✓ 0.0s

数据读入结果如下：

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city
0	47	71	通辽	一般节点	1756	585	北京
1	47	74	通辽	一般节点	1756	776	北京
2	47	240	通辽	一般节点	1756	802	北京
3	47	241	通辽	一般节点	1997	464	天津
4	47	242	通辽	一般节点	474	672	哈尔滨
...	...	...	...	...	...	...	...

2、删除多余的空行并进行过滤

采用 dropna 方法并指定参数为 any 删除多余的空行

```
primitive_data_1=primitive_data.dropna(how='any')
primitive_data_1
```

✓ 0.0s Pyth

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to
0	47	71	通辽	一般节点	1756	585	北京	
1	47	74	通辽	一般节点	1756	776	北京	
2	47	240	通辽	一般节点	1756	802	北京	
3	47	241	通辽	一般节点	1997	464	天津	
4	47	242	通辽	一般节点	474	672	哈尔滨	
...	...	...	...	...	...	...	...	
1113	1129	546	上海	网络核心	2050	502	石家庄	
1114	1129	514	上海	网络核心	2473	946	吉林	
1115	36036	499	长春	一般节点	1257	178	上海	
1116	36422	346	天津	网络核心	1997	41	天津	
1117	2701	619	大连	网络核心	2549	1070	沈阳	

接下来过滤得到 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"]!=
data_after_filter_2=data_after_filter_1.loc[data_after_filter_1["from_leve
data_after_filter_2
```

✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_le
0	47	71	通辽	一般节点	1756	585	北京	网络
1	47	74	通辽	一般节点	1756	776	北京	网络
2	47	240	通辽	一般节点	1756	802	北京	网络
3	47	241	通辽	一般节点	1997	464	天津	网络
4	47	242	通辽	一般节点	474	672	哈尔滨	一般
...	...	...	...	...	...	...	...	
1097	2473	1460	吉林	一般节点	591	586	绥化	一般
1103	36036	18	长春	一般节点	3443	650	青岛	网络
1104	63	6	通辽	一般节点	36036	20	长春	一般
1107	36036	52	长春	一般节点	1129	171	上海	网络
1115	36036	499	长春	一般节点	1257	178	上海	网络

3、一般节点对数据进行抽样，采取不同的采样方式采取 50 个样本并比较采样结果

（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

```
weight_sample_finish=weight_sample.sample(n=50,weights='weight')
#data_before_sample=data_before_sample[columns]
weight_sample_finish=weight_sample_finish[columns]
weight_sample_finish
```

✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	
330	96	336	呼和浩特	一般节点	1756	1106	北京	网络核心	51277669375	1.
374	474	421	哈尔滨	一般节点	3615	179	长沙	一般节点	50627368083	1.
84	180	214	呼和浩特	一般节点	2701	135	大连	网络核心	48901190886	1.
83	180	210	呼和浩特	一般节点	2194	450	唐山	网络核心	50514699101	1.
159	591	1250	绥化	一般节点	235	1749	北京	网络核心	49636424242	1.

（2）随机抽样

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

✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_le
327	96	157	呼和浩特	一般节点	2549	1448	沈阳	网络
418	591	96	绥化	一般节点	2549	852	沈阳	网络
126	474	1389	哈尔滨	一般节点	1756	1127	北京	网络
1021	2473	762	吉林	一般节点	1997	464	天津	网络
75	180	84	呼和浩特	一般节点	1536	86	鄂尔多斯	网络
54	96	159	呼和浩特	一般节点	2360	266	太原	网络
98	474	417	哈尔滨	一般节点	1997	41	天津	网络
660	63	224	通辽	一般节点	2701	71	大连	网络

（3）分层抽样

```
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']
after_sample=pd.concat([ybjd.sample(17),wlhx.sample(33)])
after_sample
```

✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_le
169	787	54	玉溪	一般节点	4953	725	贵阳	一般
1104	63	6	通辽	一般节点	36036	20	长春	一般
822	47	243	通辽	一般节点	474	1311	哈尔滨	一般
867	63	224	通辽	一般节点	787	54	玉溪	一般
959	36036	939	长春	一般节点	47	260	通辽	一般

（4）系统抽样

代码：

