week3-使用pandas库对数据进行过滤与抽样

201900032052 雷舒扬

实验时间：2023年9月26日

实验目的：pandas是一个强大的Python数据处理库，提供了高效的数据结构和数据分析工具。本实验旨在通过实践应用pandas库中的数据处理和分析功能，探索其在数据过滤和抽样方面的应用。

实验内容：

1. 导入numpy和pandas库。
2. 过滤：首先使用pandas库读取csv文件，并将其转换为Dataframe。接着根据特定的过滤条件condition\_1和condition\_2对数据进行筛选，以获取满足条件的子集。最后将过滤后的数据保存为新的csv文件。
3. 抽样：
4. 加权抽样

加权抽样旨在根据样本权重进行抽样。在本实验中，我们根据”to\_level”列的不同取值为样本赋予不同的权重。使用Pandas的sample函数按照20%的比例进行抽样。

1. 分层抽样

分层抽样要根据特定的层级列对数据进行分层，并在每个层次中进行抽样。我们选择“to\_level”作为层级列。通过groupby进行分层，并apply采样函数。（stratify\_sample()就是应用sample函数）

1. 随机抽样

使用sample()即可。

1. 系统抽样

系统抽样要按照固定步长选择样本，pandas库的iloc方法提供了一种基于整数位置2进行数据访问和选择的便捷方式。在systematic\_sampling()中，我们创建了一个步长为5的索引数组，并使用iloc方法按位置索引进行选取数据。

1. 整群抽样

整群抽样要求从每个群组中随机选择一个样本。我们根据”to\_city“列的不同取值进行数据分组，尝试从每个城市类别中随机抽取一个样本，具体实现方法和分层抽样相似。

代码：

import numpy as np

import pandas as pd

# 读取csv文件

data = pd.read\_csv('test.csv')

df = pd.DataFrame(data)

# 过滤

condition\_1 = data['traffic'] != 0

condition\_2 = data['from\_level'] == '一般节点'

filtered\_data = data[condition\_1 & condition\_2]

filtered\_data.to\_csv('filtered\_data.csv', index=False)

# 抽样

# 加权抽样

# 定义权重列，一般节点权重为1，网络核心权重为5

weights = df['to\_level'].map({'一般节点': 1, '网络核心': 5})

# 采样比例20%，random\_state为随机种子

sample\_1 = df.sample(frac=0.2, weights=weights, random\_state=42)

sample\_1.to\_csv('weighted\_sampling.csv', index=False)

# 分层采样

# 设置分层抽样的层级列

stratify\_by = 'to\_level'

# 定义抽样函数

def stratified\_sample(data):

return data.sample(frac=0.2, random\_state=42)

# groupby按照层级列进行分组，并使用apply方法应用采样函数

sample\_2 = df.groupby(stratify\_by, group\_keys=False).apply(stratified\_sample)

sample\_2.to\_csv('stratified\_sampling.csv', index=False)

# 随机抽样

sample\_3 = df.sample(frac=0.2, random\_state=42)

sample\_3.to\_csv('random\_sampling.csv', index=False)

# 系统抽样

step = 5

def systematic\_sampling(df, step):

# 创建索引数组，包含了以0开始、以len(df结束)，步长为step的索引

indexes = np.arange(0, len(df), step=step)

sample\_4 = df.iloc[indexes]

return sample\_4

sample\_4 = systematic\_sampling(df, step)

sample\_4.to\_csv('systematic\_sampling.csv', index=False)

# 整群抽样

def cluster\_sample(data):

return data.sample(n=1, random\_state=42) # 从每个群组中随机选择一个样本

# 按照to\_city列的值进行分组，然后应用抽样函数

sample\_5 = df.groupby('to\_city').apply(cluster\_sample)

sample\_5.to\_csv('cluster\_sampling.csv', index=False)