大 数 据 分 析 实 践
课 程 实 验 报 告

实验项目-数据采样实践

专业班级：22 级公信

学　　号：202200120100

姓　　名：徐瑞良

报告日期：2025 年 9 月 19 日

目　录

实验 1 数据采样实践

1.1 实验目的

利用 Pandas 库实现多种数据采样和过滤的方法

1.2 实验环境

python3.9，jupyter notebook

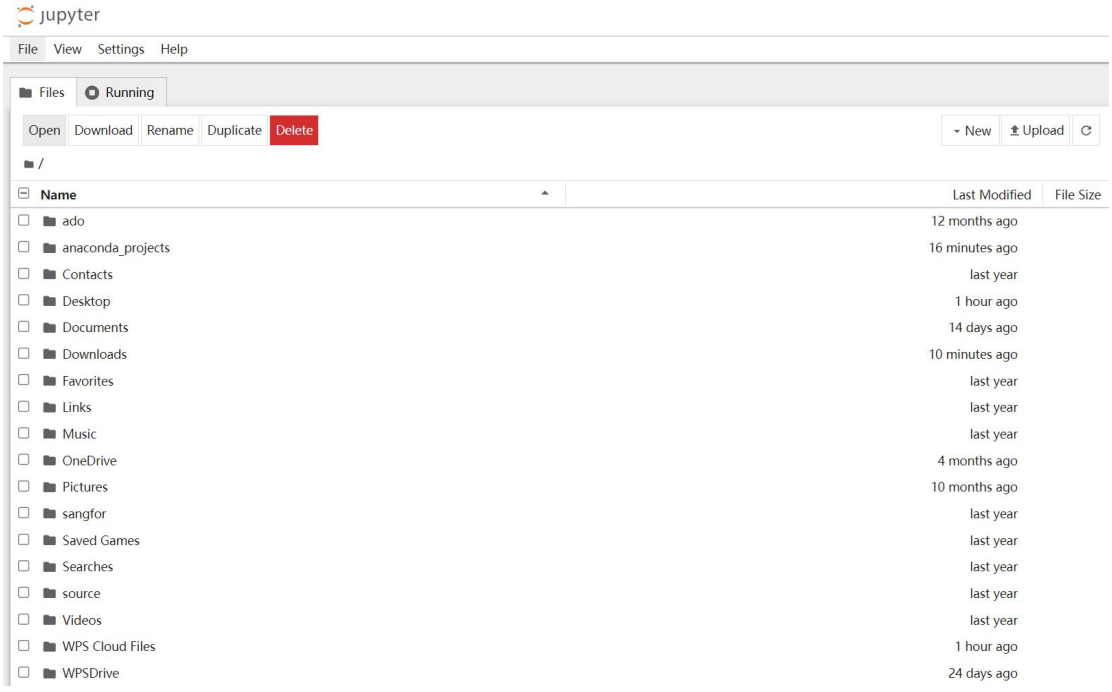
1.3 数据集

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

1.4 实验过程

1.4.1 搭建环境

首先使用 Anaconda 搭建 jupyter notebook



1.4.2 实验步骤

①首先尝试 jupyter notebook 环境搭建是否成功

```
data =123
data
```

```
123
```

成功搭建

②其次按照实验指导步骤进行库的导入与数据的读入，在这一步出现了问题

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

primitive_data = pd.read_csv('D:/大数据分析实践 实验内容/data.csv')
primitive_data
```

```
---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)
Cell In[5], line 6
      3 from pandas import DataFrame
      4 import numpy as np
----> 6 primitive_data = pd.read_csv('D:/大数据分析实践 实验内容/data.csv')
      7 primitive_data

File D:\Conda\envs\ai_clone\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
   1013 kwds_defaults = _refine_defaults_read(
   1014     dialect,
   1015     delimiter,
   (...)
   1022     dtype_backend=dtype_backend,
   1023 )
```

发现由于文件编码问题，pandas 默认使用 utf-8 编码读取 CSV 文件与 data.csv 文件使用的编码格式不符，因而解码失败。

③为解决这一问题，我引入了 chardet 库进行自动检测编码

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

# 检测文件编码
with open('D:/大数据分析实践 实验内容/data.csv', 'rb') as f:
    result = chardet.detect(f.read())

# 使用检测到的编码读取数据
primitive_data = pd.read_csv('D:/大数据分析实践 实验内容/data.csv', encoding=result['encoding'])
primitive_data
```

