What is cluster sampling and how is it implemented in pandas? Can you provide examples?

Authored by

stats writer

April 21, 2024

Cluster sampling is a statistical sampling method in which the population is divided into clusters or groups, and a random sample of these clusters is selected for study. This method is commonly used when the population is large and geographically dispersed, making it impractical to sample individuals from each location.

In pandas, cluster sampling can be implemented through the "sample" function. This function allows for the random selection of a specified number of clusters from the dataset. It also allows for stratified sampling, where the clusters can be selected based on specific criteria such as location or demographics.

For example, if a company wants to conduct a survey on customer satisfaction, they can use cluster sampling by dividing their customers into different regions or stores and selecting a random sample of these clusters to survey. In pandas, this can be done by using the "sample" function and specifying the number of clusters to be selected, along with any necessary stratification parameters.

Cluster Sampling in Pandas (With Examples)

Researchers often take samples from a population and use the data from the sample to draw conclusions about the population as a whole.

One commonly used sampling method is cluster sampling, in which a population is split into clusters and all members of *some* clusters are chosen to be included in the sample.

This tutorial explains how to perform cluster sampling on a pandas DataFrame in Python.

Example: Cluster Sampling in Pandas

Suppose a company that gives city tours wants to survey its customers. Out of ten tours they give one day, they randomly select four tours and ask every customer to rate their experience on a scale of 1 to 10.

The following code shows how to create a pandas DataFrame to work with:

```
import pandas as pd
import numpy as np

#make this example reproducible
np.random.seed(0)

#create DataFrame
df = pd.DataFrame({'tour': np.repeat(np.arange(1,11), 20),
'experience': np.random.normal(loc=7, scale=1, size=200)})

#view first six rows of DataFrame
df.head()

tour experience
1 1 6.373546
2 1 7.183643
```

3 1 6.164371

4 1 8.595281

5 1 7.329508

6 1 6.179532

And the following code shows how obtain a sample of customers by randomly selecting four tours and including every member in those tours in the sample:

```
#randomly choose 4 tour groups out of the 10
clusters = np.random.choice(np.arange(1,11), size=4, replace=False)

#define sample as all members who belong to one of the 4 tour groups
cluster_sample = df.isin(clusters)]

#view first six rows of sample
cluster_sample.head()
```

tour experience

40 3 5.951447

41 3 5.579982

42 3 5.293730

43 3 8.950775

44 3 6.490348

#find how many observations came from each tour group
cluster_sample.value_counts()

10 20
6 20
5 20
3 20
Name: tour, dtype: int64

From the output we can see that:

20 customers from tour group #10 were included in the sample.20 customers from tour group #6 were included in the sample.20 customers from tour group #5 were included in the sample.20 customers from tour group #3 were included in the sample.

Thus, this sample is composed of 80 total customers that came from 4 different tour groups.

Understanding Different Types of Sampling Methods Stratified Sampling in Pandas

Systematic Sampling in Pandas