```
def systematic_sample(df, n, *, random_state=None):
    N = len(df)
    if n >= N:
        return df.copy()

    rng = np.random.default_rng(random_state)
    k = math.floor(N / n)
    start = rng.integers(0, k)  # 随机起点
    idx = np.arange(start, start + k*n, k)
    idx = idx[idx < N]

    if len(idx) < n:
        extra = rng.choice(np.setdiff1d(np.arange(N), idx), size=n-len(idx), replace=False)
        idx = np.concatenate([idx, extra])
    return df.iloc[np.sort(idx)].copy()

sys_sample = systematic_sample(data_after_filter_2, n=50, random_state=42)
```

结果：

	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
11	47	259	通辽	一般节点	1756	245	北京	网络核心	50703793815	1.000000e+11
22	63	60	通辽	一般节点	36422	258	天津	网络核心	49920786706	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
44	96	127	呼和浩特	一般节点	1756	1027	北京	网络核心	50087522340	1.000000e+11
55	96	336	呼和浩特	一般节点	1756	1029	北京	网络核心	51600306541	1.000000e+11
66	180	26	呼和浩特	一般节点	36272	133	太原	网络核心	51023900961	1.000000e+11
77	180	98	呼和浩特	一般节点	1129	910	上海	网络核心	50330801190	1.000000e+11
88	180	254	呼和浩特	一般节点	235	1663	北京	网络核心	51477333650	1.000000e+11
100	474	422	哈尔滨	一般节点	96	141	呼和浩特	一般节点	48084671443	1.000000e+11
114	474	682	哈尔滨	一般节点	1536	585	广州	网络核心	50262691915	1.000000e+11
125	474	1374	哈尔滨	一般节点	2050	336	石家庄	网络核心	50242784823	1.000000e+11
136	591	19	绥化	一般节点	36036	18	长春	一般节点	49524524277	1.000000e+11

（5）整群抽样：抽 g 个群，群内全取，结果只展示前五行。

```
        if cluster_col not in df.columns:
            raise KeyError(f"[one_stage_cluster_sample] 列 '{cluster_col}' 不存在。可选列: {d
        clusters = df[cluster_col].dropna().unique()
        g = min(g, len(clusters))
        rng = np.random.default_rng(random_state)
        chosen = rng.choice(clusters, size=g, replace=False)
        return df[df[cluster_col].isin(chosen)].copy()

    # 用已有列当群：例如按“去向城市”分群
    cluster_sample_all = one_stage_cluster_sample(
        data_after_filter_2, cluster_col='to_city', g=5, random_state=42
    )
    len(cluster_sample_all), cluster_sample_all.head()
```

✓ 0.0s

```
(80,
     from_dev  from_port from_city from_level  to_dev  to_port to_city  \
4          47        242       通辽       一般节点     474      672     哈尔滨
43         96        124      呼和浩特       一般节点      47      243      通辽
45         96        134      呼和浩特       一般节点      47      252      通辽
48         96        141      呼和浩特       一般节点     474      422     哈尔滨
49         96        152      呼和浩特       一般节点      47      314      通辽

    to_level      traffic     bandwidth
4       一般节点  50492573662  1.000000e+11
43      一般节点  49986988230  1.000000e+11
45      一般节点  49416652053  1.000000e+11
48      一般节点  49429192047  1.000000e+11
49      一般节点  51981076188  1.000000e+11  )
```

结论分析与体会：

代码 weight_sample_finish=weight_sample[columns]会显示所有行，因为抽样结果又被覆盖回原表了。所以需要修改为 weight_sample_finish=weight_sample_finish[columns]。

通过本实验，理解了抽样方法选择与数据特点密切相关。同样的数据，采用不同的抽样方式可能会导致完全不同的结论。通过对比实验，进一步体会到大数据分析不仅是计算，更是设计。抽样方法作为数据预处理的重要环节，会直接影响后续的建模与结论可靠性。

注：实验报告的命名规则：学号_姓名_实验 n_班级