实验结果也如下所示

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47.0	71.0	通辽	一般节点	1756.0	585.0	北京	网络核心	3.677962e+10	1.000000e+11
1	47.0	74.0	通辽	一般节点	1756.0	776.0	北京	网络核心	3.660713e+10	1.000000e+11
2	47.0	240.0	通辽	一般节点	1756.0	802.0	北京	网络核心	3.603489e+10	1.000000e+11
3	47.0	241.0	通辽	一般节点	1997.0	464.0	天津	网络核心	4.233391e+10	1.000000e+11
4	47.0	242.0	通辽	一般节点	474.0	672.0	哈尔滨	一般节点	1.130008e+10	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1142	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1143	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1144	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1145	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1146	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1147 rows × 10 columns

观察到数据底部有较多的空行

④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。代码及实验结果如下。

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

⑤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"]!=0]
data_after_filter_2=data_after_filter_1.loc[data_after_filter_1["from_level"]=='一般节点']
data_after_filter_2
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47.0	71.0	通辽	一般节点	1756.0	585.0	北京	网络核心	3.677962e+10	1.000000e+11
1	47.0	74.0	通辽	一般节点	1756.0	776.0	北京	网络核心	3.660713e+10	1.000000e+11
2	47.0	240.0	通辽	一般节点	1756.0	802.0	北京	网络核心	3.603489e+10	1.000000e+11
3	47.0	241.0	通辽	一般节点	1997.0	464.0	天津	网络核心	4.233391e+10	1.000000e+11
4	47.0	242.0	通辽	一般节点	474.0	672.0	哈尔滨	一般节点	1.130008e+10	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1097	2473.0	1460.0	吉林	一般节点	591.0	586.0	绥化	一般节点	9.165302e+10	1.000000e+11
1103	36036.0	18.0	长春	一般节点	3443.0	650.0	青岛	网络核心	4.350363e+10	1.000000e+11
1104	63.0	6.0	通辽	一般节点	36036.0	20.0	长春	一般节点	1.871659e+10	1.000000e+11
1107	36036.0	52.0	长春	一般节点	1129.0	171.0	上海	网络核心	2.760267e+10	1.000000e+11
1115	36036.0	499.0	长春	一般节点	1257.0	178.0	上海	网络核心	4.117194e+10	1.000000e+11

554 rows × 10 columns

⑥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

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

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

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
674	591.0	586.0	绥化	一般节点	47.0	243.0	通辽	一般节点	2.310000e+11	1.000000e+11
51	96.0	156.0	呼和浩特	一般节点	3227.0	103.0	济南	网络核心	3.504423e+10	1.000000e+11
16	47.0	427.0	通辽	一般节点	1997.0	213.0	天津	网络核心	4.349038e+10	1.000000e+11
309	96.0	99.0	呼和浩特	一般节点	2360.0	76.0	太原	网络核心	1.810000e+11	1.000000e+11
587	96.0	141.0	呼和浩特	一般节点	3213.0	246.0	重庆	网络核心	9.794152e+10	1.000000e+11
277	47.0	240.0	通辽	一般节点	3213.0	246.0	重庆	网络核心	9.794152e+10	1.000000e+11
365	180.0	260.0	呼和浩特	一般节点	1756.0	788.0	北京	网络核心	1.280000e+11	1.000000e+11
660	63.0	224.0	通辽	一般节点	2701.0	71.0	大连	网络核心	9.786992e+09	1.000000e+11
286	47.0	259.0	通辽	一般节点	4561.0	1087.0	成都	网络核心	1.140000e+11	1.000000e+11
349	180.0	52.0	呼和浩特	一般节点	3227.0	449.0	济南	网络核心	6.987232e+10	1.000000e+11
44	96.0	127.0	呼和浩特	一般节点	1756.0	1027.0	北京	网络核心	8.917187e+10	1.000000e+11
494	47.0	252.0	通辽	一般节点	1536.0	86.0	鄂尔多斯	网络核心	4.103025e+10	1.000000e+11
452	787.0	325.0	玉溪	一般节点	2701.0	181.0	大连	网络核心	9.501373e+09	1.000000e+11
1107	36036.0	52.0	长春	一般节点	1129.0	171.0	上海	网络核心	2.760267e+10	1.000000e+11
172	787.0	63.0	玉溪	一般节点	1536.0	1882.0	广州	网络核心	1.040000e+11	1.000000e+11

