```
#Sampling
#1. Random
#importing the random module
import random
#defining the population from where sample will be created
population = list(range(1, 100))
#defining the size of sample
sample_size = 10
#perform simple random sampling by using the random.sample() function
sample = random.sample(population, sample_size)
#it will print 10 random numbers within the range provided
print("Simple random sampling of 10 numbers are: ", sample)
```

```
Simple random sampling of 10 numbers are:  [35, 92, 42, 80, 29, 82, 27, 59, 78, 99]
```

```
#Systematic
import numpy as np

# Define the population
population = np.arange(1, 100)

# Define the sample size and sampling interval
# We can provide sample size and sampling interval as per user
# Here sampling interval is 9 (99//10 = 9)

sample_size = 10
sampling_interval = len(population) // sample_size



start_point = np.random.randint(0, sampling_interval)

# Perform systematic sampling
sample = population[start_point::sampling_interval]

# Print the sample
print("Systematic or interval sampling of 10 numbers are: ",sample)
```

```
Systematic or interval sampling of 10 numbers are:  [ 6 15 24 33 42 51 60 69 78 87 96]
```

```
#Startified
import pandas as pd
from sklearn.model_selection import train_test_split

# Load the data into a Pandas DataFrame
data=pd.read_csv("https://raw.githubusercontent.com/ayan-zz/Statistics_python/main/titanic.csv")

# Specify the stratification variable
stratify_by = 'Sex'

# Split the data into training and testing sets, with stratification
train, test = train_test_split(data, test_size=0.3, stratify=data[stratify_by])

# Check the distribution of the stratification variable in the training and testing sets
print("Train dataset:\n", train[stratify_by].value_counts())
print("Test dataset:\n", test[stratify_by].value_counts())
```

```
Train dataset:
 Sex
male      403
female    220
Name: count, dtype: int64
Test dataset:
 Sex
male      174
female     94
Name: count, dtype: int64
```

```
#Clustered
import random

# create a list of population data
population_data = [10, 15, 20, 26, 29, 33, 35, 37, 41, 42, 46, 48, 50, 52, 55, 58, 61, 64, 68, 72]

# set the desired cluster size
cluster_size = 5

# randomly select a starting point for the first cluster
starting_point = random.randint(0, cluster_size + 1)

# create a list to store the sampled data
sampled_data = []

# loop through the population data by clusters of size cluster_size
for i in range(starting_point, len(population_data), cluster_size):
    # append the data from the current cluster to the sampled data list
    sampled_data.append(population_data[i+1:i+cluster_size])

# print the sampled data
print(starting_point)
print(sampled_data)
```

```
2
[[26, 29, 33, 35], [41, 42, 46, 48], [52, 55, 58, 61], [68, 72]]
```