**EX. NO:2 DEMONSTRATION THE PROBABILITY SAMPLING USING PYTHON**

**AIM:**

To write the python program demonstration the probability sampling using python.

**ALGORITHM:**

* 1. Create a sample from a set of 10 products using probability sampling to determine the population mean of a particular measure of interest.
  2. Implement Simple Random Sampling

The simple random sampling method selects random samples from a process or population where every unit has the same probability of getting selected

* 1. Implement Systematic Sampling

The systematic sampling method selects units based on a fixed sampling interval

* 1. Implement Cluster Sampling

The cluster sampling method divides the population in clusters of equal size n and selects clusters every Tth time

* 1. Implement Stratified Random Sampling

The stratified random sampling method divides the population in subgroupsand selects random samples where every unit has the same probability of getting selected

**PROGRAM:**

**2A) Create Sample:**

# Import required libraries

import numpy as np

import pandas as pd

# Set random seed

np.random.seed(42)

# Define total number of products

number\_of\_products = 10

# Create data dictionary

data = {'product\_id':np.arange(1, number\_of\_products+1).tolist(),

'measure':np.round(np.random.normal(loc=10,scale=0.5, size=number\_of\_products),3)}

# Transform dictionary into a data frame

df = pd.DataFrame(data)

# Store the real mean in a separate variable

real\_mean = round(df['measure'].mean(),3)

# View data frame

df

print(data)

**2B) Implement Simple Random Sampling**

# Obtain simple random sample

simple\_random\_sample = df.sample(n=4).sort\_values(by='product\_id')

# Save the sample mean in a separate variable

simple\_random\_mean = round(simple\_random\_sample['measure'].mean(),3)

# View sampled data frame

simple\_random\_sample

**2C) Implement Systematic Sampling**

# Import in order to use inbuilt functions

import numpy as np

import pandas as pd

 # Define total number of students

number\_of\_students = 15

 # Create data dictionary

data = {'Id': np.arange(1, number\_of\_students+1).tolist(),

        'height': [159, 171, 158, 162, 162, 177, 160, 175,

                   168, 171, 178, 178, 173, 177, 164]}

# Transform dictionary into a data frame

df = pd.DataFrame(data)

 display(df)

# Define systematic sampling function

def systematic\_sampling(df, step):

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

    systematic\_sample = df.iloc[indexes]

    return systematic\_sample

 # Obtain a systematic sample and save it in a new variable

systematic\_sample = systematic\_sampling(df, 3)

 # View sampled data frame

display(systematic\_sample)

**2D) Implement Cluster Sampling**

# code

# importing modules

import pandas as pd

import numpy as np

#creating dictionary of data

dic\_data = {'employee\_id':np.arange(1,21),

                  'value':np.random.randn(20)}

 # creating DataFrame from dictonary

df = pd.DataFrame(dic\_data)

print(df)

samples = df.sample(4).sort\_values(by='employee\_id')

  # show samples

print(samples)

**2E) Implement Stratified Random Sampling**

# Create data dictionary

data = {'product\_id':np.arange(1, number\_of\_products+1).tolist(),

'product\_strata':np.repeat([1,2], number\_of\_products/2).tolist(),

'measure':np.round(np.random.normal(loc=10, scale=0.5, size=number\_of\_products),3)}

# Transform dictionary into a data frame

df = pd.DataFrame(data)

# View data frame

df

# Import StratifiedShuffleSplit

fromsklearn.model\_selection import StratifiedShuffleSplit

# Set the split criteria

split = StratifiedShuffleSplit(n\_splits=1, test\_size=4)

# Perform data frame split

for x, y in split.split(df, df['product\_strata']):

stratified\_random\_sample = df.iloc[y].sort\_values(by='product\_id')

# View sampled data frame

stratified\_random\_sample

# Obtain the sample mean for each group

stratified\_random\_sample.groupby('product\_strata').mean().drop(['product\_id'],axis=1)