山东大学计算机科学与技术学院

可视化技术课程实验报告

学号：202302130293	姓名：李嘉欣	班级：数据科学与大数据技术
实验题目：一、数据采样方法实践		
实验学时：2	实验日期：2025/9/17	
实验目标：利用 Pandas 库实现多种数据采样和过滤的方法		
实验环境：python3.9		

实验步骤：

1. 库的导入与数据的读入

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

print('1')
primitive_data = pd.read_csv(r"实验一\data.csv")
print(primitive_data.head())
print("\n数据形状（行×列）：", primitive_data.shape)
```

输出：

```
PS D:\1\学习\大三\大三上\大数据分析实践\实验> & C:/Users/dream/AppData/Local/Programs/Python/Python313/python.exe d:/1/学
1
   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

数据形状（行×列）：  (1118, 10)
```

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

```
print('\n2_1')
primitive_data_1=primitive_data.dropna(how='any')
print(primitive_data_1.head())
print("\n数据形状（行×列）：", primitive_data_1.shape)
```

过滤得到 traffic 不等于 0 且 from_level=一般节点的数据

```
print('\n2_2')
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']
print(data_after_filter_2.head())
```

输出：

```
2_1
   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

数据形状（行×列）：  (1118, 10)

2_2
   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
```

3. 对数据进行抽样

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

```
print('\n3_1')
data_before_sample=data_after_filter_2
columns=data_before_sample.columns
weight_sample=data_before_sample.copy()
weight_sample['weight']=0
for 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

weight_sameple_finish=weight_sample.sample(n=50,weights='weight')
#data_before_sample=data_before_sample[columns]
weight_sameple_finish=weight_sameple_finish[columns]
print(weight_sameple_finish.head())
print(weight_sameple_finish.shape)
```

随机抽样

```
print('\n3_2')
random_sample=data_before_sample
random_sample_finish=random_sample.sample(n=50)
random_sample_finish=random_sample_finish[columns]
print(random_sample_finish.head())
print(random_sample_finish.shape)
```

分层抽样：根据 to_level 的值进行分层采样

```
print('\n3_3')
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)])
print(after_sample.head())
print(after_sample.shape)
```

输出：

```
3_1
     from_dev  from_port from_city from_level  to_dev  to_port to_city to_level      traffic     bandwidth
143       591         96        绥化        一般节点    3443      101       青岛     网络核心  51199279798  1.000000e+11
109       474        671       哈尔滨         一般节点    2549      919       沈阳     网络核心  50446722135  1.000000e+11
8          47        251        通辽        一般节点    2549      839       沈阳     网络核心  50755299504  1.000000e+11
135       591         17        绥化        一般节点    3443      186       青岛     网络核心  49474305249  1.000000e+11
728      2473        946        吉林        一般节点    2701      195       大连     网络核心  52184126133  1.000000e+11
(50, 10)

3_2
     from_dev  from_port from_city from_level  to_dev  to_port to_city to_level      traffic     bandwidth
326        96        156      呼和浩特        一般节点    4561     1031       成都     网络核心  50272713910  1.000000e+11
103       474        472       哈尔滨         一般节点    2050      312       石家庄     网络核心  49236653925  1.000000e+11
674       591        586        绥化        一般节点      47      243       通辽     一般节点  50565152517  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
634      2473        769        吉林        一般节点    1997      464       天津     网络核心  49319842054  1.000000e+11
(50, 10)

3_3
      from_dev  from_port from_city from_level  to_dev  to_port to_city to_level      traffic     bandwidth
863       4069       1196        宁波        一般节点     591     1290       绥化     一般节点  48726638175  1.000000e+11
1039       180        264      呼和浩特         一般节点   36036       54       长春     一般节点  49124032697  1.000000e+11
1063        47        314        通辽        一般节点      47      252       通辽     一般节点  49900452417  1.000000e+11
953        180        192      呼和浩特         一般节点      47      249       通辽     一般节点  50233070000  1.000000e+11
732         96        141      呼和浩特         一般节点   36036      499       长春     一般节点  47474335913  1.000000e+11
(50, 10)
```

实验分析与体会：

一、实验分析

本次实验以网络流量数据集为对象，先通过 dropna(how='any')删除原始数据中的空行，再筛选“traffic≠0 且 from_level='一般节点'”的有效记录，为后续采样奠定基础。三种采样方法各有适配场景：加权采样按 to_level 权重 1:5 分配，聚焦网络核心节点；随机采样反映数据整体分布；分层采样固定两类节点样本量，避免小类别被忽略。实验中还解决了路径转义（用原始字符串`r""`）、变量拼写错误等问题，凸显细节对代码运行的关键影响。

二、实验体会

此次实验让我实现理论到实践的落地，比如将加权采样的抽象逻辑拆解为“新增权重列—赋值—采样”的具体步骤，明白数据处理需转化业务逻辑。同时，每步操作后验证结果（如查 NaN 值、样本分布）的过程，让我重视严谨性的重要。此外，思考数据规模扩大后需改用 Spark 等工具，也让我意识到要根据场景动态调整方案，为后续学习积累了实用经验。