⑦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
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
691	2473	946	吉林	一般节点	1756	1117	北京	网络核心	48978564669	1.000000e+11
851	47	314	通辽	一般节点	591	1028	绥化	一般节点	50080623466	1.000000e+11
530	47	249	通辽	一般节点	2473	799	吉林	一般节点	49803820036	1.000000e+11
490	47	243	通辽	一般节点	1385	2778	广州	网络核心	50075073640	1.000000e+11
578	63	70	通辽	一般节点	235	1749	北京	网络核心	50871621460	1.000000e+11
497	47	260	通辽	一般节点	36422	350	天津	网络核心	49613775497	1.000000e+11
167	787	51	玉溪	一般节点	4561	1033	成都	网络核心	51033155364	1.000000e+11
173	787	307	玉溪	一般节点	4953	686	贵阳	一般节点	49399787960	1.000000e+11

⑧分层抽样：根据 to_level 的值进行分层采样。根据比例一般节点抽 17 个，网络核心抽 33 个

```
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
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
294	63	12	通辽	一般节点	474	417	哈尔滨	一般节点	48667628783	1.000000e+11
787	36036	54	长春	一般节点	180	256	呼和浩特	一般节点	51915256521	1.000000e+11
850	474	422	哈尔滨	一般节点	591	638	绥化	一般节点	51214123797	1.000000e+11
173	787	307	玉溪	一般节点	4953	686	贵阳	一般节点	49399787960	1.000000e+11
408	591	13	绥化	一般节点	180	264	呼和浩特	一般节点	51673456087	1.000000e+11
160	591	1258	绥化	一般节点	4448	127	无锡	一般节点	50322958171	1.000000e+11
311	96	105	呼和浩特	一般节点	2473	804	吉林	一般节点	51224734473	1.000000e+11
1057	47	243	通辽	一般节点	2473	769	吉林	一般节点	49117847542	1.000000e+11
445	787	60	玉溪	一般节点	47	314	通辽	一般节点	49484495071	1.000000e+11
392	474	1228	哈尔滨	一般节点	96	134	呼和浩特	一般节点	51278220999	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
793	180	20	呼和浩特	一般节点	474	359	哈尔滨	一般节点	50601340670	1.000000e+11

⑨系统抽样

```
def systematic_sampling(data, sample_size):
    N = len(data)
    k = N // sample_size
    start = np.random.randint(0, k)
    indices = np.arange(start, N, k)[:sample_size]
    sample = data.iloc[indices]
    return sample

systematic_sample = systematic_sampling(primitive_data,50)
systematic_sample
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
12	47	260	通辽	一般节点	2549	835	沈阳	网络核心	50220958279	1.000000e+11
34	96	99	呼和浩特	一般节点	1257	560	上海	网络核心	49753614568	1.000000e+11
56	96	346	呼和浩特	一般节点	1257	138	上海	网络核心	47759033178	1.000000e+11
78	180	188	呼和浩特	一般节点	36422	350	天津	网络核心	49047066099	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
122	474	1272	哈尔滨	一般节点	2473	1043	吉林	一般节点	49735704801	1.000000e+11
144	591	98	绥化	一般节点	2701	195	大连	网络核心	50256295026	1.000000e+11
166	591	1300	绥化	一般节点	3443	1022	青岛	网络核心	49657631257	1.000000e+11

⑩整群抽样

```
def cluster_sampling(data, num_clusters, cluster_column):
    unique_clusters = data[cluster_column].unique()
    selected_clusters = np.random.choice(unique_clusters, num_clusters, replace=False)
    sample = data[data[cluster_column].isin(selected_clusters)]
    return sample

cluster_sample = cluster_sampling(primitive_data, 3, 'from_city')
cluster_sample
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
34	96	99	呼和浩特	一般节点	1257	560	上海	网络核心	49753614568	1.000000e+11
35	96	102	呼和浩特	一般节点	2549	852	沈阳	网络核心	48368154112	1.000000e+11
36	96	105	呼和浩特	一般节点	3227	781	济南	网络核心	50585574378	1.000000e+11
37	96	108	呼和浩特	一般节点	2360	236	太原	网络核心	48210462086	1.000000e+11
38	96	111	呼和浩特	一般节点	3227	468	济南	网络核心	48575689938	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1065	3227	103	济南	网络核心	2360	215	太原	网络核心	51248496177	1.000000e+11
1070	1257	138	上海	网络核心	2050	443	石家庄	网络核心	48498470042	1.000000e+11
1091	1129	910	上海	网络核心	4515	652	西安	网络核心	48912516167	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

222 rows × 10 columns

1.5 心得与体会

本次数据采样实践实验，提升了我使用 Pandas 处理数据的实操能力，让我面对不同的分析需求，思考如何选择更合适的抽样方法、如何处理数据中的异常情况。同时，实验中遇到的问题也让我学会了查阅文档、调试代码的方法，为后续更复杂的大数据分析实验打下了坚实基础。